Neural Networks 143 (2021) 74–87

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2021 Special Issue

A brain-inspired computational model for spatio-temporal information processing

Xiaohan Lin [a,1], Xiaolong Zou [a,b,1], Zilong Ji [a,b], Tiejun Huang [a], Si Wu [a,b,*], Yuanyuan Mi [c,d,*]

[a] School of Electronics Engineering and Computer Science, Peking University, No.5 Yiheyuan Road Haidian District, Beijing 100871, PR China
[b] School of Psychological and Cognitive Sciences, IDG/McGovern Institute for Brain Research, PKU-Tsinghua Center for Life Sciences, Peking University, No.5 Yiheyuan Road Haidian District, Beijing 100871, PR China
[c] Center for Neurointelligence, School of Medicine, Chongqing University, No.174 Shazhengjie, Shapingba, Chongqing 400044, PR China
[d] AI Research Center, Peng Cheng Laboratory, No.2, Xingke First Street, Nanshan District, Shenzhen 518005, PR China

ARTICLE INFO

Article history:
Available online 16 May 2021

Keywords:
Spatio-temporal pattern
Brain-inspired
Reservoir computing
Decision-making

ABSTRACT

Spatio-temporal information processing is fundamental in both brain functions and AI applications. Current strategies for spatio-temporal pattern recognition usually involve explicit feature extraction followed by feature aggregation, which requires a large amount of labeled data. In the present study, motivated by the subcortical visual pathway and early stages of the auditory pathway for motion and sound processing, we propose a novel brain-inspired computational model for generic spatio-temporal pattern recognition. The model consists of two modules, a reservoir module and a decision-making module. The former projects complex spatio-temporal patterns into spatially separated neural representations via its recurrent dynamics, the latter reads out neural representations via integrating information over time, and the two modules are linked together using known examples. Using synthetic data, we demonstrate that the model can extract the frequency and order information of temporal inputs. We apply the model to reproduce the looming pattern discrimination behavior as observed in experiments successfully. Furthermore, we apply the model to the gait recognition task, and demonstrate that our model accomplishes the recognition in an event-based manner and outperforms deep learning counterparts when training data is limited.

1. Introduction

Spatio-temporal pattern recognition is a fundamental task in many AI applications. For instance, understanding video contents, such as human actions, requires recognizing the spatio-temporal patterns embedded in image sequences that define the contents (Laptev, 2005; Niyogi & Adelson, 1994; Xie, Sun, Huang, Tu, & Murphy, 2018). Deep neural networks (DNNs) have made great success in static image recognition, whose performances are even better than that of humans in some large image sets (LeCun, Bengio, & Hinton, 2015). However, for spatio-temporal pattern recognition, the field is still lacking promising methods (Herath, Harandi, & Porikli, 2017). There are two major challenges for recognizing complex spatio-temporal patterns. One is on extracting representative features of spatio-temporal patterns. Unlike recognizing images, where a DNN can learn the representative features of objects via supervised learning from large data, in tasks such as video analysis, the complexity of the representational space and shortage of labeled data prevent us from adopting the same strategy. The other challenge is on extracting the temporal structure, in particular, the temporal order, of image sequences, which is crucial for spatio-temporal recognition but remains unsolved yet.

The structures and functions of real neural systems can inspire us to develop efficient computational models for AI applications, for example, the hierarchical nature of information processing in the ventral visual pathway has inspired the development of DNNs (LeCun et al., 2015). Here, we delve into neural systems for inspiration to develop a model for spatio-temporal pattern recognition. A large volume of experiments has demonstrated that the brain can discriminate movement and sound patterns extremely fast. Therefore we turn to the visual and auditory neural pathways and try to emulate the underlying neural mechanisms with a canonical computational model. We find that the high efficiency of spatio-temporal pattern processing in neural systems can be largely attributed to the fact that the brain employs a pre-defined circuitry to pre-process data, which can be modeled as a reservoir network, followed by a decision-making circuitry to integrate evidence over time.

* Corresponding authors.
E-mail addresses: lin.xiaohan@pku.edu.cn (X. Lin), xiaolz@pku.edu.cn (X. Zou), jizilong@mail.bnu.edu.cn (Z. Ji), tjhuang@pku.edu.cn (T. Huang), siwu@pku.edu.cn (S. Wu), miyuanyuan0102@163.com (Y. Mi).

[1] These authors contribute equally to this work.

https://doi.org/10.1016/j.neunet.2021.05.015

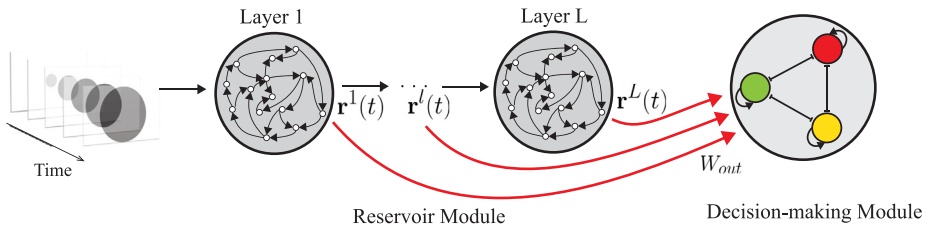

Fig. 1. The structure of RDMN. It consists of two modules, a reservoir network, and a decision-making network. A spatio-temporal pattern is first processed by the reservoir module and then read out by the decision-making module. The reservoir module consists of several forwardly connected layers, with each layer having a large number of recurrently connected neurons. The decision-making module consists of several competing neurons, with each of them representing one category. Each decision-making neuron receives inputs from the reservoir module and they compete with each other via mutual inhibition, with the winner reporting the recognition result.

With regards to the visual pathway, we note that in addition to the ventral and dorsal cortical pathways, there exists a subcortical pathway from the retina to the superior colliculus (SC), which can rapidly recognize motion patterns (De Gelder et al., 2008; Ffytche, Guy, & Zeki, 1995; Wei et al., 2015; Zeki, 1998). In this shortcut pathway, as opposed to the ventral hierarchical one, no explicit feature extraction occurs; rather it resembles the idea of reservoir computing (also called liquid state machines (Bertschinger & Natschläger, 2004) or echo state machines (Jaeger, 2001; Jaeger & Haas, 2004; Yildiz, Jaeger, & Kiebel, 2012)), where the retinal network holds the memory trace of external visual inputs via abundant recurrent connections among neurons, such that the spatio-temporal structure of a motion pattern is mapped into a specific state of the network; consequently, a linear network in SC, specifically the superficial layer of SC which receives the retina input directly, can read out the motion pattern. Notably, one of the four major types of cells in the superficial layer of SC is the wide-field vertical cell (May, 2006), whose prominent morphological characteristic is the huge dendritic tree that allows for spatial sampling and integration over a large retinal area (Gale & Murphy, 2014). The role of wide-field cells is to provide a function of integrating the retinal input and feed them into downstream areas. Also, wide-acting inhibition was observed between wide-field vertical cells (Gale & Murphy, 2014; May, 2006), possibly mediating a winner-take-all computation for discriminating the category of the input pattern.

With regards to the auditory pathway, the first two relays are the inner ear and the cochlear nuclei. In the inner ear, inner hair cells are essential for converting mechanical energy into changes of membrane potentials. Inner hair cells provide 95% of the input to spiral ganglion cells, which in turn yield the exclusive output fibers from the inner ear to the cochlear nuclei (Kiang, Rho, Northrop, Liberman, & Ryugo, 1982). Each spiral ganglion cell receives input from only one inner hair cell, while each inner hair cell synapses on about 10 spiral ganglion neurites (Spoendlin, 1974). This structure is reminiscent of the dimensionality expansion observed in various parts of the brain, such as the cerebellum and the dentate gyrus of the hippocampal formation (Cayco-Gajic & Silver, 2019), as is often touted as a key component in reservoir computing (Lukoševičius & Jaeger, 2009). In the cochlear nuclei, the octopus cell is one of the four principal cells with large dendritic trees that can receive inputs from many cochlear nerve fibers, mirroring the wide-field vertical cell in SC. These neurons are broadly tuned and respond to vowels, musical sounds, and the onset of broadband sounds found in consonants or clicks (Levy & Kipke, 1997).

To sum up, we see that both the visual and auditory pathways share a canonical circuitry mechanism for fast spatio-temporal pattern recognition, which can be captured by a reservoir network followed by a decision-making circuitry. Motivated by this observation, we propose a brain-inspired computational model for spatio-temporal pattern recognition. The model consists of two modules, a reservoir module and a decision-making module. (Buonomano & Maass, 2009; Rabinovich, Huerta, & Laurent, 2008) The reservoir module features abundant recurrent connections, serving as a substrate to hold the fading memory of external inputs. The decision-making module extracts information from the input-specific neuronal activities in the reservoir module and carries out discrimination in an event-based manner.

2. The model

The basic structure of our model, referred to as Reservoir Decision-making Network (RDMN) hereafter, is shown in Fig. 1. The model consists of two modules, a reservoir network and a decision-making network, which are introduced below.

2.1. The decision-making module

Decision-making is a fundamental function of neural systems and has been observed in many areas of the brain. The standard decision-making model was developed based on the recorded neurophysiological data in LIP when monkeys were performing a motion discrimination task (Shadlen & Newsome, 2001). In the experiment, the monkey needed to accumulate evidence over time to judge the coherent moving direction of random dots. During this process, neuron groups representing different choices receive cues from the visual input and compete with each other via mutual inhibition to determine the final choice. We adopt the mean-field decision-making model in Wong and Wang (2006) but simplify it for the application of spatio-temporal pattern discrimination.

As shown in Fig. 1, the decision-making module consists of several competing neurons ($N_{dm} = 3$ is shown in the illustration), with each of them representing one category. These neurons receive inputs from the reservoir module and compete with each other via mutual inhibition, with the winner reporting the discrimination result. Denote x_i the synaptic input received by the ith neuron, r_i the corresponding neuronal activity, and s_i the synaptic current due to NMDA receptors. The dynamics of the module is written as,

$$x_i(t) = J_E s_i + \sum_{j \neq i}^{N_{dm}} J_M s_j + I_i, \tag{1}$$

$$r_i(t) = \frac{\beta}{\gamma} \ln\left[1 + \exp\left(\frac{x_i - \theta}{\alpha}\right)\right], \tag{2}$$

$$\tau_s \frac{ds_i}{dt} = -s_i + \gamma(1 - s_i) r_i, \tag{3}$$

where the synaptic input x_i consists of three components: (1) $J_E s_i$, with $J_E > 0$, denotes the contribution of self-excitation (in a detailed model, it represents the excitatory interactions between neurons encoding the same category (Wong & Wang, 2006));

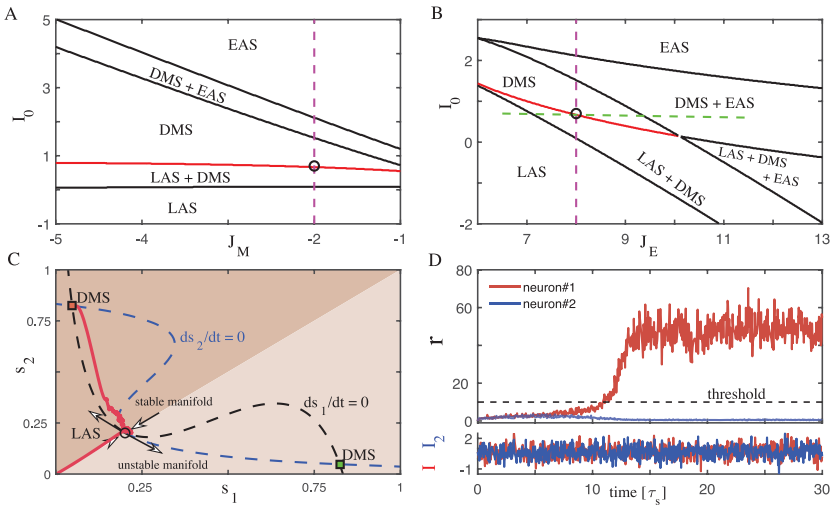

Fig. 2. The mechanism of decision-making. (A-B) The phase diagram of the decision-making network for: (A) the feedforward input I_0 vs. the mutual inhibition J_M. We fix $J_E = 8$ in this case; (B) the feedforward input I_0 vs. the self-excitation J_E. We fix $J_M = -2$ in this case. The stationary states of the network in different parameter regimes are shown. The red lines denote the DM-boundary. (C) The network dynamics when the parameters are on the DM-boundary (at the point denoted by the black circles in (A–B)). The red curve illustrates a typical example of the dynamics of decision-making neurons. (D) An example trial of discriminating two temporal sequences. Upper panel: the time courses of neuronal responses. Lower panel: two temporal sequences with slightly different means corrupted by large fluctuations, which are $I_1 = 0.7 + 0.6\xi_1(t)$ and $I_2 = 0.66 + 0.6\xi_2(t)$, with $\xi_1(t)$ and $\xi_2(t)$ denoting independent Gaussian white noises of zero mean and unit variance. The black dashed line denotes the pre-defined threshold, and a decision is made once a neuron's activity crosses this threshold. Other parameters are: $\alpha = 1.5$, $\theta = 6$, $\beta = 4$, $\gamma = 0.1$, $\tau_s = 100$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(2) $\sum_{j\neq i}^{N_{dm}} J_M s_j$ is the summed recurrent input from other decision-making neurons, with $J_M < 0$ indicating mutual inhibition; (3) I_i is the feedforward input from the reservoir module, whose form is optimized through learning (see Section 2.3). The parameters β, γ, and α control the shape of the nonlinear active function of neurons, and θ the threshold. Eq. (3) describes the slow dynamics of the synaptic current due to the activity-dependent NMDA receptors, which play a crucial role in decision making (Wong & Wang, 2006). $\tau_s \gg 1$ is the time constant, which controls the time window for integrating input information over time by decision-making neurons.

2.1.1. The mechanism of decision-making

Although the dynamics of a decision-making network has been analyzed previously for interpreting the neurophysiological data (Wong & Wang, 2006), to apply it to pattern recognition, we still need to understand its working mechanism thoroughly, in particular, to quantify its feasible parameter regime. Without loss of generality, we consider a simple case that the decision-making network has only two neurons, i.e., $N_{dm} = 2$ for discriminating two spatio-temporal patterns. The result can be straightforwardly extended to general cases of $N_{dm} > 2$ (See Appendix A).

The network dynamics is analyzed when both neurons receive the same constant feedforward inputs, i.e., $I_i = I_0$, for $i = 1, 2$. By varying the parameters, we find that the decision-making module can reach three types of stationary state: (1) Low active state (LAS), in which both neurons are at the same low-level activity; (2) Decision-making state (DMS), in which one neuron is at high activity and the other at low activity; (3) Explosively active state (EAS), in which both neurons are at the same high-level activity. Apparently, only the parameter regime for DMS is suitable for decision-making.

Fig. 2A–B show the phase diagram of the network, which guides us to set the parameters. For example, along the dashed vertical (pink) lines in Fig. 2A–B (both J_E and J_M are fixed), we see that with the increase of the input I_0, the network dynamics experiences several bifurcations: from being stable only at LAS, to at both LAS and DMS, to at only DMS, to at both DMS and EAS, and eventually to at only EAS. The parameter regime for decision-making should be set at where the network holds DMS as stable states. Further inspecting the network dynamics suggests that the optimal regime should be at the bifurcation boundary where LAS just loses its stability and DMS becomes the only stable state of the network (indicated by the red lines in Fig. 2A–B); hereafter, for convenience, we call this boundary the decision-making boundary (DM-boundary). On the DM-boundary, the network dynamics holds two appealing properties: (1) since LAS is unstable, a feedforward input with a little bias (e.g., $I_1 > I_2$) will drive the network to reach at one of DMS (e.g., neuron 1 becomes active while neuron 2 at the low activity state); (2) due to supercritical pitchfork bifurcation, the relaxation dynamics of the network is extremely slow, which endows the network with the capacity of averaging out input fluctuations over time.

To elucidate the above properties clearly, we investigate the network dynamics by setting the parameters on the DM-boundary (the black circles in Fig. 2A–B). Fig. 2C draws the nullclines of neuronal activities (for the variables s_i, for $i = 1, 2$), with their intersecting points corresponding to the unstable LAS and two stable DMSs, respectively. According to the characteristic of supercritical pitchfork bifurcation, the typical trajectory of the network state under the drive of a noisy input is as follows (illustrated by the red curve in Fig. 2C): starting from silence, the network state is attracted first by the stable manifold of LAS; while approaching to LAS closely enough, the unstable manifold of LAS starts to push the network state away, and this process is extremely slow due to that, the eigenvalue of the unstable manifold of LAS is close to zero at the supercritical pitchfork bifurcation point; but once it is far enough from LAS, the network state evolves rapidly to reach one of DMSs. Notably, due to the slow evolving process, the state that the network eventually arrives is determined by the integration of inputs over time, rather than by instant fluctuations.

To confirm the above analysis, we perform a task of discriminating two temporal sequences having slightly different means but corrupted with strong noises. To accomplish this task, it is necessary to integrate inputs over time, so that instant large fluctuations are averaged out and the subtle difference between means pops out. Fig. 2D presents a typical trial of the decision-making process: initially, the activities of two neurons are both

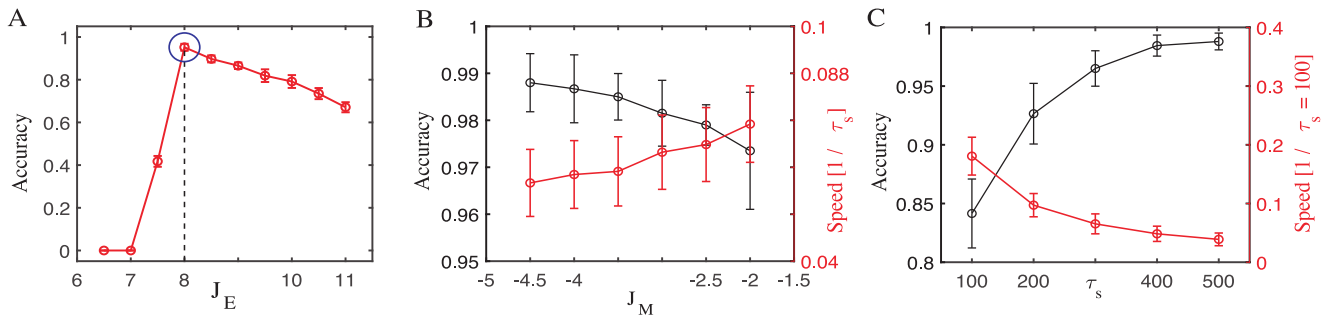

Fig. 3. Parameter setting in the decision-making module. The network performances are evaluated by the task of discriminating two temporal sequences given in Fig. 2D. The discrimination accuracy is measured by the rate that the neuron receiving the larger input wins the competition. The decision speed is measured by $1/t_{res}$, where the decision-making time t_{res} is measured as the moment when the activity of the winning neuron crosses the predefined threshold (Fig. 2D). (A) The discrimination accuracy vs. the self-excitation strength J_E. The value of J_E varies along the dashed horizontal green line in Fig. 2B, with $J_E = 8$ at the black circle on the DM-boundary. When J_E is small on the horizontal axis, the network state will eventually reach LAS, i.e. all decision-making neurons are at resting states, which is regarded as incorrect. (B-C) The speed–accuracy tradeoff of decision-making with respect to: (B) the mutual inhibition strength J_M and (C) the time constant τ_s (speed is evaluated for $\tau_s = 100$). All results are obtained by averaging over 2000 trials. Other parameters are the same as in Fig. 2. Error bars represent standard deviations.

low and intermingled with each other; as time goes on, due to integration of inputs and competition via mutual inhibition, the neuron receiving the larger mean input eventually wins.

To demonstrate that the optimal parameter regime is on the DM-boundary, we compare network performances with varying parameter values and observe that when the parameters are away from the DM-boundary, the discrimination accuracy degrades dramatically (Fig. 3A).

The above analysis reveals that the optimal parameter regime for decision-making should be on the DM-boundary. Along this boundary, there is still flexibility to select the time constant τ_s and the mutual inhibition strength J_M (the value of other parameters such as β, γ, α are chosen according to the unreduced decision-making model (Wong & Wang, 2006)). We find that by varying τ_s or J_M along the DM-boundary, the network performance exhibits a speed–accuracy trade-off (Fig. 3B-C). This is intuitively understandable. With increasing τ_s while fixing other parameters, neurons have a larger time window to average out temporal fluctuations, which increases the discrimination accuracy but postpones the decision-making time. Similarly, with increasing J_M, since the mutual inhibition between neurons becomes larger, it tends to take a longer time for a neuron to win over the competition, which postpones the decision-making time but improves the accuracy. In practice, the values of τ_s and J_M should match the statistics of input noises.

The above analysis can be extended to general cases when the decision-making module has $N_{dm} > 2$ number of neurons to discriminate N_{dm} spatio-temporal patterns. We can calculate the phase diagram of the network with a varying number of decision-making neurons and obtain the optimal parameter regime accordingly (see Appendix A). The optimal parameter regime is given by the DM-boundary in each case.

2.2. The reservoir module

The above analysis demonstrates that the decision-making module can average out input fluctuations via its slow dynamics, but this is not enough for discriminating complex spatio-temporal patterns, e.g., to discriminate temporal sequences which are only differentiable in oscillation frequencies, since the integration of these input sequences gives the same mean value. We need to induce a reservoir module to leverage the representation power of the model (Fig. 1).

Reservoir computing has been proposed as a canonical framework for neural information processing (Bertschinger & Natschläger, 2004; Jaeger, 2001; Yildiz et al., 2012). Its key idea is to project external inputs of low dimension into activity patterns of a large-size network. (Buonomano & Maass, 2009; Rabinovich et al., 2008). Here, as a pre-processing step before decision-making, we expect that the reservoir model can map different spatio-temporal patterns into spatially separated neural activities, so that the decision-making module can read out them.

The reservoir model we consider consists of L forwardly connected layers, and neurons in each layer are connected recurrently (Fig. 1). In practice, the value of L will be set depending on the temporal characteristics of spatio-temporal patterns to be discriminated (see examples in Sections 3 & 4). Denote x_i^l the synaptic current received by neuron i in layer l, for $i = 1, \ldots, N_l$; $l = 1, \ldots, L$, and N_l the number of neurons in layer l. The dynamics of the reservoir module is written as,

$$\tau_l \frac{dx_i^l}{dt} = -x_i^l + \sum_{j=1}^{N_{l-1}} W_{i,j}^{l,l-1} r_j^{l-1} + \sum_{j\neq i}^{N_l} W_{i,j}^{l,l} r_j^l + \sum_{j=1}^{N_{in}} W_{i,j}^{l,0} I_j^{ext} \delta_{l,1}, \quad (4)$$

where $r_i^l = \tanh(x_i^l)$ is the neuronal activity, τ_l the time constant of layer l, $W_{i,j}^{l,l-1}$ the forward connection from neuron j at layer $l-1$ to neuron i at layer l, and $W_{i,j}^{l,l}$ the recurrent connection from neurons j to i at layer l. I_j^{ext} represents the external input, with N_{in} the input dimension and $W_{i,j}^{l,0}$ the input connection weight; $\delta_{l,1} = 1$, for $l = 1$, and otherwise 0, indicating that only layer 1 receives the external input.

The optimal parameter regime for the reservoir module is fixed by adjusting the dynamics of the reservoir module properly. Specifically, we choose the parameters, so that each layer of the module holds two good computational properties, which are: (1) starting from different initial states, the same external input will drive the network to reach the same stationary state, satisfying the so-called echo state property (Jaeger, 2001; Yildiz et al., 2012); (2) in response to different external inputs, the network states are also significantly different, realizing the so-called computing at the edge of chaos (Bertschinger & Natschläger, 2004; Bertschinger, Natschläger, & Legenstein, 2005).

The detailed procedure for parameter setting is as follows. Firstly, we set both the feedforward connections between layers and the recurrent connections in the same layer to be sparse and random, such that neuronal responses are largely independent of each other. Specifically, for the feedforward connections $W_{i,j}^{1,0}$ and $W_{i,j}^{l,l-1}$, they only have a small probability of $p = 10\%$ to take non-zero values randomly chosen from uniform distributions in the ranges of $[-w_{ext}^{1,0}, w_{ext}^{1,0}]$ and $[-w_f^{l,l-1}, w_f^{l,l-1}]$, respectively, where $w_{ext}^{1,0}$ and $w_f^{l,l-1}$ are positive numbers controlling the overall

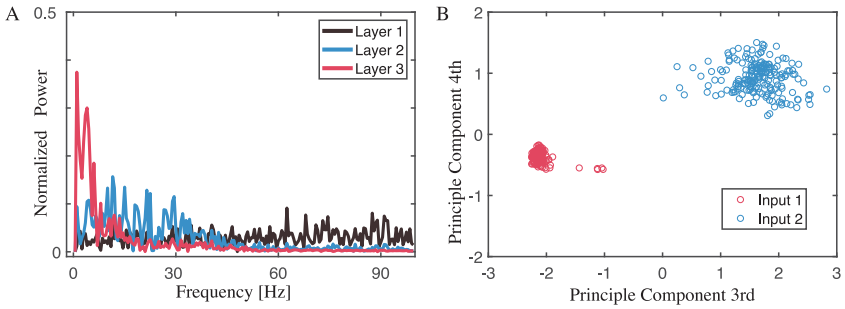

Fig. 4. Representation capacity of the reservoir module. (A) Spectral analysis of neuronal responses across layers when a sequence of Gaussian white noises of zero mean and unit variance is applied as the external input. (B) Separation of neural representations in the reservoir module for two temporal sequences of different frequencies, which are $I^{ext,1}(t) = \sin(20\pi t) + \sin(200\pi t) + 0.1\xi_1(t)$ and $I^{ext,2}(t) = \sin(40\pi t) + \sin(160\pi t) + 0.1\xi_2(t)$, where $\xi_1(t)$ and $\xi_2(t)$ are Gaussian white noises of zero mean and unit variance. Neural activities are projected onto their third and fourth principal components, which have the largest contributions on separating the two temporal sequences. The parameters are: $L = 3$, $N = [130, 130, 130]$, $\tau = [2.5, 10, 150]$, $\rho = [1.1, 1.1, 1.1]$, $w_{ext}^{1,0} = 1$, $w_f^{2,1} = 10$, $w_f^{3,2} = 25$. The detail of data processing is described in Appendix B.

connection strengths. For the recurrent connection $W_{i,j}^{l,l}$, it only has a small probability of $p = 10\%$ to take a non-zero value randomly chosen from a Gaussian distribution of zero mean and variance $(w_c^{l,l})^2$, with $w_c^{l,l}$ a positive number controlling the overall recurrent connection strengths at layer l. Note that since $W_{i,j}^{l,l}$ has equal properties to take positive and negative values, the balanced condition is roughly satisfied. Secondly, we scale the recurrent connection strengths properly, so that the largest eigenvalue of the matrix $\mathbf{M}^l = (1-\delta t/\tau_l)\mathbf{E} + \mathbf{W}^{l,l}\delta t/\tau_l$ (Yildiz et al., 2012), referred to as $\rho(\mathbf{M}^{l,l})$, is slightly larger than one. Here, $\mathbf{E}$ is the identity diagonal matrix, and $\mathbf{W}^{l,l}$ the recurrent matrix whose elements are $W_{i,j}^{l,l}$. In practice, we find that when $\rho(\mathbf{M}^{l,l})$ is in the range of (1.1, 1.3), the network has good performances.

2.2.1. Neural representation in the reservoir module

We illustrate the representation capacity of the reservoir module using some simple examples. Fig. 4A presents the spectral analysis of neuronal responses in different layers when a sequence of Gaussian white noises is applied as the external input. It shows that along the layer hierarchy, the dominating components of neuronal responses progress from high to low frequencies, indicating that the frequency information of temporal inputs is separated across layers. This implies that if two temporal sequences have different frequencies, they can be discriminated against via neural activities across layers.

We also apply two temporal sequences to the reservoir module, which are: $I^{ext,1}(t) = \sin(20\pi t) + \sin(200\pi t) + 0.1\xi_1(t)$ and $I^{ext,2}(t) = \sin(40\pi t) + \sin(160\pi t) + 0.1\xi_2(t)$. Using principal component analysis (PCA) to the neural activities at the reservoir module generated by the temporal patterns, we find that the two patterns are well separated (see Fig. 4B). Later, we further demonstrate that the decision-making module can capture this difference (see Section 2.3).

2.3. Integrating two modules

As shown in Fig. 1, the reservoir and decision-making modules are integrated via a linear read-out matrix to carry out a discrimination task, where the read-out matrix is optimized using known examples.

Denote the input from the reservoir module to a decision-making neuron as,

$$I_i = I_0^* + \sum_{l=1}^{L}\sum_{j=1}^{N^l} W_{lj}^{dm,i} r_j^l, \tag{5}$$

where the summation runs over all neurons in the reservoir module, and $W_{lj}^{dm,i}$ denotes the connection weight from neuron j in layer l of the reservoir network to neuron i in the decision-making module. I_0^* is the optimal feedforward input specified by the DM-boundary (see Fig. 2), We optimize the read-out matrix $\mathbf{W}^{dm} = \{W_{lj}^{dm,i}\}$ through minimizing the discrepancy between the actual inputs received by decision-making neurons and the target inputs, which is written as,

$$E = \frac{1}{2}\sum_{i=1}^{N_{dm}}\sum_{k=1}^{N_k}\int_0^T dt\left[f_i^k(t) - I_i^k(t)\right]^2, \tag{6}$$

where N_{dm} is the number of spatio-temporal pattern categories, i.e., the number of decision-making neurons, and N_k the number of training patterns in each category. $I_i^k(t)$ is the actual input to the decision-making neuron i obtained by evolving the network dynamics when a pattern k is presented (Eqs. (1)–(3)), and $f_i^k(t)$ is the target input function for the decision-making neuron i in response to the spatio-temporal pattern k. According to the response characteristics of decision-making neurons as observed in the experiment (Shadlen & Newsome, 2001) and also in our model (Fig. 2D), the target function $f_i^k(t)$ for the decision-making neuron representing the correct choice should be of the sigmoid-shape over time. Therefore, we set the target function for $k = i$ to be $f_i^k(t) = J_E\{\tanh[b(t - T/2)] + 1\}/2 + I_0^*$, where T represents the input duration, and the target function for $k \neq i$ to be $f_i^k(t) = J_M + I_0^*$. We optimize the read-out matrix $\mathbf{W}^{dm}$ by minimizing the error function E using backpropagation through time (BPTT) (Rumelhart, Hinton, & Williams, 1986). A biologically more plausible method, FORCE learning (Sussillo & Abbott, 2009), can also be used to get comparable results.

It has been demonstrated both in experiments and theoretical studies that with appropriate learning rules, linear read-out neurons are capable of decoding time-varying states from reservoir networks (Buonomano & Maass, 2009; Hung, Kreiman, Poggio, & DiCarlo, 2005; Mazor & Laurent, 2005; Nikolić, Häusler, Singer, & Maass, 2009). To demonstrate this property, we inspect the learned read-out matrix by considering the task of discriminating two temporal sequences as given in Fig. 4B. We first combine the neural response matrices under two temporal sequences $\mathbf{R}^1(t)$ and $\mathbf{R}^2(t)$ into one matrix $\mathbf{A}$. PCA is subsequently applied to the combined neural activity matrix (see Appendix B for details). We project the neural activity difference $\left[\mathbf{R}^1(t) - \mathbf{R}^2(t)\right]$ on each principal component (PC), and obtain the contribution of each PC on separating the two sequences, which are given by: $D_1 = 0.32$, $D_2 = 0.27$, $D_3 = 3.75$, $D_4 = 1.29$ (the first four PCs are considered). Here, PCs are calculated on the neural activity vectors concatenated in the time dimension, with the larger the absolute value of D_i, the larger the contribution of the PC on separating two temporal sequences. In this particular example,

PC3 and PC4 have the largest contributions (Fig. 4B) (this is due to that PCA is performed on the combined matrix **A**, whereby PC1 and PC2 correspond to the intra-class variability, while PC3 and PC4 correspond to the inter-class variability, and the latter is informative for pattern classification). To accomplish the recognition task, the read-out matrix should have large overlaps with those PCs having large contributions on separating two temporal sequences (i.e. PC3 and PC4 in this example). To check whether this is the case, for neuron 1, we compute the projections of the difference between two read-out vectors $\left[\mathbf{W}^{dm,1}(t) - \mathbf{W}^{dm,2}(t)\right]$ on PCs, which gives $C_1 = 0.08$, $C_2 = 0.05$, $C_3 = 0.83$, $C_4 = 0.26$, showing that indeed the projections on PC3 and PC4 are much larger than those on other PCs. This demonstrates that through learning, the read-out matrix indeed captures the key components of neural representations at the reservoir module needed for discriminating the two temporal sequences.

3. Model analysis

In this section, using synthetic data, we analyze the computational properties of the model RDMN.

3.1. Extracting frequency information of temporal inputs

We consider the task of discriminating two temporal sequences that are different in either high or low frequencies (see Table 1). To explore the influences of model parameters, we vary the number of layers, the number of neurons in each layer, and the time constant of each layer in the reservoir module, and the way the decision-making module reading out information from the reservoir module.

The results are summarized in Table 1, which shows that: (1) with the same total number of neurons, reservoir modules with 3 layers outperform ones with 1 layer. This is due to that a reservoir module with more layers can encode a broader range of frequency information (see also Fig. 4A); (2) reservoir modules with increasing time constants along the hierarchy outperform ones with invariant time constants, due to the same reason as in (1); (3) the model performance increases with the number of neurons in the reservoir module; (4) the decision-making module integrating temporal information from all layers of the reservoir module outperforms the one integrating information from only the last layer. Overall, these results demonstrate that RDMN can extract the frequency information of temporal patterns.

3.1.1. On the number of layers of the reservoir module

The number of reservoir layers in our model is an important parameter that influences the model performance. Given that the reservoir module aims to extract different frequency information in the temporal input (Fig. 4 and Table 1), we want different reservoir layers to cover different frequency bands. To this end, we typically need to have some ideas about the frequency distribution of the input, which may be obtained through applying Fourier Transform to the data. For example, signals in Table 1 are composed of three sine curves with different frequencies, therefore the numbers of reservoir layers are set to 3, whereby the temporal information in each sine curve can be represented in each layer. When faced with realistic tasks, Fourier Transform can be applied at first to find out the number of dominating frequency bands, and then use this information to determine the number of reservoir layers.

Table 1
Discriminating the frequency information of temporal patterns. Two sequences applied are: $I_1^{ext}(t) = \sum_{i=1}^{3} \sin\left[2\pi a_i^1(t+\xi_1)\right]/3 + 1$, $I_2^{ext}(t) = \sum_{i=1}^{3} \sin\left[2\pi a_i^2(t+\xi_2)\right]/3 + 1$, where ξ_1 and ξ_2 are random noises uniformly distributed in the range of $(0, 5)$. Task A: two patterns are different at low frequencies with $a^1 = [0.1, 20, 60]$ and $a^2 = [0.4, 20, 60]$. Task B: two patterns are different at high frequencies with $a^1 = [0.1, 20, 65]$ and $a^2 = [0.1, 20, 60]$. The time constant τ in the single-layer case is chosen to have the best performance. For readability, the actual time constant τ is set to be $\tau_c \times 10^{-2}$. The number of layers, the number of neurons in different layers, and the time constants of different layers in the reservoir module are varied, and their performances are compared. Parameters in the reservoir module: $\rho(M^{l,l}) = 1.1$, for $l = 1, 2, 3$; $w_f^{2,1} = 10$, $w_f^{2,1} = 25$; $w_{ext}^{1,0} = 1$. Parameters in the decision-making module are the same as in Fig. 2.

	Task A	Task B
1 Layer N = 180	51.91% τ_c = 1	69.47% τ_c = 0.25
3 Layer N = 60, 60, 60	61.90% τ_c = 5, 5, 5	90.35% τ_c = 0.25, 0.25, 0.25
3 Layers (Last) N = 60, 60, 60	86.37% τ_c = 0.25, 1, 12	91.43% τ_c = 0.25, 1, 7
3 Layer (All) N = 60, 60, 60	88.75% τ_c = 0.25, 1, 12	97.25% τ_c = 0.25, 1, 7
3 Layer (All) N = 80, 80, 80	94.55% τ_c = 0.25, 1, 12	99.62% τ_c = 0.25, 1, 7
3 Layer (All) N = 130, 130, 130	96.19% τ_c = 0.25, 1, 12	99.75% τ_c = 0.25, 1, 7
3 Layer (All) N = 130, 130, 130	45.9% τ_c = 12, 1, 0.25	40.35% τ_c = 7, 1, 0.25

Table 2
Variations in the timescale parameters do not have a significant performance impact. The first line in each row shows the model performances for two sets of timescale parameters given in the second line.

	Task A	Task B
3 Layer (All) N = 130, 130, 130	96.19% τ_c = 0.25, 1, 12	99.75% τ_c = 0.25, 1, 7
3 Layer (All) N = 130, 130, 130	96.43%/96.70% τ_c = 0.15/0.4, 1, 12	99.77%/97.30% τ_c = 0.1/0.3, 1, 7
3 Layer (All) N = 130, 130, 130	95.87%/95.20% τ_c = 0.25, 0.75/6, 12	99.77%/98.37% τ_c = 0.25, 0.25/7, 7
3 Layer (All) N = 130, 130, 130	95.23%/95.37% τ_c = 0.25, 1, 9/24	99.43%/98.47% τ_c = 0.25, 1, 5/48

3.1.2. On the timescale parameters of the reservoir module

Timescale hierarchy is a prominent feature that characterizes the information processing dynamics from sensory to association cortices (Chaudhuri, Knoblauch, Gariel, Kennedy, & Wang, 2015; Honey et al., 2012; Murray et al., 2014). While the reservoir module in our model does not correspond to cortices, we note that for each reservoir layer to extract different frequency bands, the best practice is to set timescale parameters hierarchically (Table 1). Nonetheless, since the hierarchical architecture of the reservoir module already has the general property of extracting increasing frequencies from shallow to deep layers, the timescale parameters only need to loosely follow a hierarchical pattern. This can be seen from Table 2, where the precise values of timescale parameters are of little consequence. However, if the timescale parameters are not chosen increasingly, the performance of the model would severely degrade, as in the last row of Table 1 (note that the performance can be less than 50% because the cases where all decision-making neuron activities are below the threshold are regarded as failures).

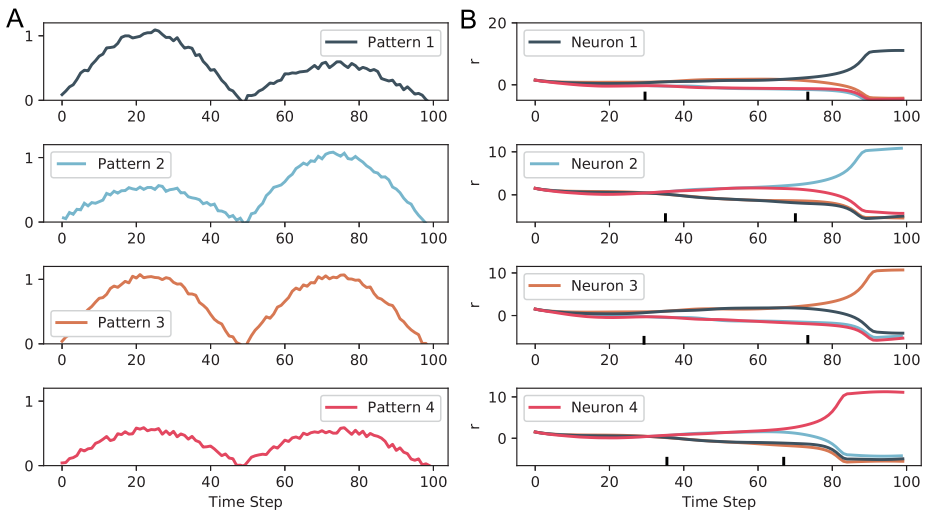

Fig. 5. Extracting the temporal order information of patterns. A. Four temporal patterns formed by two bumps in different configurations are given by $I_1(t) = I_A(t) + I_B(t - T/2)$, $I_2(t) = I_B(t) + I_A(t - T/2)$, $I_3(t) = I_B(t) + I_B(t - T/2)$, and $I_4(t) = I_A(t) + I_A(t - T/2)$, where $T = 100$, $I_A(x) = 0.5 \sin[\pi x/(T/2)]$ for $x \in [0, T/2]$ and otherwise $I_A(x) = 0$, and $I_B(x) = \sin[\pi x/(T/2)]$ for $x \in [0, T/2]$ and otherwise $I_B(x) = 0$. B. The activities of four neurons in the decision-making module when each pattern is presented. The four neurons learn to discriminate the four patterns successfully. The small black ticks indicate the moments when neuronal activities start to separate. Parameters in the reservoir module: $L = 1$, $dt = 1$, $N = 100$, $\tau = 20$, $\rho(\mathbf{M}^{I,I}) = 1.1$, $w_{ext}^{1,0} = 1$. Parameters in the decision-making module: $J_E = 10$, $J_M = -6$, $\tau_s = 10$, $I_0^* = 1.52$.

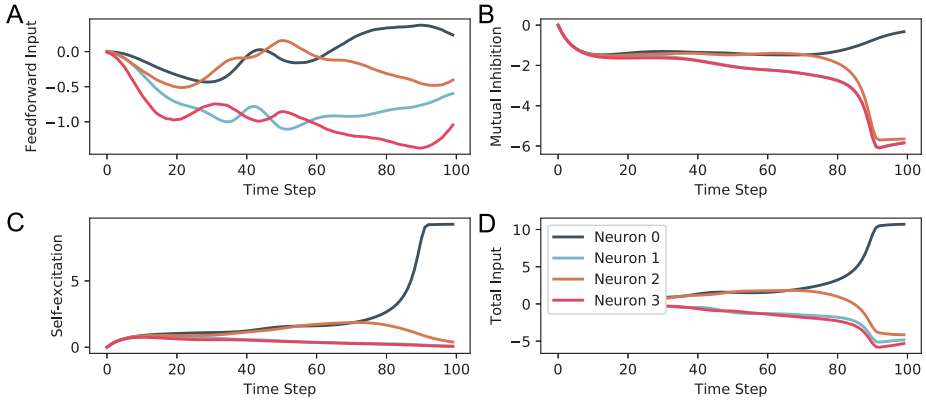

Fig. 6. The input currents to decision-making neurons over time as in the task in Fig. 5. Pattern 1 is presented and decision-making neuron 1 is designated to encode this pattern. A. The feedforward inputs. B. The inhibitory currents from other neurons due to mutual inhibition. C. The excitatory currents due to self-excitation. D. The total inputs, the sum of all currents. The parameters are the same as in Fig. 5.

3.2. Extracting order information of temporal inputs

Consider a task of differentiating four temporal sequences as shown in Fig. 5A, where patterns 1 and 2 are formed by two bumps of different sizes in distinct orders. To avoid that the model learns to differentiate them relying only on the first bump, we add another two patterns 3 and 4 formed by same-sized bumps. As shown in Fig. 5B, the model accomplishes the task, in terms of that when a temporal pattern is presented, the corresponding neuron representing this pattern always generates higher responses than others.

Remarkably, the neurons hierarchically make their decisions, capturing the fact that the temporal differences between four patterns are unfolded gradually over time. Take the top panel in Fig. 5B as an example, where pattern 1 is presented to the model. It shows that: (1) around time step 30, which is the moment when patterns 1 and 3 become sufficiently different from patterns 2 and 4 (see Fig. 5A), the activities of neurons 1 and 3 start to surpass those of neurons 2 and 4; (2) around time step 75, which is the moment when the accumulated information is sufficient to judge the input pattern is 1, the activity of neuron 1 starts to surpass that of neuron 3. This hierarchical decision-making process is also reflected in the amplitude changes in the input currents received by different neurons (see Fig. 6), where through the joint effects of feedforward connections and mutual inhibition, the total current received by the winning neuron gradually surpasses those to other neurons, exhibiting an event-based decision-making manner. This hierarchical decision-making process agrees with the experimental finding, which showed that monkeys were able to integrate information in parallel for multiple decisions and during the information integration, decisions influence each other such that incorrect decisions gradually subside (Lorteije et al., 2015).

3.3. Flexible event-based pattern recognition

We have demonstrated so far two appealing properties of the brain-inspired decision-making network, one is that by adjusting the strengths of mutual inhibitions between neurons, the model can achieve a speed–accuracy trade-off in decision-making (Fig. 3B), which offers a valuable flexibility for pattern recognition

in practice; the other is that the model can hierarchically discriminate multiple patterns according to their similarities as shown in Figs. 5–6. In addition to these two features, we find that our model has another appealing property, that is, it accomplishes pattern recognition in an event-based manner, in terms of that the model can recognize an input pattern whenever it appears without the need for applying signal detection beforehand. We will demonstrate this property using the gait recognition task below (see Fig. 10 in Section 4.3).

4. Model application

In addition to synthetic data, we also apply our model to real-world problems. One is mimicking the looming pattern experiment where the subcortical visual pathway underlies fast defensive behaviors and corresponds with our model, and the other is the gait recognition task, a typical scenario of spatio-temporal pattern discrimination.

4.1. Looming pattern discrimination

The cortical ventral and dorsal visual pathways are important for visual perception (Milner & Goodale, 1995). However, they are not the only pathways the brain relies on to process visual signals. The subcortical visual pathway also plays significant roles in fear detections (Maior et al., 2011; Tamietto & De Gelder, 2010; Wei et al., 2015), particularly in the fast detection of expanding shadows (looming patterns) from aerially approaching predators (De Franceschi, Vivattanasarn, Saleem, & Solomon, 2016; Yilmaz & Meister, 2013). Moreover, human patients with cortical blindness could still navigate around obstacles (De Gelder et al., 2008; Ffytche et al., 1995; Zeki, 1998) or detect unconscious fearful signals (Morris, DeGelder, Weiskrantz, & Dolan, 2001; Morris, Öhman, & Dolan, 1999; Van den Stock et al., 2011), presumably using the subcortical visual pathway.

Looming patterns (dark expanding shadows) could elicit defensive responses in animals, yet not every looming pattern can induce this behavior. Dark contracting shadows and white contracting/expanding disks are found to be unable to evoke similar behaviors (Kim, Shen, Hsiang, Johnson, & Kerschensteiner, 2020; Yilmaz & Meister, 2013). Mimicking the experimental protocol, we construct looming patterns as follows. The size of the image frame is 11×11, and a light spot is always at the center of the frame. The pixel values for the light spot are 1 and for the background are 0. To make the task more difficult, we add Gaussian white noises of zero mean and standard deviation of 0.1 to each image. Denote the radius of the light spot to be D, which increases over time from D_{min} to D_{max}. $D_{min} = 1$ is used in the present study and D_{max} specifies the size of a looming pattern. Analogous to the experimental setting, we choose the value of D_{max} to be $2k+1$, for $k = 0, 1, \ldots, 5$. For a given pattern of $D_{max} = 2k + 1$, we divide its time duration into $k + 1$ segments, and the size of light spot in the ith segment is $2i - 1$, for $i = 1, \ldots, k + 1$. The speed of a looming pattern is controlled by the number of light spot of the same size presented in each segment, denoted as m, which gives the time duration of the ith element to be $m\Delta t$, and the speed of the looming pattern is quantified to be $v = 1/m$.

We construct three categories of looming patterns with varying sizes and speeds. They are: (1) looming patterns of appropriate sizes and speeds, with $(D_{max}, v) = (9, 0.167)$, $(9, 0.2)$, or $(11, 0.2)$, respectively, and they trigger the innate response; (2) looming patterns of appropriate sizes but small or large speeds, with $(D_{max}, v) = (9, 0.1), (11, 0.5)$, or $(11, 1)$, respectively, and they cannot trigger the innate response; (3) looming patterns of appropriate speeds but small sizes, with $(D_{max}, v) = (1, 0.2), (3, 0.2)$ or $(3, 0.25)$, respectively, and they cannot trigger the innate response. 150 looming patterns are generated for each class. We use n of them for each case as training examples, and the rest as test examples to evaluate the discrimination accuracy of the model. The results for $n = 1, 3, 5$ are shown in Fig. 7. A two-layer reservoir module is applied in this task, with the timescales of the first and second layers set to accommodate the fastest and slowest speeds of the expanding shadows ($v = 1$ and $v = 0.1$) respectively.

Notably, our model shows remarkable generalization ability even when labeled sequences are scarce. This indicates that these pre-defined network structures emerged from millions of years of evolution serve the needs of fast learning of dangerous signals to avoid predators timely. Moreover, our model corresponds well to the subcortical visual pathway that underlies looming pattern detection. The retina holds memory traces of the input optical flow, acting as a reservoir network; while the superior colliculus (SC) acts as a decision-making network. More specifically, VG3 amacrine cells in the retina respond robustly to looming patterns and modulate downstream ganglion cells (Kim et al., 2020); the wide-field vertical cells (May, 2006) in SC can thus integrate information regarding looming patterns and send the information to downstream areas regulating defensive behaviors, such as the amygdala.

4.2. Gait recognition

For the gait recognition task, we collect a gait dataset consisting of 100 subjects, and each subject has 50 gait sequences, with each sequence lasting for 2 seconds and containing 50 image frames.[2] Three tasks of gait discrimination between 5, 10, or 15 people are performed. As in the typical application for gait recognition, where the number of training examples is small, we consider only 5 trials per person as training examples.

To avoid that recognition relies on some side information of subjects, such as the height and shape of the subject, we extract skeletons from the images and normalize them using AlphaPose (Fang, Xie, Tai, & Lu, 2017). To further eliminate the spatial location bias of skeletons, we place them at the center of each frame (Fig. 8). This forces a classifier to carry out recognition relying purely on the spatial–temporal structure of input patterns. Each frame is reshaped to 48×30 before used as input.

We first compare our model with LSTMs and GRUs. The GRU is a variant of the LSTM. LSTMs and GRUs consist of a recurrent hidden layer and a linear readout. We also vary the size of the hidden layer to see the influence of increasing the number of parameters in LSTMs and GRUs. For example, LSTM(20) refers to that the hidden layer in LSTM has 20 neurons and LSTM(50) has 50 hidden neurons. In the 5-class classification scenario, LSTM(20) has a structure of 1440-20-5.

During training, we randomly select 5, 10, or 15 subjects from the dataset to construct tasks of discriminating 5, 10, or 15 people. For RDMN, only one layer ($L = 1, N = 1000$) is used in the reservoir module, as the temporal frequencies of the gaits of different subjects are not significantly different and a single reservoir layer suffices to extract the temporal information. RDMN is trained using Force Learning (Sussillo & Abbott, 2009). Since Force Learning converges very fast, we set the training epoch to 1 with a batch size of 1. On the other hand, LSTMs and GRUs are trained using backpropagation through time (BPTT) to minimize the cross-entropy loss with the ADAM optimizer (learning rate of 0.01, gradient clip 1.0). The number of training epochs for LSTMs and GRUs is 50 with a batch size of 16. The cross-entropy loss is accumulated at every time step, as this gives a better performance than minimizing the cross-entropy loss at

[2] Dataset available online: https://tinyurl.com/usnsasr.

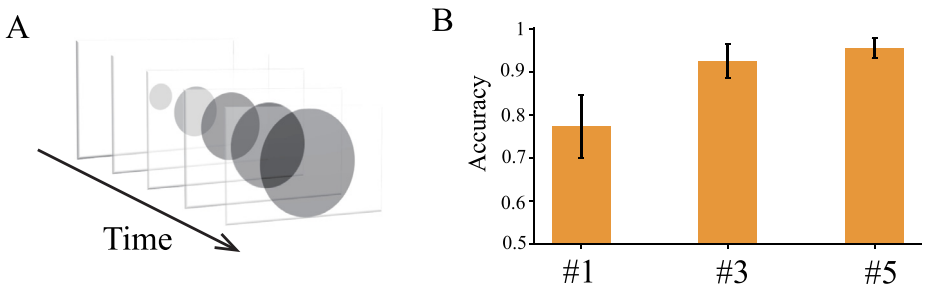

Fig. 7. Looming pattern discrimination. (A) An example of a looming pattern. (B) The accuracies of discriminating three categories of looming patterns. #n means that n trials for each class are used as training examples. The results are obtained by averaging over 100 testing trials. The parameters are: $N_{dm} = 3$, $L = 2$, $N = [200, 200]$, $\tau = [10, 100]$, $\rho = [1.1, 1.1]$, $w_{ext}^{1,0} = 10$, $w_f^{2,1} = 25$, $\alpha = 1.5$, $\beta = 4$, $\theta = 5$, $\gamma = 0.1$, $J_E = 8$, $J_M = -3$, $I_0^* = 0.7$, $\tau_s = 200$. Error bars represent standard deviations.

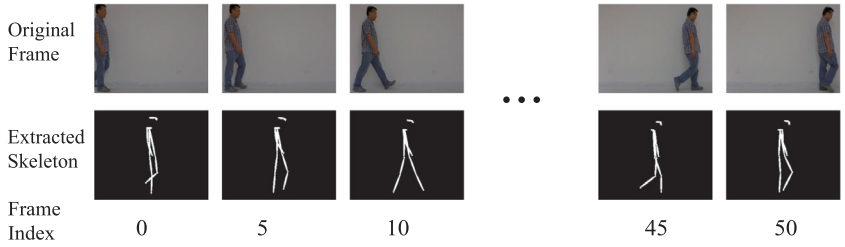

Fig. 8. An example of the gait of a subject.

Table 3
Comparing gait recognition performances (%) of different methods. Three tasks of discriminating 5,10 or 15 subjects are performed. 5 training examples per subject are used. LSTM(20) or GRU(50) refers to that the number of units in the hidden layer is 20 or 50, respectively. For RDMN, the reservoir module has one layer ($L = 1$) with $N = 1000$ neurons. The averaged neural activities over all time steps are used to calculate LSTM/GRU performance, as this practice gives better results.

Model	5 classes	10 classes	15 classes
LSTM(20)	92.4 ± 3.9	83.9 ± 3.3	79.5 ± 3.9
LSTM(50)	94.3 ± 2.0	85.7 ± 2.9	81.5 ± 3.1
LSTM(100)	90.6 ± 3.6	79.5 ± 3.2	76.6 ± 2.1
GRU(20)	92.4 ± 2.5	82.2 ± 3.7	81.3 ± 2.9
GRU(50)	95.4 ± 2.1	88.2 ± 3.2	85.7 ± 1.8
GRU(100)	96.4 ± 2.0	90.5 ± 2.1	89.7 ± 1.9
RDMN	**98.3 ± 1.0**	**93.4 ± 2.2**	**92.4 ± 2.5**

Table 4
The number of trainable parameters in different methods.

Model	5 classes	10 classes	15 classes
LSTM-20	123,385	123,490	123,595
LSTM-50	314,455	314,710	314,965
LSTM-100	616,905	617,410	617,915
GRU-20	87,765	87,870	87,975
GRU-50	223,905	224,160	224,415
GRU-100	462,805	463,310	463,815
RDMN	5000	10,000	15,000

Table 5
The parameters of RDMN used for the gait recognition task. L is the number of reservoir module layers. N is the number of units in the reservoir module layer. The rest symbols are annotated in the text.

	Hyper-params	5 Classes	10 Classes	15 Classes
Reservoir parameters	τ	3	3	3
	ρ	1.1	1.1	1.1
	L	1	1	1
	N	1000	1000	1000
Decision-making parameters	τ_s	10	10	10
	β	4	4	4
	α	1.5	1.5	1.5
	γ	0.1	0.1	0.1
	θ	3	1	1
	J_E	4	6	8
	J_M	−4	−4	−4
	I_0^*	1.6	2	4

only $t = 50$. When testing, RDMN is evaluated by reporting the neuron identity with the largest response. LSTMs and GRUs are evaluated by averaging their choices over every time step, as this gives slightly better results than only evaluating the last time step results.

The number of examples used for training, validation, and testing is 5, 15, and 30 respectively. Since the number of training examples is very small and over-fitting occurs easily, we use validation examples to select the best-trained model. The model performances are obtained by averaging over 20 training experiments. The results are summarized in Table 3, which shows significant performance improvement of our model compared to LSTM/GRU, especially when the task gets harder.

It is also remarkable that our model contains much fewer numbers of trainable parameters than LSTMs and GRUs. For example, the number of trainable parameters in our model in the case of 5-class classification is 5000; whereas they are 123 385, 314 455, 87 765 and 223 905 for LSTM (20), LSTM (50), GRU (20) and GRU (50), respectively (see Table 4). The parameters of RDMN used in this study are summarized in Table 5. We also check whether sparse connections can be adopted when reading-out information from the reservoir module (as this decreases the number of trainable parameters), but find that sparse connections degrade the model performance dramatically (see Appendix C).

Overall, we observe that our model works well for gait recognition, in particular when the number of training examples is small.

4.3. Event-based gait recognition

A key characteristic of biological decision-making is its event-based nature, i.e., the neural system will automatically detect and recognize the presence of an input pattern. This property is appealing for real-world applications, as it saves the effort of signal detection. We explore whether our model has this nice property.

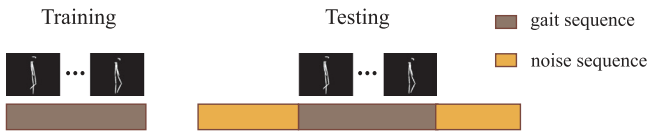

Fig. 9. The event-based gait recognition task. The training patterns are the same as in Fig. 8, while the testing patterns are padded gait sequences, in which the starting point of gait is randomized and the signal is corrupted with Gaussian white noises of zero mean and 0.1 standard deviation.

Table 6
Performance comparison in the event-based gait recognition task (%). Three tasks of discriminating 5,10 or 15 subjects are performed.

Model	5 classes	10 classes	15 classes
Linear	93.3 ± 3.0	79.6 ± 2.7	76.4 ± 2.5
RDMN	**98.3** ± **0.7**	**93.2** ± **2.4**	**92.3** ± **2.7**

To demonstrate this, we construct a new gait recognition task, in which the training patterns remain to be the same as illustrated in Fig. 8, whereas, for testing patterns, we randomize the starting moments of gaits and corrupt signals with Gaussian white noises. Specifically, we pad the original gait sequences of length $T = 50$ to $T = 200$ with gaits start at anytime between $[0, 150]$, and the whole sequence is corrupted with Gaussian noises (see Fig. 9).

Since LSTMs and GRUs do not work in the event-based recognition task, as they require the training and testing sequences have the same structure, we construct an event-based linear classifier as a baseline method for performance comparison. Specifically, we consider that the linear classifier makes decisions by integrating information over a time interval of 50 (i.e., the length of gait sequences) and reports results frequently using a sliding time window. This linear classifier inherits some merits of RDMN (i.e. moving-average, evidence-integration) but lacks mutual inhibition and self-excitation between neurons. For details of the linear classifier, see Appendix D.

Examples of the decision-making process of RDMN are presented in Fig. 10. The recognition is accomplished whenever the activity of a neuron crosses the threshold. We see that the moment of neuronal firing agrees well with the appearance of the gait, exhibiting the event-based nature. Table 6 compares the performances of RDMN with that of the linear classifier, demonstrating that RDMN outperforms the linear classifier significantly.

5. Conclusion and discussion

In this study, we have proposed a model of RDMN for spatio-temporal pattern recognition, inspired by the structures and functions of the subcortical visual pathway and early stages of the auditory pathway. We showed that this model could potentially explain the fast defensive behaviors in animals under looming pattern presentations. We demonstrated that our model outperforms LSTMs/GRUs when the number of training examples is small, and this is achieved using a much fewer number of trainable parameters. Furthermore, our model accomplishes pattern recognition in an event-based manner. These advantages make our model potentially applicable for spatio-temporal pattern recognition in edge computing devices.

The RDMN model consists of two modules, a reservoir network, and a decision-making network. Different from prevalent feature extraction (followed by feature aggregation) methods, features are not explicitly extracted in the reservoir network. Rather, the reservoir module holds a fading memory of the input via its recurrent transient dynamics, essentially mapping entangled low dimensional inputs into a high dimensional activity space to be linearly separable. The decision-making network aims to discriminate these patterns by integrating neural activities over time. We gave a detailed explanation of how the two modules work jointly.

It is worth noting that it is well acknowledged that the brain employs parallel computing strategies in both visual and auditory pathways (Nassi & Callaway, 2009; Rauschecker, 1998; Rauschecker, Tian, Pons, & Mishkin, 1997), with each pathway serving different cognitive purposes. Thus, while the pathways we modeled in this study do not extract input features explicitly, spatio-temporal patterns are almost always processed by some other feature-extraction pathways simultaneously. Evolutionally speaking, these different pathways aim to serve different functions under different circumstances. For example, the subcortical visual pathway has been found to mediate fear response (Morris et al., 1999; Shang et al., 2015; Wei et al., 2015), crucial for fast response to predators and other dangers, whereas the ventral and dorsal visual pathways are more suitable for fine-grained visual information extraction. Surviving in nature requires animals to selectively take advantage of the processed information from different pathways in different scenarios. Consequently, we do not claim our model is better than LSTMs/GRUs in general but emphasize that it performs better than these feature extraction methods when training samples are limited, and it also requires fewer trainable parameters. This is because millions of years of evolutionary pressure have forced the brain to come up with a pathway for fast and efficient spatio-temporal information processing.

Although in our experiments, RDMN has demonstrated its ability to memorize order information thanks to the recurrence in the reservoir network, it does not explicitly encode order information. This is a defect in both our model and deep learning methods. The order of a spatio-temporal pattern contains important cause–effect information and temporal correlations, which we humans actively exploit to discriminate spatio-temporal patterns. Thus, incorporating this prior knowledge into models should significantly improve the performance of spatio-temporal pattern discrimination tasks. Theoretically, order information can be encoded by a neural trajectory visiting a set of saddle–nodes in order (heteroclinic channels (Rabinovich et al., 2008)). It remains unclear how to map spatio-temporal patterns onto these heteroclinic channels, and this will be our future work direction.

5.1. Related works

Spatio-temporal patterns in the machine learning context usually refer to videos. In this domain, action recognition is one of the most frequently considered tasks. Aided by deep learning, current models for action recognition usually take a feature extraction with a subsequent feature aggregation approach. Before the booming of deep learning, features are usually obtained by either using hand-crafted features (Dalal & Triggs, 2005; Lowe, 1999; Wang & Schmid, 2013) or via unsupervised learning (Hinton, Osindero, & Teh, 2006; Le, Zou, Yeung, & Ng, 2011; Lee, Battle, Raina, & Ng, 2007). Deep learning methods, on the other hand, take advantage of a large amount of labeled data, and train models in a supervised manner. These models either extract single frame features using convolutional neural networks and later fuse them together (Jiang, Wu, Wang, Xue, & Chang, 2017; Karpathy et al., 2014; Yue-Hei Ng et al., 2015), or use 3D convolution (Baccouche, Mamalet, Wolf, Garcia, & Baskurt, 2011; Ji, Xu, Yang, & Yu, 2012; Tran, Bourdev, Fergus, Torresani, & Paluri, 2015), with the additional dimension accounting for the temporal information. Some studies also use LSTM architecture for better handling temporal structures of various lengths (Donahue et al., 2015; Lee, Kim, Kang, & Lee, 2017; Wu, Wang, Jiang, Ye, & Xue, 2015).

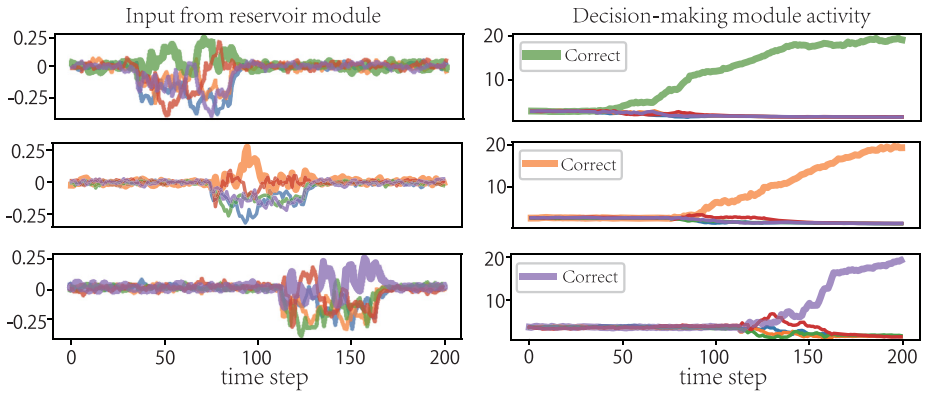

Fig. 10. The neural dynamics of the decision-making module in the event-based gait recognition task. The decision-making module integrates inputs from the reservoir network nicely, such that when the gait of a subject appears, the corresponding neuron starts to integrate information and eventually fires. Left column: the inputs from the reservoir module to the decision-making module. Right column: the responses of decision-making neurons.

Several deep learning models are also brain-inspired. Karpathy et al. (2014) used a multi-resolution stream the mimics the fovea and periphery vision to effectively reduce the number of parameters. Simonyan and Zisserman (2014) proposed a two stream architecture consisting of a spatial stream to extract features of still images and a temporal stream to extract features of motion fields, mimicking the ventral and dorsal pathways in cortical visual stream. Other deep learning methods mostly engineer their way through combining 3D convolution with two-stream models (Carreira & Zisserman, 2017) or improving performance by introducing more streams (Zhu, Lan, Newsam, & Hauptmann, 2018) or incorporating 3D convolution with more powerful network structures (Qiu, Yao, & Mei, 2017) or normalization techniques (Tran, Wang, Torresani, & Feiszli, 2019).

Our model does not extract features explicitly, rather it adopts a reservoir module to hold the temporal information of image sequences. While feature extraction methods tend to behave better when training data is abundant, methods such as ours are nontrivial, in the sense that they partly explain the data-efficient property of our brain thanks to millions of years of evolution. Our model follows the structure and function of the subcortical visual pathway and early stages of the auditory pathway, both are very conservative under evolution. The brain arguably employs both evolutionarily tuned architecture with an experience-dependent fine-tuning process to be data efficient.

There have been a lot of researches on reservoir networks for spatio-temporal pattern processing, but these works typically considered very simple tasks on pattern discrimination or pattern generation, with a focus on exploring the biologically plausible learning rules and/or the biological implications of the models (DePasquale, Cueva, Rajan, Escola, & Abbott, 2018; Jaeger, Lukoševičius, Popovici, & Siewert, 2007; Karmarkar & Buonomano, 2007; Kim & Chow, 2018; Laje & Buonomano, 2013; Maes, Barahona, & Clopath, 2020; Miconi, 2017; Rombouts, Bohte, & Roelfsema, 2015). They have not included a decision-making module to facilitate pattern discrimination as we do in this work, let alone solving real-world problems. Decision-making networks have been widely used in the neuroscience community for interpreting experimental data (Shadlen & Newsome, 2001; Wong & Wang, 2006), but they have not been applied to spatio-temporal pattern recognition, nor introduced to the machine learning community. Kurikawa, Haga, Handa, Harukuni, and Fukai (2018) used a similar architecture with a reservoir network followed by a decision-making model trained with reinforcement learning to explain the individual variability in the decision-making process of monkeys. However, their model is restricted to discriminating between 2 classes and aims at explaining experimental data. To our knowledge, our model is the first one that integrates reservoir computing and decision-making as a unified framework for spatio-temporal pattern discrimination. We hope this work can be a stepping stone for utilizing brain-inspired algorithms to process spatio-temporal data in the future.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by Guangdong Province with grant (No. 2018B030338001, S. Wu, Y. Mi), Huawei Technology Co., Ltd (No. YBN2019105137 S. Wu; No. YBN2020095106 Y. Mi), Beijing Nova Program (No. Z181100006218118, Y. Mi), the National Natural Science Foundation of China (No. 4861425025, T.J. Huang; No. 31771146, No. 11734004, Y. Mi) and the Fundamental Research Funds for the Central Universities (2020CDJQY-A073, Y. Mi).

Appendix A. The phase diagram of the decision-making module with $N_{dm} > 2$ neurons

We calculate the phase diagrams of the network with a varying number of decision-making neurons ($N_{dm} = 5, 10, 15, 20$) as shown in Fig. A.1. The optimal parameter regime in each case is chosen under the same rationale as described in Section 2.1.1, which is the DM-boundary in each case.

Appendix B. Data processing for neural representation analysis in Fig. 4B

Consider two temporal sequences are in the time interval $(0, T)$, we discretize time with bins of size Δt. Denote $\mathbf{R}^k(m)$ a vector representing the overall activity of the hierarchical reservoir module at moment $t = m\Delta t$ in response to spatiotemporal sequence k, for $k = 1, 2$, whose element is given by $R_i^k(m) = r_j^l(m)$, for $j = 1, \ldots, N_l$ and $l = 1, \ldots, L$, where L and N_l denote the total number of layer and the total number of neurons in each layer, respectively. $r_j^l(m)$ refers to the neuronal activity in the time interval $[(m-1)\Delta t, m\Delta t]$. The dimensionality of $\mathbf{R}^k(m)$ is N, with $N = \sum_{l=1}^{L} N_l$. Combining the network activities at all moments, we obtain a matrix $\mathbf{S}^k = \{\mathbf{R}^k\}$, whose dimensionality is $N \times M$, with $M = T/\Delta t$. Define a variance matrix $\mathbf{A} =$

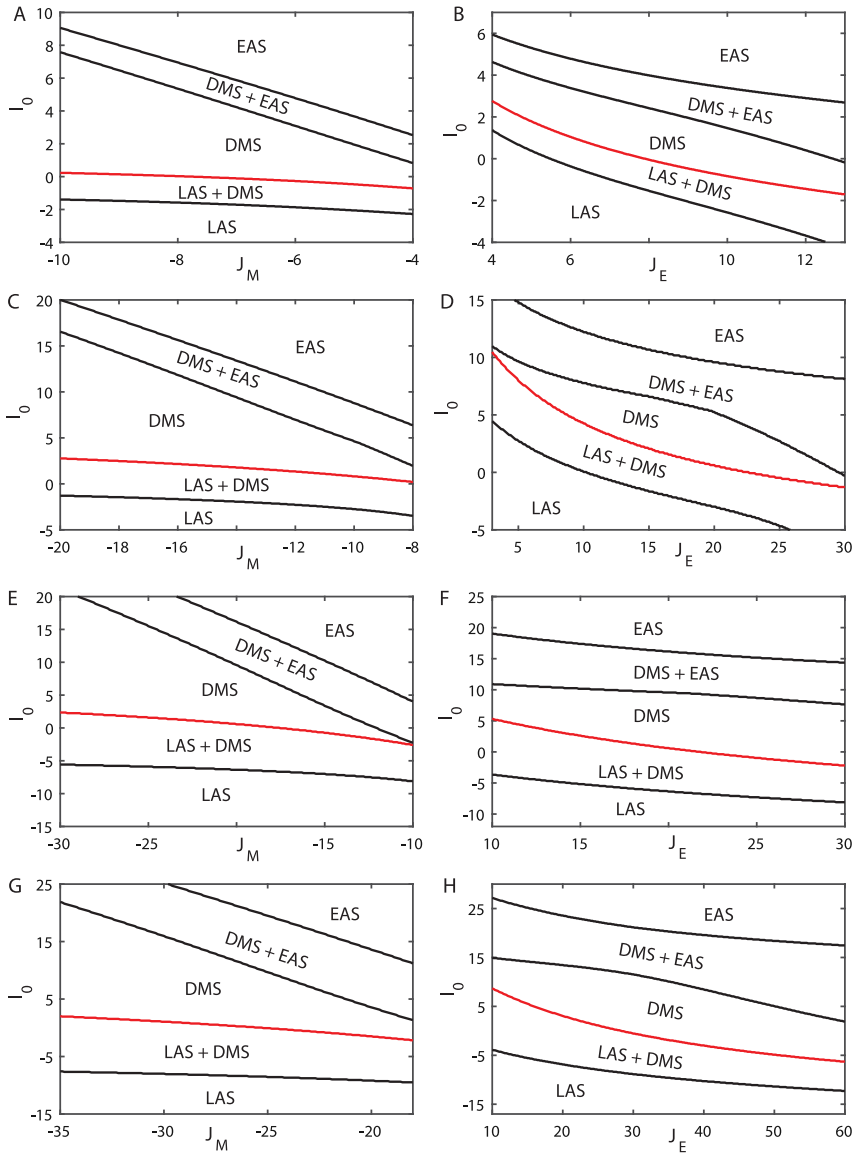

Fig. A.1. The phase diagrams of the decision-making module with varying number of neurons. The stationary states of the network in different parameter regimes are shown. LAS: all neurons are at low-level activity states; DMS: only one neuron is at a high-level activity state; EAS: two or more than two neurons are at high-level activity states. The red lines denote the DM-boundary in each case. (A, C, E, G) the feedforward input I_0 vs. the mutual inhibition J_M. (B, D, F, H) the feedforward input I_0 vs. the self-excitation J_E. The parameters are: (A–B) $N_{dm} = 5$, $\theta = 3$, $\beta = 3.2$; (C–D) $N_{dm} = 10$, $\theta = 1$, $\beta = 1$; (E-F) $N_{dm} = 15$, $\theta = -5$, $\beta = 0.8$; (G–H) $N_{dm} = 20$, $\theta = -9$, $\beta = 0.5$; (A) $J_E = 9$; (B) $J_M = -5$; (C) $J_E = 18$; (D) $J_M = -11$; (E) $J_E = 20$; (F) $J_M = -20$; (G) $J_E = 27$; (H) $J_M = -27$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$\sum_{k=1}^{2} \left(\mathbf{S}^k - \langle \mathbf{S} \rangle\right) \left(\mathbf{S}^k - \langle \mathbf{S} \rangle\right)^T$, where $\langle \mathbf{S} \rangle$ is obtained by averaging neuronal activities over time and over two inputs.

We apply principle component analysis (PCA) to the variance matrix **A** and obtain a set of eigenvectors with decreasing eigenvalues. The eigenvalue of each component reflects the variability of neural responses over time in that direction. By projecting the network activities on a few chosen principle components, it helps us to visualize whether neural representations for different inputs are separated in the reservoir module.

To obtain Fig. 4B, we first use the network activities in response to each temporal sequence in one trial to carry out PCA as described above, and then we apply the obtained PCs to reduce the dimensionality of neural representations in other trials. By reducing dimensionality, it means to project neural representations on the chosen PCs. In this particular example, we find that by projecting network activities on the third and fourth PCs, the two temporal sequences are well separated. This indicates that if the read-out matrix has large overlaps with these two PCs, the decision-making module can discriminate between the two temporal sequences, which is confirmed by our study.

Appendix C. The impact of sparse reading-out connections

The trainable parameters in our model contain solely the full connections from the reservoir module to the decision-making module. It is natural to ask whether adopting sparse connections could further decrease the number of trainable parameters. We construct sparse connections from the reservoir to decision-making modules by randomly imposing a portion of the reading-out matrix to be zero. We find that sparse connections severely degrade the model performance, as shown in Table A.1. This suggests that for the size of the reservoir module we use, the neuronal information is not redundant.

Table A.1
The model performances under different sparsity of the reading-out matrix. FC denotes full connections.

Sparsity	5 classes	10 classes	15 classes
1.0 (FC)	97.61 ± 1.88	92.37 ± 1.91	93.04 ± 1.32
0.8	86.99 ± 10.62	28.13 ± 13.69	12.62 ± 5.57
0.6	66.45 ± 19.38	18.33 ± 7.34	7.90 ± 2.69
0.4	51.20 ± 16.63	13.36 ± 4.36	7.34 ± 2.52
0.2	42.95 ± 13.23	14.25 ± 4.55	7.75 ± 2.20

Appendix D. The event-based linear classifier

The event-based linear classifier we constructed consists of a weight matrix $\mathbf{W}_l$ of shape $N \times C$ with no bias vector, where N is the number of neurons in the reservoir module and C is the number of subjects. During training, the linear classifier is optimized, such that the inputs of the correct and wrong identities to the decision-making module are 0.1 and -0.1 respectively. For instance, if the correct identity for an input sequence is the first neuron, we optimize $\mathbf{W}_l$, such that $\mathbf{W}_l^T r(t) = [0.1, -0.1, \ldots, -0.1]^T$ at every time point t, where $r(t)$ is a $N \times 1$ vector representing the activity of the reservoir module.

The linear classifier is evaluated by counting votes of all output units. Concretely, we treat the largest output at each time point as a vote for the corresponding identity. Whenever an identity has 20 more votes than other categories in a sliding time window of length 50 (the length of a gait sequence), a gait event is detected and the recognition is reported. The threshold 20 for the linear classifier is set, since it gives the best performance on the validation set.

References

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., & Baskurt, A. (2011). Sequential deep learning for human action recognition. In *International Workshop on Human Behavior Understanding* (pp. 29–39). Springer.

Bertschinger, N., & Natschläger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, *16*(7), 1413–1436.

Bertschinger, N., Natschläger, T., & Legenstein, R. A. (2005). At the edge of chaos: Real-time computations and self-organized criticality in recurrent neural networks. In *Advances in Neural Information Processing Systems* (pp. 145–152).

Buonomano, D. V., & Maass, W. (2009). State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, *10*(2), 113–125.

Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6299–6308).

Cayco-Gajic, N. A., & Silver, R. A. (2019). Re-evaluating circuit mechanisms underlying pattern separation. *Neuron*, *101*(4), 584–602.

Chaudhuri, R., Knoblauch, K., Gariel, M.-A., Kennedy, H., & Wang, X.-J. (2015). A large-scale circuit mechanism for hierarchical dynamical processing in the primate cortex. *Neuron*, *88*(2), 419–431.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *1*, In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (pp. 886–893). IEEE.

De Franceschi, G., Vivattanasarn, T., Saleem, A. B., & Solomon, S. G. (2016). Vision guides selection of freeze or flight defense strategies in mice. *Current Biology*, *26*(16), 2150–2154.

De Gelder, B., Tamietto, M., Van Boxtel, G., Goebel, R., Sahraie, A., Van den Stock, J., et al. (2008). Intact navigation skills after bilateral loss of striate cortex. *Current Biology*, *18*(24), R1128–R1129.

DePasquale, B., Cueva, C. J., Rajan, K., Escola, G. S., & Abbott, L. (2018). Full-FORCE: A target-based method for training recurrent networks. *PloS One*, *13*(2).

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., et al. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2625–2634).

Fang, H.-S., Xie, S., Tai, Y.-W., & Lu, C. (2017). RMPE: Regional multi-person pose estimation. In *ICCV*.

Ffytche, D. H., Guy, C., & Zeki, S. (1995). The parallel visual motion inputs into areas V1 and V5 of human cerebral cortex. *Brain*, *118*(6), 1375–1394.

Gale, S. D., & Murphy, G. J. (2014). Distinct representation and distribution of visual information by specific cell types in mouse superficial superior colliculus. *Journal of Neuroscience*, *34*(40), 13458–13471.

Herath, S., Harandi, M., & Porikli, F. (2017). Going deeper into action recognition: A survey. *Image and Vision Computing*, *60*, 4–21.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Honey, C. J., Thesen, T., Donner, T. H., Silbert, L. J., Carlson, C. E., Devinsky, O., et al. (2012). Slow cortical dynamics and the accumulation of information over long timescales. *Neuron*, *76*(2), 423–434.

Hung, C. P., Kreiman, G., Poggio, T., & DiCarlo, J. J. (2005). Fast readout of object identity from macaque inferior temporal cortex. *Science*, *310*(5749), 863–866.

Jaeger, H. (2001). *The "echo state" approach to analysing and training recurrent neural networks-with an erratum note, Vol. 148* (34), (p. 13). Bonn, Germany: German National Research Center for Information Technology GMD Technical Report.

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

Jaeger, H., Lukoševičius, M., Popovici, D., & Siewert, U. (2007). Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Networks*, *20*(3), 335–352.

Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 221–231.

Jiang, Y.-G., Wu, Z., Wang, J., Xue, X., & Chang, S.-F. (2017). Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(2), 352–364.

Karmarkar, U. R., & Buonomano, D. V. (2007). Timing in the absence of clocks: encoding time in neural network states. *Neuron*, *53*(3), 427–438.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1725–1732).

Kiang, N., Rho, J., Northrop, C., Liberman, M., & Ryugo, D. K. (1982). Hair-cell innervation by spiral ganglion cells in adult cats. *Science*, *217*(4555), 175–177.

Kim, C. M., & Chow, C. C. (2018). Learning recurrent dynamics in spiking networks. *ELife*, *7*, Article e37124.

Kim, T., Shen, N., Hsiang, J.-C., Johnson, K., & Kerschensteiner, D. (2020). Dendritic and parallel processing of visual threats in the retina control defensive responses. *Science Advances*, *6*(47), eabc9920.

Kurikawa, T., Haga, T., Handa, T., Harukuni, R., & Fukai, T. (2018). Neuronal stability in medial frontal cortex sets individual variability in decision-making. *Nature Neuroscience*, *21*(12), 1764–1773.

Laje, R., & Buonomano, D. V. (2013). Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nature Neuroscience*, *16*(7), 925.

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, *64*(2–3), 107–123.

Le, Q. V., Zou, W. Y., Yeung, S. Y., & Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR 2011* (pp. 3361–3368). IEEE.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lee, H., Battle, A., Raina, R., & Ng, A. Y. (2007). Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems* (pp. 801–808).

Lee, I., Kim, D., Kang, S., & Lee, S. (2017). Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1012–1020).

Levy, K. L., & Kipke, D. R. (1997). A computational model of the cochlear nucleus octopus cell. *The Journal of the Acoustical Society of America*, *102*(1), 391–402.

Lorteije, J. A., Zylberberg, A., Ouellette, B. G., De Zeeuw, C. I., Sigman, M., & Roelfsema, P. R. (2015). The formation of hierarchical decisions in the visual cortex. *Neuron*, *87*(6), 1344–1356.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *2*, In *Proceedings of the Seventh IEEE International Conference on Computer Vision* (pp. 1150–1157). Ieee.

Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, *3*(3), 127–149.

Maes, A., Barahona, M., & Clopath, C. (2020). Learning spatiotemporal signals using a recurrent spiking network that discretizes time. *PLoS Computational Biology*, *16*(1), Article e1007606.

Maior, R. S., Hori, E., Barros, M., Teixeira, D. S., Tavares, M. C. H., Ono, T., et al. (2011). Superior colliculus lesions impair threat responsiveness in infant capuchin monkeys. *Neuroscience Letters*, *504*(3), 257–260.

May, P. J. (2006). The mammalian superior colliculus: laminar structure and connections. *Progress in Brain Research*, *151*, 321–378.

Mazor, O., & Laurent, G. (2005). Transient dynamics versus fixed points in odor representations by locust antennal lobe projection neurons. *Neuron*, *48*(4), 661–673.

Miconi, T. (2017). Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *Elife*, *6*, Article e20899.

Milner, A., & Goodale, M. (1995). *Oxford psychology series, No. 27. The visual brain in action*. Oxford University Press New York.

Morris, J. S., DeGelder, B., Weiskrantz, L., & Dolan, R. J. (2001). Differential extrageniculostriate and amygdala responses to presentation of emotional faces in a cortically blind field. *Brain*, *124*(6), 1241–1252.

Morris, J. S., Öhman, A., & Dolan, R. J. (1999). A subcortical pathway to the right amygdala mediating "unseen" fear. *Proceedings of the National Academy of Sciences*, *96*(4), 1680–1685.

Murray, J. D., Bernacchia, A., Freedman, D. J., Romo, R., Wallis, J. D., Cai, X., et al. (2014). A hierarchy of intrinsic timescales across primate cortex. *Nature Neuroscience*, *17*(12), 1661–1663.

Nassi, J. J., & Callaway, E. M. (2009). Parallel processing strategies of the primate visual system. *Nature Reviews Neuroscience*, *10*(5), 360–372.

Nikolić, D., Häusler, S., Singer, W., & Maass, W. (2009). Distributed fading memory for stimulus properties in the primary visual cortex. *PLoS Biology*, *7*(12), Article e1000260.

Niyogi, S. A., & Adelson, E. H. (1994). Analyzing gait with spatiotemporal surfaces. In *Proceedings of 1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects* (pp. 64–69). IEEE.

Qiu, Z., Yao, T., & Mei, T. (2017). Learning spatio-temporal representation with pseudo-3d residual networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 5533–5541).

Rabinovich, M., Huerta, R., & Laurent, G. (2008). Transient dynamics for neural processing. *Science*, 48–50.

Rauschecker, J. P. (1998). Parallel processing in the auditory cortex of primates. *Audiology and Neurotology*, *3*(2–3), 86–103.

Rauschecker, J. P., Tian, B., Pons, T., & Mishkin, M. (1997). Serial and parallel processing in rhesus monkey auditory cortex. *Journal of Comparative Neurology*, *382*(1), 89–103.

Rombouts, J. O., Bohte, S. M., & Roelfsema, P. R. (2015). How attention can create synaptic tags for the learning of working memories in sequential tasks. *PLoS Computational Biology*, *11*(3).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Shadlen, M. N., & Newsome, W. T. (2001). Neural basis of a perceptual decision in the parietal cortex (area LIP) of the rhesus monkey. *Journal of Neurophysiology*, *86*(4), 1916–1936.

Shang, C., Liu, Z., Chen, Z., Shi, Y., Wang, Q., Liu, S., et al. (2015). A parvalbumin-positive excitatory visual pathway to trigger fear responses in mice. *Science*, *348*(6242), 1472–1477.

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems* (pp. 568–576).

Spoendlin, H. (1974). Neuroanatomy of the cochlea. In *Facts and Models in Hearing* (pp. 18–32). Springer.

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

Tamietto, M., & De Gelder, B. (2010). Neural bases of the non-conscious perception of emotional signals. *Nature Reviews Neuroscience*, *11*(10), 697–709.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4489–4497).

Tran, D., Wang, H., Torresani, L., & Feiszli, M. (2019). Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 5552–5561).

Van den Stock, J., Tamietto, M., Sorger, B., Pichon, S., Grézes, J., & de Gelder, B. (2011). Cortico-subcortical visual, somatosensory, and motor activations for perceiving dynamic whole-body emotional expressions with and without striate cortex (v1). *Proceedings of the National Academy of Sciences*, *108*(39), 16188–16193.

Wang, H., & Schmid, C. (2013). Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 3551–3558).

Wei, P., Liu, N., Zhang, Z., Liu, X., Tang, Y., He, X., et al. (2015). Processing of visually evoked innate fear by a non-canonical thalamic pathway. *Nature Communications*, *6*(1), 1–13.

Wong, K.-F., & Wang, X.-J. (2006). A recurrent network mechanism of time integration in perceptual decisions. *Journal of Neuroscience*, *26*(4), 1314–1328.

Wu, Z., Wang, X., Jiang, Y.-G., Ye, H., & Xue, X. (2015). Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM International Conference on Multimedia* (pp. 461–470).

Xie, S., Sun, C., Huang, J., Tu, Z., & Murphy, K. (2018). Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 305–321).

Yildiz, I. B., Jaeger, H., & Kiebel, S. J. (2012). Re-visiting the echo state property. *Neural Networks*, *35*, 1–9.

Yilmaz, M., & Meister, M. (2013). Rapid innate defensive responses of mice to looming visual stimuli. *Current Biology*, *23*(20), 2011–2015.

Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4694–4702).

Zeki, S. (1998). Parallel processing, asynchronous perception, and a distributed system of consciousness in vision. *The Neuroscientist*, *4*(5), 365–372.

Zhu, Y., Lan, Z., Newsam, S., & Hauptmann, A. (2018). Hidden two-stream convolutional networks for action recognition. In *Asian Conference on Computer Vision* (pp. 363–378). Springer.