博士論文

Self-supervised learning models for temporal feature analysis

(時系列情報の特徴検出のための
自己教師あり学習モデル)

朝吹　俊丈

Abstract

Interpretation and execution of complex sequences is crucial for various cognitive tasks such as language processing and motor control. The brain deals with complex sequential inputs by concatenating frequently repeated segments into single unis. While this process, called "chunking" is fundamental to time-series analysis in biological and artificial information processing systems, the underlying mechanism of chunking is still poorly understood. Classical statistical method for sequence learning relies on surprise over inputs, which is based on the bias of transition probabilities over sequence elements. However, recent experimental findings suggest that the brain is unlikely to adopt this method, as human subjects can chunk sequences with uniform transition probabilities.

In this thesis, I examined the neural mechanisms of chunking hierarchical sequences and overcome the limitations of traditional frameworks through modelling studies. The thesis consists of three works related to sequence learning.

In the first work, I examined the effect of independent components extracted from the local field potential (LFP) recorded of rat motor cortex during reward-motivated movement learning. I used the four major independent components to train a recurrent network model for the same lever movements as the rats performed. I found that the independent components differently contribute to the formation of various task-related activities, but they also play overlapping roles in motor learning. Although this research is not directly related to chunking, it shows that reservoir computing, a kind of recurrent network, is effective in studying the neural representation of sequences.

In the second work, I proposed a class of reservoir computing for detecting chunks by predicting dynamical response patterns to sequence input rather than to directly learn transition patterns. In this model, a pair of reservoir computing systems, each of which comprises a recurrent neural network and readout units, supervise each other to detect frequently recurring segments in sequences with

various degrees of complexity. In addition, I demonstrate that background noise plays a crucial role in correctly learning chunks in this model. In particular, the model can successfully chunk sequences that conventional statistical approaches fail to chunk due to uniform transition probabilities. In addition, the neural responses of the model exhibit an interesting similarity to those of the basal ganglia observed after motor habit formation.

In the last work, by analogy with the effect of backpropagating action potentials on synaptic plasticity in the dendrites of cortical pyramidal neurons, I proposed a self-supervising learning rule in single neurons, minimizing the differences in the probabilistic responses between somatic and dendritic activities. I further show that a family of networks composed of the dendritic neurons performs a surprisingly wide variety of complex unsupervised learning tasks including the chunking of temporal sequences and the blind source separation of correlated mixed signals. Common methods applicable to the two different temporal feature analyses were previously unknown. These results suggest the powerful ability of networks of dendritic neurons to analyze temporal features.

Although this thesis consists of model simulation research, yet the experimentally testable prediction of my results are consistent with the experimental results. Therefore, the results of this study may provide a new perspective on the mechanism of sequence learning in the brain.

Acknowledgements

It is my great pleasure to thank my supervisor, Dr. Tomoki Fukai for guiding the PhD study. I thank to Dr. Gonzalo Martín-Vázquez, Dr. Yoshikazu Isomura and Dr. Naoki Hiratani for collaborating my projects. I would like to acknowledge Dr.Tomoki Kurikawa for mentoring project on motor cortex, and Dr. Masato Okada for general mentoring. In addition, I would like to acknowledge Dr. Shunichi Amari, Dr. Takashi Hayakawa, Dr. Tatsuya Haga, Dr. Chi Chung Alan Fung, Dr. Anthony DeCostanzo, Dr. Keita Watanabe for helpful discussion. I would also like to thank to Mr.Yuan Chieh Ling for technical assistance. Furthere, I would like to acknowledge Mr. Kento Suzuki, Mr. Thomas Burns, Mr. Gastón Sivori and Ms. Milena Menezes Carvalho for helpful discussion.

Nomenclature

Acronyms/Abbreviations

FORCE	First-Order Reduced and Controlled Error
ICA	Independent Component Analysis
iSTDP	Inhibitory Spike-Timing-Dependent Plasticity
MIM	Mutual information maximization
MSNs	Medium Spiny Neurons
NMDA	N-Methyl-D-aspartic Acid
PCA	Principal Component Analysis
PETH	Peri-Event Time histogram
PSP	Postsynaptic Potential
STDP	Spike-Timing-Dependent Plasticity

Copyright Statement

Most contents of chapter 2 and 3 are already published in following papers. In the reference (i), I am listed as equally contributed first author. These papers are open to public under CC-BY licence, hence copy and distribution of these works do not come under copyright infringement upon this attribution. Also, the content of chapter 4 is currently under review and will be published later.

(i) Martín-Vázquez, G., Asabuki, T., Isomura, Y., & Fukai, T. (2018). Learning task-related activities from independent local-field-potential components across motor cortex layers. Frontiers in neuroscience, 12, 429.

(i) Asabuki, T., Hiratani, N., & Fukai, T. (2018). Interactive reservoir computing for chunking information streams. PLoS computational biology, 14(10), e1006400.

Contents

List of Figures

Chapter 1

Background

1.1 Temporal feature analysis in the brain

Temporal sequence learning plays an extremely important role in the high-order functions of the brain, such as motor-skill learning and language acquisition. The brain has the ability to encode the temporal structure of sequences and predict the upcoming sensory input, which allows us to perform higher-level cognitive activities. Although deep insights on these sequence learning is crucial, but the underlying learning mechanism is still unknown.

To deal with complex sequential events, the brain seems to have an ability to transform the temporal information into simpler one. This process, called "chunking" is thought to occur in two consecutive processes. Long and complex sequences are first segmented into short and simple sequences, and frequently repeated contiguous events are grouped into a single unit [117].

Direct evidences of chunking has been found in the striatum and prefrontal cortex [34, 52]. In [52], Mice were trained in the task of fast sequential pushing of four levers. After training, Peri-event time histogram (PETH) of medium spiny neurons

(MSNs) in basal ganglia were calculated based on the first, second, third, fourth, or last press timing. As a result, they found that activity patterns of each cell drastically changes through acquisition of a sequential movements. Specifically, two types of activities appeared after learning. In the former, MSNs showed phasic response which was phased locked to the start or end of a sequential movements. This type of activities had already been observed in nigrostriatal circuits [51], and are believed to involved in segmenting sequences. In the latter, a large fraction of MSNs were inhibited during the execution of the whole sequential movements. Although the proportion was lower than that of the inhibited neurons, the rest of the MSNs showed high activation during the whole sequential movements. These results suggest that even single neuron can encode the chunks of sequences.

A widely accepted hypothesis of the mechanism of chunking is that it relies on prediction errors or surprise. Evidence supporting this hypothesis is typically obtained from studies of mismatch negativity, a brain signal (such as from electroencephalogram and functional magnetic resonance imaging) indicating the detection of deviance from a regular temporal pattern of sensory stimuli [7, 20, 100, 114]. In this framework, deviant stimuli are detectable if the brain has learned to predict a normal stimulus pattern. In other words, the brain should be able to identify recurring sequence patterns before it can perceive deviance. In fact, chunking favors an account based on the temporal community detection, in which stimuli that frequently go together in random sequence are grouped into a chunk [97]. In their setting, 15 visual images were sequentially presented, each of which generated by a random walks on a graph. The critical feature of their setting is that since each node in the graph is connected to four other nodes, the transition probabilities are uniform over every nodes. Due to this characteristic, detecting community structures is a difficult task for traditional chunking model, while the human subjects can. They showed the critical evidence that the brain does not rely on the information of local transition probabilities during chunking. However, the physiological

mechanisms of the community detection remain difficult to pursue.

Several models related to chunk learning have been proposed. Below, I explain the main theoretical models.

Predictive coding

The brain seems to predict a stimulus that will be input in the near future, and processes the difference from the actual input to realize efficient information processing. In the cortex, bidirectional connections exist between higher and lower order regions. In predictive coding, it is assumed that the prediction generated in the higher-order region propagates to the lower-order region, and the difference between actual and predicted input (i.e., prediction error) is fed back from the lower-order region to the higher-order region. As a result, a series of processes are performed in which the prediction model in the brain is updated, making it possible to make more accurate predictions.

Predictive coding has gained widespread attention as a general framework for unsupervised learning of sequences. Wacongne et al. (2012) has proposed a spiking neuron model that detects the regularity of sequences and mismatch detection based on the predictive coding [114]. In their model, the transition probabilities of each element in the sequences are stored in synaptic matrix and the model learns the conditional probabilities of current input given the past one. Based on this, the model generates mismatch responses when the expected sequential event is violated.

Stable Heteroclinic Channel

Another powerful theory is based on a chain of sequentially changing metastable states [55, 86]. In that theory, the metastable states are represented by a chain of saddle points (stable heteroclinic channel; SHC) in a low-dimensional dynamical system. While the theory of SHC is not directly related to the mechanisms of "learning", a local synaptic plasticity rule has been proposed to perform segmen-

tation of sequences in a network with metastable fixed points [31].

In the above frameworks, learning of sequences relies on the bias of transition probabilities between each elements. However, as mentioned above, the brain chunks temporally organized elements even if the transition probabilities are uniform. What mechanisms enables the brain to detect temporal features without relying on predictive uncertainty or surprise? To approach this mechanism, let us suppose a network receiving information streams as a input and generates corresponding outpus. The goal here is to transform external information into a more compact representation, and if this is accomplished, the representation of the network should becoms much simpler through training, and the uncertainty of its response should be decreased. Based on this assumption, I proposed a framework for learning sequences by minimizing the uncertainty over the output, not the input transition. The major contribution of my work is to bridge the gap between several sequence learning. I showed that the proposed model exhibits nice performance over a variety of unsupervised learning tasks, including detection of frequently appeared into distractors, grouping temporal community structures, and even the blind source separation of mutually correlated signals, which were previously performed by specialized networks and learning rules.

In Chapter 2, I trained the reservoir model with independent components of local field potentials to reproduce the voluntary forelimb-movement of rats. The results in this chapter will show that reservoir computing is useful for understanding the principles of sequence learning in the brain. On the other hand, since supervised learning is used in conventional reservoir computing, it turns out that it cannot be used directly to learn the characteristics of inputs flexibly. To overcome this shortage, in Chapter 3, I will propose the new framework for training reservoir computing, which we term as "dual reservoir computing". In this framework, a pair of networks supervise each other, which enables the network to rely on unsupervised learning. While this framework shows nice performance

in chunking, it requires some biologically implausible assumptions. Therefore, in Chapter 4, I will propose much realistic model for sequence learning based on two-compartment neuron models. The results will show that single dendritic neuron model can solve a variety of sequence learning tasks. In Chapter 5, all results from previous chapters will be summarized and the discussed the future directions. In Chapter 2 and 3, a powerful learning algorithm for training reservoir computing, called "FORCE learning" was used. I reviewed the details of FORCE in Appendix A.

Chapter 2

Learning Independent Signals in Reservoir Computing

Cortical microcircuits generally receive synaptic inputs from several different brain regions and local field potentials are thought to represent the total input to a local volume of cortex. In motor cortex, how task-related neural activities are formed for adequately associating local synaptic inputs to motor sequences remains largely unclear. In this chapter, I examined the effect of independent components extracted from the local field potential (LFP) recorded of rat motor cortex during reward-motivated movement learning. The results in this chapter will show that the representation of sequence information in a dynamic network model is similar to that in the brain, and is effective in studying the neural mechanisms of sequence learning.

2.1 Introduction

While task-related neural activities have been studied in different layers of cortical microcircuits [25, 39, 48, 68, 72, 108], direct recordings of presynaptic inputs to

local cortical circuits are still technically challenging in behaving animals. The lack of input information makes it difficult to address how inputs from different brain regions contribute to the learning of task-related cortical activities. Here, we ask this question by using the local field potentials (LFPs) recorded from the motor cortex of the rats performing a voluntary sequential arm movement [48].

The major sources of LFP activity are widely thought to be synaptic inputs to local cortical areas. Synaptic inputs from different brain regions generally project to different layers of local cortical circuits. To segregate inputs to the motor cortex, we conducted independent component analysis (ICA) on the LFP data recorded at different cortical depths. If neural activities in different regions targeting the motor cortex are partly correlated with one another, the components extracted by ICA would not represent exact inputs from different brain areas. However, they can be, at least approximately, regarded as independent inputs converging to the primary motor cortex through multiple synaptic pathways.

As the LFP is produced mainly by synaptic currents we assume that the sources are stable in space as it correspond to the transmembrane currents fixed in different dendritic domains determined by anatomy [17, 80]). Thus we are assuming spatial independence for the signal' s sources without any temporal constrains, that allow us to perform temporal and coherence analysis between the extracted components [8, 21, 46, 98]. In the case of spatially identical sources or spatially different sources that are perfectly coherent no separation of sources can be obtained with ICA [71]. But even in situations of highly correlated activity, as in the case of coupled oscillations in feedback/feedforward circuits in CA1 [98] and CA3 [71], ICA can separate different sources successfully.

In this study, I recruited reservoir computing for exploring the roles of the independent compponents (ICs) in motor learning. A dynamical system consisting of a recurrent neural network and readout units is called reservoir computing when readout connections, but no other connections including recurrent connections,

are modifiable. In this system, the recurrent network is referred to as reservoir [99]. I trained the network model receiving the independent inputs to replicate the experimentally observed lever movements of the rats. The rat motor cortex shows several functional subtypes of neurons, i.e., hold-related, pre-movement, movement-related, and post-movement neurons, during the push-pull-hold movement [48], and their relative ratio in population is different between the superficial and deep cortical layers [47]. Results of our modeling study clarify whether the contributions of different ICs to the formation of these functional subtypes are similar or different, giving interesting insight into input- and layer-specific motor information processing.

2.2 Results

2.2.1 Training Reservoir Computing Model with Each ICs Generated Distinct Patterns of Activity

In order to separate and extract the independent signals contained in the LFP, we performed independent component analysis (ICA). ICA is a mathematical technique that separates mixed signals into statistically independent ones with some assumptions on the signal. First, each signal should be independent each other, and it is assumed that they are linearly summed, as is the case for the LFP [30]. Also, it assumes that each signal does not follow the Gaussian distribution, typical feature of brain dynamics [16].

We found four major ICs (i.e., IC1, IC2, IC3 and IC4), with different and recognizable time dynamic and layer stratification along the motor cortex. To examine whether the four ICs obtained from the rat motor cortex are sufficient for motor learning, we trained a reservoir computing model of rate-coding neurons without layer structure to generate experimentally observed arm movements (Figures

2.1A,B; see Methods), and evaluated the contribution of each IC to organizing functionally different neural activities. Training a realistic cortical microcircuit model is beyond the scope of this study. The reservoir received the ICs (10 Hz cut-off) as external inputs (Fig.2.1C) and FORCE learning was used for the training [106]. Because rate-coding neurons are insensitive to input changes faster than the membrane time constant, the ICs were low-pass filtered without changing the essential results. A related modeling study has been performed in the monkey motor cortex by using movement-preparatory activity as input [107]. Here, we aimed at generating all functional subtypes of neurons by using the ICs as input.

The reservoir model was trained simultaneously with all four ICs or separately with each IC. The post-learning responses of individual neurons are shown for each case (Fig.2.1D). In all the cases, the resultant population activity contained neural responses which are similar to various task-related neurons found in experiment. Examples of the Hold-related (or Movement-off), Movement-related and Pre-movement activities learned by the network model are in Fig.2.1E.

2.2.2 Activity Properties of the Reservoir Neurons

Depending on the input conditions, the model exhibited a different fraction of Movement-related neurons and Hold-related (or Movement-off) neurons (Fig.2.2A). The fractions obtained from the simulations are compared with those observed in experiment (Fig.2.2B, superficial plus deep layers). In the model, contributions of individual ICs exhibited interesting differences. For IC1 and IC3, the fraction of Hold-related neurons was small compared to that of Movement-related neurons, while opposite was true for IC4. In experiment, movement-related neurons were found dominantly in the superficial layer (Fig.2.2B), suggesting that the major drivers of the superficial layer involve IC1 and IC3. In contrast, the deep layers contained nearly identical fractions of these functional subtypes, from which a

reliable assessment of the relative contributions of ICs seems to be difficult.

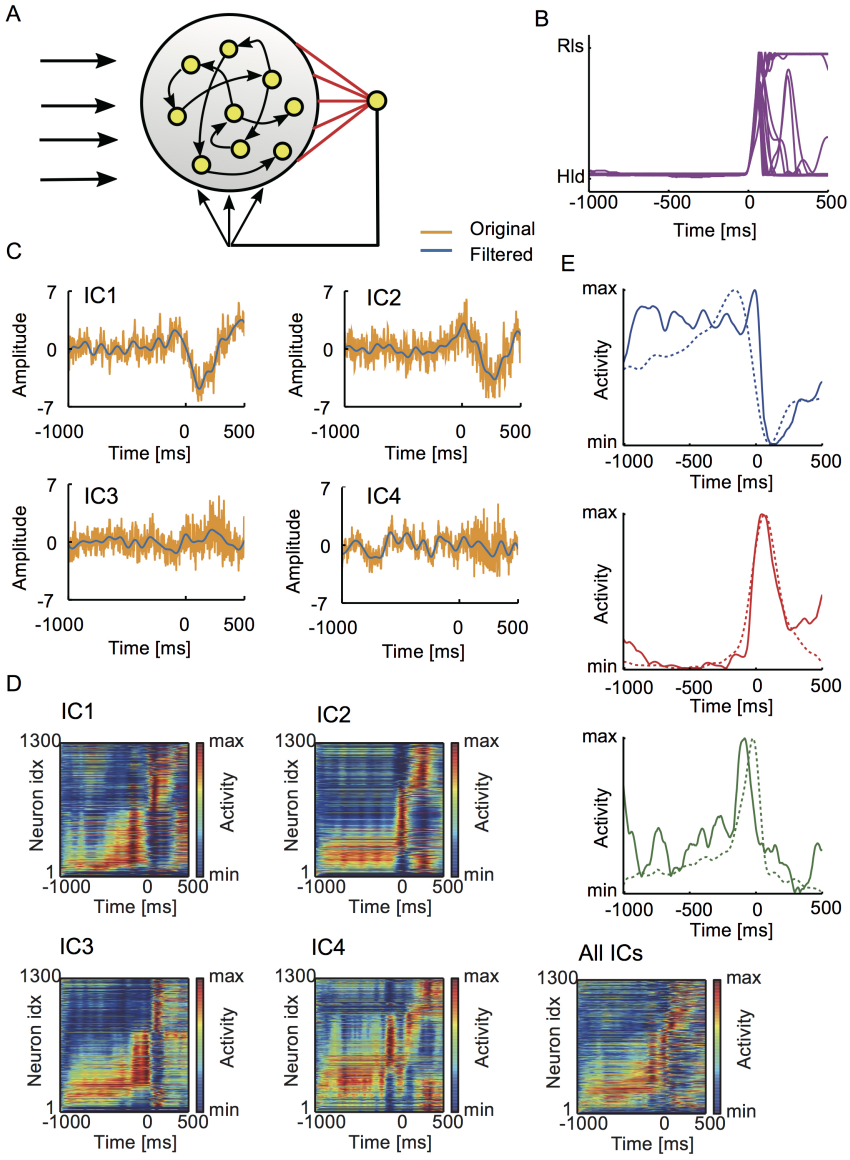

Figure 2.1: **Learned activity patterns in the reservoir network model.** (A) The model is schematically illustrated. (B) Lever movements in 18 trials provided teaching signals for learning. Time 0 refers to the onset of lever pull. (C) Typical examples of the independent components of LFP (yellow) and their low-pass-filtered versions (blue) are shown. (D) The average normalized responses of all neurons in five different conditions: external input to the reservoir involved all ICs or just one IC. The activities of individual neurons were averaged over repeated trials and sorted according to the onset time of activation (see Methods). (E) Examples of the average responses are shown for Hold-related or Movement-off (top), Movement-related (middle) and Pre-movement (bottom) neurons.

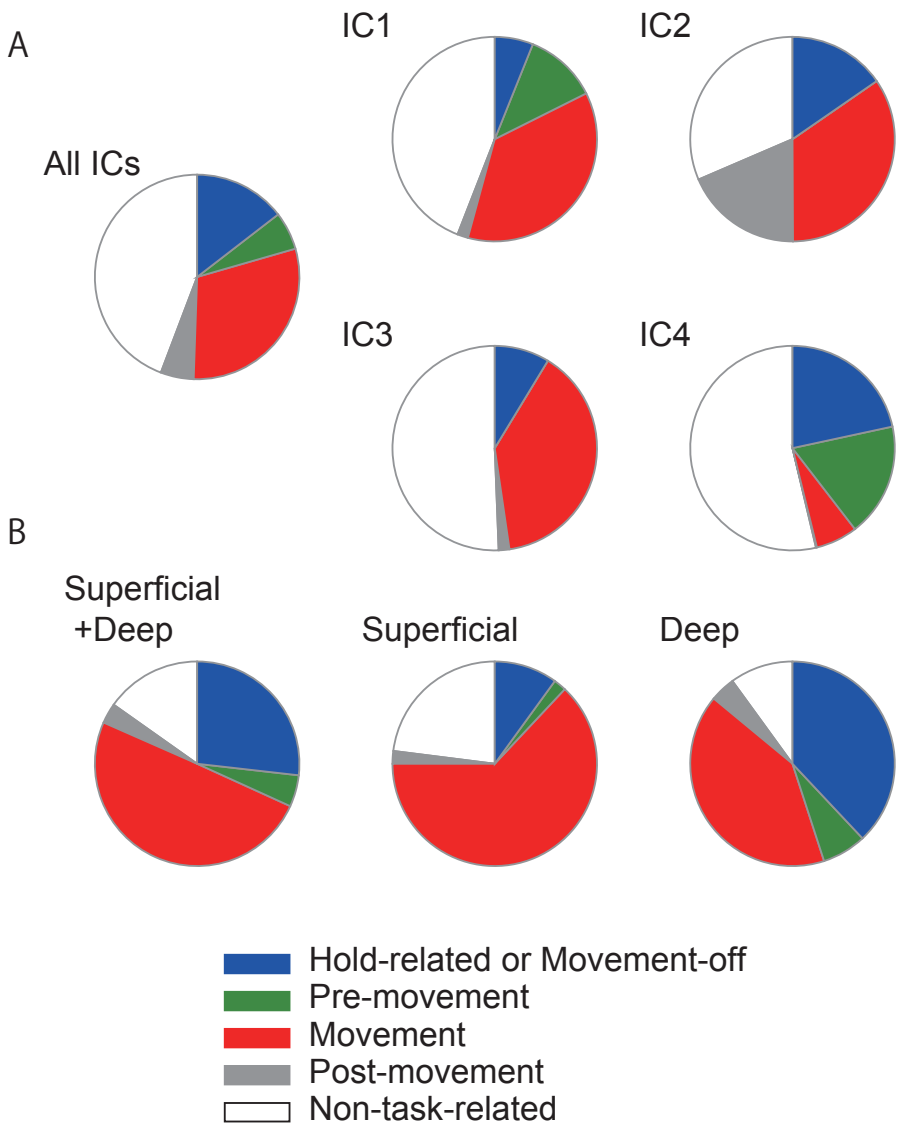

Figure 2.2: Population fraction of each functional subtype.
(A) The population ratios of functional subtypes were calculated in the models with different settings of input. (B) Similar population ratios were obtained from the experimental data reported in [47].

2.3 Discussion

In this study, we showed that training a reservoir model with the ICs successfully replicated the various functional subtypes of task-related motor cortical neurons with relative portions similar to those obtained experimentally, except that the model exhibited a larger portion of non-task-related neurons (Fig.2.2). The contributions of different ICs to the generation of Hold-related and Movement- related neurons were qualitatively similar, suggesting that they play overlapping rather than specific roles in motor learning. In the modeling, we low-pass-filtered the ICs as the model is insensitive to high frequency signals. This, however, should not be interpreted as the unimportance of high-frequency oscillations to motor learning. Gamma oscillations have been implicated in cross-area communication [18, 23, 118], and a computational study suggests that dendritic low-pass filtering makes gamma-band synchronized activity an optimal carrier of analog information between cortical neurons [35].

2.4 Methods

2.4.1 Neural Network Model

Our reservoir network consists of N_G neurons and the activity of each neuron x_i obeys the equation A.1 in Appendix A. In the present simulation, $N_G = 1300$ and $\tau = 50$ms. Each neuron in the reservoir received an input to which one of the IC1, IC2, IC3, and IC4 was randomly assigned. In some simulations, different ICs projected to approximately the same numbers of neurons without overlaps. In other simulations, all neurons received the same IC and the contributions of each IC to motor learning were separately evaluated. The inputs were normalized between -1 and +1 and were low-pass-filtered at 10Hz because the evolution of the present rate model is not sensitive to high-frequency components of inputs.

Neuron pairs were randomly connected with the connection probability $p = 0.1$, and the weights of non-modifiable recurrent connections J_{GG} were determined by a normal distribution with mean 0 and variance $1/(pN_G)$. The overall factor of recurrent inputs was set as $g_G = 1.5$ such that the pre-training network showed chaotic activity [106]. All neurons in the reservoir projected to readout unit $z(t)$ and were projected back to by the readout unit. The feedback connections had fixed weights J^{G_z} , which were taken randomly from a uniform distribution between -1 and +1. Synaptic weights to the readout were modifiable and trained by FORCE learning algorithm as in [106].

2.4.2 Simulations and Data Analysis

Training data for arm trajectory and LFP data were obtained in [48]. In the behavioral task, rats spent the majority of task period for lever-hold and generated movements only during a small portion of the task period. To enable efficient learning of movements, we only used data segments containing lever pull in each trial, namely, from 1 s before to 500 ms after lever pull onset. The data set used for learning was obtained from 18 trials. After learning, activity of each neuron in the reservoir was averaged over the 18 trials and normalized between its minimum and maximum values. Then, activity was categorized into five distinct functional subtypes as in [48]. Briefly, Movement-related activity was a phasic activation during movements, whereas Hold-related and Movement-off neurons exhibited a phasic decrease of activity during movements. The latter two types of functional activity were not distinguished in this study. Pre-movement was a phasic activation starting earlier than 500 ms before movement onset and rapidly dropping by more than the half of its peak activation after movement onset. Post-movement was a phasic activation starting from movement onset and dropping within 350 ms from the onset. Activity profiles that were not categorized into any of these functional subtypes were categorized as “others.” Phasic activity was defined as

activity of each neuron that increased or decreased beyond $\mu \pm 3\sigma$ for more than 60 ms, where μ and σ stand for the average and standard deviation of its activity during 1,000 - 250 ms before movement onset, respectively. In Fig.2.1C, neurons were sorted according to the serial order of activation time, which was calculated as

$$\hat{t}_i = \frac{T}{\pi} \arg \left[\frac{\sum_{t'=1}^{T} \bar{r}_i(2t' - 1000) \exp\left(i\frac{2\pi t'}{T}\right)}{\sum_{t'=1}^{T} \bar{r}_i(2t' - 1000)} \right] \text{[ms]},$$

where $\bar{r}(t)$ is the normalized average response of each cell, $T = 750$ms and the activation time was adjusted such that it falls within the range [-1000, +500] ms.

Chapter 3

Chunk Learning by Interactive Reservoir Computing

In Chapter 2, I showed that specific neural activity was generated at each stage of the movement by learning the recurrent network using animal behavior as a teaching signal. In particular, some reservoir neurons showed increasing activities at the beginning and end of the lever movement, which may be related to chunk segmentation, and the model used in the previous chapter can be considered to be a candidate for building chunk detection model. However, our goal is to build a model for feature detection without external teachers, and the further extension of the model is necessary. Therefore, in this chapter, I proposed a model that uses two independent reservoir circuits and performs learning using each other's outputs as teaching signals. The proposed model was able to learn the chunks in the time series without a teacher even in complicated situations such as when there was overlap between chunks.

3.1 Introduction

When a sequence of stimuli is repeated, they may be segmented into "chunks," which are then processed and stored as discrete units. This process, called "chunking" or "bracketing" [38], takes place during various cognitive behaviors that require hierarchical sequence processing[22, 29, 77, 82]. For instance, in motor learning, a sequence of smaller movements may be executed as one compound movement after repetitive practice [34, 38, 51, 52, 104]. During language acquisition, continuous vocal sounds are segmented into familiar groups of contiguous sounds that are processed as words [15, 36]. Chunking is believed to reduce the complexity of sequence processing and hence the associated computational cost [38, 87, 113]. In this regard, chunking constitutes a crucial step in representing the hierarchical structure of sequential knowledge in neural circuits [26].

Chunking is believed to occur through two consecutive processes. Long and complex sequences are first segmented into shorter and simple sequences, while frequently repeated segments are concatenated into a single unit [117]. Various mechanisms of chunking have been proposed based on Bayesian computation [55, 82], statistical learning guided by prediction errors [89], and a bifurcation structure (stable heteroclinic orbits) in nonlinear dynamical systems [31, 86]. In addition, a neuromorphic hardware has been proposed [61]. However, none of these mechanisms have been shown to chunk with the level of flexibility that the brain offers. It also remains unclear whether a bifurcation theoretic mechanism exists that enables the chunking of arbitrary complex sequences. Many studies evaluating event segmentation in biological and artificial systems have focused on mechanisms to detect boundaries between events by transient increases in surprise signals, which are thought to form based on unequal transition probabilities among sequence elements [26, 82, 84, 88]. However, human subjects can segment sequences of visual stimuli that have uniform transition probabilities and hence

cannot be chunked by any variation of such mechanisms [97]. These findings suggest that biological neural networks favor a mechanism of chunking that is based on temporal community detection, in which stimuli that frequently go together are grouped into a chunk. A similar mechanism may also account for the brain's ability to detect repetitions of patterned stimuli in random sequences [2, 44, 93].

However, the logic and neural mechanism of flexible and automatic chunking by the brain remain unknown. In this study, we propose a novel mechanism of unsupervised chunk learning based on a unique computational framework that differs from any of the previous proposals. In this mechanism, neural networks learn the low-dimensional dynamical trajectories embedding stereotyped responses to recurring segments (chunks) of a temporal input. We achieve this mechanism in a framework of cortical computation [49, 65] by extending reservoir computing (RC) to unsupervised learning. RC is a high-dimensional dynamical system consisting of a recurrent neural network, readout units, with feedforward and feedback projections between them, and supervised learning in its original form [106]. We were able to attain unsupervised learning in a pair of independent RC modules supervising each other without any external instructive signal. As a consequence, they learned to mimic, or predict, the preferential responses of partner modules to chunks in a given temporal input.

The primary interest of this study was determining the novel computational mechanism to segment information streams. However, an unexpected finding included a surprising similarity between the temporal response patterns of readout units in our model and a functional subtype of basal ganglia neurons, called stop cells, which are observed after habituation [51, 52, 104]. This finding suggests that the proposed paradigm of sequence processing has a biological relevance.

3.2 Results

3.2.1 Reservoir computing modules with mutual supervision

I will describe the basic framework of our model. Our model receives the input sequence which contains chunks and random sequences of discrete items. Here, we first start with the simple case where the sequence has only a single chunk of letters (i.e., a-b-c-d). The random sequences were chosen from the remaining 22 letters of the English alphabet (e to z) (Fig.3.1A). In a realistic setting, these letters correspond to a brief stimulus in any sensory modality, but we assume each of them gives phasic activation of a specific input neuron ($I_\mu(t)$ in Eq 3.4 in the Methods) with slow rise and decay constants (Fig.3.1B). Thus, the number of input neurons coincides with that of letters. The random sequence components are introduced to unambiguously define the initial and end points of a chunk, and their lengths vary with every repetition cycle within the length range of 5 to 8.

Our network model consists of two independent rate-based and all-to-all connected reservoir computing (RC) modules. Each module receives an identical input sequence (Fig.3.1C) and feed back the readout activity to the same reservoir. To make each reservoir neuron selective to its preferred stimulus, we assumed that each reservoir neuron receives input from one of the input neurons although this constraint is not essential for chunk learning and can be relaxed, as shown later. Although the two reservoirs have same structures, each of them has its own wiring patterns and not identical. The activity of each readout unit $z(t)$ is given as a weighted sum of the activities $\mathbf{r}(t)$ of the reservoir neurons projecting to the readout: $z(t) = \mathbf{w}^T\mathbf{r}(\mathbf{t})$. Although we will consider more complex case later, we first test in a case that each of the RC modules connects to only a single readout unit. We trained our model by FORCE learning algorithm, which is described in Appendix A. Following their setup, we trained only the readout connections $\mathbf{w}$ of both reservoirs, whereas the recurrent and feedback connections are always

fixed. The initial states of the reservoirs are weakly chaotic as in the previous model [106]. See the Methods for details of the model and values of the relevant parameters.

The previous frameworks of RC relays on supervised learning, hence it requires teacher signals from outside. In our model, on the other hand, the output of each readout unit is used as a teacher signal to train the readout weights of the other reservoir module. Although each module is supervised, the entire system undergoes unsupervised learning, since the teacher signals are generated within the model. The details of the teaching signals will be shown later.

3.2.2 Chunk Learning from a Random Sequence

The form of teaching signal is important for successful chunk learning in the proposed model. The teaching signal is symmetric with respect to the interchange of the two readout units and must be determined so that the two systems stop learning when the two readout units output similar response patterns. In other words, the teaching signal will eventually be the same between the two RC modules during learning. The teaching signal f_i for successful chunk learning in the dual RC system has the form:

$$f_i(t) = [\tanh(\hat{z}_j(t)/\beta)]_+. \ (i, j = 1, 2; i \neq j) \tag{3.1}$$

where $\hat{z}_i$ is the standalized output of the i-th readout unit (Methods), the threshold linear function $[x]_+$ returns 0 if $x \leq 0$ and $[x]_+ = x$ if $x > 0$, and the constant was set as $\beta = 3$. Defining error signals as $e_i(t) = z_i(t) - f_i(t)$, we trained RC module pairs with the FORCE learning algorithm until the error was sufficiently small (typically, about 0.01) and the readout weights converge to equilibrium values (within small fluctuations). The saturation part of nonlinear function prevents the model from responding too strongly to a specific chunk and makes it easier

to detect all the chunks embedded in the input sequence. This activity regulation is particularly important in the learning of multiple chunks studied later. The threshold linear function makes the outputs positive; these nonlinear transformations greatly improved the performance of learning. Note that the teaching signal does not explicitly contain information about the temporal structure of the input sequence. This dual RC system converged to a stable operating state when the two RC systems generated teaching signals consistent with the temporal structure of the input sequence (Fig.3.2). The readout units did not respond to the chunk before learning (Fig.3.1D). After learning, the responses of the readout units were tested for the input sequences that had not been used for the training. The test sequences contained the same chunk "a-b-c-d," but the random sequence part was different. Given these inputs, the readout units exhibited steady phasic responses time-locked to the chunk (Fig.3.1E). The readout activity piled up gradually in the beginning of the chunk, rapidly increased at its end, and then returned to a baseline level. The selective responses to the chunk were also successfully learned when each reservoir neuron was innervated by multiple input neurons. As shown in Fig.3.1F, the system succeeded in learning when randomly-chosen 10% or 40% of input neurons projected to each reservoir neuron, but failed when the fraction was 70%. Thus, responses of the individual reservoir neurons should be sufficiently independent of each other to robustly capture the recurrence of chunks.

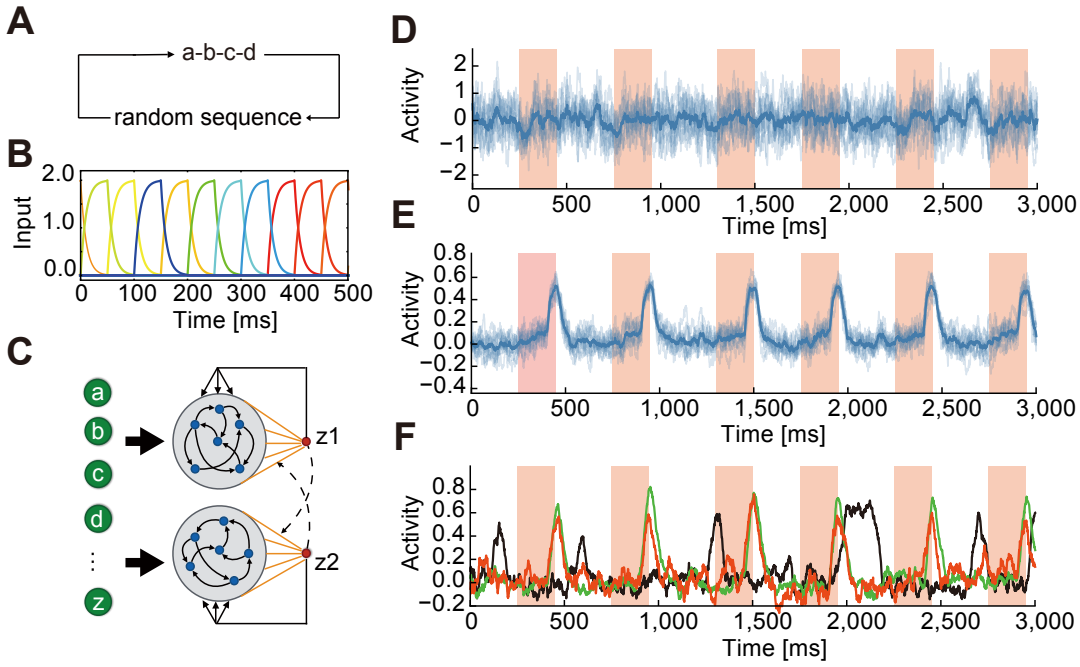

Figure 3.1: **Learning of a single chunk repeated in random sequence.** (A) Input sequence repeating a single chunk. In this example, the chunk is composed of four alphabets (a, b, c, d). The components and lengths of random sequences varied during the repetition of chunks. (B) Example responses are shown for input neurons. (C) In the dual RC model, two non-identical reservoirs are activated by the same set of input neurons. Readout weights of each RC system undergo supervised learning with a teaching signal given by the output of the partner network. (D) and (E) Pre- and post-learning trial averaged activities of a readout unit are shown, respectively. Shaded intervals designate the presentation periods of the chunk. The other readout unit exhibited a similar activity pattern. (F) Readout activity was trained with many-to-one input projections. The fraction of input neurons projecting to a reservoir neuron was 10% (red), 40% (green) and 70% (black).

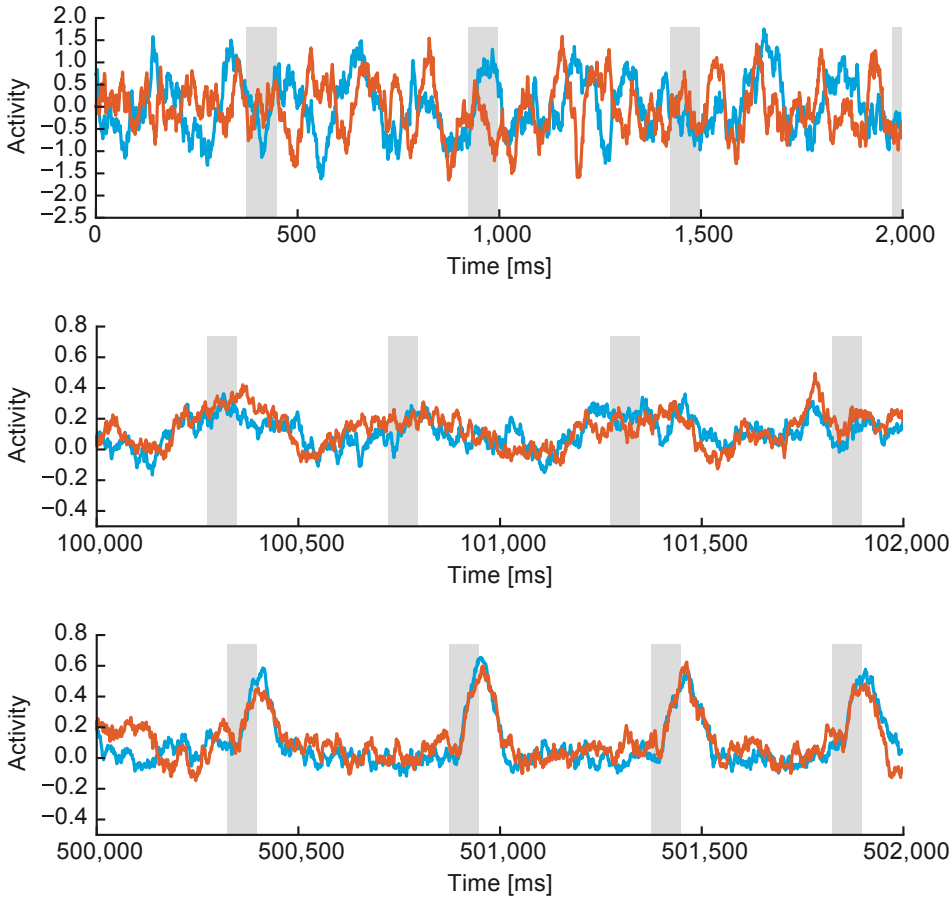

Figure 3.2: **Responses of readout neurons during learning procedure.** Below, numerical results are shown for the model simulated in Fig.3.1. The responses of two readout neurons were initially incoherent (top). They gradually developed strong coherent responses to the repetition of a chunk as learning proceeded (middle and bottom).

3.2.3 Learning of Multiple Chunks

The proposed model can learn multiple chunks by extending previous learning rules. To show this, we embedded three chunks in a random input sequence (Fig.3.3A, top). The occurrence probability of the three chunks was 1/3. To handle this complex input sequence, two changes were made to the previous model. First, each reservoir was connected to three readout units (z_1, z_2, z_3 for the 1st reservoir and z_4, z_5, z_6 for the 2nd reservoir), each responsible for the learning of one of the three chunks (Fig.3.3B). Second, we modified the teaching signals as follows:

$$f_a(t) = [\tanh((\hat{z}_{a'}(t) - \gamma \sum_{c=4,5,6}{}' \hat{z}_c(t))/\beta)]_+ \ (a = 1, 2, 3) \tag{3.2}$$

$$f_b(t) = [\tanh((\hat{z}_{b'}(t) - \gamma \sum_{c=1,2,3}{}' \hat{z}_c(t))/\beta)]_+ \ (b = 4, 5, 6) \tag{3.3}$$

where a' and b' refer to the corresponding readout units of the partner RC modules (i.e., $a' = a + 3$ and $b' = b - 3$), and dashes in the second term indicate that the corresponding readout unit should be excluded from the summation. The constant γ was set as 0.5. Therefore, as in the previous case, the teaching signal was exchanged between the RC modules. Each readout unit receives three sets of teaching signals from the partner network, one being cooperative and the other two being competitive (Fig.3.4A). These signals cause each read unit to respond to a particular chunk, but the chunks that the read unit learns are not pre-specified because the teacher signal is symmetric with respect to the permutation of the index per reservoir. A further extension of the learning rule to an arbitrary number of chunks is straightforward.

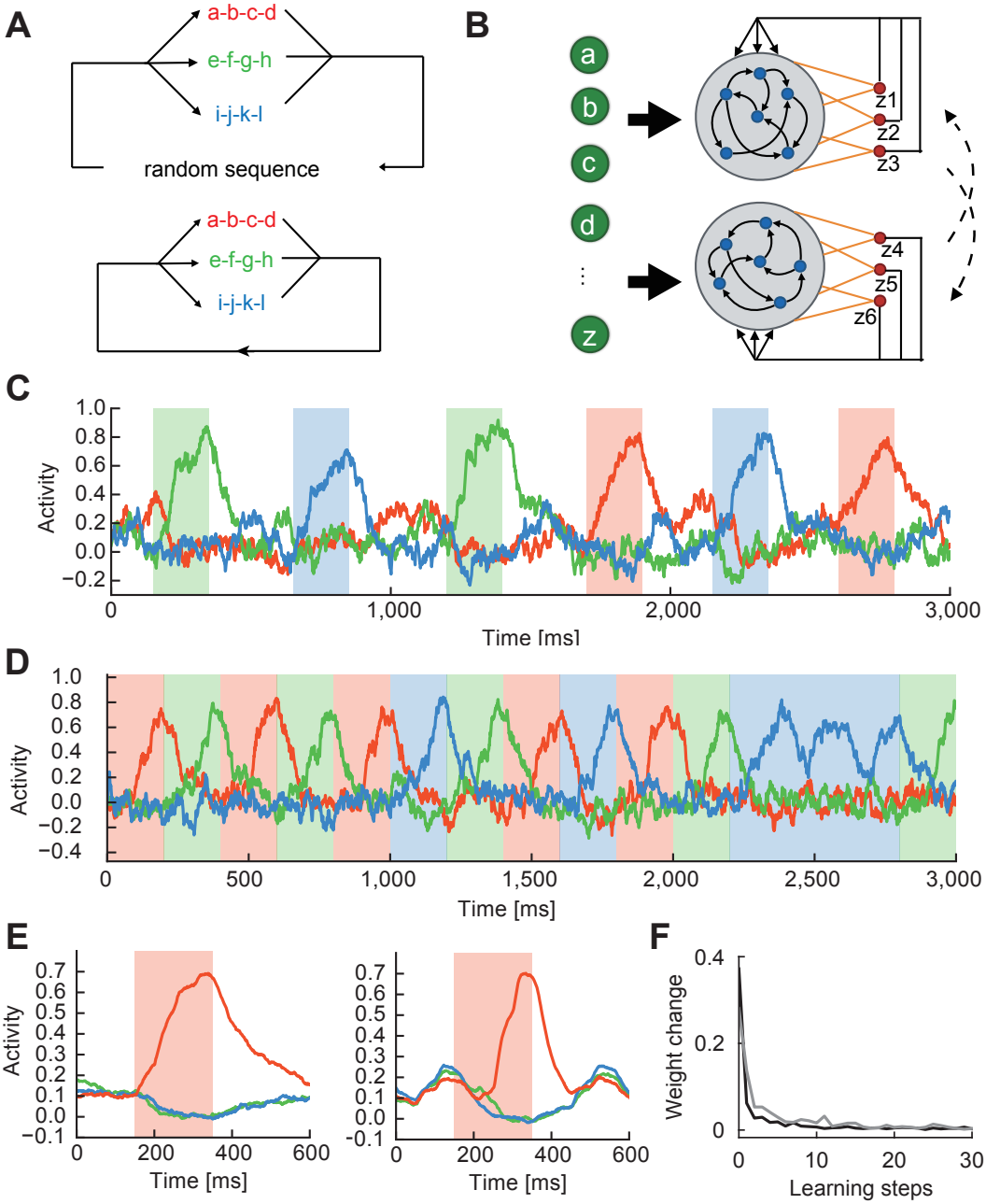

Figure 3.3: Learning of multiple chunks repeated in random sequence.
(A) Top, Three chunks a-b-c-d (red), e-f-g-h (green), and i-j-k-l (blue) separated by random sequences are recurred at equal frequencies in input. Bottom, The three chunks are repeated without the intervals of random sequences. (B) Each reservoir was connected to three readout units. (C) Selective readout responses to the individual chunks (colored intervals) were self-organized. Input contained random sequences. The responses are colored according to their selectivity to the chunks. (D) The same chunks were repeated without breaks by random sequences. Previous models of chunking typically processed such input sequences. (E) Readout activities formed with (left) and without (right) random sequence intervals were averaged over the recurrence of chunk "a-b-c-d". (f)FTime evolution of average readout weights is shown at every step of learning with (gray) and without (black) random sequence intervals.

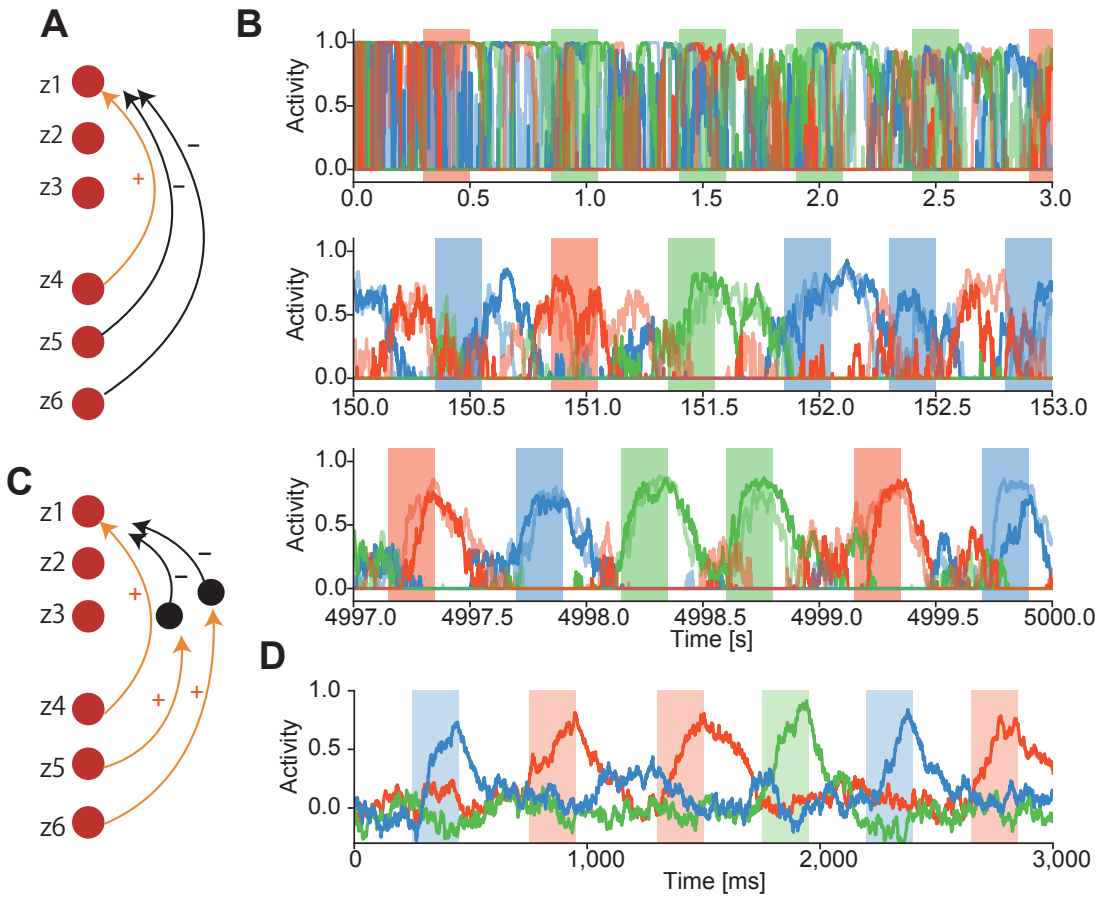

Figure 3.4: **Structure of teaching signals for multiple chunk learning.** (A) A schematic illustration for the structure of teaching signals for z1. Since the partner node of z1 is z4, the sign of the corresponding term in the teaching signal is positive whereas the other terms are negative. (B) Teaching signals show incoherent activities (top) before learning, while the learning procedure makes these activities much coherent (middle and bottom). Thick and thin lines represent teaching signals from the pair of the readouts. (C) Lateral inhibitions by interneurons are modeled. (D) The activities of readouts after learning with interneurons.

As in the case with a single chunk, each readout unit displayed a ramping activity selective to a specific chunk, signaling successful chunk learning (Fig.3.3C). During this learning, teaching signals also self-organized such that each pair of the readout units eventually exhibited a selective response to a specific chunk, indicating that the teaching signals work adequately (Fig.3.4B). The complex form of teacher signals looks somewhat biologically unrealistic, but they can easily be implemented by inhibitory neurons (Fig.3.4C: see Methods) to generate chunk-selective phasic readout responses (Fig.3.4D). Below, inhibitory neurons are not explicitly modeled.

The question then arises whether the RC system could also learn multiple chunks when they occur continuously without temporal separations by random sequences. To study this, we trained the model by using input sequences in which three chunks appear randomly and consecutively with equal probabilities, without any interval of random sequences (Fig.3.3A, bottom). Thus, the same RC system as before could easily learn multiple chunks (Fig.3.3D). A notable difference was that, outside of the chunks, the readout activity decayed faster for undisturbed sequences than for temporally separated ones (Fig.3.3E). In fact, learning proceeded faster for the former sequences (Fig.3.3F), suggesting that learning is more effective when chunks are not disrupted by random sequences. Throughout this study, one learning step corresponds to 15 sec.

3.2.4 Selective Recruitment of Reservoir Neurons for Chunk Learning

Next, we investigated how the activities of reservoir neurons encode chunks. Here, the network was trained on sequences containing three chunks and random sequences. In each reservoir, a subset of neurons selectively responded to each chunk after learning (Fig.3.5A). Therefore, we classified reservoir neurons into three ensembles according to the selectivity of their responses to each chunk

(Methods). Some reservoir neurons responded to more than one chunk, but they were excluded from the following analysis for the sake of simplicity. Each neural ensemble received slightly stronger inputs from the specific chunk it encoded, which then determines the selectivity of the encoding ensemble (Fig.3.5B).

Through learning, the neural ensemble encoding a particular chunk developed stronger projections to the corresponding readout unit compared with other neural ensembles (Fig.3.5C). Consistent with this, the distribution of readout weights was more positively skewed in encoding ensembles than in non-encoding ensembles (Fig.3.5D). Moreover, the readout unit projected back to the corresponding encoding neuron ensemble more strongly than to the other ensembles (Fig.3.5E). Because feedback connections were not modifiable, these results imply that readout connections were strengthened between readout units and reservoir neurons that happened to receive relatively strong feedback from the readout unit.

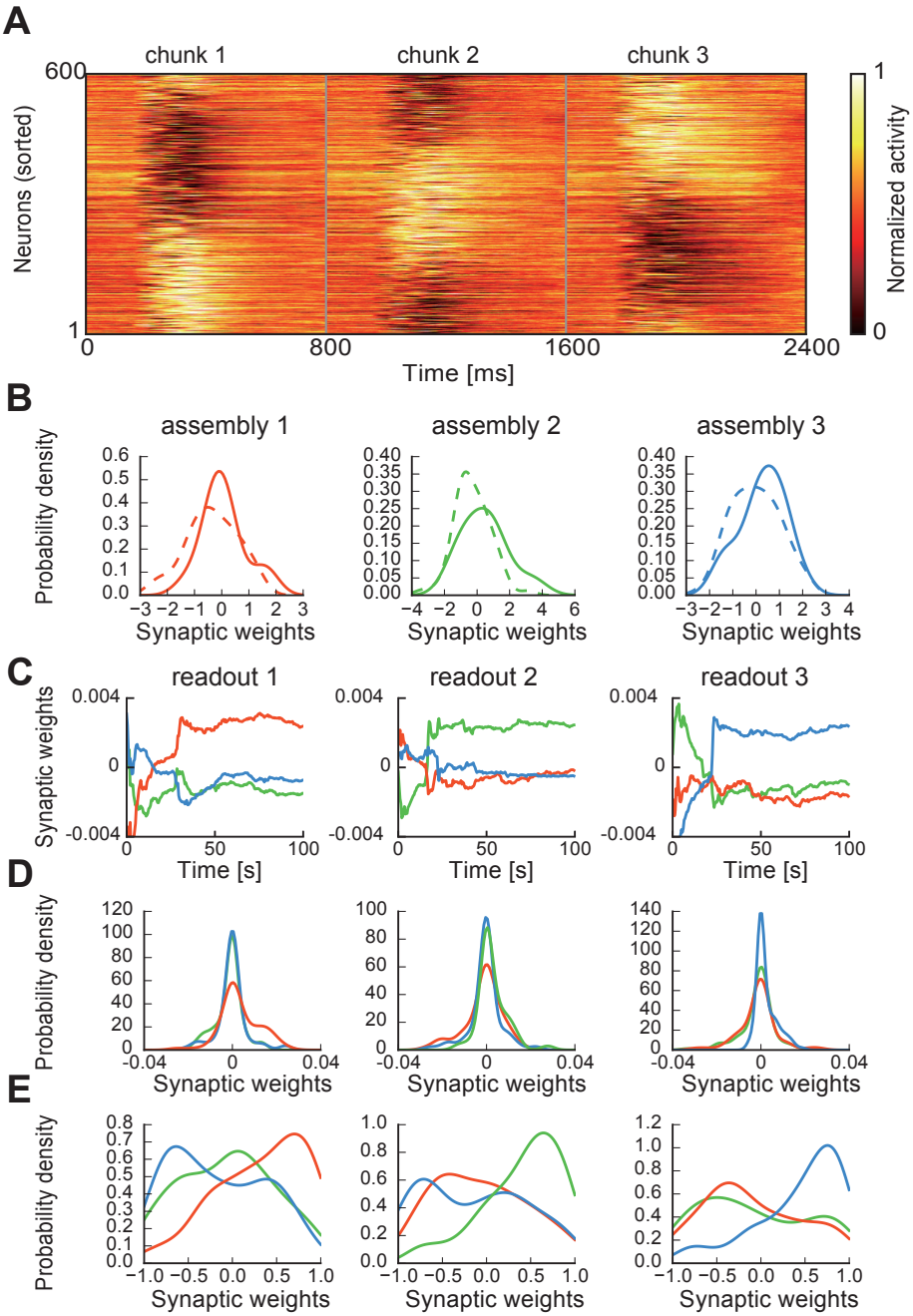

Figure 3.5: Cell assemblies selected in the reservoirs.
(A) The activity of each reservoir neuron was averaged over repeated trials and normalized by its maximum activity. Neurons were sorted according to the onset times of their activations to reveal the cell assemblies encoding the three chunks (Methods). (B) The distributions of input weights onto each cell assembly are shown for input neurons belonging to the corresponding chunk (solid) and the others (dashed). The solid and dashed distributions summed over all cell assemblies were significantly different ($p = 0.011$, t-test). (C) Temporal evolution is shown for average weights from encoding cell assemblies to the corresponding readout units. (D) Normalized distributions are shown for readout weights from each cell assembly. (E) The distribution of feedback weights from readout units to each cell assembly is shown.

3.2.5 The Role of Low-dimensional Network Dynamics in Chunk Learning

To gain further insight into the mechanism of chunking, we explored the low-dimensional characteristics of the dynamics of reservoir networks. In our model, the two RC modules, termed R1 and R2, are thought to mimic others. This would be possible when the two recurrent networks receiving the same input sequence predict the responses of other modules well. To see how this prediction is formed, we calculated the principal components (PCs) of the post-learning activity of trained recurrent networks in the example shown in Fig.3.1. After learning, the lowest principal component (PC1) but not the other PCs, of each reservoir resembled the phasic response of the corresponding readout unit during the presentation of chunks (Fig.3.6A, left). The learned trajectories wandered in the low-dimensional PC space outside the chunks where teacher signals vanished, while, inside the chunks, non-vanishing teacher signals rapidly constrained both trajectories in narrower regions showing similar PC1 values (Fig.3.6A, right). This behavior is understandable because the eigenvalues of PCs decay rapidly (Fig.3.6B). Interestingly, the correlation coefficient between each PC and the readout activity decayed more dramatically (Fig.3.6C). Accordingly, the direction of readout weight vector was more strongly correlated with that of PC1 compared to other PCs (Fig.3.6D). These results suggest that the low-dimensional characteristics of neural dynamics play a pivotal role in encoding the chunks.

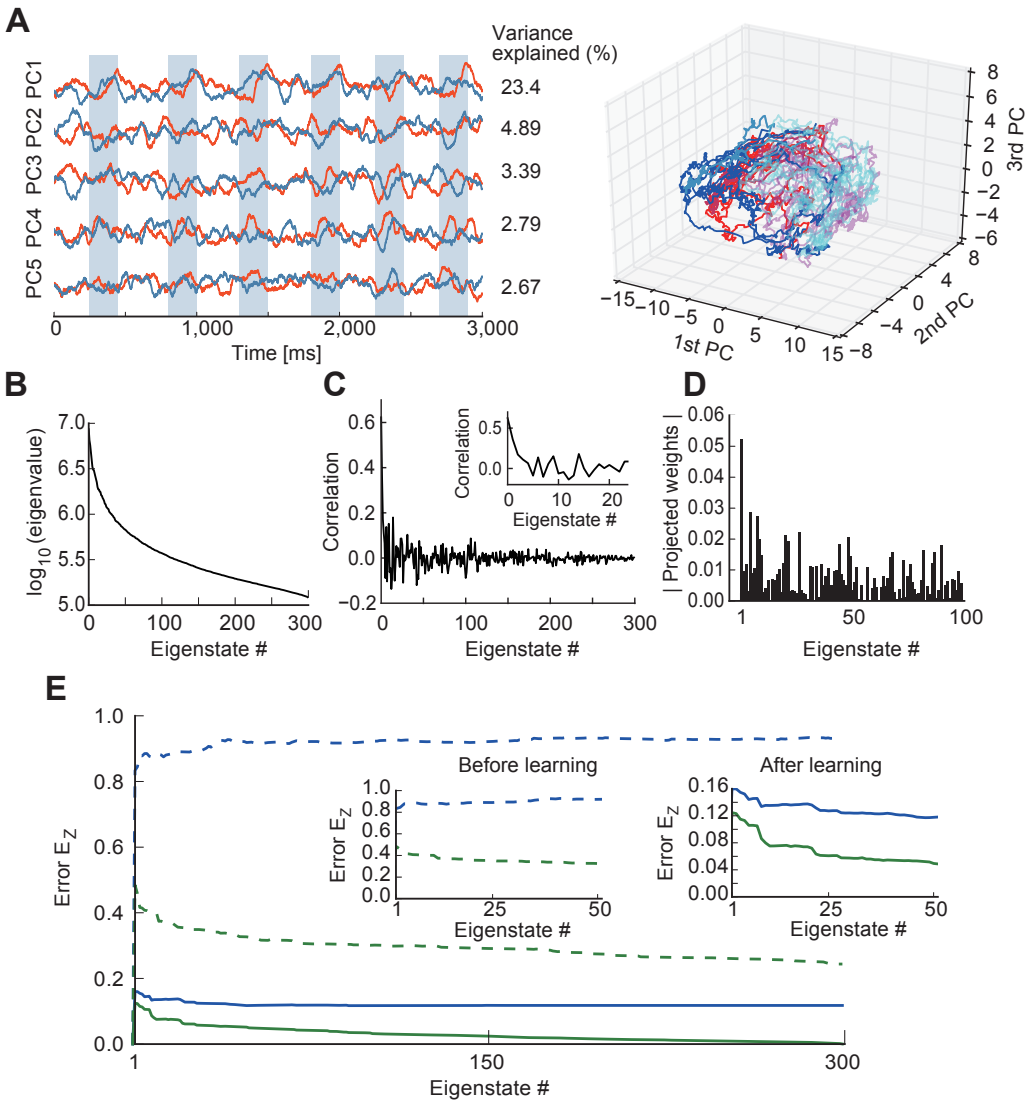

Figure 3.6: **Principal component analysis of recurrent networks.** Each recurrent network consists of 300 neurons. (A) Left, Activities of two reservoir networks are projected onto the top five eigenvectors of the correlation matrix. Shaded areas indicate the intervals of the presentation of chunks. Numerals on the right side show the variances explained. Right, The low-dimensional trajectories of the two reservoir modules are shown in the space spanned by PC1 to PC3. Red/blue or magenta/cyan portions show trajectories during the epoch of non-vanishing or vanishing teacher signals, respectively. (B) The eigenvalues of PCs are shown in a logarithmic scale. (C) The correlation coefficient between each PC and the readout activity is shown. (D) The length of readout weights projected onto each eigenvector is shown for first 100 eigenstates. (E) "Within-self" difference between the R1-output and the projected R1-output (green) and "between-partner" difference between the R2-output and the projected R1-output (blue) are shown for all the eigenstates before (dashed) and after (solid) learning. Insets display magnified versions for major eigenstates.

We then determined to what extent the responses of R1 and R2 are represented by the low-dimensional dynamical characteristics of R1. We calculated the PCs of recurrent network dynamics in R1, and expanded its population rate vector and readout weight vector up to the M-th order of these PCs (M ≦ NG). Then, we reconstructed the output of R1 by using the M-th order rate vector and the M-th order weight vector on the low-dimensional subspace spanned by the first M PCs (Methods). In Fig.3.6E, we calculated the differences between the reconstructed R1-output and the full outputs of R1 (within-self difference) and R2 (between-partner difference). Before learning, both differences remained large as M was increased. After learning, the "within-self" difference rapidly decreased for $M < 30 - 40$ and then gradually approached zero. The "between-partner" difference also rapidly dropped for relatively small values of M, but it stopped decreasing for $M > 50$, remaining at relatively large values. These results suggest that R1's reservoir, as well as R2's reservoir, learns to mimic the partner's response by using the low-dimensional characteristics of its recurrent neural dynamics.

The role of low-dimensional neural dynamics in a broad range of computation was recently explored in a class of recurrent network models with a minimal connectivity structure [74], which is a combination of a low-rank structured matrix and a random unstructured matrix. The low-rank matrix may be trained to give task-related low-dimensional dynamics whereas the random matrix may generate chaotic fluctuations useful for learning. The RC system can be approximately viewed as such a network, where the product of readout weight vector and feedback weight vector $(\mathbf{J}^{GZ})^T\mathbf{W}$ defines a rank-one matrix and recurrent connections in the reservoir gives a random matrix. It will be intriguing to study the present chunk learning in the theoretical framework.

3.2.6 Network- and Chunk-size Dependences of Learning

Chunk learning may be easier and more accurate if chunks were shorter or network size is larger. However, we did not find a sharp drop of performance when the size of chunks was increased. To observe this, we first measured learning performance for two chunks with the sizes 4 and 7 by varying the network size. Instantaneous correlations were calculated between the activity of a readout unit and a reference response pattern, which takes the value 1 during the presentation of a chunk and is 0 otherwise, every 15 s during learning and were averaged over 20 independent simulations. Note that the maximum value of the correlation was 0.5 if the readout activity grows linearly from 0 to 1 during the chunk presentation. Fig.3.7A shows the correlations for input sequences containing the short or long chunk in networks of sizes $N_G = 30, 300$, and 500. Correlations were nearly zero before learning, but reached similar maximum values approximately within ten steps of learning. The average value of the correlations was generally larger for chunk size 4 than for chunk size 7, but the differences were not significant (Fig.3.7B).

Second, we measured learning performance by varying the size of chunks with the network size fixed (N_G = 300). In this simulation, we alternately presented a single chunk with the size s and random sequences of the sizes $s + 2$ to $s + 5$, where each element of the random sequences was chosen from a set of 4s elements. Thus, the dual RC system had 5s input neurons. When the chunk size exceeded 10 (Fig.3.7C), the value of correlation rapidly dropped, suggesting the existence of a critical chunk size beyond which learning performance is degraded. For $s = 4$, learning performance showed unexpectedly large fluctuations due to some unknown reason. The explicit evaluation of the critical chunk size requires an analytic approach, which is beyond the scope of this study.

In addition, a larger network did not necessarily show better performance. The

magnitude of the post-learning instantaneous correlation was not significantly increased when the network size was 200 or greater (Fig.3.7B). Thus, the performance of chunk learning does not scale with the network size. This is not so surprising because increasing the size of the reservoirs does not necessarily increase the variety of neural responses useful for learning if the size is already sufficiently large. This seems to be particularly the case in the proposed mechanism because it heavily relies on the low-dimensional characteristics of neural dynamics (Fig.3.6A).

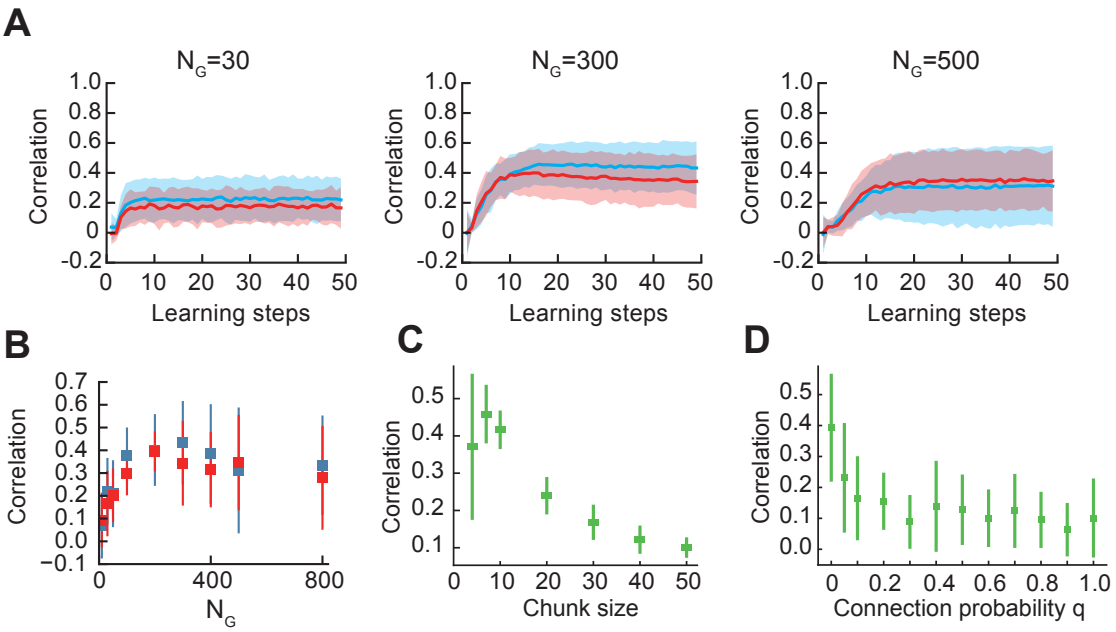

Figure 3.7: Learning with different sizes of reservoirs and chunks.
(A) The time course and performance of learning are shown for an input sequence involving a single chunk of the length 4 (blue) or 7 (red). Three networks with different sizes ($N_G s = 30, 300, 500$) were tested. (B) The values of the correlation after learning are plotted as a function of the network size. (C) The dependence of the correlation on the chunk size is shown. (D) The dependence of the correlation on the connection probability between the two reservoirs is presented.

3.2.7 Crucial Role of Noise in Chunk Learning

We found that external noise plays an active role in successful chunking. We demonstrated this in the case where the input only contained a single chunk (Fig.3.8A). In the absence of noise readout units, phasic responses were still observed, but these responses were not necessarily time-locked to chunks (Fig.3.8A, vertical arrow). As shown later, the two RC modules in principle may agree on an arbitrary feature of the input sequence, which implies the RC system may converge to a local minimum of learning. Noise may help the system to escape from the local minima. On the other hand, too strong of noise completely deteriorated the phasic responses to chunks. Thus, the RC system could learn chunks only when a modest amount of external noise existed (Fig.3.8B). In the presence of adequate noise (σ = 0.25), the average weight of the readout connections rapidly decreased to a small equilibrium value during learning (Fig.3.8C), leaving some readout weights much stronger than the majority (Fig.3.8D). This reduction was expected because external noise gives a regularization effect on synaptic weights in error-minimization learning [12]. The strong weights were obtained for readout connections from the reservoir neurons responding to the chunk, hence they were crucial for chunk detection. However, this was not the case in the absence of noise (σ = 0). We counted the fraction of strong readout connections that emerged from chunk-encoding reservoir neurons, where strong connections included those that were greater than the standard deviation of the weight distribution. Such a fraction was significantly larger in the presence of adequate noise than in the absence of noise. Under strong noise (σ= 1), although the weight distribution becomes more bimodal, the noise disrupted learning and the system failed to capture the chunks (Fig.3.8E).

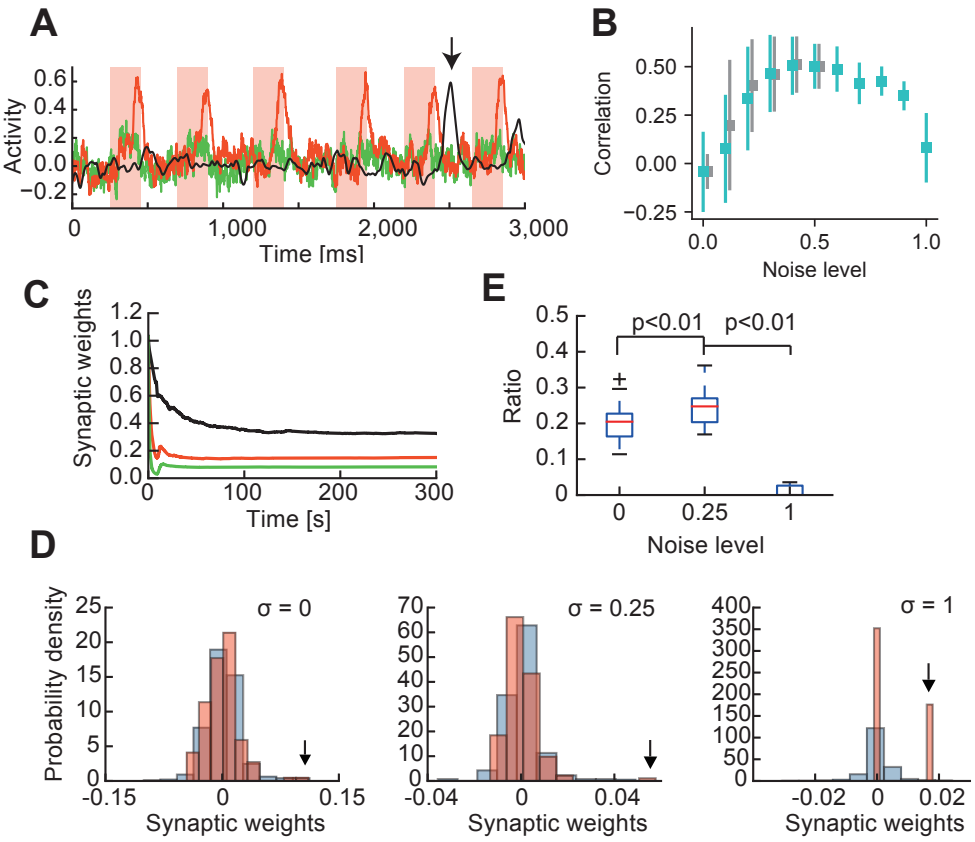

Figure 3.8: **Effects of noise on successful chunk learning.**
(A) Activity of a readout unit after learning a chunk at different noise levels: σ = 0 (black), 0.25 (red) and 1 (green). Without noise, the readout unit still learned to respond to a portion of input, but this portion did not necessarily belong to a chunk (vertical arrow). (B) Learning performance is a non-monotonic function of the noise level. The optimal performance was obtained at σ = 0.4 - 0.6 when the scaling factor in Equation 4 was set as g_G = 1.5 (cyan). The effect of noise on the learning performance was not significantly changed when the scaling factor was simultaneously reduced with the noise level (gray). (C) Evolution of the norm of readout weights during learning is shown for σ = 0 (black), 0.25 (red) and 1 (green). (D) The distributions of readout weights from chunk-encoding (red) and non-encoding (blue) reservoir neurons are shown after learning at different noise levels. Arrows indicate the maximum weight values from the chunk-encoding neurons. (E) The fraction of strong readout weights (see the main text) from the encoding neurons is shown for different noise levels. The fraction is significantly larger for σ = 0.25 compared with σ = 0 and 1 ($p < 0.01$, Mann - Whitney U test).

Another possible mechanism in which the external noise would improve the learning performance is that the dynamics of RC modules with weak noise are too far in the chaotic regime and the external noise suppresses chaos to enable proper chunk learning [101]. To test this possibility, we compensated a decrease of σ by decreasing the strength of recurrent conections g_G, which weakens the influences of chaos, and investigated whether the deterioration of performance is suppressed. The noise intensity was decreased from a modest level (σ = 0.5), and the values of σ and g_G were decreased at the same rate. Although the improvement was not significant, the dual RC system better resisted the performance deterioration (Fig.3.8B), suggesting that proper chunk learning requires a certain balance between external noise and chaotic dynamics.

Though our results so far suggest that mutual supervision enables the RC system to learn recurring groups of items in a sequence, these results do not indicate how the system chooses particular groups for learning. The question then arises whether our model detects a "chunk" if a sequence merely repeats each letter randomly without temporal grouping. To study this, we constructed a set of input sequences of ten letters, where all the letters appeared equally often in each sequence. We then exposed the RC system with a readout unit to these sequences. We found that the system learned to respond to one of the letters with approximately equal probabilities (Fig.3.9A). We then made the occurrence probability of letter "a" twice as large as the occurrence probabilities of the others and found that the system detected "a" about twice as frequent as the others (Fig.3.9B). These results indicate that the learning performance of the dual RC system relies on the occurrence frequency of repeated features if there are no other characteristic temporal features in the input sequence.

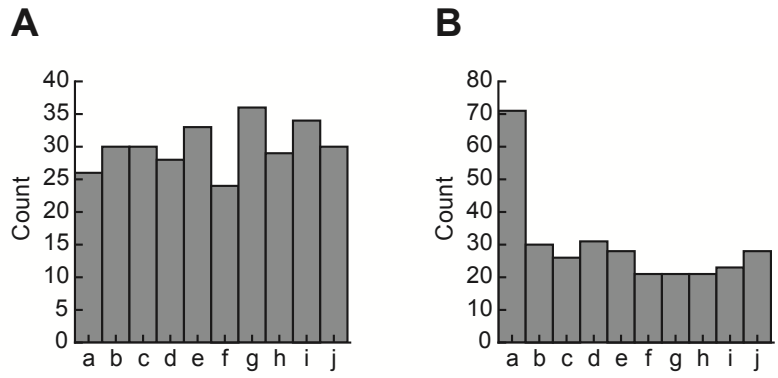

Figure 3.9: **Learning random sequences of single characters.**
The model shown in Fig.3.1 was exposed to random sequences consisting of 10 characters (a, b, ⋯, j). Input sequences had no apparent temporally grouped subsequences. (A The counts of simulation trials in which each character was learned. All characters appeared equally often in input sequences. In total, 300 trials were performed. (B) Similar trial counts were taken when character "a" appeared twice as often as others.

The frequency dependence of our model partially accounts for the features of sequences that are grouped into chunks. As demonstrated in Fig.3.6, a pair of RC modules engage in the mutual prediction of the partners' response. This prediction would be easier for the items in the input that repeatedly occur in a fixed temporal order. However, the explicit role of temporal grouping in chunking remains to be further clarified.

We then demonstrate that the RC system can simultaneously chunk multiple sequences with overlaps, where input sequences share some letters as common items. In some sequences, common subsequences appeared in the beginning or the end of chunks (Fig.3.10A), whereas other sequences involved common subsequences in the middle of chunks (Fig.3.10D). In both cases, the RC system (with two readout units) successfully chunked these input sequences without difficulty (Fig.3.10B and 10E). Interestingly, the activity of the readout units averaged over

repetitive presentations ceased to increase during the presentation of the overlapping part of the chunks (Fig.3.10C and 10F). This seems reasonable as overlapping in part does not contribute to the prediction of the following items in the chunks and hence needs not be learned.

3.2.8 Chunking Sequences of Realistic Inputs

So far, we have only studied discrete sequences of letters with varying complexity. However, the applicability of the proposed mechanism is not restricted to this relatively simple class of temporal inputs. We first showed the potential advantage of this mechanism over the conventional statistical methods, considering a system with three readout units (per RC module) for processing sequence inputs generated by a random walk through a graph (Fig.3.11A, left). This was previously used in examining the learning ability of event recognition by human subjects [97]. Each node of this graph has exactly four neighbors, and hence is visited by random walk with uniform transition probabilities over all neighbors. Despite this uniformity, the graph has three clusters of densely connected nodes, which define the communities in the graph [32, 79]. Human subjects and our model (Fig.3.11A, right) can easily chunk these clusters according to community structure, but machine-learning algorithms based on surprise signals (e.g., [84]) cannot [97].

We further demonstrated that the proposed system can learn to detect two images recurring in visual input streams with (Fig.3.11B) and without (Fig.3.11C) random intervals of Gaussian noise stimuli. We examined whether learning speed depends on the resolution of images and found that such a dependence was weak if the network size was unchanged (Fig.3.11D). Our results show the potential ability of the proposed mechanism in analyzing the community structure of a broad class of temporal inputs.

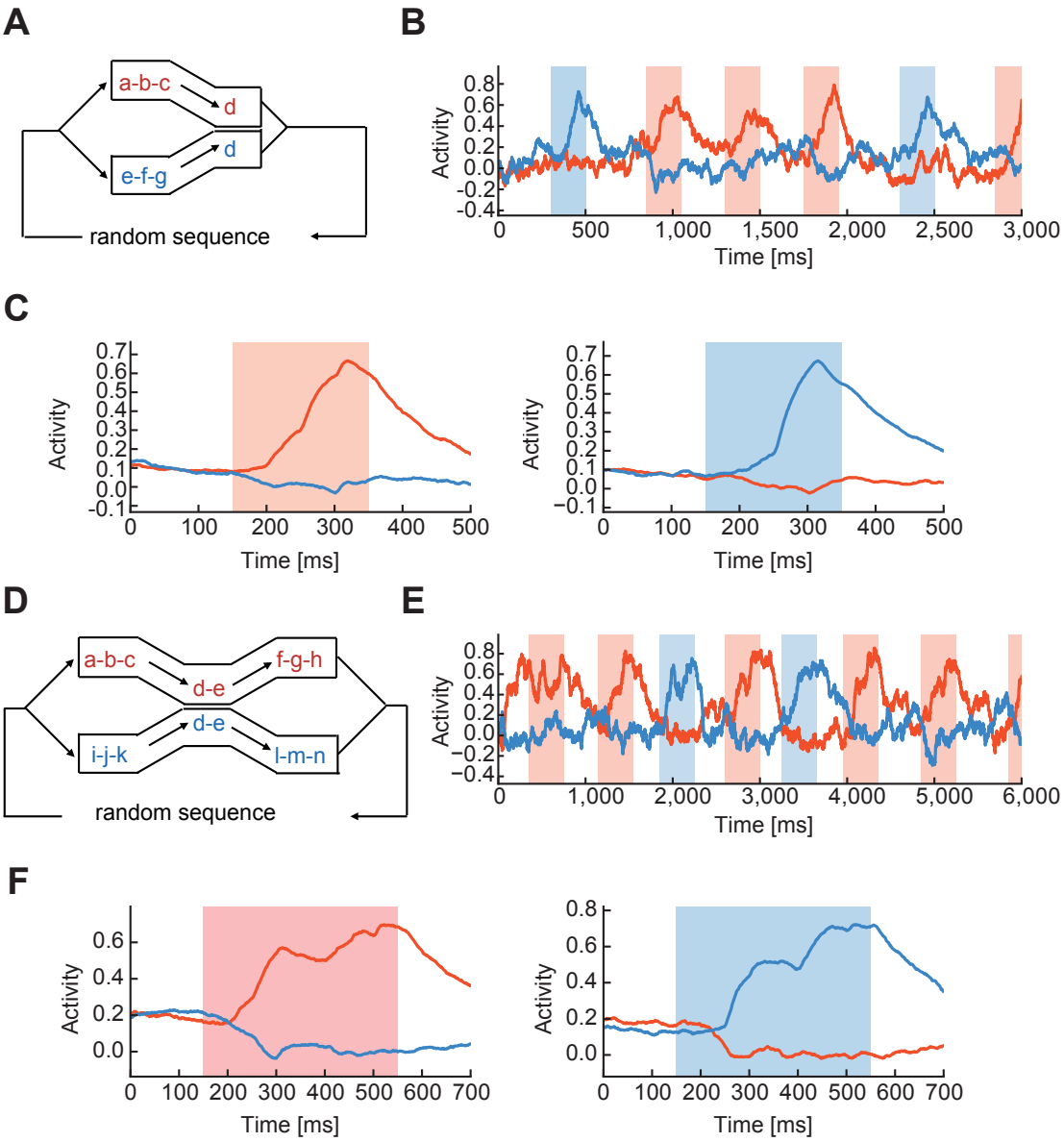

Figure 3.10: Learning chunks with mutual overlaps.
(A) Two chunks shared the last component "d" in a random input sequence. (B) Activities of two readout units were selective to different chunks after learning. (C) The average response profiles are shown for the two readout units. (D) Two chunks shared the middle components "d-e" in a random input sequence. (E) and (F), Activities of two readout units and the average response profiles are shown, respectively.

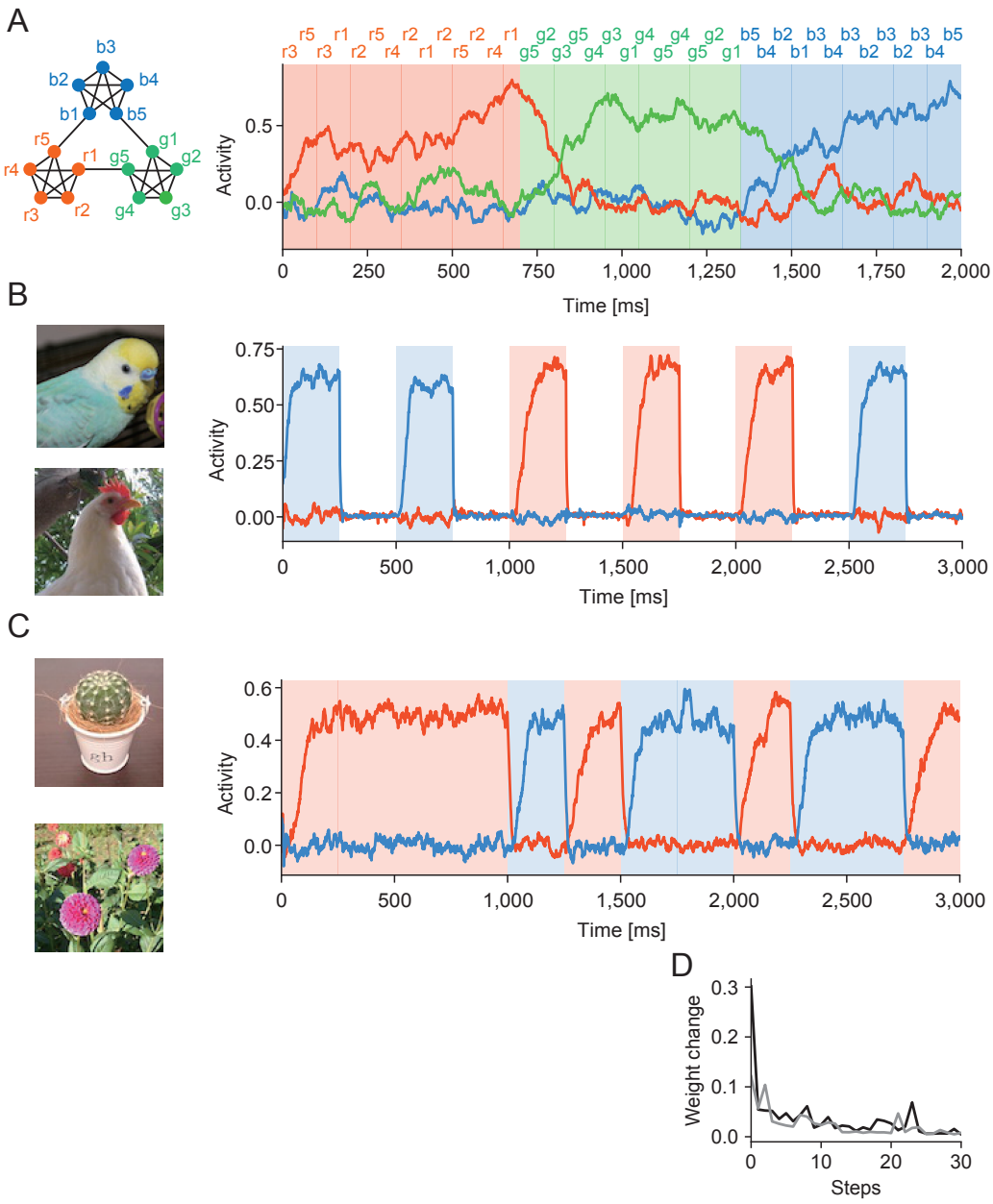

Figure 3.11: **Chunking complex temporal inputs.**
(A) Sequence inputs were generated by a graph with uniform transition probabilities and community structure. The graph was modified from [97]. (B) Sequence of high -resolution (97x97x3) visual stimuli, where the factor 3 represents the three RGB channels, was chunked. White intervals show periods of Gaussian noise. (C) Sequence of high-resolution (97x97x3) visual stimuli was chunked. (D) Learning curves are compared for the images shown in (C) between high (black) and low (gray) resolution versions. The images were repeatedly presented without noise intervals.

3.3 Discussion

Conventional statistical methods of chunking use unequal transition probabilities between sequence elements as cues for sequence segmentation. In contrast, we propose a conceptually novel framework in which the neural system self-organizes its internal dynamics to respond preferentially to chunks (i.e., frequently recurring segments) with a temporal input, rather than attempts to predict the temporal patterns of input sequences. We achieved this unsupervised learning in a network of paired RC modules mutually learning the responses of the partners. Sequence leaning with RC has been studied in motor control [27, 56, 57] and decision making [102, 107]. Theoretical extensions to spiking neuron networks [69] and/or reward-based learning [19] have thus been proposed. In this study, we showed that RC can be used for the unsupervised learning of hidden structure of continuous information streams.

Chunking has often been accounted for by predictive uncertainty or surprise [1, 5, 42, 89]. However, recent evidence suggests the existence of an alternative mechanism of chunking in which events are segmented based on the temporal community structure of sequential stimuli [97]. Indeed, it has been shown that individual items in a sequence are concatenated into an event when they frequently go together in the sequence. This dual RC system automatically chunks a continuous flow of stimuli based on temporal clustering structure and the occurrence probabilities of the stimuli without relying on predictive uncertainty or surprise. In addition, the model can chunk clusters of sequence elements that cannot be chunked by conventional statistical methods based on unequal transition probabilities (Fig.3.11A). Unsupervised chunk learning was previously modeled by using heteroclinic orbits in a dynamical neural system [31]. Though this mechanism enables the learning of prescribed chunks, whether it also offers flexible learning of arbitrary chunks remains unclear.

Our model has some advantages over the previous models of chunking. Our model can detect multiple chunks embedded into random background sequences. To our knowledge, the detection of multiple chunks has not been seriously attempted in the presence of various types of input noise on chunking. Further, as shown in Fig.3.10 our model can learn multiple partially overlapping chunks without additional mechanisms, which was also previously difficult. On the other hand, a weak point is that our model requires specially designed teaching signals, which depend on the structure of chunks. Related to this, mutually inhibitory teaching signals were introduced in an ad-hoc manner to prevent multiple readout units from learning the same chunk. A more flexible mechanism of learning should be further explored.

The dual RC system described here shows good performance in the presence of external noise. Without noise, the system also learns to respond to certain segments of a sequence, but these segments may not coincide with any of the frequently repeated chunks. An adequate amount of external noise eliminates such spurious responses and enables the system to respond to the most prominent features of a sequence, namely repeated chunks. This finding is interesting because the initial state of the dual RC system is chosen on the so-called "edge of chaos," on which weakly chaotic neural dynamics provide an adequate amount of flexibility for supervised learning [96, 106, 120]. Moreover, the present system assumes a similar initial state, but additionally requires the regularization of synaptic weight dynamics by noise (Fig.3.8C). Training a recurrent neural network with an explicit regularization term is known to eliminate the strange neuronal responses that are not observed in the motor cortex [57].

The dual RC system learns sequence in an unsupervised fashion by using two neural networks and, in this sense, is similar to Generative Adversarial Network (GAN) in deep learning [110]. A critical difference, however, exists between the two models. In GANs, a generative network learns to mimic the structure of

training data and a discriminative network learns to distinguish between samples from the training data and those generated by the generative network. Because the generative model learns to deceive the discriminative model, GANs learn the structure of data distribution under a conflicting cost function. By contrast, in the dual RC system, two neural networks learn to help each other for the formation of a consensus about the structure of temporal inputs. Therefore, our model is conceptually different from GANs.

The structure of our model has an interesting similarity to cortico-basal ganglia loops, where two reservoirs may represent bi-hemispheric cortical networks and readout units may correspond to striatal neurons. The responses of readout units and those of striatal neurons in the formation of motor habits also look similar. Sequential motor behavior becomes more rigid and automatic over the course of learning and practice, and the basal ganglia is thought to play a pivotal role in habit formation [92, 104]. For instance, in rats running in a T maze, the majority of dorsolateral striatal neurons exhibit burst firing when the run is initiated or completed, or both [37]. Similarly, in mice an increased population of striatal neurons selectively responds to the initial (Start cells), the last (Stop cells), or both actions in the trained behavioral sequence [51, 52]. In our model, readout units respond strongly to the last component of each chunk, similar to the Stop cells. Our model predicts that the Stop-cell's response may decrease when two motor chunks have overlapping portions (Fig.3.10). On the other hand, our model does not show Start cell-like responses. Whether and how Start cells are formed is an intriguing open question.

The proposed learning scheme works most efficiently when two RC modules are not interconnected, but rather work independently. In fact, the performance of chunk learning drops below 50% of the initial level when the connection probability between the two reservoirs exceeds about 10% (Fig.3.7D, see the Methods). This suggests that each RC module can obtain maximum information about tem-

poral input when it receives the teaching signal completely from its outside. The existence of inter-reservoir connections implies that some portion of the teaching signal originates from its inside. Where can such independent networks be located in the brain? One possibility is that they are represented by mutually disconnected recurrent neuronal networks in a local cortical area. Because they are functionally equivalent, it is unlikely that they are implemented in functionally distinct areas. Another intriguing possibility is that they are distributed to functionally equivalent cortical areas in different hemispheres. Indeed, the inferior frontal gyrus and the anterior insula are bilaterally activated when human subjects chunk visual information streams [13, 97]. Whether subnetworks of pyramidal cells perform chunking in these or other cortical areas [119] remains an intriguing open question.

In summary, we propose an unsupervised learning system by combining two independent reservoir computing modules. During learning, the two systems supervise each other to generate coincident outputs, which in turn allows the entire system to consistently learn chunks hidden in irregular input sequences. As chunking is a fundamental step in the analysis of sequence information, our results have significant implications for understanding how the brain models the external world.

3.4 Methods

3.4.1 Details of the Neural Network Models

In this study, the proposed model is composed of two recurrent networks (reservoirs). Each recurrent network is composed of N_G neurons. Each neuron follows

the following dynamics as $i = 1, 2, \ldots, N_G$

$$\tau \dot{x_i}(t) = -x_i(t) + g_G \sum_{j=1}^{N_G} J_{ij}^{GG} r_j(t) + J_i^{GZ} z(t) + \sum_{\mu=1}^{N_I} J_{i\mu}^{GI} I_\mu(t) + \sigma \xi_i(t), \tag{3.4}$$

$$r_i(t) = \tanh(x_i(t)), \tag{3.5}$$

where $I_\mu(t)$ is the activity of input neurons, $\xi_i(t)$ is a random (Wiener) process and σ is the standard deviation. N_I is the number of input neurons. The parameter $g_G$determines the complexity of the behavior of the reservoir, and shows chaotic spontaneous activity if $g_G > 1$. The instantaneous output is given by $z(t) = \mathbf{w}^T \mathbf{r}(t)$, where w is the readout weight vector. The readout unit is connected with n reservoir neurons by the readout weights w. The readout weights are modified according to the FORCE learning rule in which the error between the actual output and the teaching signal is minimized [106]. The activity of the readout unit is transmitted to the reservoir via the feedback.

The initial values of the readout weights w are generated by a Gaussian distribution with the mean 0 and variance $1/n$. Each element of the feedback coupling J^{Gz} is randomly sampled from a uniform distribution $[-1, +1]$. In the connection matrix J^{GG} of the reservoir, each element is taken from a Gaussian distribution with mean 0 and variance $1/(pN_G)$, where p is the connection probability of the reservoir neurons. In the connection matrix J^{GI} between input neurons and the reservoir, each row has only one non-zero element drawn from a normal distribution of mean 0 and variance 1. We simulated the model with time steps of 1 [ms].

The values of parameters used in simulations are as follows: in Figs. 3.1, 3.2, 3.6, 3.7C, 3.7D, 3.8 and 3.9, $N_G = 300, p = 1, n = 300$ and $\sigma = 0.3$; in Figs. 3.3, 3.4, 3.5, 3.9 and 3.11, $N_G = 600, p = 0.5, n = 300$ and $\sigma = 0.1$; in Fig.3.7A and Fig.3.7B, $p = 1, \sigma = 0.3$ and $n = N_G$ while the values of N_G were varied; in Fig. 3.10,

$p = 1, n = 300$, and $N_G = 800, \sigma = 0.15$(B) or $N_G = 500, \sigma = 0.1$ (E). The number of input neurons was $N_I = 26$ in all simulations except Fig.3.7C, in which $N_I$was $5s$ with s being the size of the chunk. In all simulations, $\tau = 10$ [ms] and $g_G = 1.5$. The learning rate was set as $\alpha = 100$ because larger values could cause instability in the learning process. The network was trained typically for several hundreds of seconds except in Figs. 3.3, 3.10B and 3.10E where the simulation time was 5000, 2500 and 25,000 [s], respectively.

Teaching Signals Mediated by Interneurons

In Fig.3.4D, the teaching signals were generated as

$$[\tanh((\hat{z}_{a'}(t) - \gamma \sum_{c=4,5,6}{}' y_c(t))/\beta)]_+ \ (a = 1, 2, 3) \tag{3.6}$$

where the activities of interneurons were calculated as

$$\tau \dot{y_c}(t) = -y_c(t) + \hat{z}_c(t) \tag{3.7}$$

A similar formula applied to the partner network. Note that a dash in the second term of Equation 3.6 indicates that the corresponding readout unit should be excluded from the summation (Fig.3.4C).

3.4.2 Connections Between the Reservoirs

In Fig.3.7D, the weights of recurrent connections in each reservoir module and those of connections between the modules were sampled from an identical Gaussian distribution with mean 0 and variance $1/\{(1+q)N_G\}$, where q is the connection probability of inter-module connections. The recurrent connections were all-to-all. The value 1 in the denominator was introduced such that the limit $q \to 0$ gives the disconnected RC modules studied in other panels in Fig.3.7.

Normalized Output for Teaching Signals

In our learning rule, we changed the outputs of readout units such that the mean outputs coincide with zero and the standard deviation becomes unity:

$$z(t) \to \hat{z}(t) = (z(t) - \mu(t))/\sigma(t), \tag{3.8}$$

where $\mu(t)$ and $\sigma(t)$ were calculated as

$$\mu(t) = \frac{1}{T}\int_{t-T}^{t} z(t')dt', \tag{3.9}$$

$$\sigma(t) = \sqrt{\frac{1}{T}\int_{t-T}^{t} z(t')^2 dt' - \mu(t)^2}, \tag{3.10}$$

with a sufficiently long period $T(= 15$ [s]). The modified output $\hat{z}(t)$ was then transformed by two nonlinear functions to generate the teaching signal shown in the Results.

3.4.3 Selectivity of Reservoir Neurons

In Fig.3.5A, the activities of all reservoir neurons were first averaged and then normalized. To define the response selectivity of neurons, we sorted all of the neurons by their mean activation phases defined as,

$$\hat{t_i} = \frac{T}{\pi}\arg\left[\frac{\sum_{t'=1}^{T}\bar{r_i}(t')\exp\left(i\frac{2\pi t'}{T}\right)}{\sum_{t'=1}^{T}\bar{r_i}(t')}\right] \text{[ms]},$$

where $\bar{r}(t)$ is the normalized average response of each cell and $T = 2400$[ms]. Each reservoir neuron generally showed a significantly large and prolonged phasic response to a particular chunk, which determined the selectivity of the reservoir neuron. We defined a phasic response as such transient activity that exceeded the threshold value $\mu + 3\sigma$ for more than 100 [ms], where μ and σ stand for the

average and standard deviation of its activity during the presentation of input sequence. Neurons that were not related to any chunks or responded to multiple chunks were discarded in the analysis.

3.4.4 Analysis of the Low-dimensional Dynamics of Reservoirs

In Fig. 3.6, we projected the neural responses $\mathbf{r}_{R1}(t)$ of recurrent network in R1 onto the $M(\leq NG)$ dimensional subspace:

$$\mathbf{r}_{R1,M}(t) = \mathbf{V}_M^T \mathbf{r}_{R1}(t). \tag{3.11}$$

Here, the $(N_G \times M)$-dimensional matrix $\mathbf{V}_M$ is defined as $\mathbf{V}_M = (\boldsymbol{\phi}_1^{(R1)} \boldsymbol{\phi}_2^{(R1)} \dots \boldsymbol{\phi}_M^{(R1)})$ in terms of the λ-th eigenvector of R1 reservoir $\boldsymbol{\phi}_\lambda^{(R1)}$. Similarly, we projected the readout weight vectors from R1 onto the same subspace as

$$\mathbf{w}_{R1,M} = \mathbf{V}_M^T \mathbf{w}_{R1}. \tag{3.12}$$

We then calculated the difference between the actual output of R1 and the output reconstructed on the subspace as

$$E_z = \sqrt{\frac{1}{T} \int_0^T |z_{R1}(t) - \mathbf{w}_{R1,M}^T \mathbf{r}_{R1,M}(t)|^2 dt}. \tag{3.13}$$

The difference between the actual output of R2 and the projected R1-output was calculated in a similar fashion.

3.4.5 Simulations of Visual Information Streams

In Fig. 3.11B and 3.11C, we constructed a pair of RC modules each having two readout units. A stream of two images with high (97x97 pixels x 3 RGB channels) or low (32x32 pixels x 3 RGB channels) resolutions was used as input, in which

the presentations of two images (and Gaussian noise images in Fig. 3.11B) were randomly switched at every 250 ms. Each reservoir neuron received input from randomly chosen 10% of pixels. In Fig. 3.11D, the low-resolution versions of the images used in Fig. 3.11C were created at the reduced size of 32 x32 pixels (x 3 RGB channels).

Chapter 4

Temporal Feature Learning by Somatodendritic Mismatch Detection

The mutually supervising reservoir network model proposed in Chapter 3 successfully learning chunks from sequences without any teaching signals from outside. However, the previous model is not biologically plausible in at least two respects. First, the number of outputs in each reservoir must be the same as the number of chunks in the sequences. This means that it is necessary to know the number of features prior to learning. Second, as shown in the previous chapter, learning fails as the number of connections between circuits increases. This indicates that the two circuits need to be sufficiently independent, but a completely independent structure in the real neural circuit is unnatural. To overcome these disadvantages, in this chapter, I will propose novel neuron model, which consists of two compartments: dendrites and soma. In this model, dendrites change synaptic connections so that they learn a statistical model of somatic activity. We will show that this model can perform not only chunking, but also a wide range of sequence learning,

including blind source separation of correlated signals.

4.1 Introduction

Cognitive functions of the brain entail modeling of externally or internally driven dynamic processes. For this modeling, the brain has to identify the salient temporal features of continuous information streams. How the brain conducts this time-series analysis remains unknown, but the component processes necessary for the analysis are partly known. The process by which frequently recurring segments of temporal sequences are concatenated into single units that are easy to process is called "chunking" or "bracketing" [26]. Chunking underlies sensory scene analyses, motor learning, episodic memory and language processing [34, 51, 52, 97, 119]. In predictive coding [6, 33, 54], the brain may chunk bottom-up and top-down information flows to identify variables relevant to the hierarchical Bayesian modeling of mental processes. Another important class of temporal feature analysis is blind source separation (BSS: the so-called cocktail party effect) in which the brain separates mixed sensory (typically auditory) signals from multiple sources for recognition [75]. Despite of the fundamental importance, however, the methods by which neural circuits in the brain analyze and learn temporal features remain largely unclear. Whether the different temporal feature analyses require specialized network architectures and learning rules is also unknown.

In this study, we provide a conceptually novel solution to these fundamental problems of the brain computing. We show in a two-compartment neuron model that the minimization of information loss between dendritic synaptic input and neuron's own output spike trains enables efficient learning of clustered temporal events in a completely unsupervised manner. This learning proceeds intracellularly and can be viewed as a self-supervising process in which a single neuron (more precisely, the soma) generates an appropriate supervising signal to learn

the spatiotemporal firing patterns repeated in upstream neurons (projecting to the dendrite). The resultant learning rule conceptually resembles Hebbian learning with backpropagating action potentials, which crucially contribute to synaptic plasticity in cortical neurons [59, 62, 67, 70, 103]. Furthermore, our model predicts that the gain and threshold of somatic responses should be modulated in an activity history-dependent manner.

To our surprise, a family of competitive networks of the proposed neuron model can perform a variety of unsupervised learning tasks ranging from chunking to BSS, which were previously performed by specialized networks and learning rules. Members of this family have the same network architecture but different network parameters (e.g., synaptic weights). We emphasize that some chunking tasks solvable with our model (and also by humans) are difficult for the conventional machine learning methods due to uniform transition probabilities between-consecutive items [97]. Furthermore, the same network model successfully separates the mixed signals of highly correlated sources, namely, musical instruments playing their own parts of the same note. BSS has been extensively studied in machine learning [3, 24, 46, 53], but how the brain solves this problem has not been fully understood. Our results suggest the computational principles that underlie the wide range of subconscious temporal feature analyses by cortical networks and the active role of dendrites in these processes.

4.2 Results

4.2.1 Neural implementation of minimization of regularized information losss

Our model entails the learning of temporal features of an input based on a novel learning rule, which we term “minimization of regularized information loss (MRIL).”

Suppose that the dendrite attempts to predict the responses of soma. In short, MRIL achieves this by minimizing the information loss (within a certain recent period) when the somatic activity is replaced with its model, that is, the dendritic activity driven by given synaptic inputs. The loss can be easily minimized if the somatic responses are well predictable. This will be the case when the neuron learns to selectively respond to temporal patterns recurring in synaptic input. Fig.4.1 schematically illustrates the present learning rule in a two-compartment spiking neuron model. Mathematically, MRIL minimizes the Kullback - Leibler (KL) divergence between the probability distributions of somatic and dendritic activities. Note that in the resultant learning rule the somatic response is fed back to the dendrite to train dendritic synapses. Although the underlying biological mechanisms are not modeled here, backpropagating action potentials may provide the feedback signal in cortical pyramidal neurons [58].

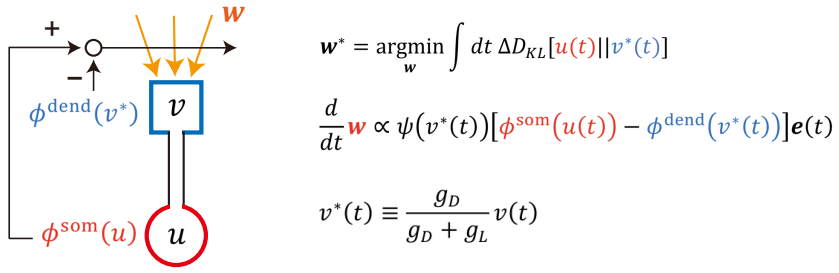

Figure 4.1: **Two-compartment neuron model for MRIL.**
The model neuron consists of somatic and dendritic compartments and undergoes MRIL learning. The dendritic component receives Poisson spike trains, and the somatic membrane potential is given as an attenuated version of the dendritic membrane potential. Output of the soma backpropagates to dendritic synapses as a self-teaching signal. Learning stops when the dendrite minimizes the error between its prediction and the actual somatic firing rate.

The division of labor between the soma and dendrite was previously modeled with a teaching signal given explicitly or implicitly to the soma [111]. Unlike the previous model, our model modulates the gain and threshold of somatic responses according to the recent history of somatic responses. These modulations enable the model to avoid a trivial solution to the learning rule, and therefore ensure successful learning of nontrivial temporal features. Differences between the present and previous models will be further discussed later. Before going to the detail of results, I will explain the details of consepts of our learning mdel.

4.2.2 Neural Network Model

Each output neuron has two compartments, i.e., somatic and dendritic compartments. The dendritic membrane potential of output neuron $i \in 1, 2, ..., N_{out}$ is calculated as

$$v_i(t) = \sum_j w_{ij} e_j(t), \tag{4.1}$$

where w_{ij} is the weight of synapse between output neuron i and input neuron j. The somatic activity integrates the dendritic potential, and it evolves as

$$\dot{u}_i(t) = -g_L u_i(t) + g_D[-u_i(t) + v_i(t)] - \sum_j G_{ij} \phi_i^{som}(u_j(t)), \tag{4.2}$$

where $\tau = 15$ ms and the conductance between the two compartments $g_D = 0.7$. The last term describes lateral inhibition with synaptic weights $G_{ij} (\geq 0)$. We calculated the inhibitory input in terms of the firing rates of output neurons. However, as explained below, spike trains of these neurons were also generated for simulating modifications of G_{ij} by spike-timing-dependent plasticity. We assume that the soma of neuron i generates a Poisson spike train with the instantaneous

firing rate $\phi_i^{som}(u_i(t))$ in terms of the nonlinear response function

$$\phi_i^{som}(u_i) = \phi_0[1 + \exp(\beta_i(-u_i + \theta_i))]^{-1}. \tag{4.3}$$

The parameters β_i and θ_i are defined as follows:

$$\beta_i = \sigma_i(t)^{-1}\beta_0, \tag{4.4}$$

$$\theta_i = \mu_i(t) + \sigma_i(t)\theta_0, \tag{4.5}$$

where $\mu_i(t)$ and $\sigma_i(t)$ are the mean and variance of the membrane potential over a sufficiently long period t_0:

$$\mu_i(t) = \frac{1}{t_0}\int_{t-t_0}^{t} u_i(t')dt', \tag{4.6}$$

$$\sigma_i(t) = \sqrt{\frac{1}{t_0}\int_{t-t_0}^{t} u_i(t')^2 dt' - \mu_i(t)^2}. \tag{4.7}$$

We set $\beta_0 = 5$ throughout this study, but the values of ϕ_0 and θ_0 are task-depend and described later. We note that the slope of nonlinearity β_i and the threshold value θ_i are modified as the values of $\mu_i(t)$ and $\sigma_i(t)$ change during learning. As described later, these online modifications of the somatic response function maintain the dynamic range of output firing rate within a certain range, preventing synaptic weights from converging to a trivial solution through learning.

In our model, sensory information given to the network is first encoded into Poisson spike trains. Input neuron $i \in 1, 2, ..., N_{in}$ generates a Poisson spike train

$$X_i(t) = \sum_f \delta(t - t_{i,q}), \tag{4.8}$$

where δ is the Dirac' delta function and $t_{i,q}$ denotes the time of the q-th spike of

input neuron i. The presynaptic spikes induce the following synaptic current $I_i(t)$:

$$\tau_{syn}\dot{I}_i = -I_i + \frac{1}{\tau}X_i, \tag{4.9}$$

where the synaptic time constant $\tau_{syn} = 5$ ms ($\tau_{syn} = 50$ ms in Fig.4.7G and Fig.4.8C). The synaptic currents in turn evoke postsynaptic potential (PSP) $e_i(t)$ as

$$\dot{e}_i = -\frac{e_i}{\tau} + I_i. \tag{4.10}$$

4.2.3 Optimal learning rule for minimization of regularized information loss (MRIL)

To extract the characteristic features of temporal input, our model compresses the high dimensional data carried by the input sequence onto a low dimensional manifold of neural dynamics. The model performs this by modifying the weights of dendritic synapses to minimize the time-averaged mismatch between the somatic and dendritic activities over a certain interval $[0, T]$. We assumed that an attenuated version of dendritic potential $v_i^*(t)$ well describes the somatic membrane potential as $u_i(t) \approx v_i^*(t) = \alpha v_i(t)$ with the degree of attenuation $\alpha = g_D/(g_D + g_L)$ [111]. Explicitly representing the dependency of u_i and v_i^* on $\mathbf{X}$, we define the cost function for synaptic weights $\mathbf{w}$ as

$$E(\mathbf{w}) = \int_{\Omega_\mathbf{X}} d\mathbf{X} \int_0^T dt P^*(\mathbf{X}) \sum_i D_{KL}\left[\phi_i^{som}(u_i(t;\mathbf{X}))\middle|\middle|\phi^{dend}(v_i^*(t;\mathbf{X}))\right] \tag{4.11}$$

where $P^*(\mathbf{X})$ stands for the true distribution of input spike trains, $\Omega_\mathbf{X}$ for the space spanned by all possible combinations of input spike trains, and D_{KL} for the KL-divergence between the two Poisson distributions:

$$\begin{aligned} D_{KL}\left[\phi_i^{som}(u_i(t;\mathbf{X}))\middle|\middle|\phi^{dend}(v_i^*(t;\mathbf{X}))\right] \equiv \phi_i^{som}(u_i(t;\mathbf{X}))\log\frac{\phi_i^{som}(u_i(t;\mathbf{X}))}{\phi^{dend}(v_i^*(t;\mathbf{X}))} \\ +\phi^{dend}(v_i^*(t;\mathbf{X})) - \phi_i^{som}(u_i(t;\mathbf{X})) \end{aligned} \tag{4.12}$$

with $\phi^{dend}(x) = \phi_0[1 + \exp(\beta_0(-x + \theta_0))]^{-1}$.

We minimize the cost function (i.e., the averaged KL-divergence) with respect to $\mathbf{w}$ such that the responses of the two compartments become consistent with each other. Thus, the unsupervised learning rule of "somatodendritic mismatch detection" resolves the mismatch between the somatic and dendritic responses to temporal input. We search the optimal weight matrix by the gradient descent as

$$\begin{aligned}
\Delta w_{ij} &\propto -\frac{\partial}{\partial w_{ij}} E \qquad (4.13)\\
&= -\frac{\partial}{\partial w_{ij}} \int_{\Omega_{\mathbf{X}}} d\mathbf{X} \int_0^T dt P^*(\mathbf{X}) \sum_{i'} D_{KL}\left[\phi_{i'}^{som}(u_{i'}(t;\mathbf{X}))\middle\|\phi^{dend}(v_{i'}^*(t;\mathbf{X}))\right]\\
&= -\int_{\Omega_{\mathbf{X}}} d\mathbf{X} \int_0^T dt P^*(\mathbf{X}) \frac{\partial}{\partial w_{ij}} \left[\phi_i^{som}(u_i(t;\mathbf{X})) \log \frac{\phi_i^{som}(u_i(t;\mathbf{X}))}{\phi^{dend}(v_i^*(t;\mathbf{X}))}\right.\\
&\quad \left. + \phi^{dend}(v_i^*(t;\mathbf{X})) - \phi_i^{som}(u_i(t;\mathbf{X}))\right] \qquad (4.14)\\
&= \int_{\Omega_{\mathbf{X}}} d\mathbf{X} \int_0^T dt P^*(\mathbf{X}) \frac{\partial \log(\phi^{dend}(v_i^*(t;\mathbf{X})))}{\partial w_{ij}} \left[\phi_i^{som}(u_i(t;\mathbf{X})) - \phi^{dend}(v_i^*(t;\mathbf{X}))\right].
\end{aligned}$$

Note that the identity $d\phi^{dend}(x)/dx = \phi^{dend}(x) d\log\phi^{dend}(x)/dx$ was used in deriving the last expression. Since $v_i^*(t) = \alpha \sum_j w_{ij} e_j(t)$, the local learning rule is written in a vector form as

$$\Delta \mathbf{w}_i \propto \int_{\Omega_{\mathbf{X}}} d\mathbf{X} \int_0^T dt P^*(\mathbf{X}) \psi(v_i^*(t;\mathbf{X})) \left[\phi_i^{som}(u_i(t;\mathbf{X})) - \phi^{dend}(v_i^*(t;\mathbf{X}))\right] \mathbf{e}(t;\mathbf{X}) \qquad (4.15)$$

where $\mathbf{w}_i = [w_{i1}, \cdots, w_{iN_{in}}]$ and the function $\psi(x)$ is defined as

$$\psi(x) = \frac{d}{dx} \log(\phi^{dend}(x)). \qquad (4.16)$$

Note that the i-dependence of $\Delta\mathbf{w}_i$ arises in our network model from activity-dependent modifications of recurrent inhibitory connections among output neurons (see equation 4.2). The inhibitory connections are modifiable by STDP (see Fig.4.3B).

In all simulations, we added the regularization term $-\gamma \mathbf{w}_i$to equation (4.15)

to prevent the diverging growth of synaptic weights. Thus, the following online learning rule was used:

$$\dot{\mathbf{w}}_i(t) = \eta \left\{ \psi(v_i^*(t)) \left[\phi^{som}(u_i(t)) - \phi^{dend}(v_i^*(t)) \right] e_j(t) - \gamma \mathbf{w}_i \right\}, \tag{4.17}$$

where η is the learning rate. The parameter γ controls the strength of regularization and was adjusted in a task-dependent manner. The initial values of w were generated by a Gaussian distribution with mean zero and standard deviation $1/\sqrt{N_{in}}$. Note that the above learning rule coincides with the Bienenstock-Cooper-Munro (BCM) theory except a sign difference [11]. In BCM theory the threshold between potentiation and depression is an unstable fixed point, while in our model this point is a stable fixed point. The online modifications shown in equations (4.3)-(4.5) avoids a trivial fixed point.

A similar learning rule was previously considered in a supervised learning model in which the average "surprise" of somatic spike output driven by dendritic synaptic input and a teaching signal given to the soma was minimized [111]. In the framework, our learning rule may be interpreted as "self-consistent surprise minimization" in which the teaching signal itself is provided by the somatic response to make the learning rule for dendritic neurons unsupervised. This summarizes the essential difference between our model and the previous model.

4.2.4 Inhibitory plasticity

We modified lateral inhibitory connections through a symmetric anti-Hebbian spike-timing-dependent plasticity (STDP): If a pair of presynaptic and postsynaptic spikes occur at the times t_{pre} and t_{post}, respectively, weight changes were calculated as

$$\Delta G_{ij} = C_p \exp\left(-\frac{|t_{pre} - t_{post}|}{\tau_p} \right) - C_d \exp\left(-\frac{|t_{pre} - t_{post}|}{\tau_d} \right), \tag{4.18}$$

where τ_p and τ_d are the decay time constants of LTP and LTD, respectively. Typically, we used $\tau_p = 40$ ms, $\tau_d = 20$ ms, $C_p = 0.00525$ and $C_d = 0.0105$. Inhibitory weights G_{ij} were modified between zero and an upper bound $G_{max}(\propto 1/\sqrt{N_{out}})$

Our learning rule (Eq. 4.17) looks similar to the maximum likelihood estimation [85], a well-studied framework of supervised learning. However, there is a conceptual difference between them. In the maximum likelihood estimation, the target data distribution (somatic activity) is provided externally as teaching signals. By contrast, our model simultaneously learns the probability distributions of input and output data without teaching signals. The consistency between the two data sets constrains the self-supervised learning, thereby avoiding an overly redundant or an overly simplistic categorization of temporal inputs. We emphasize that MRIL fits particularly well with dendritic neurons, but the principle is generic and applicable to a broad range of information processing systems.

4.2.5 Learning patterned temporal inputs in single neurons

We first demonstrate that the dendritic neuron model detects the salient temporal features recurring in synaptic input. Learning to detect and discriminate repeated temporal input patterns is crucial for various cognitive functions such as language acquisition [15, 36] and motor sequence learning [34, 51, 52, 104]. In Fig.4.2A, presynaptic spike trains intermittently repeated three fixed spatiotemporal patterns with equal probabilities of occurrence. These patterns may be regarded as chunks. As learning of temporal input proceeds through the mismatch check between the soma and dendrite, a single neuron gradually learned to respond selectively to an input pattern (Fig.4.2B, C). The neuron learned one of the input patterns with approximately equal probabilities among the trials, although it responded to more than one input pattern in some trials (Fig.4.2D). We note that all presynaptic neurons had the same average firing rates that were constant dur-

ing the entire task period (Methods). Therefore, the discrimination does not rely on differences in firing rate. Cortical neurons are actually capable of discriminating temporal inputs and generating sequence-selective spike output, although the synaptic sequences tested in the experiment were relatively simple [14].

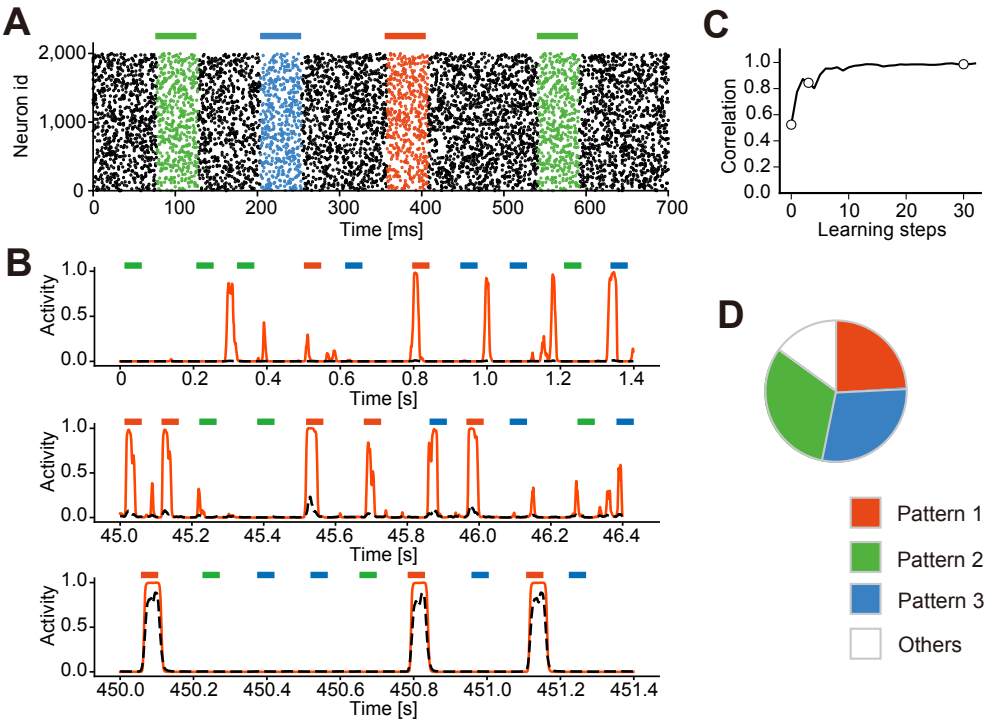

Figure 4.2: **Unsupervised learning in two-compartment neurons.**
(A) Three frozen spatiotemporal patterns (red, blue, and green) were repeated in irregular spike trains from 2,000 input neurons. (B) A dendritic neuron selectively learned one of the recurring patterns. Examples of the somatic (red) and dendritic (black dashed lines) activities are shown at the initial (top), middle (middle) and final (bottom) stages of learning. (C) Leaning curve is shown, with circles indicating the time points at which the examples were drawn. Instantaneous correlations were calculated between the activities of dendrite and soma, every 15 s during learning. (D) The fraction of trials in which a single neuron model learned a selective response to one of the three repeated spike patterns is shown. The number of trials was 100. "Others" indicates trials in which the neuron had more than one preferred pattern, i.e., the peak responses to the second preferred pattern was greater than 50% of those to the most preferred pattern.

4.2.6 Automatic chunking with MRIL and inhibitory STDP

Next, we considered a competitive network of the two-compartment model neurons receiving similar presynaptic spike trains (Fig.4.3A). To study whether chunk-specific cell assemblies can be formed, we made recurrent inhibitory connections among these neurons modifiable by inhibitory spike timing-dependent plasticity (iSTDP; Fig.4.3B). This rule weakens inhibition between two dendritic neurons when both of them respond to the same temporal feature. The use of this plasticity rule for lateral inhibition is realistic because this type of STDP has been found at cortical excitatory synapses on inhibitory interneurons [64] and at inhibitory synapses in the hippocampus [116]. In either case, inhibitory circuits will exhibit the desired changes. During learning, each neuron gradually increased coherence between the somatic and dendritic activities (Fig.4.3C). The postsynaptic neurons self-organized into three neuron ensembles, each detecting one of the input activity patterns (Fig.4.3D), and iSTDP enabled mutual inhibition between the neural ensembles (Fig.4.3E). The strength of lateral inhibition, however, needs to be within an appropriate range, as too strong (Fig.4.4A) or too weak (Fig.4.4B) inhibition failed to generate chunk-specific cell assemblies. The regularization parameter γ (see Methods) also has to be in an appropriate range to enable the unsupervised learning of chunk-specific cell assemblies, as values that were too large suppressed all neural responses and those that were too small did not generate selective responses to chunks (Fig.4.4C).

The ability of the network model to learn recurring input patterns was assessed with various types of biological noise. Background presynaptic spikes degraded the performance as the signal-to-noise ratio decreased (Fig.4.3F), whereas learning was optimal at finite noise levels with synaptic transmission failure (Fig.4.3G) and with jitters in presynaptic spike timing (Fig.4.3H). We speculate that this disparity may reflect the different noise structures. Background spikes were uncorrelated with the recurring input patterns and merely contaminated the signals, whereas

transmission failures and timing jitters yielded the noise patterns that were correlated with the input and thus enhanced the sampling for learning. Presynaptic noise may also induce a regularization effect during learning [12]. However, this effect was unlikely to be prominent in our model, as not all types of presynaptic noise improved the learning.

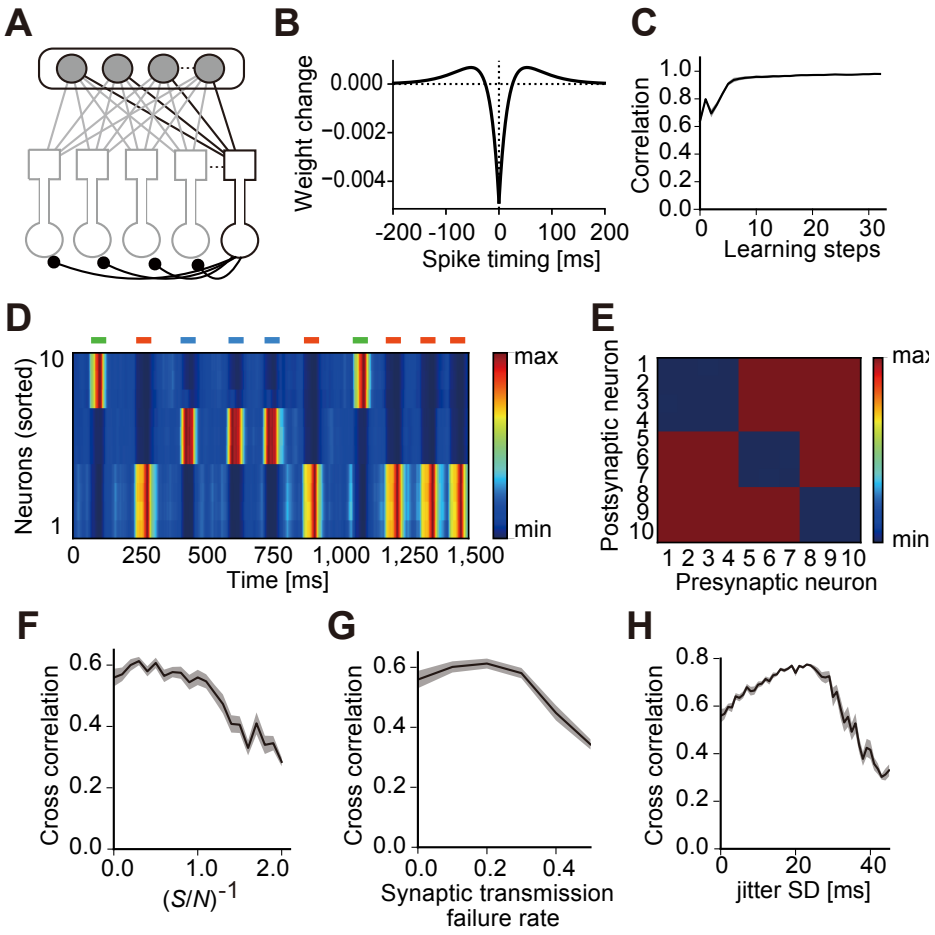

Figure 4.3: **Formation of temporal feature-specific cell assemblies.**
(A) A competitive network of dendritic neurons was used throughout this study. The input layer consists of Poisson spiking neurons and the output layer comprises the dendritic neuron models. In this particular example, input neurons received presynaptic spikes trains similar to those shown in Fig.4.2A. (B) Window function of the iSTDP implemented at lateral inhibitory connections. Here, spike timing refers to the time advance of postsynaptic firing from the preceding presynaptic input. (C) The average correlation between the somatic and dendritic activities is plotted against learning step. Shaded area represents the S.D. (D) Phasic responses of output neurons are shown. Horizontal bars show the intervals in which three chunks (green, red, blue) were presented. The responses sorted according to the neurons' onset response times indicate the emergence of chunk-specific cell assemblies. (E) Post-learning synaptic weight matrix of lateral inhibition is shown. The correlations between reference responses and actual output responses were evaluated in the presence of (F) contamination by background presynaptic spikes, (G) failure in synaptic transmissions and (H) timing jitters in the target spiking patterns. Each reference response takes unity during the presentation of the corresponding chunk and zero otherwise. The ordinates refer to the inverse of the number ratio of background spikes to target-specific spikes in (F) and the s.d. of spike timing jitters in (H). The mean (thick line) and s.d. (shaded area) over 20 trials are shown. The correlations are shown for the maximally correlated pairs of cell assemblies and chunks (i.e., preferred chunks).

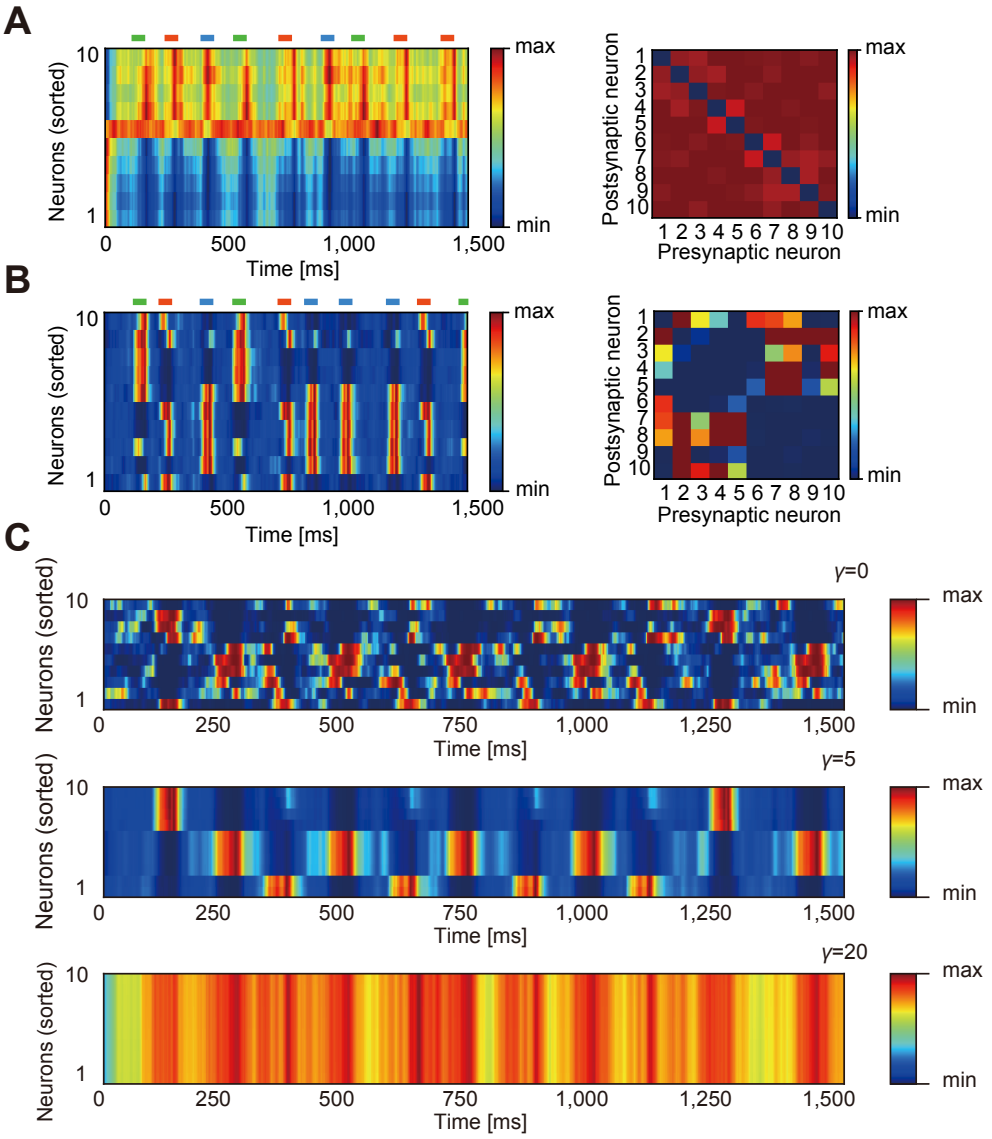

Figure 4.4: **Roles of inhibitory STDP and regularization parameter.** Here, simulations were performed with irregular spike trains of 200 input neurons. The other settings were the same as in Figure 4.3. (A) Post-learning responses of output neurons are shown for strong lateral inhibition (left). About half of the neurons responded to all three chunks without stimulus selectivity, but the others showed almost no responses. Synaptic weight matrix developed no clustering structures (right). (B) Post-learning responses are shown for weak lateral inhibition (left). The majority of the neurons had more than one preferred stimulus. Synaptic weight matrix developed overlapping clustering structures (right). (C) The strength of the regularization term was changed in Equation 4.17 in the task shown in Figure 4.3. Too weak regularization term impaired the self-organization of selective responses to chunks (top), a moderate range of the regularization term resulted in successful learning (middle), and too strong regularization term prohibited the learning of features (bottom).

The above results may account for the following perceptual ability of humans. It has been shown that human subjects can detect the recurrence of frozen noise patterns embedded in a noisy auditory signal [2]. As in Fig. 4.2A, both repeated and background auditory signals may be represented by irregular synaptic inputs to the auditory cortex. However, the subjects learned the noise without extensive learning, indicating the possibility that the learning mechanisms might differ from the method presented here.

We may use the present network model in analyzing large-scale neural activity data. To show this, we performed similar simulations using synthetic data in which only a small fraction of (total 500) presynaptic neurons constituted a recurring pattern (Fig.4.5A). It is unlikely that a large portion of recorded neurons participate in recurring cell assemblies in real data. Learning was successful when the fraction was 10% or 5%, but unsuccessful when the fraction was 3% (Fig.4.5B, C). Then, we considered the case where the total number of presynaptic neurons was 1000 and 25 neurons (2.5% of all neurons) belonged to a patterned activity. Interestingly, the network still succeeded to learn the pattern, indicating that successful learning requires a minimal number, but not a minimal fraction, of pattern-encoding presynaptic neurons (Fig.4.5D).

Previously, STDP was used to detect repeated spike sequences [73, 78]. We compared the detection performance between the present model and a STDP-based model ([73]: see Fig.4.6A, B). Both models exhibited high success rates when recurring cell assemblies comprised a large portion of presynaptic neurons. An interesting difference was found when only a small portion of presynaptic neurons participated in the cell assemblies. In such cases, our model outperformed the previous model (Fig.4.6C). We also note that the STDP-based model requires a somewhat unrealistic implementation of STDP, in which postsynaptic neurons need to know the times of future presynaptic spikes.

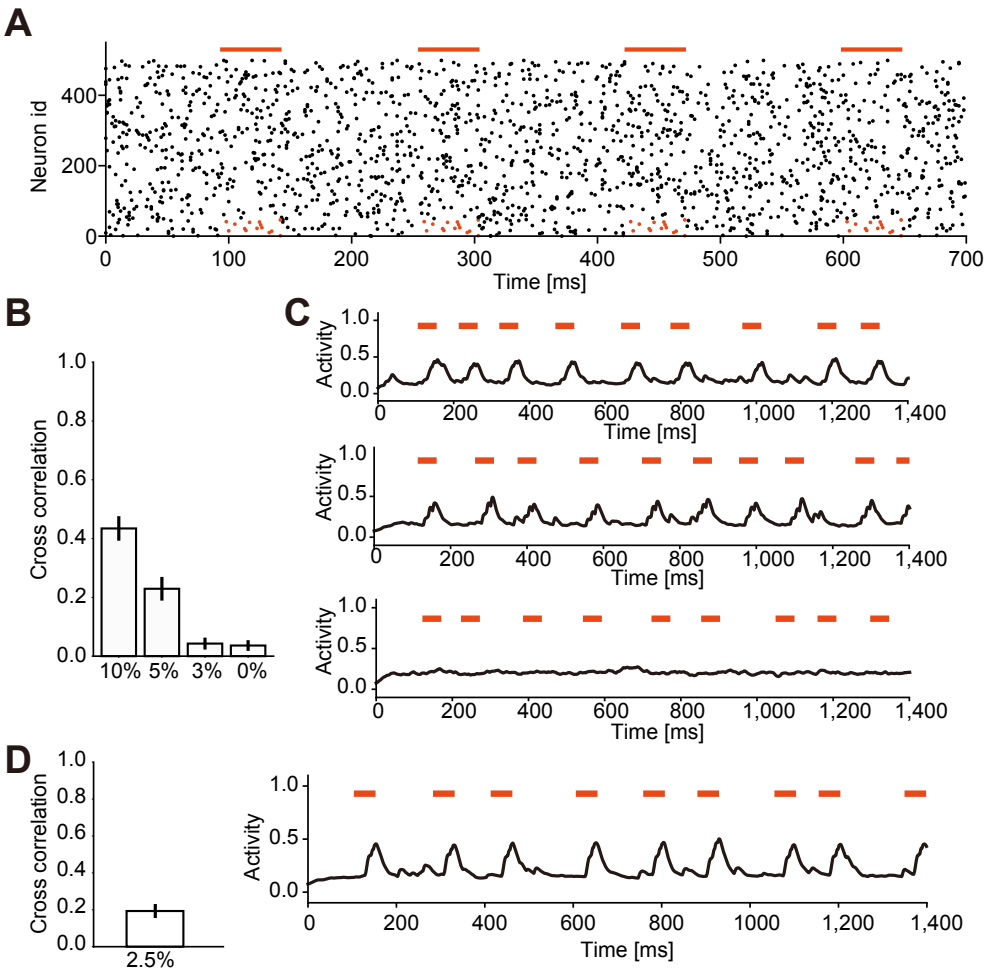

Figure 4.5: **Detection of cell assembly patterns from neural population data.** (A) Spike sequences with a fixed spatiotemporal pattern (red) of 50 (10%), 25 (5%) and 15 (3%) neurons recurred (red horizontal bars) in Poisson spike trains of 500 presynaptic neurons. This example shows the 10% case. (B) Average correlations over 40 independent trials are shown between chunk-selective responses and the corresponding reference patterns. Vertical bars are standard errors. (C) Examples of chunk-selective neuronal responses in the 10, 5 and 3% cases (from the top). (D) A recurring firing pattern of 25 neurons was embedded into input spike trains of 1000 presynaptic neurons. The average and standard error of the input-output correlation (left) and a typical response after learning (right) are shown.

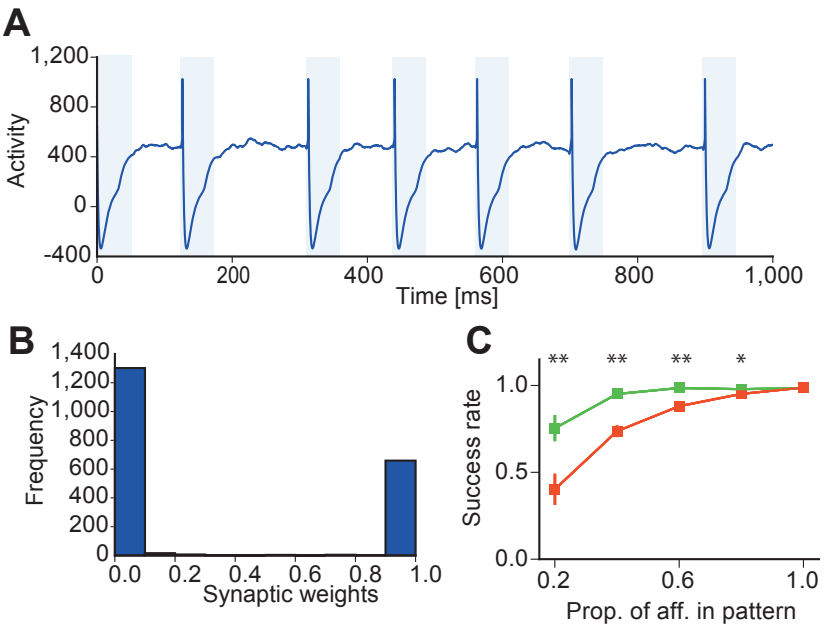

Figure 4.6: Comparison with the STDP-based model in spike sequence detection.
(A) The previous model [73] generates a single spike when it successfully detects recurring input spike patterns (shaded areas). (B) Bimodal distributions of self-organized synaptic weights. (C) Success rates are plotted for the STDP model (red) and our model (green) against the proportions of input neurons encoding the recurring pattern.

We further examined the ability of our network model in learning a variety of information streams. First, we applied random sequences of three chunks comprising four characters each (Fig.4.7A) to a network model with 10 output neurons and 1,000 input neurons. Each input neuron generated a 30 ms 10 Hz burst in response to a randomly assigned preferred character (Fig.4.7B). This resulted in the formation of three neuron ensembles that selectively responded to the chunks (Fig.4.7C). Principal-component analysis of the low-dimensional dynamics of output neurons revealed the emergence of the three chunks after learning (Fig.4.7D). The network also learned the chunks when they were embedded with distractors, which were given as random sequences comprising arbitrary English characters (Fig.4.8). Then, we examined whether the model can learn partially overlapping chunks. In this case, some characters were shared between three chunks (Fig.4.7E) and learning was more difficult than in the previous case. The original model with fast synaptic current failed to generate selective responses to the chunks (Fig.4.7F). However, making the decay constant of synaptic current slower (50 ms: see Methods) enabled the model to sense temporal inputs on a longer timescale and to successfully learn the overlapping chunks (Fig.4.7G). The results suggest that slow synaptic current such as NMDA receptor-mediated current is important for chunking.

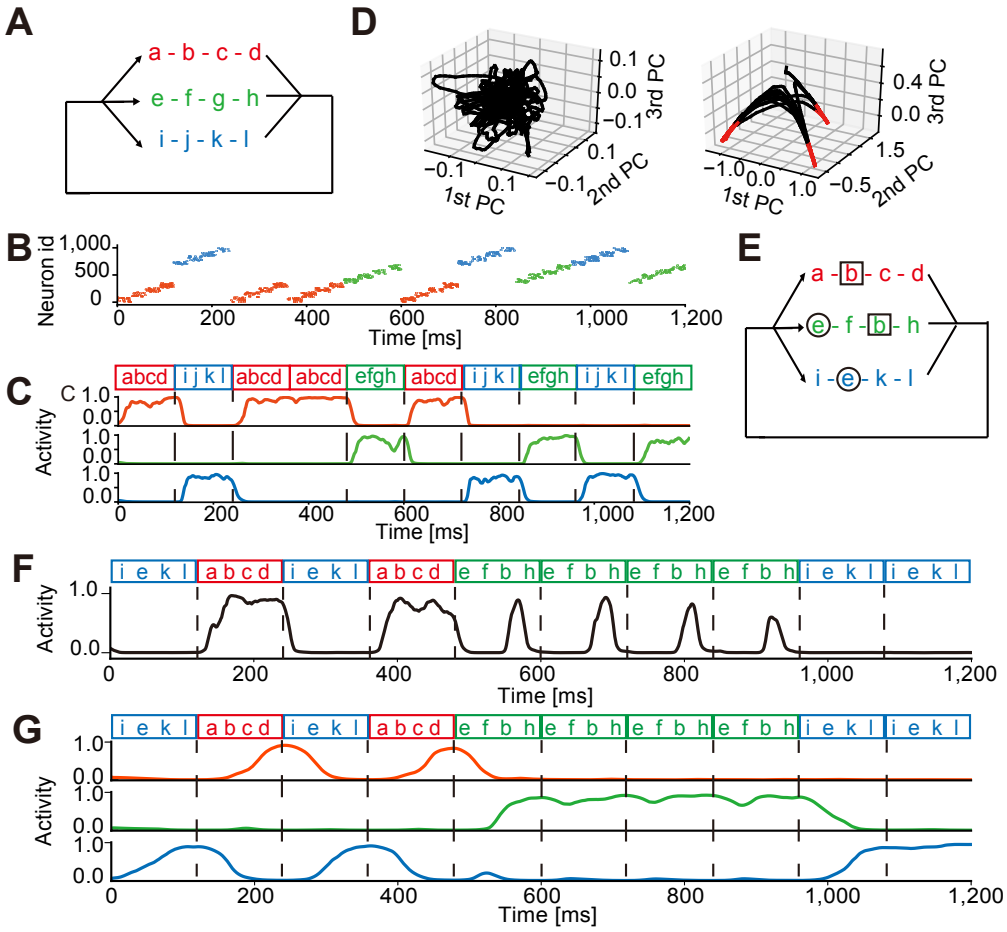

Figure 4.7: **Segmentation and concatenation of various sequences.** Ten output neurons were connected with all-to-all inhibitory synapses modifiable by iSTDP. (A) Three chunks (a-b-c-d [red], e-f-g-h [green], and i-j-k-l [blue]) recurred in the input sequence with equal probabilities. (B) Each input neuron fired at 10 Hz to encode one of the chunks. Neurons were sorted according to their preferred stimuli. (C) Typical normalized responses of three output neurons are shown after learning. Colors indicate the epochs of the corresponding chunks. (D) Responses of output neurons were projected onto the three leading principal-component (PC) vectors before (left) and after (right) learning. More than 99% of the variance was explained by the three PCs. Epochs of high normalized responses ($f > 0.8$ in all neurons) are indicated in red. (E) Character "b" is shared by the red and green chunks, and character "e" appears in the green and blue chunks. (F) Response of an output neuron to the overlapping chunks is shown. The time constant of synaptic current was 5 ms. (G) Selective responses of output neurons to the overlapping chunks are shown when the synaptic time constant was 50 ms.

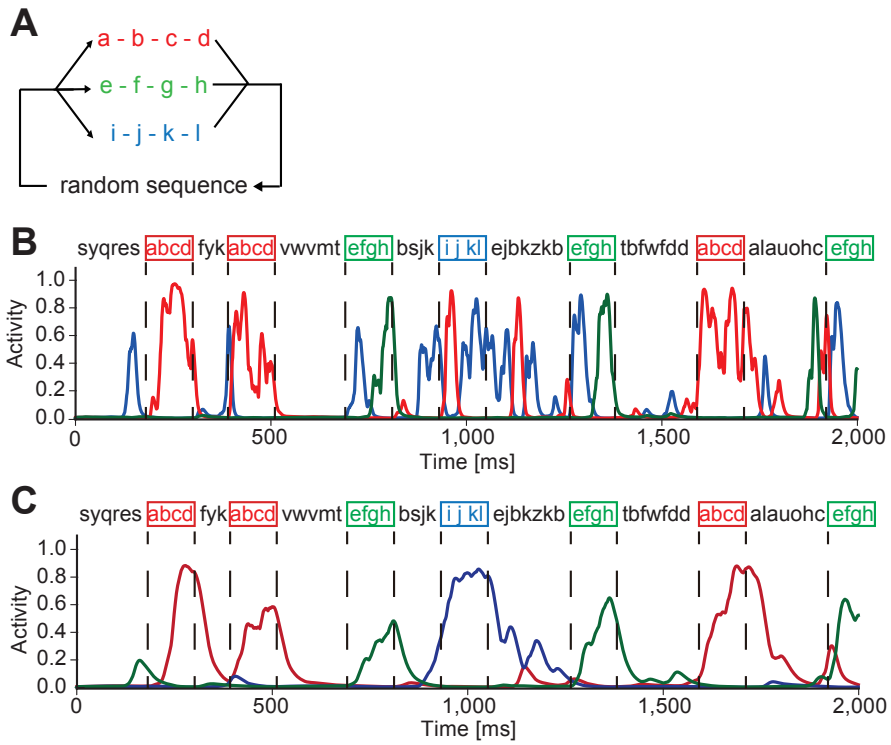

Figure 4.8: **Chunking in the presence of distractors.**
(A) Three chunks were separated by random sequences of arbitrary English characters. The length of random sequences ranged from 3 to 7. (B) Typical responses of output neurons are shown.

Because the word segmentation shown above is relatively easy for other methods as well [83], we next tested the same model with more complex input sequences generated by a random walk on a graph with a community structure, where the connection of each node to the other four occurred with an equal probability of 0.25 (Fig.4.9A). The detection of this community structure was easy for human subjects but was difficult for the conventional machine learning methods that rely on nonuniform transition probabilities between elements [97]. To our surprise, each output neuron easily learned to respond selectively to members within a temporal community (Fig.4.9B).

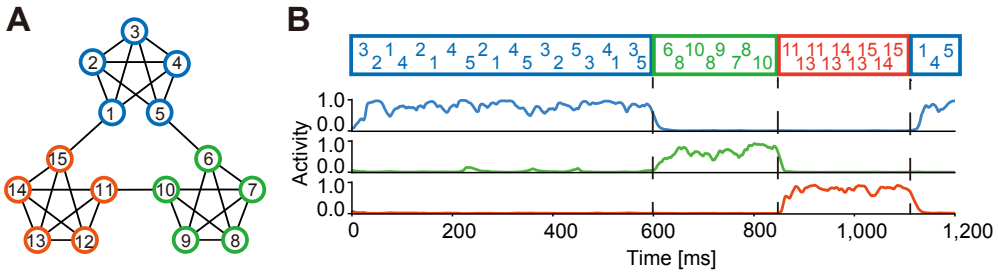

Figure 4.9: **Detection of temporal community.**
(A) The input sequence represented a random walk with uniform transition probabilities on a graph with community structure (modified from reference [97]). (B) Normalized responses of three output neurons to input sequences defined in panel A are shown.

The network model could also learn feature detection maps from continuous sensory streams. All sensory features, either static or dynamic, arrive at the brain essentially in sequence. Therefore, we asked whether MRIL enables neural networks to learn the static features of input when they are repeatedly presented in a temporal sequence. We applied a random sequence of noisy images of oriented bars presented for 40 ms at 30 ms intervals to the network model (Fig.4.10A). The output neurons, which initially had no preferred orientations (Fig.4.10B), developed well-defined preferences for specific orientations after learning (Fig.4.10C), resembling a visual orientation map (Fig.4.10D) [43, 81].

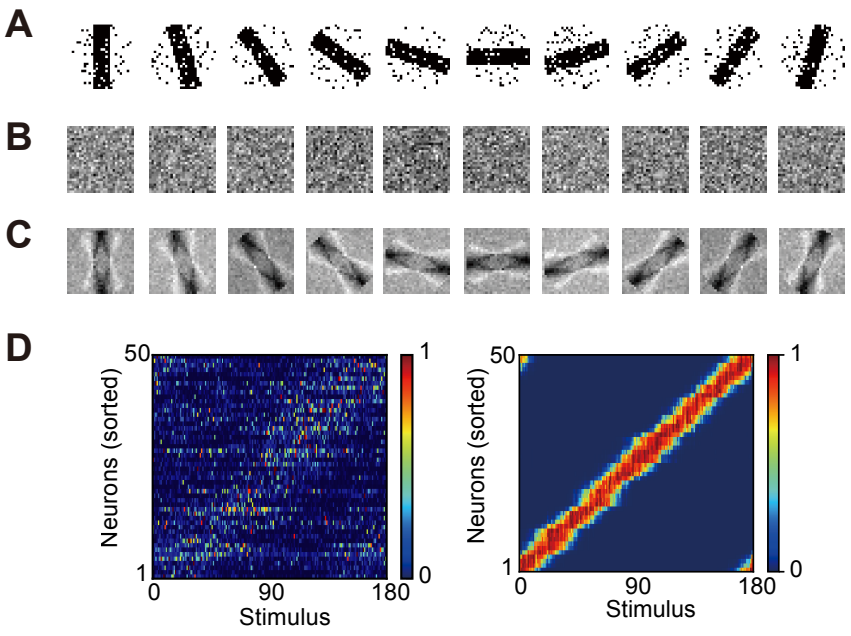

Figure 4.10: **Learning an orientation tuning map.**
(A) Examples of noisy images of oriented bars used for the training. Each image was presented for 40 ms in a random order with intervals of 30 ms between images. (B and C) The feedforward synaptic weights before and after learning are shown for the example stimuli shown in panel A. (D) The responses of all dendritic neurons before (left) and after (right) learning are shown. The neurons were sorted according to the onset times of responses to their preferred stimuli. See Methods for further details on the simulations.

4.2.7 BSS of mutually correlated signals

The results shown above demonstrate that the MRIL successfully chunks a variety of temporal inputs by detecting repeated temporal features. The question then arises whether this ability of the MRIL enables learning of other types of sequence processing tasks. Sequence processing of cognitive importance also involves the so-called cocktail party problem [75]. We examined the performance of our network model in the blind separation of signals within mixtures from multiple sources. BSS is an extensively studied problem in auditory processing [3, 24, 46], and various methods have been proposed for mixtures of mutually independent signals. However, methods are limited when the original signals comprise mutually correlated signals [53].

We applied the MRIL to sound mixtures from two music instruments, i.e., a bassoon and a clarinet (Bach10 Dataset) [28], playing their respective parts of the same score (Fig.4.11A); thus the two sound sources are correlated. The mixtures of signals were encoded into irregular spike trains (Fig.4.11B), which in turn were applied to output neurons. After training, these neurons self-organized into two subgroups, each responding selectively to one of the true sources (Fig.4.11C). The original sounds were then decoded from the average firing rates of these subgroups. Although some high-frequency components were lost due to the low-pass filtering effect corresponding to membrane dynamics (Fig.4.12), the decoded sounds are easily identifiable with the original sounds. We compared our model with a naive independent-component analysis (FastICA) [45, 46] and temporal ICA (Second Order Blind Identification or SOBI) [9]. When the source signals are mutually independent, all three methods showed excellent performance although the ICA-based methods slightly outperformed our biology-inspired model (Fig.4.11D, top). When the source signals are non-independent (i.e., mutually correlated), SOBI and our model exhibited significantly better performance than FastICA (Fig.4.11D, bottom).

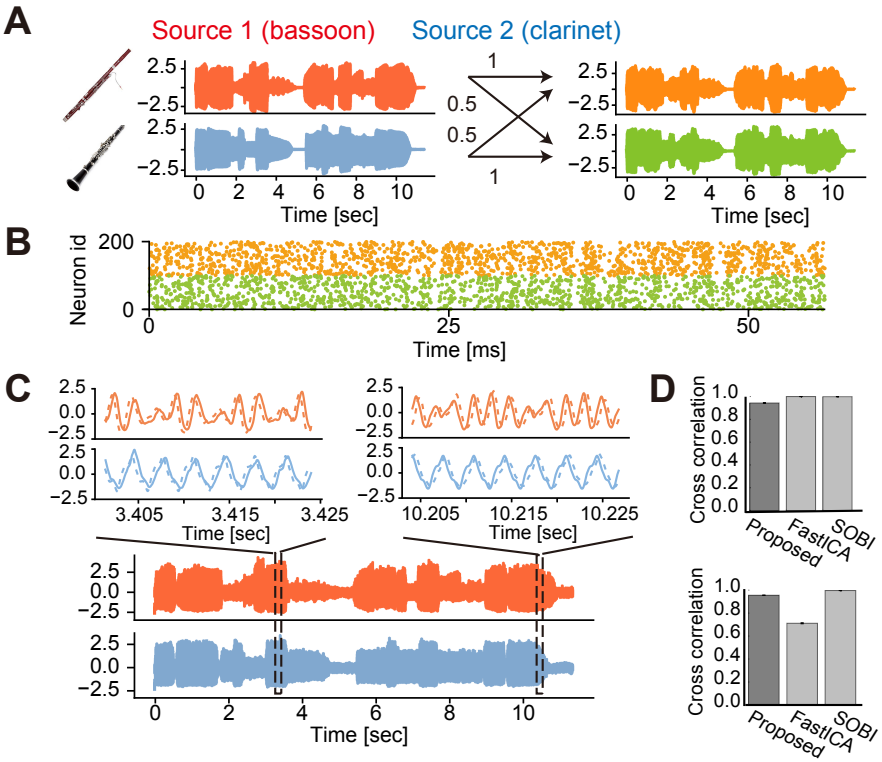

Figure 4.11: **BSS of correlated auditory streams.**
(A) Sound waveforms of a bassoon and a clarinet (left) were linearly transformed to two mixture signals (right). The diagonal and off-diagonal elements of the mixing matrix were 1 and 0.5, respectively. (B) Nonstationary Poisson spike trains of 200 input neurons (out of the total 500) are shown. The instantaneous firing rates were proportional to the amplitudes of the mixed signals normalized between 0 Hz and 10 Hz. Each input neuron encodes either of the two mixed signals. (C) Separated waveforms (bottom) are shown together with magnified versions (top, solid) and true sources (top, dashed). The waveforms were averaged over 20 trials with different realizations of input spike trains and the same initial weights. (D) Cross-correlations between the separated and true sources are compared between our model and independent-component analysis (ICA) for independent (top) and dependent (bottom) auditory signals (see Methods). Error bars show S.D.s (invisible).

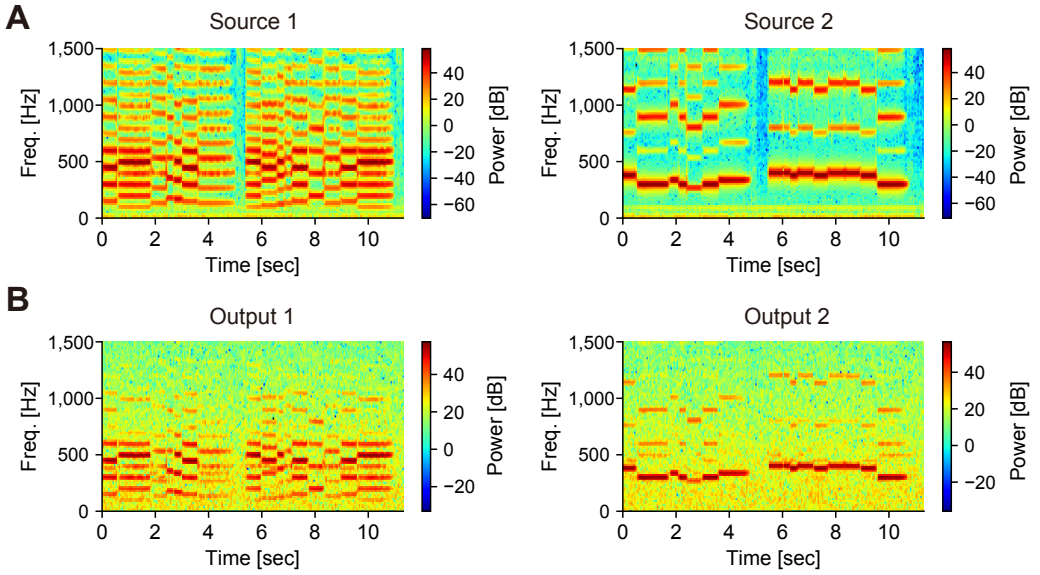

Figure 4.12: **Spectrogram of true and estimated signals in BSS.**
(A) The spectrograms of the true sources. The two sources were taken from the same piece of music. (B) Example spectrograms of the estimated signals. The network model cut off the high-frequency components above 60 Hz. This was because the membrane dynamics act as a low-pass filter with the cut-off frequency being the inverse of the membrane time constant.

Although SOBI slightly outperformed our model in the present examples, SOBI was unable to chunk the previous sequences of English characters (Fig.4.13), which our model could easily solve (see Fig.4.7). The result demonstrates a virtue of the present brain-inspired model, which exhibits high levels of task performance in a wide range of temporal feature analysis. In addition, the model does not require highly task-specific network architectures.

Finally, we examined the performance of the model by varying the magnitudes of cross-talk noise between the two mixture sounds (Methods). We also performed similar tests for the mixtures comprising two identical music instruments playing different notes. In all cases, high performance was attained only at an intermediate level of cross-talk noise, implying that performance drops not only for strong noise but also for weak noise (Fig.4.14A, dashed curves). Nevertheless, we could rescue the model from this counterintuitive defect for weak cross-talk noise by including another noise component (see Methods) in the somatodendritic interaction (Fig.4.14A, solid curves). We speculate that the additional noise could suppress learning from harmful interferences between the original signals when both signals were weak. However, this point requires further clarification. Next, we examined whether the improved model trained on the original signals (i.e., vanishing cross-talk noise) exhibits better performance for other mixtures that were not used in the training. The pre-training actually made the decomposition of unexperienced mixtures easier (Fig.4.14B).

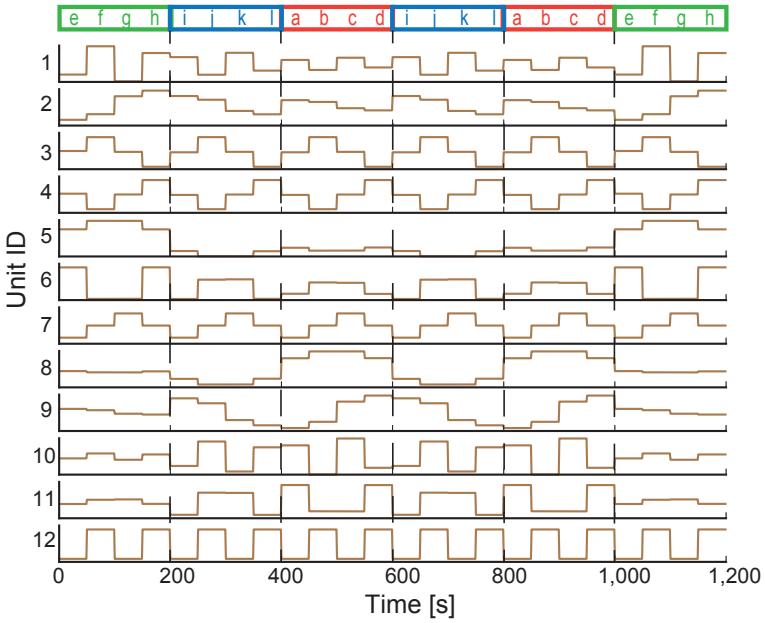

Figure 4.13: **Chunking of character sequences by SOBI.**
We attempted to learn three chunks each consisting four non-overlapping English characters. Post-learning outputs of 12 output units are shown. Note that the number of output units needs to be coincide with that of English characters involved in inputs (a, b, ⋯, l) because SOBI is derived based on a linear algebra.

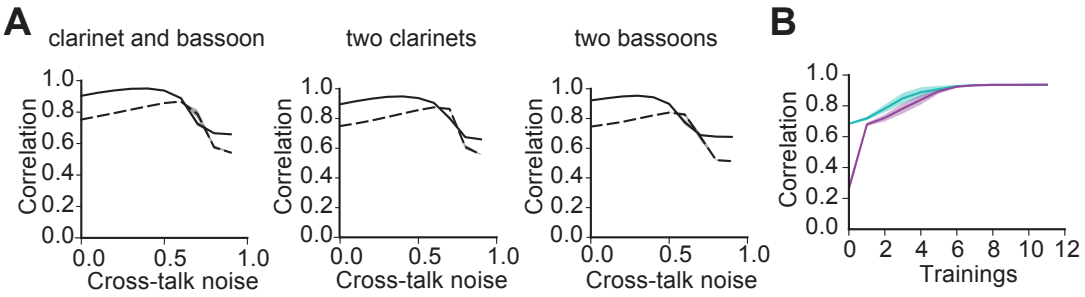

Figure 4.14: **Robustness of performance in BSS.**
Curves represent the averages over five trials with different initial weights and different realizations of noise, and shaded areas represent S.D.s. (A) Correlations between the original and separated signals are plotted as a function of cross-talk noise (Methods) in the cases where the mixtures consist of different (left) or identical (middle and right) instruments. Error bars are invisible. (B) Performance is compared between the networks pre-trained (cyan) or untrained (magenta) on an original signal. Cross-talk noise in the tests was 0.5.

4.3 Discussion

We proposed the novel principle called "minimization of regularized information loss" for enabling the self-supervised learning of multiple recurring temporal features in a family of competitive networks of two-compartment neuron models. Our model not only performs chunking but also achieves BSS from mixtures of mutually correlated signals. Importantly, although different values of parameters were learned in different tasks, the circuit structure was essentially the same. It is surprising that simple neural networks with identical circuit structures can perform the broadly different tasks. In particular, our brain-inspired model can solve some of the hard tasks, e.g., the detection of temporal community structure (Fig.4.9) and the BSS of mixed correlated signals (Fig.4.11). To our knowledge, such a multifunctional model was previously unknown in the learning of information streams.

4.3.1 Comparison with other computational principles

Our learning rule minimizes the information loss between synaptically driven dendritic activity and somatic output. The rule enabled mutually inhibiting dendritic neurons to learn the repetition of spatiotemporal activity patterns on a slow timescale (typically, several tens to several hundreds of milliseconds). While the aim of many previous methods for chunking is to predict input sequences [55, 114], our model entails a different principle in which a neural system learns to predict its own responses to a given input. To this end, the MRIL minimizes the discrepancy between input data and output data to produce a predictable low-dimensional representation of high-dimensional input data. This learning continues until an agreement is formed by the somatic output and dendritic input regarding the low-dimensional features (i.e., chunks).

We previously used paired reservoir computing for chunking, in which two

recurrent networks supervise each other to mimic the partner's responses to a common temporal input [4]. Although the previous model also learns self-consistency between input data and output data, performance of the previous model was severely limited because exactly the same number of output neurons as chunks had to be preconfigured. In contrast, the present model self-organizes output neurons according to the number of temporal features, outperforming the previous one.

Mutual information maximization (MIM) has often been hypothesized to describe the transfer of information between neurons [91], and Hebbian synaptic plasticity may approximately follow MIM [63]. However, the aim of MRIL also differs from that of MIM. As mentioned above, the MRIL attempts the detection of recurring, and hence salient, temporal features without caring the loss of other pieces of information, whereas the MIM principle ultimately implies that messages are faithfully copied at all layers of hierarchical processing. In other words, MIM does not account for the compression or abstraction of temporal inputs, but the MRIL aims at describing how these processes may proceed in the brain. Our results suggest that such processes can even occur at the level of single cortical neurons.

The method called "information bottleneck" also compresses data streams [109]. The method contains a free parameter to determine the degree of information loss between the original and compressed data. To clarify whether there is a relationship between information bottleneck and the proposed method is an intriguing open question.

4.3.2 Relationship to previous models

A previous model [111] used a learning rule similar to the present one. However, while the somatic response function undergoes activity history-dependent

modulations in our model (see equations (4.3)-(4.5)), such modulations were not included in the previous model. Importantly, our model without these modifications (hence the previous model also) could not solve the present unsupervised learning tasks. Networks of the previous model were shown to perform "semi-unsupervised" learning, for instance, when recurrent synaptic input was configured as an effective teaching signal to the soma. In contrast, our model indicates that the recent history of somatic activity is sufficient for "self-supervising" the learning of temporal features.

Dendritic computing has been studied from the various viewpoints of neural computing. Capacity of leaky integrate-and-fire neurons was derived to implement desired transformations from streams of input spikes into desired output spike sequences [76]. The capacity was estimated by calculating the available volume of state space for generating the desired spike outputs, and an error-correcting supervised learning rule was presented to attain the desired input-output associations (this rule does not require dendrites). The role of nonlinear dendritic processing in performing various logic operations was studied [60]. Their model combines the branch-strength potentiation of dendrites and spike-timing-dependent plasticity to discriminate spatial activity patterns represented in presynaptic neuron ensembles. A dendritic model to implement a classical error backpropagation algorithm for supervised learning, in which deviations between top-down predictive signals and bottom-up sensory signals provided an error signal was proposed [95]. Redundant synaptic connections between neuron pairs were utilized to implement Bayesian filtering algorithm to infer input-output associations in single dendritic neurons [41]. The model showed that small number of redundant synapses is sufficient for an optimal inference if the model includes both the Hebbian learning of synaptic weights and structural plasticity on the dendrites. All these models of dendritic processing are biologically more realistic compared to the present model, but these studies did not address the ability of

dendritic neurons in analyzing temporal features.

On the other hand, memory-related sequential activities of hippocampal neurons were modeled in terms of nonlinear amplification of synchronous inputs [50]. Furthermore, the discrimination of sequences on behavioral time scales was recently formulated in terms of the reaction-diffusion processes triggered by sequential inputs along dendrites [10]. While these processes were implemented in morphologically realistic neuron models, whether such models can perform complex temporal feature analyses is yet to be clarified. In [40], they modeled sequence processing in a cortical microcircuit model of formal neurons, each of which receives top-down feedback inputs on apical synapses, feedforward inputs on proximal synapses and lateral inputs from nearby neurons on multiple dendritic segments. Through coincidence detection and segment-basis Hebbian learning, the network learns sparse activity to predict next spikes in input sequence. While their model emphasizes the role of dendrites and cortical microcircuit structure in predicting spike sequences, our model demonstrates the ability of single dendritic neurons to learn recurring temporal input patterns.

To determine which neuron or synapse has credit for learning a desired output of a hierarchical neural circuit is a difficult problem. Solutions to this 'credit assignment problem' require feedback signals conveying to neurons or synapses information on credit. In cortical pyramidal neurons, feedforward signal carrying sensory data and feedback signals possibly carrying credit information project onto the basal and apical dendrites, respectively. It was recently argued that the spatial separation between the two pathways enables these neurons to solve the credit assignment problem through dendritic computing [90]. The current version of our model does not solve the credit assignment problem, but this problem arises on multiple timescales in hierarchical brain computation. How morphologically complex neurons implement the proposed temporal feature analysis and how this analysis helps the brain to solve hierarchically organized credit assignment prob-

lems are intriguing open questions.

4.4 Methods

4.4.1 Improved learning rule with additional noise

In Fig.4.14, we included an additional noise term at each time step of learning as follows:

$$\dot{\mathbf{w}}_i(t) = \eta \left\{ \psi(v_i^*(t)) \left[f(\phi^{som}(u_i(t)) + g\phi_0\xi_i) - \phi^{dend}(v_i^*(t)) \right] e_j(t) - \gamma \mathbf{w}_i \right\}, \quad (4.19)$$

where ξ_i is a random variable obeying a normal distribution. The parameter g controls the strength of the noise, and we set $g = 0.6$ in Fig.4.14. The piecewise linear function f is defined as

$$f(x) = \begin{cases} 0 & (x < 0) \\ x & (0 \leq x < \phi_0) \\ \phi_0 & (x \geq \phi_0) \end{cases}$$

Negative signals should be eliminated to suppress the learning during noise-dominant epochs.

4.4.2 Evaluation of the degree of independency between signals

ICA was not valid for the auditory signals used for the simulations of BSS. This was because the signals were not independent. In addition to the standard correlations between two analog signals, negentropy (≥ 0) was used to evaluate the independency of signals. Negentropy measures the deviation of a target distribution from a Gaussian distribution: negentropy vanishes if the target distribution is Gaussian but otherwise takes a positive value; the larger the deviation is, the

larger the value of negentropy is. The calculation of negentropy $J(Y)$ for the statistical variable Y requires the true distribution, but it is unknown in the present study. Therefore, we made the following approximation in the evaluation of $J(Y)$ using a certain function Q:

$$J(Y) \propto [E\{Q(Y)\} - E\{Q(\rho)\}]^2, \tag{4.20}$$

where $E(x)$ refers to the expectation value of x and ρ obeys a Gaussian distribution. Typically, the logarithm of hyperbolic cosine function is used for Q [45]:

$$Q(u) = \frac{1}{a} \log \cosh(au), \tag{4.21}$$

where $1 \leq a \leq 2$. In this study, we set as $a = 1$.

4.4.3 Simulation details

In Figs 4.2, 4.3, synaptic input to an output neuron during the presentation of a target input pattern was the superposition of a target-specific Poisson spike train of rate $r_{sig} = r/(1 + (S/N)^{-1})$ and a background Poisson spike train of rate $r - r_{sig}$, where S/N refers to the signal-to-noise ratio $r_{sig}/(r - r_{sig})$. Note that $r_{sig} < r$. The temporal patterns of target-specific spike trains were kept unchanged across the repetition of target patterns, whereas background input changed their temporal patterns for each repeat. Outside the target patterns, Poisson spike trains of rate r were given as input. Thus, the spike rate of synaptic input was always r, which was fixed at 5 Hz. In Fig.4.3G, we did not model the realistic process of failure in synaptic transmissions, i.e., failure in evoking postsynaptic potentials by presynaptic spikes. Instead, we simulated transmission failure in the following way. We first generated Poisson spike trains with a fixed Poisson rate $r/(1 - p_{fail})$, where p_{fail} is the failure probability of presynaptic transmissions. Then, assuming that the failure rate is small, we eliminated spikes with the probability

p_{fail}. In Fig.4.3H, we introduced trial-by-trial jitters in presynaptic spikes. First, we generated the reference spike trains used for the presentation of a chunk to individual input neurons. Then, in each presentation of the chunk, spike times were sifted by the amounts drawn by a Gaussian distribution with mean zero.

In Fig.4.7, the numbers of input and output neurons were 1000 and 10, respectively and each input neuron selectively responded to a letter at the firing rate of 10 Hz. The individual chunks consisting of four English letters had the same length of 30 ms and appeared in the input sequence with the same occurrence probabilities of 1/3. Letters in each chunk appeared in a fixed order. Both feedforward connections and lateral inhibitory connections were trained.

In Fig.4.10, the size of each image was $28 \times 28 (= 784)$ pixels and each bar has a width of 7 pixels. Each pixel took its value on either 1 or 0. Noise was generated by flipping pixels with the probability of 0.1, and the images were blurred by circular masks to prevent artifacts from the edges of the images. Each input neuron received input from an image pixel without overlaps between neurons, and the neurons responding to the pixels of value 1 generated Poisson spike trains with the mean firing rate of 40 Hz. The total time of training was 400 sec. The weights of lateral inhibitory connections were not modifiable.

In Fig.4.11, the network had 500 input neurons and two output neurons with non-modifiable lateral inhibition. Each input neuron generated a Poisson spike train at an instantaneous firing rate equal to the waveform of the mixture signal assigned randomly to the neuron. During training, the network model was repeatedly exposed to mixture signals 60 times, where each presentation had 500,000 time steps. In sampling spiking activities, we normalized the mixture signals between the minimum (0 Hz) and maximum (10 Hz) rates. Note that, input spike trains varied from trial to trial although their rate profiles were unchanged. The sampling rate of audio files was 44.1 kHz and the unit time step of network simulations was 1 ms. For comparison with ICA, we used the FastICA function of

Python library scikit-learn with a log cosh function, and tolerance on update at each iteration was 0.0001. FastICA is efficient and popularly used. Independence of auditory signals was evaluated with the negentropy (negative entropy), which measures the deviations of sampled signals from a Gaussian random process. Correlations between signals were not sensitive enough to discriminate between our model and ICA in performance. We compared our model and ICA in two cases. In the first case, we used the sounds of two music instruments playing different pieces of music. In the second case, we used the sounds of the same two instruments playing their own repertoires of the same piece of music. The summed negentropy of two source signals was 0.001 and 0.0004 for the former and latter cases, respectively, indicating that the two signals were less independent in the latter case than in the former case. Generally, the two source signals were highly correlated. Our model gives output activities that are generally delayed behind the input (Fig.4.11C). The delay was adjusted at about 13 ms to obtain the maximal correlations between true sources and output neuron activities. We paired each output activity with the true source that yields the largest correlation. In ICA, the signs of the estimated signals can be opposite to those of the true sources. Therefore, we calculated the correlations between all pairs and all possible combinations of signs (when the number of sources is two, $2! \times 22 = 8$ patterns) to adopt the maximal value. In (C), each waveform was reconstructed from the firing rate of each output neuron averaged over 20 trials with an identical set of initial weights after standardization. Namely, we subtracted the temporal average of the (trial-averaged) firing rate from the instantaneous values and divided the resultant differences by the standard deviation. In (D), simulations were performed 400 times for all combinations of 20 different sets of input spike trains and 20 different sets of initial weights.

Chapter 5

Conclusion

In this thesis, I first made a hypothesis that minimization of uncertainty of networks responses enables them to learn temporal features without any additional knowledge. I proposed novel neural network models to test this hypothesis. The proposed frameworks can be viewed as self-supervised learning rules and I showed that the rule enables the same network to learn surprisingly wide variety of tasks, including chunking of recurring spatiotemporal patterns, detecting temporal community structures of sequences with uniform transition probabilities, and blind source separation of non-independent signals. Although my models in Chapter 3 and 4 have different structures, yet they rely on common principle, in which the models are trained to maximize the self-consistency of their responses to minimize the uncertainty of its responses.

5.1 Summary of Results

In Chapter 2, I investigated the dynamics of reservoir model, trained with independent components extracted from the LFP activity recorded at multiple depths of rat motor cortex during reward-motivated movement. I showed that the trained

model exhibits various functional subtypes of task-related neurons with relative ratio is similar to the one confirmed in experiment. My study suggests that the reservoir model is valid for probing the representation of sequences in the brain, despite the learning rule used here (i.e., the FORCE learning rule) is biologically implausible.

In Chapter 3, I extended the reservoir computing framework, which undergoes supervised learning to unsupervised framework for automatic chunking of complex streams of information. To this end, I proposed the novel structure of neural network, which consists of two independent reservoir modules. In this framework, the two network, which receive the same input sequences mutually supervise each other. Since the teacher signal for training each reservoir modules is generated by partner's output, my novel learning model relies on unsupervised learning framework. The proposed model showed chunk-related activities, which is similar to the one observed in experiment. I tested the model with various types of tasks, including separation of chunks with mutual overlaps, chunking community structures of sequences, or even two images recurring in visual input streams with noisy intervals. My model revealed a unique dynamical mechanism for embedding the temporal community structure of inputs into low-dimensional neural trajectories. From a biological viewpoint, my model based on pairwise computing suggests the computational implications of brain's bi-hemispheric information streams. Owing to recent technological advances, the implementation of this model by electronic devices should be straightforward.

In Chapter 4, I proposed the novel learning rule for two-compartment neuron models. In this model, the teaching signals for training dendritic synapses are generated by somatic compartments, thus it enables the unsupervised learning of temporal features. I showed that the proposed model successfully learns temporal community structures of complex sequences with uniform transition probabilities, and separates mixtures of non-independent signals. These tasks are difficult for

most of the previous machine learning techniques, and my model will provide new insights for neural mechanism of chunking that does not require any prior knowledge about temporal structure of sequences.

5.2 Future directions

In Chapters 3 and 4, I proposed two neural network models that learns salient features in temporal sequences. Although these two models have different structures, yet they rely on common principle, in which the models are trained to maximize the self-consistency of their responses. One of the essential future works is to understand the underlying mechanism of such principles in sequence learning. In particular, adding the Gaussian noise seems to have a large impact on training processes in both of them, and a thorough theoretical analysis is essential.

Another interesting direction is to investigate the biological plausibility of the model proposed in Chapter 4. My current dendritic learning rule is too mathematical and limited to simple structure of dendritic compartments. Since this is biological unrealistic, we are planing to propose a new learning rule for multi-branch dendritic trees.

The final remark is on a extension of my dendritic learning model to the recurrent network model with excitatory and inhibitory connections. Since my current models detects the community structures of sequences, much powerful model for sequence learning is necessary.

Appendix A

Review: Reservoir Computing and FORCE Learning Rule

In this section, l will overview pros and cons of recurrent neural networks for sequence learning. Since the recurrent neural networks have feedback connections to hold the state of the model, they have been widely used in learning and recognition of time series information. However, training the recurrent neural networks is extremely difficult for some reasons. First, it is difficult due to their complex structure of their connections. Although some powerful learning rules for recurrent neural networks have been proposed [94, 115], it is still difficult and complex compared to training feedforward networks. Moreover, chaotic spontaneous activity is widely observed in the brain, but in most learning rules, training recurrent neural networks with chaotic activity is unstable. Since spontaneous activities are thought to be useful in learning complex dynamics [112], conventional methods have some disadvantages.

In order to solve the problems of such recurrent learning, so called the reservoir computing was proposed [49, 66]. In the reservoir computing, input signals are transformed into a high dimensional space to which the neurons are randomly

coupled. The advantage of reservoir computing is that the recurrent connections are always fixed and only the synaptic weights to the outputs are modified even in the learning phase. Reservoir computing has been developed from echo-state networks [49], in which the teaching signals, not the actual output is fed back to the reservoir for stabilization of learning, and seemed to be biologically implausible.

Recently, the FORCE learning [106] was proposed to solve the problems appeared in reservoir computing training. In the FORCE learning, training is performed with the feedback of actual output, and it works well even if the pretrained network shows chaotic activities. In the FORCE learning, the magnitude of weight change at each time step is very large, which differs from previous methods. Therefore, the error between output and the desired one must be small at any time during learning. Due to these characteristics, the goal of FORCE learning is not to bring the output close to the desired output as much as possible, but to reduce the number of time steps required to reduce the error.

A.1 Details of Training Recurrent Neural Networks with FORCE Learning Rule

The reservoir is composed of N_G neurons. Each neuron follows the following dynamics as $i = 1, 2, ..., N_G$,

$$\tau \dot{x}_i(t) = -x_i(t) + g_G \sum_{j=1}^{N_G} J_{ij}^{GG} r_j(t) + J_i^{G_z} z(t) + \sum_{\mu=1}^{N_I} J_{i\mu}^{GI} I_\mu(t) \tag{A.1}$$

Where x_i is the potential of each neuron, N_I is the number of neurons used in input layer and $r_i = [\tanh(x_i)]_+$ is a firing rate, where $[\,]_+$ is rectify (returns 0 if the argument is negative, otherwise the argument itself). The parameter g determines the complexity of the behavior of the reservoir, and shows chaotic spontaneous activity if $g > 1$ [105].

All neurons in the reservoir are connected to the readout neuron, and the output at each time is given by $z = \mathbf{w}^T\mathbf{r}$, where $\mathbf{w}$ is the synaptic weight from reservoir to readout neuron. The initial value of $\mathbf{w}$ was $\mathbf{0}$, and these weights are modified so that the error between output of the readout neuron and teaching signal is minimized according to the learning rule described later. The activity of readout neuron is transmitted to the reservoir via the feedback. Each element of the feedback coupling follows a uniform distribution and randomly takes a value between -1 and 1. The connection matrix of the reservoir is a sparse matrix and each element follows a gaussian distribution with zero mean and the standard deviation is $1/(pN_G)$ where p is a connection probability. In addition, J^{GI} is the connection matrix between input neurons and the reservoir, and each row of it has only one nonzero element according to the normal distribution of zero mean and unit variance.

In the FORCE learning, an error between the actual output and the teaching signal is used for modifying the synaptic weights between the reservoir and the the readout neuron. Here, the output is calculated using the weights before update, that is, in each learning step, the error is calculated as follows.

$$e_-(t) = W^T(t - \Delta t)r(t) - f(t) \tag{A.2}$$

Here, Δt is the time step of learning.

To realize the rapid modification of readout weights, $\mathbf{w}$ is updated by the following algorithm with recursive least squares method (RLS) shown in below,

$$W(t) = W(t - \Delta t) - e_-(t)P(t)r(t). \tag{A.3}$$

Here, P is a correlation matrix with size $N_G \times N_G$, and is updated by the following

procedure at the same time.

$$P(t) = P(t - \Delta t) - \frac{P(t - \Delta t)r(t)r^T(t)P(t - \Delta t)}{1 + r^T(t)P(t - \Delta t)r(t)} \tag{A.4}$$

The initial value of P is I/α, where I is the identity matrix, and α corresponds to the learning rate. Larger α enable the quick learning, but the too large learning rate cause the learning unstable. On the contrary, smaller learning rate makes learning stable, but it takes too long time for learning. Usually, α is set in the range 1 to 100.

Bibliography

[1] L.F. Abbott, B. DePasquale, and R.M. Memmesheimer. Building functional networks of spiking model neurons. *Nature neuroscience*, 19(3):350, 2016.

[2] T.R. Agus, S.J. Thorpe, and D. Pressnitzer. Rapid formation of robust auditory memories: insights from noise. *Neuron*, 66(4):610–618, 2010.

[3] S.I. Amari and J.F. Cardoso. Blind source separation-semiparametric statistical approach. *IEEE Transactions on Signal Processing*, 45(11):2692–2700, 1997.

[4] T. Asabuki, N. Hiratani, and T. Fukai. Interactive reservoir computing for chunking information streams. *PLoS computational biology*, 14(10):e1006400, 2018.

[5] D. Baldwin, A. Andersson, J. Saffran, and M. Meyer. Segmenting dynamic human action via statistical structure. *Cognition*, 106(3):1382–1407, 2008.

[6] A.M. Bastos, W.M. Usrey, R.A. Adams, G.R. Mangun, P. Fries, and K.J. Friston. Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711, 2012.

[7] T.A. Bekinschtein, S. Dehaene, B. Rohaut, F. Tadel, L. Cohen, and L. Naccache. Neural signature of the conscious processing of auditory regularities. *Proceedings of the National Academy of Sciences*, 106(5):1672–1677, 2009.

[8] A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[9] A. Belouchrani, K. Abed-Meraim, J.F. Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on signal processing*, 45(2):434–444, 1997.

[10] U.S. Bhalla. Synaptic input sequence discrimination on behavioral timescales mediated by reaction-diffusion chemistry in dendrites. *Elife*, 6:e25827, 2017.

[11] E.L. Bienenstock, L.N. Cooper, and P.W. Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2(1):32–48, 1982.

[12] C.M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

[13] D. Bor, J. Duncan, R.J. Wiseman, and A.M. Owen. Encoding strategies dissociate prefrontal activity from working memory demand. *Neuron*, 37(2): 361–367, 2003.

[14] T. Branco, B.A. Clark, and M. Häusser. Dendritic discrimination of temporal input sequences in cortical neurons. *Science*, 329(5999):1671–1675, 2010.

[15] M. Buiatti, M. Peña, and G. Dehaene-Lambertz. Investigating the neural correlates of continuous speech computation with frequency-tagged neuroelectric responses. *Neuroimage*, 44(2):509–519, 2009.

[16] G. Buzsáki and K. Mizuseki. The log-dynamic brain: how skewed distributions affect network operations. *Nature Reviews Neuroscience*, 15(4):264, 2014.

[17] G. Buzsáki, C.A. Anastassiou, and C. Koch. The origin of extracellular fields

and currents—eeg, ecog, lfp and spikes. *Nature reviews neuroscience*, 13(6): 407, 2012.

[18] R.T. Canolty and R.T. Knight. The functional role of cross-frequency coupling. *Trends in cognitive sciences*, 14(11):506–515, 2010.

[19] F. Carnevale, V. de Lafuente, R. Romo, O. Barak, and N. Parga. Dynamic control of response criterion in premotor cortex during perceptual detection under temporal uncertainty. *Neuron*, 86(4):1067–1077, 2015.

[20] R. Čeponienė, T. Lepistö, A. Shestakova, R. Vanhala, P. Alku, R. Näätänen, and K. Yaguchi. Speech–sound-selective auditory impairment in children with autism: they can perceive but do not attend. *Proceedings of the national academy of sciences*, 100(9):5567–5572, 2003.

[21] S. Choi, A. Cichocki, H.M. Park, and S.Y. Lee. Blind source separation and independent component analysis: A review. *Neural Information Processing-Letters and Reviews*, 6(1):1–57, 2005.

[22] M.H. Christiansen and N. Chater. The now-or-never bottleneck: A fundamental constraint on language. *Behavioral and Brain Sciences*, 39, 2016.

[23] L.L. Colgin, T. Denninger, M. Fyhn, T. Hafting, T. Bonnevie, O. Jensen, M.B. Moser, and E.I. Moser. Frequency of gamma oscillations routes flow of information in the hippocampus. *Nature*, 462(7271):353, 2009.

[24] P. Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[25] C.P. de Kock and B. Sakmann. Spiking in primary somatosensory cortex during natural whisking in awake head-restrained rats is cell-type specific. *Proceedings of the National Academy of Sciences*, 106(38):16446–16450, 2009.

[26] 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*, 88(1):2–19, 2015.

[27] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[28] Z. Duan and B. Pardo. Soundprism: An online system for score-informed source separation of music audio. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1205–1215, 2011.

[29] K.A. Ericcson, W.G. Chase, and S. Faloon. Acquisition of a memory skill. *Science*, 208(4448):1181–1182, 1980.

[30] J. Fermaglich. Electric fields of the brain: the neurophysics of eeg. *JAMA*, 247(13):1879–1880, 1982.

[31] J. Fonollosa, E. Neftci, and M. Rabinovich. Learning of chunking sequences in cognition and behavior. *PLoS computational biology*, 11(11):e1004592, 2015.

[32] S. Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[33] K. Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127, 2010.

[34] N. Fujii and A.M. Graybiel. Representation of action sequence boundaries by macaque prefrontal cortical neurons. *Science*, 301(5637):1246–1249, 2003.

[35] T. Fukai. Neuronal communication within synchronous gamma oscillations. *NeuroReport*, 11(16):3457–3460, 2000.

[36] T.Q. Gentner, K.M. Fenn, D. Margoliash, and H.C. Nusbaum. Recursive syntactic pattern learning by songbirds. *Nature*, 440(7088):1204, 2006.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[38] A.M. Graybiel. The basal ganglia and chunking of action repertoires. *Neurobiology of learning and memory*, 70(1-2):119–136, 1998.

[39] K.D. Harris and T.D. Mrsic-Flogel. Cortical connectivity and sensory coding. *Nature*, 503(7474):51, 2013.

[40] J. Hawkins and S. Ahmad. Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *Frontiers in neural circuits*, 10:23, 2016.

[41] N. Hiratani and T. Fukai. Redundancy in synaptic connections enables neurons to learn optimally. *Proceedings of the National Academy of Sciences*, 115 (29):E6871–E6879, 2018.

[42] G.M. Hoerzer, R. Legenstein, and W. Maass. Emergence of complex computational structures from chaotic neural networks through reward-modulated hebbian learning. *Cerebral cortex*, 24(3):677–690, 2012.

[43] D.H. Hubel and T.N. Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574–591, 1959.

[44] S.A. Huettel, P.B. Mack, and G. McCarthy. Perceiving patterns in random series: dynamic processing of sequence in prefrontal cortex. *Nature neuroscience*, 5(5):485, 2002.

[45] A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks*, 10(3):626–634, 1999.

[46] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[47] J. Igarashi, Y. Isomura, K. Arai, R. Harukuni, and T. Fukai. A θ–γ oscillation code for neuronal coordination during motor behavior. *Journal of Neuroscience*, 33(47):18515–18530, 2013.

[48] Y. Isomura, R. Harukuni, T. Takekawa, H. Aizawa, and T. Fukai. Microcircuitry coordination of cortical motor information in self-initiation of voluntary movements. *Nature neuroscience*, 12(12):1586, 2009.

[49] H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science*, 304(5667):78–80, 2004.

[50] S. Jahnke, M. Timme, and R.M. Memmesheimer. A unified dynamic model for learning, replay, and sharp-wave/ripples. *Journal of Neuroscience*, 35(49): 16236–16258, 2015.

[51] X. Jin and R.M. Costa. Start/stop signals emerge in nigrostriatal circuits during sequence learning. *Nature*, 466(7305):457, 2010.

[52] X. Jin, F. Tecuapetla, and R.M. Costa. Basal ganglia subcircuits distinctively encode the parsing and concatenation of action sequences. *Nature neuroscience*, 17(3):423, 2014.

[53] H. Kameoka, L. Li, S. Inoue, and S. Makino. Semi-blind source separation with multichannel variational autoencoder. *arXiv preprint arXiv:1808.00892*, 2018.

[54] G.B. Keller and T.D. Mrsic-Flogel. Predictive processing: a canonical cortical computation. *Neuron*, 100(2):424–435, 2018.

[55] S.J. Kiebel, K. Von Kriegstein, J. Daunizeau, and K.J. Friston. Recognizing sequences of sequences. *PLoS computational biology*, 5(8):e1000464, 2009.

[56] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[57] R. Laje and D.V. Buonomano. Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nature neuroscience*, 16(7):925, 2013.

[58] M. Larkum. A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends in neurosciences*, 36(3):141–151, 2013.

[59] M.E. Larkum, J.J. Zhu, and B. Sakmann. A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature*, 398(6725):338, 1999.

[60] R. Legenstein and W. Maass. Branch-specific plasticity enables self-organization of nonlinear computation in single neurons. *Journal of Neuroscience*, 31(30):10787–10802, 2011.

[61] G. Li, L. Deng, D. Wang, W. Wang, F. Zeng, Z. Zhang, H. Li, S. Song, J. Pei, and L. Shi. Hierarchical chunking of sequential memory on neuromorphic architecture with reduced synaptic plasticity. *Frontiers in computational neuroscience*, 10:136, 2016.

[62] D.J. Linden. The return of the spike: postsynaptic action potentials and the induction of ltp and ltd. *Neuron*, 22(4):661–666, 1999.

[63] R. Linsker. Perceptual neural organization: Some approaches based on network models and information theory. *Annual review of Neuroscience*, 13(1): 257–281, 1990.

[64] J.t. Lu, C.y. Li, J.P. Zhao, M.m. Poo, and X.h. Zhang. Spike-timing-dependent plasticity of neocortical excitatory synapses on inhibitory interneurons depends on target cell type. *Journal of Neuroscience*, 27(36):9711–9720, 2007.

[65] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.

[66] W. Maass, P. Joshi, and E.D. Sontag. Computational aspects of feedback in neural circuits. *PLoS computational biology*, 3(1):e165, 2007.

[67] J.C. Magee and D. Johnston. A synaptically controlled, associative signal for hebbian plasticity in hippocampal neurons. *Science*, 275(5297):209–213, 1997.

[68] S. Manita, T. Suzuki, C. Homma, T. Matsumoto, M. Odagawa, K. Yamada, K. Ota, C. Matsubara, A. Inutsuka, M. Sato, et al. A top-down cortical circuit for accurate sensory perception. *Neuron*, 86(5):1304–1316, 2015.

[69] V. Mante, D. Sussillo, K.V. Shenoy, and W.T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature*, 503(7474): 78, 2013.

[70] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science*, 275(5297): 213–215, 1997.

[71] G. Martín-Vázquez, N. Benito, V.A. Makarov, O. Herreras, and J. Makarova. Diversity of lfps activated in different target regions by a common ca3 input. *Cerebral Cortex*, 26(10):4082–4100, 2016.

[72] Y. Masamizu, Y.R. Tanaka, Y.H. Tanaka, R. Hira, F. Ohkubo, K. Kitamura, Y. Isomura, T. Okada, and M. Matsuzaki. Two distinct layer-specific dynamics of cortical ensembles during learning of a motor task. *Nature neuroscience*, 17(7):987, 2014.

[73] T. Masquelier, R. Guyonneau, and S.J. Thorpe. Competitive stdp-based spike pattern learning. *Neural computation*, 21(5):1259–1276, 2009.

[74] F. Mastrogiuseppe and S. Ostojic. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron*, 99(3):609–623, 2018.

[75] J.H. McDermott. The cocktail party problem. *Current Biology*, 19(22):R1024–R1027, 2009.

[76] R.M. Memmesheimer, R. Rubin, B.P. Ölveczky, and H. Sompolinsky. Learning precisely timed spikes. *Neuron*, 82(4):925–938, 2014.

[77] G.A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[78] B. Nessler, M. Pfeiffer, L. Buesing, and W. Maass. Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS computational biology*, 9(4):e1003037, 2013.

[79] M.E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[80] P.L. Nunez, R. Srinivasan, et al. *Electric fields of the brain: the neurophysics of EEG*. Oxford University Press, USA, 2006.

[81] B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996.

[82] G. Orbán, J. Fiser, R.N. Aslin, and M. Lengyel. Bayesian learning of visual chunks by human observers. *Proceedings of the National Academy of Sciences*, 105(7):2745–2750, 2008.

[83] P. Perruchet. What mechanisms underlie implicit statistical learning? transitional probabilities versus chunks in language learning. *Topics in cognitive science*, 11(3):520–535, 2019.

[84] P. Perruchet and A. Vinter. Parser: A model for word segmentation. *Journal of memory and language*, 39(2):246–263, 1998.

[85] J.P. Pfister, T. Toyoizumi, D. Barber, and W. Gerstner. Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. *Neural computation*, 18(6):1318–1348, 2006.

[86] M.I. Rabinovich, P. Varona, I. Tristan, and V.S. Afraimovich. Chunking dynamics: heteroclinics in mind. *Frontiers in computational neuroscience*, 8:22, 2014.

[87] P. Ramkumar, D.E. Acuna, M. Berniker, S.T. Grafton, R.S. Turner, and K.P. Kording. Chunking as the result of an efficiency computation trade-off. *Nature communications*, 7:12176, 2016.

[88] G. Remillard. Implicit learning of fifth-and sixth-order sequential probabilities. *Memory & Cognition*, 38(7):905–915, 2010.

[89] J.R. Reynolds, J.M. Zacks, and T.S. Braver. A computational model of event segmentation from perceptual prediction. *Cognitive science*, 31(4):613–643, 2007.

[90] B.A. Richards and T.P. Lillicrap. Dendritic solutions to the credit assignment problem. *Current opinion in neurobiology*, 54:28–36, 2019.

[91] F. Rieke, D. Warland, R.d.R. Van Steveninck, W.S. Bialek, et al. *Spikes: exploring the neural code*, volume 7. MIT press Cambridge, 1999.

[92] A. Rivkind and O. Barak. Local dynamics in trained recurrent neural networks. *Physical review letters*, 118(25):258101, 2017.

[93] A.R. Romberg and J.R. Saffran. Expectancy learning from probabilistic input by infants. *Frontiers in psychology*, 3:610, 2013.

[94] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[95] J. Sacramento, R.P. Costa, Y. Bengio, and W. Senn. Dendritic error backpropagation in deep cortical microcircuits. *arXiv preprint arXiv:1801.00062*, 2017.

[96] J.R. Saffran, R.N. Aslin, and E.L. Newport. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928, 1996.

[97] A.C. Schapiro, T.T. Rogers, N.I. Cordova, N.B. Turk-Browne, and M.M. Botvinick. Neural representations of events arise from temporal community structure. *Nature neuroscience*, 16(4):486, 2013.

[98] E.W. Schomburg, A. Fernández-Ruiz, K. Mizuseki, A. Berényi, C.A. Anastassiou, C. Koch, and G. Buzsáki. Theta phase segregation of input-specific gamma patterns in entorhinal-hippocampal networks. *Neuron*, 84(2):470–485, 2014.

[99] B. Schrauwen, D. Verstraeten, and J. Van Campenhout. An overview of reservoir computing: theory, applications and implementations. In *Proceedings of the 15th european symposium on artificial neural networks. p. 471-482 2007*, pages 471–482, 2007.

[100] E. Schröger, A. Bendixen, S.L. Denham, R.W. Mill, T.M. Bőhm, and I. Winkler. Predictive regularity representations in violation detection and auditory stream segregation: from conceptual to computational models. *Brain topography*, 27(4):565–577, 2014.

[101] J. Schuecker, S. Goedeke, and M. Helias. Optimal sequence memory in driven random networks. *Physical Review X*, 8(4):041029, 2018.

[102] K.V. Shenoy, M.T. Kaufman, M. Sahani, and M.M. Churchland. A dynamical systems view of motor preparation: implications for neural prosthetic system design. In *Progress in brain research*, volume 192, pages 33–58. Elsevier, 2011.

[103] P.J. Sjöström and M. Häusser. A cooperative switch determines the sign of synaptic plasticity in distal dendrites of neocortical pyramidal neurons. *Neuron*, 51(2):227–238, 2006.

[104] K.S. Smith and A.M. Graybiel. A dual operator view of habitual behavior reflecting cortical and striatal dynamics. *Neuron*, 79(2):361–374, 2013.

[105] H. Sompolinsky, A. Crisanti, and H.J. Sommers. Chaos in random neural networks. *Physical review letters*, 61(3):259, 1988.

[106] D. Sussillo and L.F. Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

[107] D. Sussillo, M.M. Churchland, M.T. Kaufman, and K.V. Shenoy. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature neuroscience*, 18(7):1025, 2015.

[108] M. Takeda, K.W. Koyano, T. Hirabayashi, Y. Adachi, and Y. Miyashita. Top-down regulation of laminar circuit via inter-area signal for successful object memory recall in monkey temporal cortex. *Neuron*, 86(3):840–852, 2015.

[109] N. Tishby, F.C. Pereira, and W. Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[110] T. Toyoizumi and L. Abbott. Beyond the edge of chaos: Amplification and temporal integration by recurrent networks in the chaotic regime. *Physical Review E*, 84(5):051908, 2011.

[111] R. Urbanczik and W. Senn. Learning by the dendritic prediction of somatic spiking. *Neuron*, 81(3):521–528, 2014.

[112] C. Van Vreeswijk and H. Sompolinsky. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, 274(5293):1724–1726, 1996.

[113] W.B. Verwey and E.L. Abrahamse. Distinct modes of executing movement sequences: reacting, associating, and chunking. *Acta psychologica*, 140(3): 274–282, 2012.

[114] C. Wacongne, J.P. Changeux, and S. Dehaene. A neuronal model of predictive coding accounting for the mismatch negativity. *Journal of Neuroscience*, 32(11):3665–3678, 2012.

[115] R.J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[116] M.A. Woodin, K. Ganguly, and M.m. Poo. Coincident pre-and postsynaptic activity modifies gabaergic synapses by postsynaptic changes in cl- transporter activity. *Neuron*, 39(5):807–820, 2003.

[117] N.F. Wymbs, D.S. Bassett, P.J. Mucha, M.A. Porter, and S.T. Grafton. Differential recruitment of the sensorimotor putamen and frontoparietal cortex during motor chunking in humans. *Neuron*, 74(5):936–946, 2012.

[118] J. Yamamoto, J. Suh, D. Takeuchi, and S. Tonegawa. Successful execution of working memory linked to synchronized high-frequency gamma oscillations. *Cell*, 157(4):845–857, 2014.

[119] J.M. Zacks, T.S. Braver, M.A. Sheridan, D.I. Donaldson, A.Z. Snyder, J.M. Ollinger, R.L. Buckner, and M.E. Raichle. Human brain activity time-locked to perceptual event boundaries. *Nature neuroscience*, 4(6):651, 2001.

[120] J.M. Zacks, C.A. Kurby, M.L. Eisenberg, and N. Haroutunian. Prediction error associated with the perceptual segmentation of naturalistic events. *Journal of Cognitive Neuroscience*, 23(12):4057–4066, 2011.