Discovering Novel Circuit Mechanisms in Higher Cognition through Interpretable Recurrent Neural Network Training

Yiteng Zhang[1,2], Xingyu Li[1], Xuewen Shen[1,3], Jianfeng Feng[2,4,5], Gouki Okazawa[6], Liping Wang[6] and Bin Min[1]

1 Lingang Laboratory, Shanghai, China

2 School of Data Science, Fudan University, Shanghai, China

3 School of Physics, Center for Quantitative Biology, Peking University, Beijing, China

4 Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai, China

5 Key Laboratory of Computational Neuroscience and Brain-Inspired Intelligence, Fudan University, Ministry of Education, Shanghai, China

6 Institute of Neuroscience, Key Laboratory of Brain Cognition and Brain-Inspired Intelligence Technology, CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai, China.

Abstract

Training recurrent neural networks (RNNs) has revolutionized the way how systems neuroscientists form hypothesis when studying circuit mechanisms in various problems. However, the trained RNNs oftentimes are difficult to be interpreted, inconsistent with neural data or not necessarily comprising the full set of biological solutions. Here, we developed an interpretable RNN training framework, namely Restricted-RNN, capable of generating interpretable circuit hypothesis through a multilevel proposing-and-testing procedure that seamlessly integrates computational-, collective- and implementational-level descriptions. We demonstrated its validity in identifying data-compatible circuit mechanisms through a variety of macaque cognitive tasks, including parametric working memory, sequence working memory and perceptual decision-making. The key derived predictions were confirmed by monkey prefrontal and parietal neurophysiological data. Critically, the interpretable nature of Restricted-RNN endowed us a unified theory to explain the seemingly disparate phenomena across different tasks with a novel neural control state space, providing an intriguing geometric understanding for the ubiquitous control in cognitive processes.

Introduction

Elucidating circuit mechanisms underlying complex behaviors arguably is one of the major endeavors in neuroscience. The enormous complexity exhibited in single neural activities (Rigotti et al., 2013; Tye et al., 2024), however, poses a major challenge towards fulfilling this grand endeavor. One promising and influential idea to address this challenge is the computation-through-dynamics framework in which the neural computation underlying behavior is casted as a dynamical process in high-dimensional neural state space, capable of accommodating both single-neuron-level complexity and collective-level simplicity (Fig. 1A; (DePasquale et al., 2023; Vyas et al., 2020)). This framework is further corroborated with a new modeling approach that leverages the artificial recurrent neural networks (RNNs) to emulate cognitive and motor tasks (Fig. 1B), which has been widely adopted to generate new circuit mechanism hypotheses (Cueva & Wei, 2018; Driscoll et al., 2024; Langdon & Engel, 2025; Mante et al., 2013; Sussillo et al., 2015; Yang et al., 2019). While extremely powerful, this RNN-based modeling approach also inherits the "black-box" property from deep-learning approaches, oftentimes leading to models difficult to be interpreted, inconsistent with neural data or not comprising the full space of biological solutions (Pagan et al., 2024). This key limitation hinders the further applications of this approach in generating circuit mechanism hypotheses for challenging problems, calling for a major revision or an alternative with higher interpretability.

We interrogated the origin of this "black-box" issue from a cross-level modeling perspective—a core concept in cognitive science proposed to account for the cross-level nature of complex neural systems (Marr, 2010). In the lens of cross-level modeling, while the collective-level computation indeed arises from the RNN with trained single-neuron-level connectivity, it is unclear what kinds of properties of the trained connectivity (with about 10^6 parameters for an RNN with 1000 neurons) determine the collective-level computation (i.e., the interactions between a handful of latent factors). In other words, there is a major gap from the trained single-neuron-level connectivity to the collective-level understanding (Fig. 1B), the origin of model interpretability and other related issues.

Here, we propose an interpretable RNN-based modeling approach, termed as Restricted-RNN, capable of closing this gap. Restricted-RNN is based upon a factor-centric view of neural computation which regards neural populations as the substrate mediating the communication between factors (DePasquale et al., 2023). By introducing a novel generative model of network connectivity (Fig. 1C), Restricted-RNN can directly train "*collective-level connectivity weights*" that mediate the communication between factors (Fig. 1D), which is in sharp contrast to the single-neuron-level connectivity weight training in conventional RNN (Fig. 1B). By doing so, the trained models in Restricted-RNN are automatically endowed with high interpretability as both connectivity training and model understanding occurs at the same collective level. Importantly, the connectivity priors (a form of connectivity restriction) can be flexibly imposed at the collective level in Restricted-RNN (Fig. 1D), rather than at the implementational level in conventional RNN (Fig. 1B). This is crucial as this flexible collective-level connectivity restriction allows us to systematically explore the hypothesis space of circuit mechanisms, which is impossible in conventional RNN.

We demonstrated the validity of Restricted-RNN through identifying data-compatible circuit mechanisms in three representative tasks, including parametric working memory (Hernández et al., 1997; Machens et al., 2005; V. Mountcastle et al., 1990; V. B. Mountcastle et al., 1992; Romo et al., 1998, 1999, 2002, 2004), sequence working memory (Chen et al., 2024; Xie et al., 2022) and perceptual decision-making (Okazawa et al., 2021). In particular, Restricted-RNN uncovered *de nova* circuit mechanisms underlying both the sequence working memory gating problem and the counter-intuitive firing rate reversal phenomenon, with the key derived predictions being confirmed by monkey neural physiological data. More importantly, the interpretable nature of Restricted-RNN endowed us a unified theory to explain the seemingly disparate phenomena across different tasks with a novel *neural control state space*, providing an intriguing geometric understanding for the ubiquitous control in cognitive processes. Together, these results strongly suggest that restricted-RNN holds the great promise to uncover novel circuit mechanisms underlying challenging higher cognition problems.

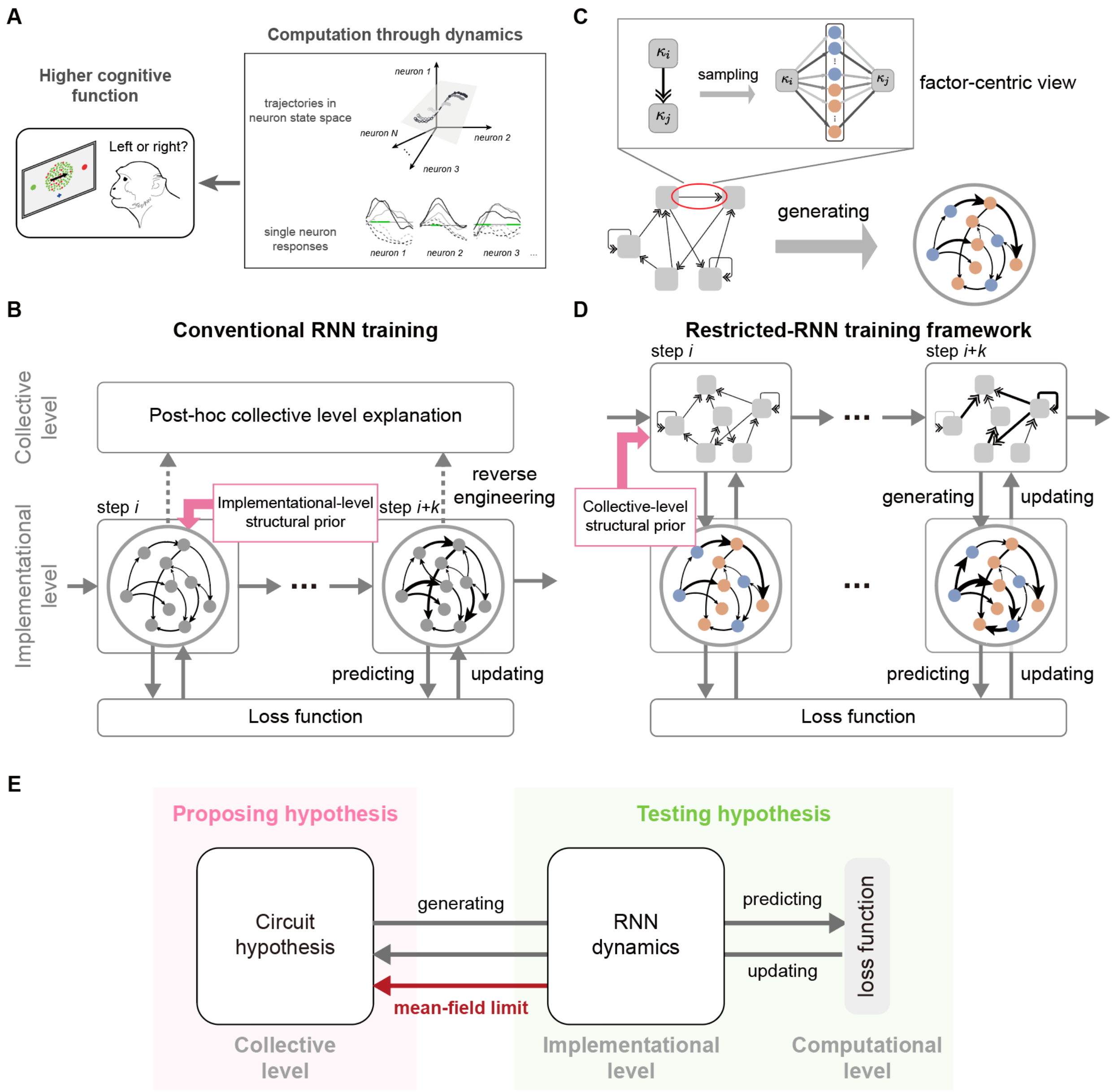

Figure 1. Restricted-RNN: an interpretable RNN training framework for systematic hypothesis proposing and testing.

(A) Dynamical modeling helps to reveal the circuit mechanism underlying cognitive functions, accommodating both single-neuron-level activities and collective-level representations.

(B) In convectional RNN training, one directly updates the connectivity weights among neurons to optimize the performance. The resulting model is a "black box," and interpreting it is hard and often necessitates extensive reverse engineering. Furthermore, prior knowledge of the targeting task can only be integrated at the implementation level by constraining the connection between individual neurons.

(C) Restricted-RNN model. First, a collective-level circuit is built up to specify the factor dynamics. In the circuit, each edge (a double arrow) can be expanded into neuronal basis with modular structure, describing how the two factors communicate through neuron populations (top box). For instance, κ_i first connects to two population modules (blue and red), which then connect to κ_j. The connection from the source factor κ_i to a module and the one from that module to the targetting factor κ_j consist of a module-mediated pathway for the communication between κ_i and κ_j. By expanding and combining all the edges, one produces a generative model that maps the collective-level circuit of factors to an implementational-level circuit of neurons, i.e., a Restricted-RNN.

(D) Restricted-RNN training framework. To train a Restricted-RNN, one directly updates the collective-level circuit. Restricted-RNN automatically provides a collective-level interpretation due to the direct mapping between collective-level and implementational-level circuits. In addition, one can constrain the collective-level circuit by imposing structural prior on it, enabling systematic exploration of hypothesis space of circuit mechanism.

(E) Overall picture of Restricted-RNN training framework. The close relationship between the collective-level and implementational-level circuits in Restricted-RNN allows for systematic exploration of the circuit mechanisms underlying cognitive tasks through hypothesis generation and testing.

Results

Overview of Restricted-RNN

To start with, we first introduce the major assumptions made in Restricted-RNN. In comparison with conventional RNN typically assuming a connectivity matrix without a clear statistical structure, Restricted-RNN is based on a statistical description of the connectivity matrix with two major assumptions—the low rank structure and the modular structure (Dubreuil et al., 2022; Mastrogiuseppe & Ostojic, 2018; Yang et al., 2019) (see more details in Methods), both of which are well-supported by empirical experimental data (Hirokawa et al., 2019; Xie et al., 2022).

Once equipped with such two structures, the associated RNNs can be conveniently summarized as a directed dynamic graph (Fig. 1C) characterizing the dynamics of collective-level latent factors (Beiran et al., 2021; Dubreuil et al., 2022; Zhang et al., 2024). That is, given a connectivity matrix with R ranks and P modules, the latent state $\boldsymbol{x}$ of the associated RNN can be decomposed into a linear combination of latent factors $\{\kappa_j\}_{j=1}^{R}$. In the mean-field limit (i.e., when there are enough neurons in each module), these latent factors can be described by the following dynamical systems:

$\tau \frac{\mathrm{d}\kappa_j}{\mathrm{dt}} = -\kappa_j + \sum_i \kappa_i E_{\kappa_i \to \kappa_j} + \sum_l \nu_l E_{\nu_l \to \kappa_j}$, where $\{\nu_l\}_{l=1}^{R_{inp}}$ are external input factors and $E_{\kappa_i \to \kappa_j} = \sum_{p=1}^{P} E_{\kappa_i \to \kappa_j, p}$ (shown as ↠ in Fig. 1C) is the effective coupling strength of the $\kappa_i \to \kappa_j$ pathway, with each $E_{\kappa_i \to \kappa_j, p}$ standing for the effective coupling strength mediated by the p-th module (see Methods for more details and why multiple modules are required).

Built upon this theoretical framework, one major innovation of Restricted-RNN is that we introduce a novel *pathway-based* generative model for connectivity matrix (Fig. 1C). In this pathway-based generative model, the connectivity along each pathway (say, the $\kappa_i \to \kappa_j$ pathway), mediated by P modules with N neurons in each module, is consisted of one PN-dimensional input connectivity vector $(\mathcal{T}^{in}_{\kappa_i \to \kappa_j, 1}, \dots, \mathcal{T}^{in}_{\kappa_i \to \kappa_j, PN})^T$ and one PN-dimensional output connectivity vector $(\mathcal{T}^{out}_{\kappa_i \to \kappa_j, 1}, \dots, \mathcal{T}^{out}_{\kappa_i \to \kappa_j, PN})^T$, of which the N-dimensional sub-vectors $(\mathcal{T}^{in}_{\kappa_i \to \kappa_j, 1+(p-1)N}, \dots, \mathcal{T}^{in}_{\kappa_i \to \kappa_j, pN})^T$ and $(\mathcal{T}^{out}_{\kappa_i \to \kappa_j, 1+(p-1)N}, \dots, \mathcal{T}^{out}_{\kappa_i \to \kappa_j, pN})^T$ are sampled according to the connectivity statistics of the p-th module. Concretely, the input and output connectivity $\mathcal{T}^{in}_{\kappa_i \to \kappa_j, n}$ and $\mathcal{T}^{out}_{\kappa_i \to \kappa_j, n}$ of neuron n from the p_n-th module is parameterized by $S^{in}_{\kappa_i \to \kappa_j, p_n} \epsilon_{\kappa_i \to \kappa_j, n}$ and $S^{out}_{\kappa_i \to \kappa_j, p_n} \epsilon_{\kappa_i \to \kappa_j, n}$, respectively, where $S^{in}_{\kappa_i \to \kappa_j, p_n}$ and $S^{out}_{\kappa_i \to \kappa_j, p_n}$ are trainable "*collective-level input and output connectivity weights*", respectively, and $\epsilon_{\kappa_i \to \kappa_j, n}$ is sampled from the normalized Gaussian noise shared by the input and output connectivity. Additionally, when pathway blocking is required, a pathway-precise mask without affecting other pathways can be conveniently applied (See Methods for more details). Note that once the input and output connectivity vectors are determined, an associated RNN can be naturally induced (Fig. 1C; See Methods for more details).

This pathway-based generative model has the following three major properties that enable Restricted-RNN to be a powerful task-driven RNN training approach with *high model* *interpretability, minimal training parameters and systematic hypothesis space exploration ability* (see Methods for a complete description). Firstly, when the number of neurons in each module is sufficiently large, the generated RNN can be well-described by a dynamic graph with a gain modulation formulation for the effective coupling strength $E_{\kappa_i \to \kappa_j}$ in the mean-field limit ((Dubreuil et al., 2022); see Methods for details). Moreover, such parameterization enables all the statistical parameters (i.e., $S^{in}_{\kappa_i \to \kappa_j,p}$ and $S^{out}_{\kappa_i \to \kappa_j,p}$) of connectivity matrix differentiable and thereby trainable through backpropagation (see Methods for more details), akin to the reparameterization trick used in variational auto-encoder (Kingma & Welling, 2022), ensuring the validity of mean-field theory throughout the whole training process and endowing the trained model with high interpretability. Secondly, while this pathway-based generative model seems complex (i.e., with the order of PR^2 collective parameters), it can be proved that this is the minimal requirement to account for arbitrary interactions between factor pairs. This is because given R latent factors, it requires the order of R^2 number of parameters to account for R^2 kinds of factor pair interactions (see Methods for more details). Thirdly and most importantly, such kind of generative model allows us to independently parameterize different pathways, enabling a systematic exploration of circuit mechanism hypothesis space. That is, collective-level structural priors can be conveniently imposed to the collective-level dynamic graph through masking (Fig. 1D).

In the following, we will validate Restricted-RNN as a powerful circuit hypothesis generator through three examples. In the first example, we showed that while conventional RNN cannot generate models well-aligned with the biological solution (Romo et al., 1999), Restricted-RNN can leverage the existing model structure (Machens et al., 2005) to generate interpretable circuit models compatible with biological data (Fig. 2). In the second example, we demonstrated that Restricted-RNN can generate *de nova* circuit hypothesis for the sequence working memory gating problem (Chen et al., 2024; Xie et al., 2022), with the key predictions being experimentally confirmed by monkey frontal cortex data (Fig. 3). In the last example, we showed that through systematic exploration, Restricted-RNN can even generate circuit hypothesis for counter-intuitive phenomenon, with the key predictions being confirmed by monkey parietal cortex data (Fig. 4). Interestingly, the interpretable nature of Restricted-RNN endowed us a unified theory to explain all these examples with a novel neural control state space, providing an intriguing geometric understanding for the ubiquitous control in cognitive processes (Fig. 5).

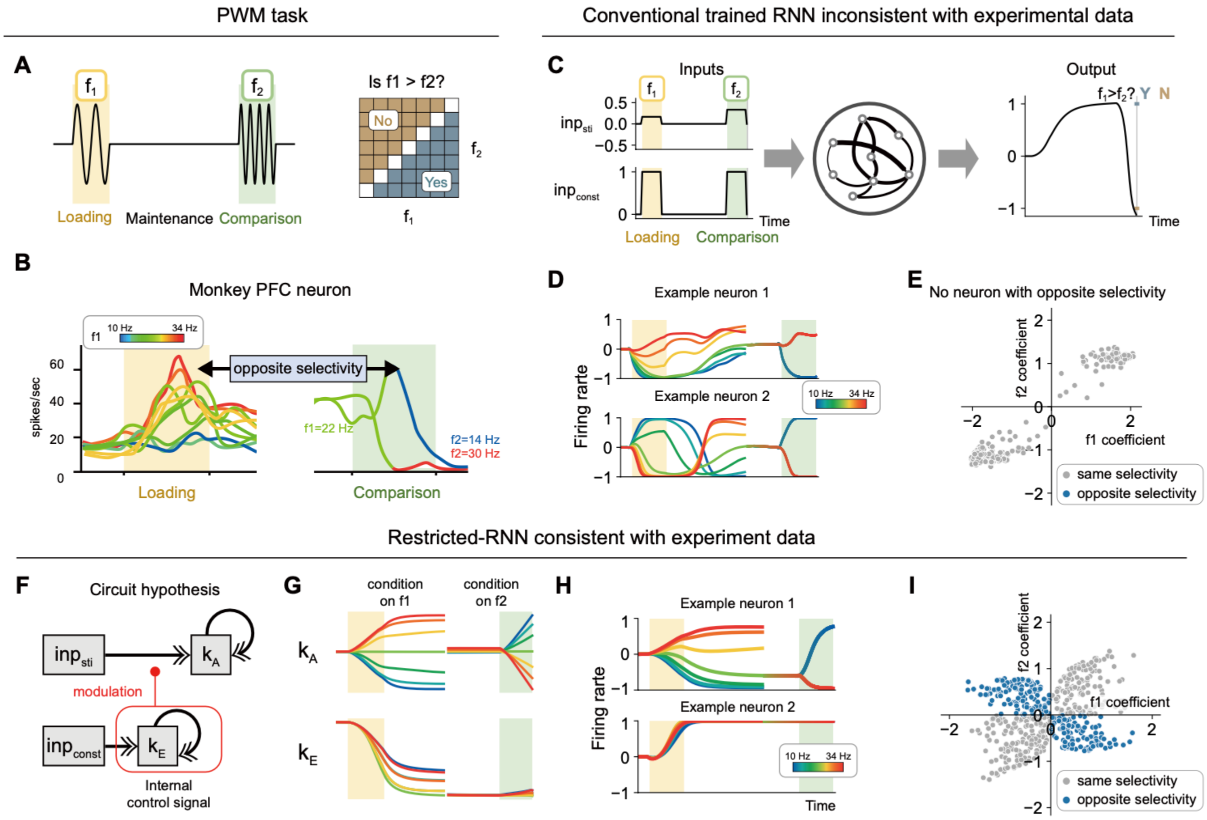

Figure 2. Restricted-RNN reproduces the opposite selectivity of monkey PFC neurons in parametric working memory task, while conventional RNN fails to.

(A) Parametric working memory (PWM) task. The monkey receives a vibratory stimulus at frequency f_1 during the Loading epoch and retains it during the maintenance epoch. In the Comparison epoch, a second stimulus at frequency f_2 is presented. The monkey must compare their magnitudes and indicate whether f_1 is greater than f_2.

(B) Example Monkey PFC neuron. The neuron shows gradient tuning for f_1 during the Loading epoch, with a higher firing rate for increased f_1, whereas it exhibits opposite selectivity for f_2 in the Comparison epoch, firing more for decreased f_2.
(C-E) Conventional RNN modeling for PWM task. (C) RNN settings. The input includes two channels: $\mathrm{inp}_{\mathrm{sti}}$ carries the frequency information, which is non-zero in the Loading and Comparison epochs; $\mathrm{inp}_{\mathrm{const}}$ indicates the presence of input stimuli. A full-rank RNN is used as the model, which outputs 1 when $f_1 > f_2$ and -1 otherwise immediately after the Comparison epoch. (D) Example model neurons. Example neuron 1 does not demonstrate opposite selectivity in the Loading and Comparison epochs. Example neuron 2 switches its selectivity for f_1 during the maintenance epoch. Both are inconsistent with the example PFC neuron in (B). (E) Conventional RNN neurons' tuning distribution for f_1 and f_2. No neurons are found in the second and fourth quadrants, indicating a lack of opposite selectivity and failing to replicate the experimental observation.
(F-I) Restricted-RNN modeling for PWM task. (F) Collective-level circuit of factor dynamics based on Machens et al. (2005). There are two factors: κ_A integrates the stimulus inputs for later comparison, and κ_E recieves and accumulates a constant input. It acts as an internal control signal to modulate the evidence integration of κ_A. (G) Behavior of factors. κ_A (upper) shows opposite responses to f1 and f2 at the Loading and Comparison epochs, respectively. The κ_E (bottom) response mainly during the Loading and maintenance epochs and is insignificant at the Comparison epoch, thus providing distinct modulation effects on the evidence integration. (H) Example neurons of Restricted-RNN. The upper one accounts for the stimulus accumulation and replicates the example PFC neuron in (B). The firing rate of the bottom neuron changes over time and does not tune to the stimulus. (I) Restricted-RNN neurons tuning distribution for f_1 and f_2. A significant number of neurons show opposite selectivity (blue dots).

Restricted-RNN can leverage the existing model structure to generate interpretable models compatible with biological data

The first example we investigated is the classical parametric working memory (PWM) task (Romo et al., 1999). In this experiment (Fig. 2A), macaques first received a vibratory stimulus (f_1) during the loading period. After a memory maintenance period, a second vibratory stimulus (f_2) was presented during the comparison period, during which the macaques were required to compare the two frequencies and decide which one was larger. The experiment revealed a rich set of neuronal activity patterns in prefrontal cortex (Machens et al., 2005; Romo et al., 1999). Among these neurons, there is one type of neurons exhibiting an intriguing firing profile—opposite selectivity to the incoming sensory input during the loading period versus the comparison period (e.g., preferring a higher f_1 during the loading period while preferring a lower f_2 during the comparison period, and verse versa; Fig. 2B).

To begin with, we asked whether this kind of neurons can naturally emerge from conventional RNN training (Fig. 2C). To this end, we trained vanilla RNNs to perform the PWM task and found that neurons from the trained RNNs typically exhibited the same selectivity to the sensory input over the loading and comparison periods (Fig. 2D). To better examine if there exist neurons with opposite selectivity in these RNN models, we performed linear regression for each neuron at each time step and defined its f1 (f2) coefficient as the regression coefficient for stimulus strength f_1 (f_2) averaged over the loading period (the comparison period) (see Methods for details). For neurons with opposite selectivity, f1 and f2 coefficients should show opposite signs. Our analysis revealed that no neurons within the trained RNNs displayed this property (Fig. 2E), a result entirely inconsistent with observations in PFC neurons. Similar discrepancies were observed in vanilla RNNs trained with different hyperparameter settings and in low-rank RNNs. These findings suggest that blind backpropagation-based RNN training does not necessarily generate models well-aligned with biological solutions, which was also shown in a recent work in a different cognitive task (Pagan et al., 2024).

In the following, we showed that by leveraging the existing model structure (Machens et al., 2005), Restricted-RNN can generate circuit models compatible with biological data. Let us first revisit the overall structure of Machens model. Basically, there are two working components in this model, of which the first component, a sensory-to-accumulator (S-A) pathway, sends the sensory input to the accumulator while the second component utilizes an external control signal (E) to modulate how sensory input is projected to the accumulator. By changing the external control signal from high to low, the same frequency input produces opposite effects on the accumulator during the loading and comparison periods, enabling the network to compare the inputs across the two stages (see (Machens et al., 2005) for more details).

Interestingly, we found that this high-level model structure knowledge can be conveniently implemented in Restricted-RNN in a form of collective-level structural priors (Fig. 2F). Specifically, we introduced two pathways in our model. The first pathway, denoted as $\nu_{sti} \rightarrow \kappa_A$, sends the stimulus-strength-dependent input (ν_{sti}) to the accumulator variable (κ_A), mimicking the S-A pathway in Machens model. The second pathway, denoted as $\nu_{sti} \rightarrow \kappa_A$, sends the stimulus-strength-independent input (ν_{const}) to an additional variable κ_E that is hypothesized to play a similar role as the external control signal (E) in Machens model. The underlying intuition is that since κ_E can monitor the task stage procession (loading or comparison period) through integration over time, in principle, it can modulate the information pathway from sensory input to the accumulator in a task-stage-dependent manner through module-based gain modulation (Dubreuil et al., 2022).

Note that this structural prior only specified the overall low-rank structure of connectivity patterns. The minimal number of modules required to perform the task is yet to be determined through learning. Specifically, by gradually increasing the number of modules starting from $P = 1$, we identified the minimal number of modules that are required to successfully minimize the task objectives through backpropagation (see Methods for more details). Following this procedure, we found that a minimal two-module network model with the prescribed structural prior is required to perform the task. When applying the same single neuron analysis pipeline, we found that many neurons in this minimal network model exhibited the opposite selectivity to the incoming input during the loading period versus the comparison period, consistent with the biological data (Fig. 2I; see Fig. 2H for an example neuron and other types of neurons). In fact, this opposite selectivity also exhibited at the collective level, as shown by the dynamics of the accumulator κ_A (Fig. 2G).

Together, this simple example demonstrated that when conventional RNN did not necessarily generate circuit models well-aligned with biological solutions, Restricted-RNN can leverage the existing high-level model knowledge to generate circuit models compatible with biological data.

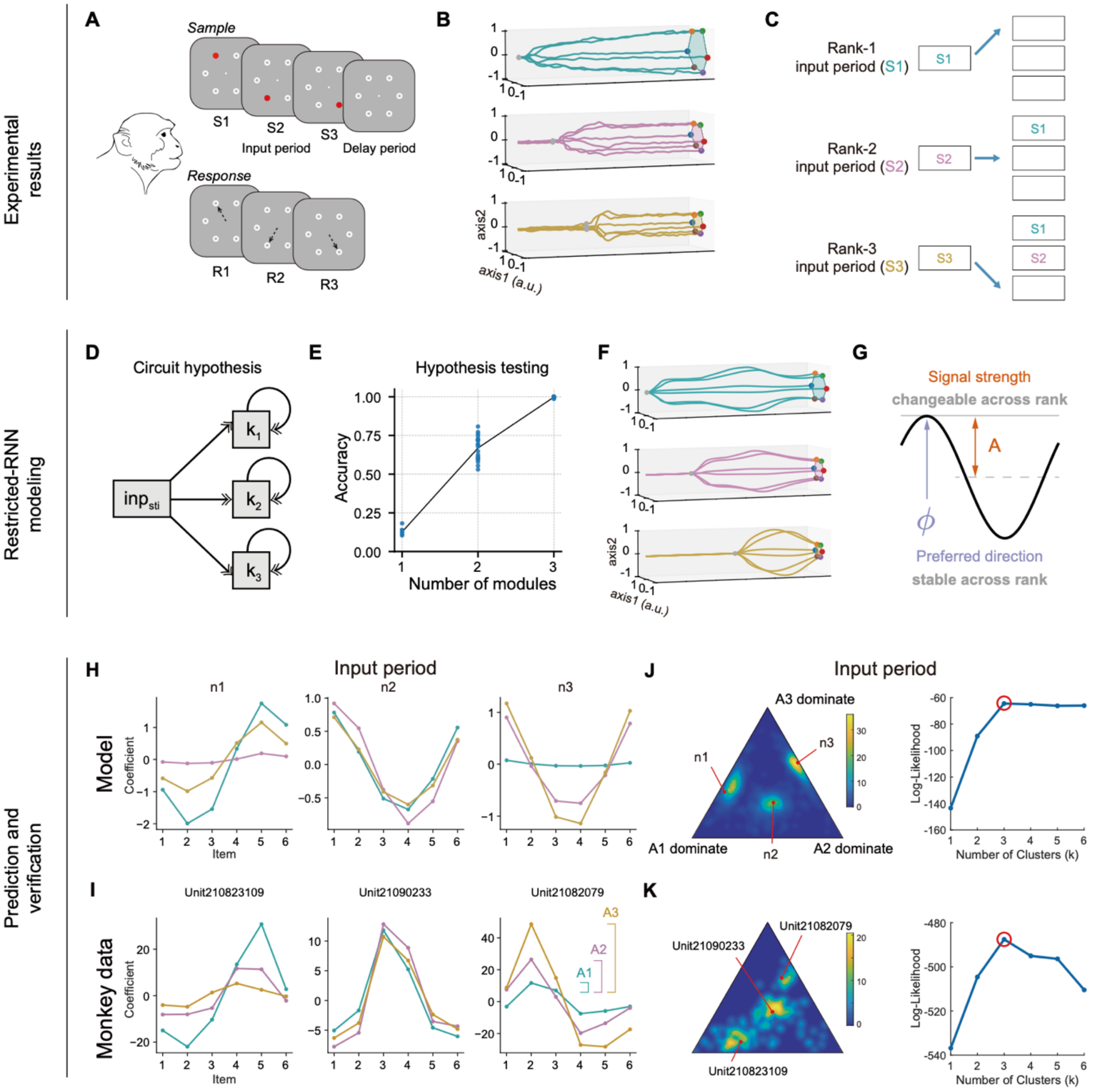

Figure 3. Restricted-RNN modeling for sequence working memory task predicts the population structure underlying memory gating.

(A) Sequence working memory (SWM) task. The monkey receives a sequence of stimuli (of length 1-3) each being a location drawn from a hexagon. The task requires the monkey to remember both the locations and their order for an indefinite period (Delay) and then reports the locations in the presented order.
(B) Neural trajectories in the memory subspaces for the length 3 SWM case. We show the trajectories during a certain period, and the trajectories are colored according to the dominant ordinal rank (rank 1, 2, 3 for memory subspace 1, 2, 3, respectively).
(C) A summary of the memory gating process of SWM (Chen et. al. 2024). At first the gateway to memory subspace 1 opens while the other two are closed, so that the first item in the sequence enters memory subspace 1. Then this gateway shuts down and the one to memory subspace 2 opens. Therefore, the second item enters memory subspace 2. Finally, the same process guides the last item entering memory subspace 3.
(D-F) Restricted-RNN modeling for SWM. (D) Collective-level circuit of factor dynamics based on previous findings in (C). There are three factors ($\kappa_1, \kappa_2, \kappa_3$) for integrating the memory singals of the three ranks, respectively. They share the same stimulus input. (E) We tested hypotheses with increasing number of modules and found that it requires at least three modules for Restricted-RNN to accomplish the length 3 SWM task. (F) Neural trajectories in the memory subspaces of the trained Restricted-RNN, which replicates the experimental observation in (B).
(G) Quantification of neuron tuning pattern. To measure a neuron's tuning to the six items, we first compute the conditional average firing rates and fit them with a cosine function. This gives us two quantities: the amplitude A and the phase ϕ, which represent the sensitivity of the neuron to stimulus and its prefered tuning direction. A and ϕ quantify the neuron's tuning pattern to items.
(H-K) Model predictions and verification. (H) and (I) show example neurons of Restricted-RNN and Monkey PFC data. During the input period, item-selective Restricted-RNN neurons exhibit stable tuning phases across different ordinal ranks (H). Similar patterns can be found in Monkey PFC data (I). Therefore, the amplitude distribution of different ordinal ranks (distribution of A_1, A_2, A_3) captures the essence of neuron tuning pattern during the input period. We visualize (see Methods) the amplitude distribution across neurons in panels (J) and (K), using color to indicate neuron density—brighter colors represent higher densities. Clustering analysis (see Methods) reveals that there are three modules in Restricted-RNN model (J), and the same is true for Monkey PFC data (K).

Restricted-RNN uncovered a *de nova* neural mechanism underlying sequence working memory gating

To further test the validity of Restricted-RNN, we then applied it to a recent sequence working memory experiment (Chen et al., 2024; Xie et al., 2022). In this experiment, monkeys were presented a spatial sequence and required to reproduce the sequence after a variable delay period (Fig. 3A). Neural data analysis revealed that spatial items in different ordinal ranks were routed to the corresponding rank subspaces upon the stimulus presentation and stably maintained throughout the delay period (Fig. 3B). However, how such a precise information control—opening the pathway from the input to the right rank subspaces and closing other pathways at each input period (Fig. 3C)—remains elusive. In other words, what kind of modular structure is required to fulfill such information control is unclear.

Enlighted by the experimental findings (Chen et al., 2024), we introduced a minimal collective-level prior in Restricted-RNN with four factors (Fig. 3D), including one input factor (inp_{sti}) and three factors ($\kappa_r, r = 1,2,3$) for sequence working memory. In addition, there are three feedforward pathways ($inp_{sti} \to \kappa_r, r = 1,2,3$) sending the input factor to working memory factors and three recurrent pathways ($\kappa_r \to \kappa_r, r = 1,2,3$) for memory maintenance. We then ask what the minimal number of modules is required to perform the sequence working memory task (see Methods for more details). By gradually increasing the number of modules starting from $P = 1$, we identified the minimal number of modules is 3 for length-3 sequence working memory gating problem (Fig. 3E) and the three working memory factors in this minimal model exhibited the similar dynamics with the experiment (Fig. 3F).

Then, what did these three modules look like? An intuitive possibility is that each module is activated only in one input period (e.g., module 1 for rank-1 period). Simply examining neural firing patterns in different input periods, however, rejected this possibility. Instead, most of neurons were activated during multiple input periods (Fig. 3H). Importantly, this activation pattern is not totally random. Firstly, through examining the preferred location during the input periods, we found that all of neurons in the model showed the same location preference at different input periods—a gain modulation profile (Fig. 3H). Secondly, when each neuron is represented by the combination of the activation amplitudes (see the definition of amplitude in Fig. 3G) at different input periods, a collective-level structure with 3 clusters prevailed (Fig. 3J). We note that the intuitive possibility corresponds to three clusters at each corner of the triangle Fig. 3J. We then asked if there is any evidence for supporting such a minimal model. Following the same analysis, we found that 1) single neurons in monkey frontal cortex indeed showed a gain-modulation activity pattern (Fig. 3I) and 2) neurons in monkey frontal cortex as a population exhibited a modular structure with three clusters during length-3 sequence working memory control (Figs. 3, I and K for one monkey), which validated the model predictions.

Together, this example demonstrated that Restricted-RNN can generate novel circuit hypothesis and make testable predictions. Notably, these minimal model-based predictions were well-supported by monkey frontal data, which is surprised given the complexity of biological data, suggesting that our modeling results may capture key aspects of the biological neural computation.

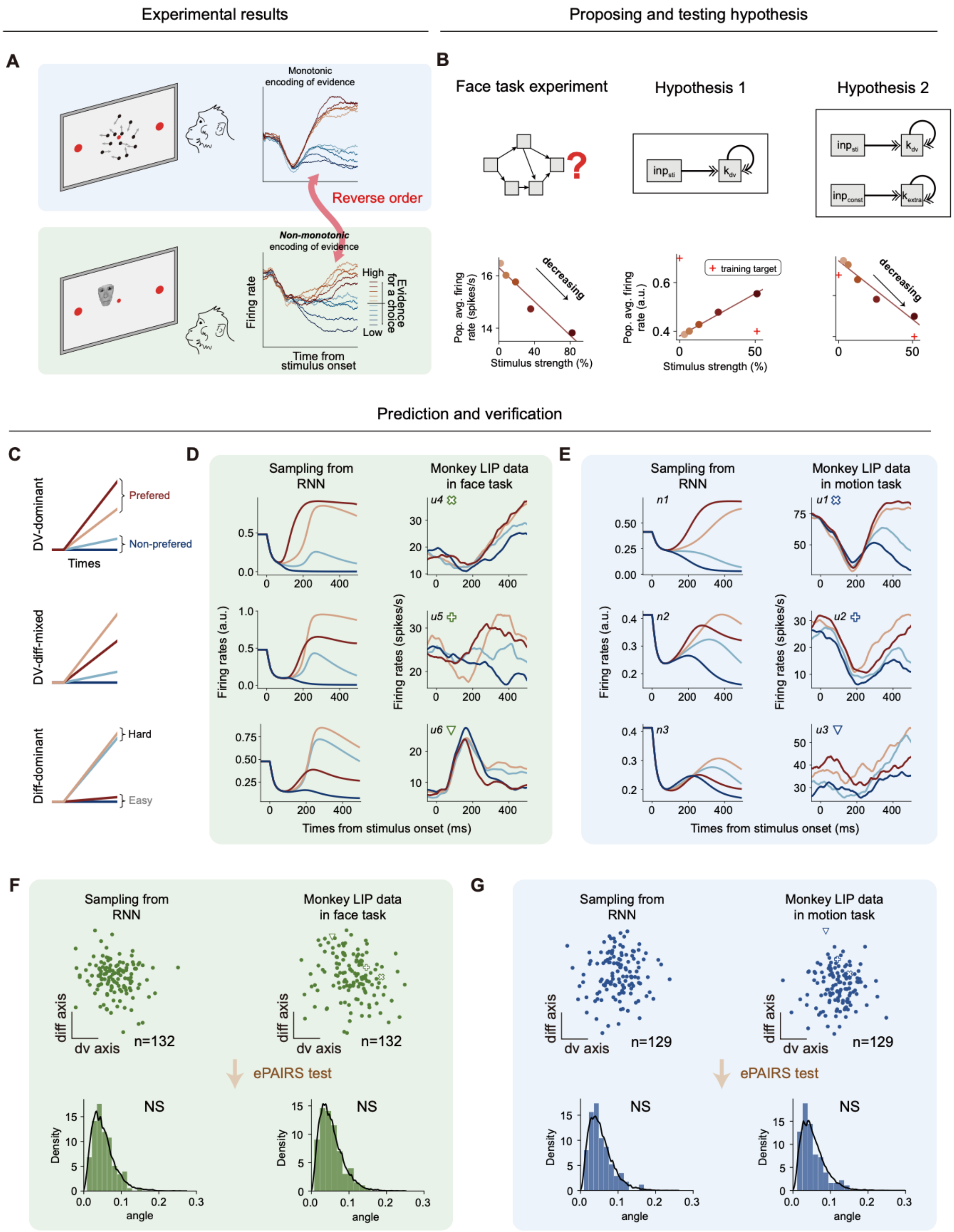

Figure 4. Restricted-RNN modeling helps to explain the reversal of mean firing rates in the face discrimination task.

(A) Two perceptual decision-making (PDM) tasks (Okazawa et al. 2021): (upper) the motion PDM task, where monkey responds based on the overall movement of random dots. (bottom) The face PDM task, where monkey responds according to how closely a presented face resembles either a human or a monkey face. Previous research found that, unlike the motion PDM task, the mean firing rates in the face PDM task demonstrate non-monotonic tuning to the evidence strength, i.e., mean firing rates reversal.
(B) Developing the collective-level circuit of Restricted-RNN model through hypothesis proposing and testing. We aim to identify the simplest collective circuit that can replicate the mean firing rates reversal in the face PDM task. Starting from the basic circuit (Hypothesis 1) that only involves one factor κ_{dv} for the evidence accumulation, we systematically tested a number of Hypotheses (see Methods) and found that it is necessary to include another stimulus-irrelevant factor, κ_{extra} (Hypothesis 2). The final collective circuit has two factors: κ_{dv}integrates the input evidence, while κ_{extra}recieves and accumulate a constant input.
(D-H) Model predictions and verification. Restricted-RNN model predicts that both the motion and face PDM tasks consist of three types of neurons (D): i) DV-dominant neurons that monotonically respond to evidence strength. ii) Diff-dominant neurons that respond to the coherence (difficulty) of stimuli. And iii) DV-diff-mixed neurons that display mixed tuning and non-monotonical responses to evidence strength. We show example neurons for all the three types from the face PDM task (E) and motion PDM task (F). In each panel, the left column are Restricted-RNN neurons and the right column shows neurons from Monkey LIP data. From top to bottom, the example neurons belong to the DV-dominant, DV-diff-mixed, and Diff-dominant type, respectively. Panels (G) and (H) investigate the population structure in the face and motion PDM tasks, respectively. Population structure is measured by projected neural activities along the κ_{dv} and κ_{extra}(also known as κ_{diff}) axes (upper plots in each panel), reflecting neurons' roles in coding these factors. In these plots, each point stands for a neuron from either Restricted-RNN (left) or Monkey LIP data (right). In all cases, there exist only one neural population (ePAIRS test, see Methods).

Restricted-RNN uncovered a *de nova* neural mechanism explaining the counterintuitive firing rate reversal in macaque parietal cortex

One best validation for a new approach is to test if it can explain the counter-intuitive phenomenon. In this regard, we applied Restricted-RNN to a recent perceptual decision-making experiment showing the counterintuitive task-dependent response geometry of perceptual decisions in monkey parietal cortex (Okazawa et al., 2021). In this experiment, the mean firing rates of lateral intraparietal (LIP) neurons were found to show non-monotonic encoding of sensory evidence during a novel face discrimination task (Fig. 4A, bottom panel), challenging the classical monotonic evidence encoding view supported by decades of motion discrimination task studies (Fig. 4A, top panel). These contrasting results were attributed to the curved manifold coding property in neural state space, in which both decision variable (denoted as dv) and stimulus difficulty (a factor encoding the unsigned strength of sensory inputs; denoted as diff) were simultaneously encoded. However, the key question of how this curved manifold representation arises from a biological neural circuit remains unanswered, providing an ideal case for testing the validity of Restricted-RNN.

In the following, we demonstrated that restricted-RNN offers a systematic solution to addressing this challenging problem. This is achieved through a series of model proposing and training within the Restricted-RNN framework, in which model proposing is referred as to propose the model structure in terms of dynamic graphs (e.g., those in Fig. 4B) while model training is referred as to test if the proposed model structure can be trained to reproduce the prescribed firing pattern (e.g., the mean firing rate reversal shown in Fig. 4B) through back-propagation learning. By gradually increasing the proposed model complexity, this approach provides a principled way to identify the minimal circuit model explaining the experimental result.

Specifically, we first proposed the simplest perceptual decision-making circuit (Fig. 4B, middle panel; termed as H1), with only one pathway sending the input variable (inp_{sti}) to the decision variable (κ_{dv}). After model training, we found that H1 can generate a curved manifold but cannot reproduce mean firing rate reversal (Fig. 4B, middle panel). What is the underlying mechanism? The high interpretability of H1 enabled us to ask this question. First, while the latent variable κ_{dv} in H1 indeed forms a straight line, the observed neural activity is not the latent variable κ_{dv} per se. Instead, it corresponds to a nonlinear transformation (i.e., the nonlinear activation function) of the latent variable, effectively transforming the straight line into a curved manifold (see Methods for more details). Second, while this simple nonlinear transformation explains the origin of curved manifold, generating mean firing rate reversal is far more non-trivial. This is because as a monotonic-increasing function, the nonlinear activation function along cannot change the monotonicity of mean firing rate (see Methods for more derivations).

To replicate this non-trivial firing rate reversal result, we then increased the model complexity by adding an additional latent factor (κ_{extra}) with extra pathways and found that none of these models can reproduce the mean firing rate reversal result. Enlighted by the classical Wong-Wang model (Wong & Wang, 2006), we also introduced a constant input (inp_{const}), independent of the stimulus input strength, to mimic the overall excitation during the task, and found that this kind of circuit still failed to generate mean firing rate reversal. Surprisingly, when a pathway conveying the constant input (inp_{const}) to the additional latent factor (κ_{extra}) is added into the model structure (termed as H2), the mean firing rate reversal can be conveniently reproduced with only one module (Fig. 4B, right panel; see Methods for more details) and κ_{extra} in the trained model behaves like a difficulty variable—encoding the absolute value of input strength. If H2 holds, it will lead to the following novel testable predictions at both single-neuron and collective levels.

First, at the single-neuron level, H2 predicts that the decision variable (DV) and the difficulty variable (Diff) are represented independently. Depending on the relative encoding strength of these two task variables, individual neurons in H2 are expected to exhibit three distinct response patterns (Fig. 4C). The DV-dominant pattern is referred to as the monotonic evidence encoding, corresponding to those individual neurons with large encoding for decision variable but small encoding for difficulty variable (Fig. 4C, top panel and see Fig. 4D, top left panel for example model neuron). In contrast, the Diff-dominant pattern is referred to as a response pattern with high (low) firing rate for hard (easy) trials, representing stimulus difficulty and corresponding to those individual neurons with small decision variable encoding and large difficulty variable encoding (e.g., Fig. 4C, bottom panel and see Fig. 4D, bottom left panel for example model neuron). The

DV-diff-mixed pattern is referred to as the non-monotonic evidence encoding pattern like the meaning firing rate reversal pattern in the face task (Fig. 4A, bottom panel), corresponding to those individual neurons with comparable decision variable and difficulty variable encoding (Fig. 4C, middle panel and see Fig. 4D, middle left panel for example model neuron). Through re-examining the single neuron data (Okazawa et al., 2021), we indeed found all three single neuron firing patterns in face task (Fig. 4D, right panel), verifying our model predictions.

Second, in the collective-level, our modeling result suggested that one neural population suffices to explain the task-dependent representational geometry of perceptual decisions in monkey parietal cortex. To test this prediction, we performed the EPAIRS analysis (Dubreuil et al., 2022; Hirokawa et al., 2019; Raposo et al., 2014) and found that these seemingly diverse neural firing patterns in face task can be well-explained only by one functional neural cluster (Fig. 4F; see Methods for details), perfectly aligning with H2 and thereby verifying our collective-level model prediction.

Importantly, when the same single neuron and population analysis were applied to the motion task, we found the almost same firing pattern: there were three different kinds of single neuron firing profiles (Fig. 4E, right panel) and one functional cluster (Fig. 4G, right panel). We shall emphasize that while the DV-diff-mixed pattern was reported in the face task (Okazawa et al., 2021), the existence of both DV-diff-mixed and Diff-dominant single neurons in LIP in the motion task is particularly surprising as neurons in LIP in motion task is traditionally considered to be of DV-dominant type (Gold & Shadlen, 2007). From the modeling perspective, the existence of Diff-dominant single neuron cannot be explained by H1, thereby challenging the canonical circuit model for motion task (see Methods for more details). Interestingly, the same H2 but with a set of different collective connectivity parameters can conveniently explain all these patterns (Fig. 4E, left panel and Fig. 4G, left panel). Therefore, H2 provided a unified yet simple neural mechanism for explaining the neurophysiological data in both motion and face decision-making tasks.

Taking together, through systematic exploration, Restricted-RNN cracked the hidden circuit mechanism underlying the counter-intuitive firing rate reversal in monkey LIP, strongly suggesting that Restricted-RNN indeed can help uncover novel circuit mechanisms for challenging higher cognition problems.

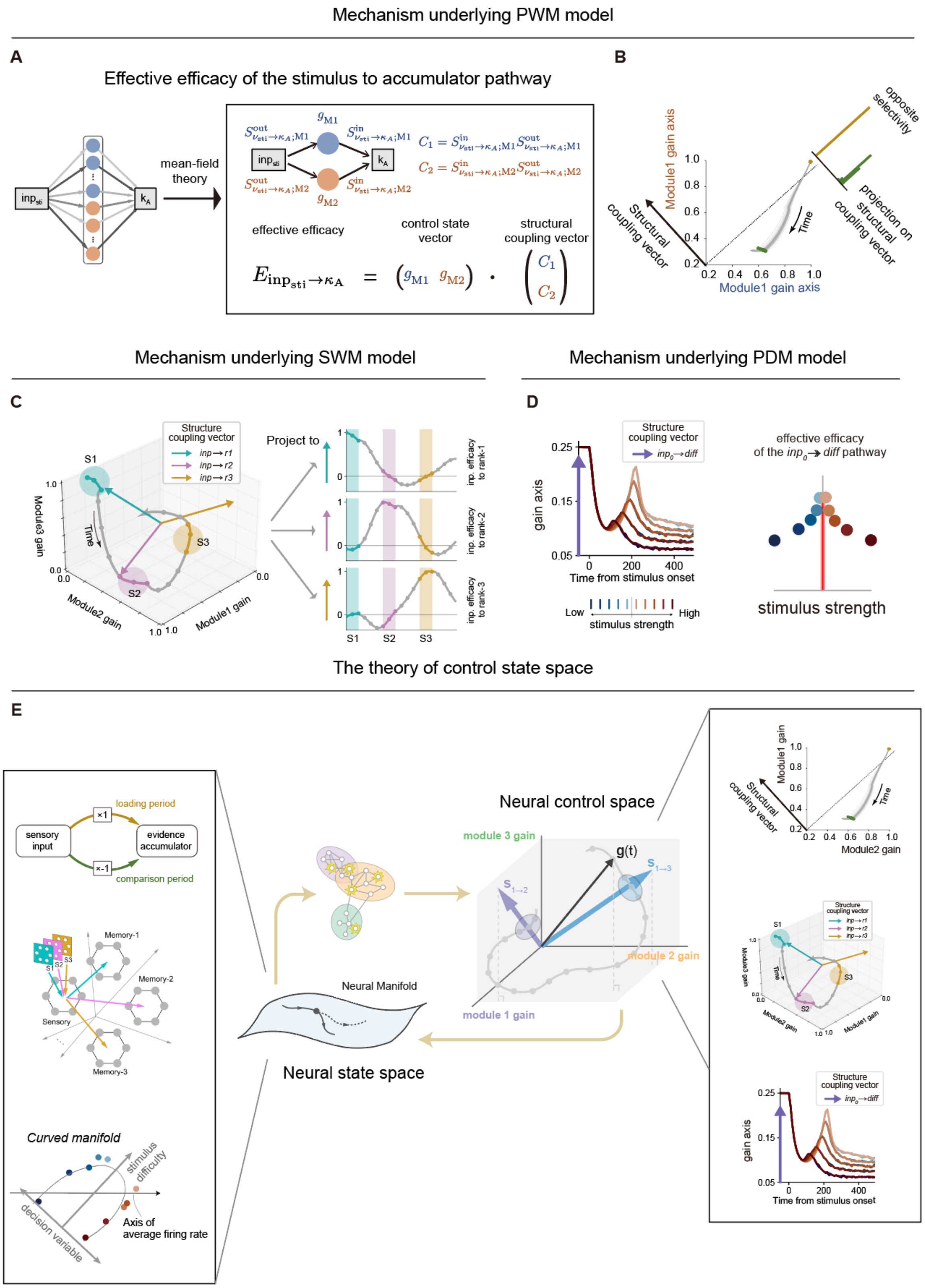

Figure 5. The control state space provides a unified framework for interpreting the control process.

(A-B) Introducing the control state space description and mechanism underlying opposite selectivity at different epochs in PWM model. (A) At collective-level, the module-mediated communication between factors depends on two properties: the coupling strength and module gain. The coupling C equals to the multiplication of the output and input connectivity strength to the module, i.e., C=$S^{out} * S^{in}$. This indicates the capacity of the pathway. Module gain g reflects the overall excitability of neurons in the module, which modulate the actual amount of information that pass through the pathway. By combining all modules, we can create a control state vector consisting of module gains and structural coupling vectors for all the pairs of communicating factors. The inner product of the control state vector and structural coupling vector indicates the strength of information flow between the factors, referred to as effective efficacy. (B) During the Loading and Comparison epochs, the overlapping between module gain vector and the structural coupling vector changes from positive to negative, leading the change of sign in neuron selectivity.
(C) Mechanism underlying memory gating in SWM model. We plot the evolution of the module gain vector (grey line) in the three-dimensional control state space. During input periods S1, S2 and S3, the module gain vector aligns exclusively with rank 1, rank 2 and rank3 structural coupling vectors, respectively, thus opening the gateway to corresponding memory subspaces successively.
(D) Mechanism behind difficulty representation in the PDM model. Despite there is no direct difficulty input, the module gain evolves differently with different absolute stimulus strengths. This will modulate how factor κ_{diff} integrates the stimulus-irrelevant input and generates difficulty representation.
(E) The control state space provides a concise and intuitive way to understand how the flow-field in the neural state space is formed and guide the evolution of neural representations. Specifically, neuron connectivity establishes structural coupling vectors. The arrangement of the module gain vector relative to these coupling vectors in the control state space governs information flow within the Restricted-RNN model (right box). This interaction influences the evolution of representations in the neural state space (left box), altering the excitation of individual neurons and thereby affecting the module gain vector.

Restricted-RNN endowed us a unified theory to understand the hidden control representation

Through these examples, we have demonstrated Restricted-RNN as a powerful hypothesis generator. In fact, as a modeling approach grounded in rigorous theoretical basis (Beiran et al., 2021; Dubreuil et al., 2022; Zhang et al., 2024), Restricted-RNN can also endow us a unified theory to explain the disparate phenomena across different tasks. Let us first recall the computation-through-dynamics framework (DePasquale et al., 2023; Vyas et al., 2020), in which hypothesis regarding cognitive computation is expressed in terms of flow-field shaping the trajectory of factors in the high-dimensional neural state space. In general, it is hard to get a good understanding of the flow-field in a high-dimensional space, the key to the “black-box” issue.

Interestingly, the interpretable nature of Restricted-RNN can provide us a valuable means for understanding how the flow-field itself is formed to perform the computation required by the task.

In the example of parametric working memory here, understanding how the flow-field shapes the trajectory of the accumulator κ_A is equivalent to modeling the effective coupling strength of the pathway $\nu_{sti} \rightarrow \kappa_A$. In the mean-field limit of Restricted-RNN, this coupling strength can be concisely expressed as the inner product between the structural coupling vector $(C_1, C_2)^T$ of the $\nu_{sti} \rightarrow \kappa_A$ pathway and the module gain vector $(g_1, g_2)^T$, of which C_p $(p = 1,2)$ stands for the structural coupling strength of the p-th module while g_p $(p = 1,2)$ is the average gain of the p-th module (Fig. 5A; see Methods for more details). When introducing a new state space with each axis standing for the gain of one distinct module, such an inner product formula then can be interpreted in a geometric manner: 1) for each trial, the trajectory of the state-dependent module gain vector in this new state space represents the dynamic state of network excitability; 2) the overlap between this dynamic state and the static structural coupling vector determines the effective coupling of the $\nu_{sti} \rightarrow \kappa_A$ pathway. For instance, during the loading onset, the overlap between the dynamic state and the static structural coupling vector is positive (Fig. 5B). With the procession of the task, the dynamic state evolves, and the resultant overlap becomes negative during the comparison onset, which is the desired effective coupling strength required by the comparison operation of the task (see Methods for more details). Taken together, this set of analyses provides an intuitive geometric understanding for how the state-dependent flow-field is formed to regulate the information flow required by the task.

Importantly, such a set of analyses can be equally applied to both the SWM model and the perceptual decision-making model. For the SWM model, as there were three modules, the associated new state space is of three dimensions (Fig. 5C). Moreover, instead of investigating the effective coupling of one pathway (the $\nu_{sti} \rightarrow \kappa_A$ pathway) in parametric working memory model, there are three pathways of interest, including $inp_{sti} \rightarrow \kappa_r, r = 1,2,3$, in the SWM model. By plotting out both the structural coupling vectors of these three pathways and the dynamic state through the whole trial in this new state space (Fig. 5C, left panel), we found that 1) during the rank-1 stimulus period (cyan circle), the dynamic state stays in a subregion with-aligned with the structural coupling vector of the $inp_{sti} \rightarrow \kappa_1$ pathway; 2) with the precession of task stage, the dynamic state enters in a subregion with-aligned with the structural coupling vector of the $inp_{sti} \rightarrow \kappa_2$ pathway during the rank-2 stimulus period (purple circle); 3) as expected, during the rank-3 stimulus period (yellow circle), the dynamic state enters in a subregion with-aligned with the structural coupling vector of the $inp_{sti} \rightarrow \kappa_3$ pathway. A more quantitative visualization can be obtained by projecting the dynamic state into the three structural coupling vectors, revealing a substantial overlap with the structural coupling vector of the relevant pathway while near-zero overlaps with other irrelevant structural coupling vectors at each input period (Fig. 5C, right panel)—the desired geometric property required by sequence working memory control (Fig. 3C). Notably, this geometric portrait also provides an intuitive understanding why three modules are required for the length-3 sequence working memory control problem simply because three dimensions are required to have the dynamic state be overlapping with one vector while simultaneously orthogonal to the other two.

For the perceptual decision-making model, the major issue is to understand how the difficulty variable emerges from a circuit without explicitly modeling the difficulty variable (Fig. 4B, right panel). To resolve this issue, we applied the same set of analyses to the perceptual decision-making model. In this model, as there is only one module, the associated new state space is of one dimension. By plotting out the trajectories of the dynamic state (i.e., the module gain) with different stimulus strengths, we found that upon the onset of stimulus, the module gains with larger absolute stimulus strengths dropped faster than those with smaller ones, resulting in a difficulty-like code (Fig. 5D). According to the model, as κ_{diff} represents the code of inp_0 (a constant code independent of stimulus strengths) multiplied by the effective coupling strength of $inp_0 \twoheadrightarrow diff$ pathway, this explains why κ_{diff} behaves like a difficulty variable. In contrast, as κ_{dv} represents the code of inp (encoding stimulus strengths) multiplied by the effective coupling strength of $inp \twoheadrightarrow diff$ pathway, κ_{dv} still exhibited the monotonic code of stimulus strengths. Together, this set of analyses provides a mechanistic understanding for the emergence of difficulty variable in perceptual decision-making tasks.

Taken together, we propose this new state space, named as *the neural control state space*, as an important complement to the concept of the predominant neural state space (Fig. 5E). In the neural state space, the trajectory of multiple factors can be conveniently investigated, providing a geometric understanding regarding the factor representation but leaving the issue of how these factor representations are formed largely open. Augmented with the modular structure, the neural control state space provides the much-needed concept to account for the intriguing interactions between factors and thereby explain how the trajectories of multiple factors are formed, which is well-supported by all of three examples.

Discussion

As a deep-learning-based dynamical model, RNN naturally implements the computation-through-neural-dynamics framework and plays a major role in generating circuit mechanism hypothesis in systems neuroscience. However, it also inherits the "black-box" property from deep-learning approach. Through integrating the multi-level descriptions of neural systems and leveraging the recent theoretical progress in neural computation, Restricted-RNN provides an alternative modeling approach featuring *high interpretability, minimal number of parameters and systematic hypothesis space exploration ability*. The validity of Restricted-RNN in novel circuit hypothesis generation was demonstrated through a variety of macaque cognitive tasks, with the key derived predictions being confirmed by monkey neurophysiological data. Critically, based on Restricted-RNN, a new concept—namely the neural control state space—was proposed to provide a unified geometrical understanding for the ubiquitous control in cognitive processes. Together, these results strongly demonstrated the great promise of Restricted-RNN in generating novel interpretable circuit mechanism hypothesis for challenging higher cognitive problems.

Post hoc reverse-engineering versus theory-based training

In conventional RNN, each connection weight is trained, endowing the model with great fitting power (with about one million free parameters for an RNN with 1000 neurons). While many insights can be gained from these trained models through a variety of reverse-engineering approaches, it is difficult to understand a model with millions of free parameters in general. Crucially, this black-box issue could be exaggerated in challenging problems. For example, recent work showed that conventional RNN failed to reproduce the individual variability of neural computations underlying flexible decisions (Pagan et al., 2024). Instead of training each connection weight between neurons, Restricted-RNN directly trained “the collective-level connectivity weights” through introducing a novel generative model for connectivity matrix. By doing so, Restricted-RNN endowed the trained model with high interpretability and conveniently generated data-compatible circuit models for a variety of cognitive tasks.

Neural state space versus neural control state space

Neural state space has been playing a dominant role in revealing the dynamic evolution of task-related factors. In other words, it provides the appropriate concept for characterizing the representation of factors, which is undeniably important. However, it is conceivable that the emergence of these factor representations involves a delicate control process that enables a contextual dependent information flow regulation required by the task. What is the right language to describe such a delicate control process remains unknown (Badre et al., 2021; Cohen, 2017; Miller & Cohen, 2001). As demonstrated in this work, introducing the neural control state space enabled us to address the following key issues: 1) How is control represented in a neural system? What is the dimensionality of control representation? How does control representation get updated? How does control representation regulate the information flow?

Therefore, we proposed that the neural control state space may provide the appropriate concept for accounting for the delicacy of control representation. Further systematic investigation is warranted to better test the generality of this new concept.

Factor-centric view versus neuron-centric view

Whether brain computation is factor-centric or neuron-centric is hotly debated (Barack & Krakauer, 2021). As Restricted-RNN is task-driven, it naturally adopted the factor-centric view—regarding neural populations as the substrate mediating the interactions among different task-related factors. The introduced pathway-based generative model for connectivity matrix is a natural way to account for the interactions among different factors (Fig. 1C). In fact, in Restricted-RNN, the two views are not contradictory but complimentary: factor interact with each other through neural populations while neural populations communicate with each other through subspace (coined as communication-through-subspace in the literature).

Restricted-RNN as a systematic hypothesis generator

To be qualified as an ideal hypothesis generator, an approach should be able to systematically explore the hypothesis space. Previous works have showed that conventional RNN may not be able to explore the whole hypothesis space (Pagan et al., 2024). The pathway-based generative model of connectivity introduced here enables pathway-precise blocking and thereby flexible connectivity prior imposition, providing a systematic approach to identify the minimal circuit model in terms of the number of connectivity ranks, modules and pathways. By identifying *de nova* circuit mechanisms for both sequence working memory and counter-intuitive firing rate reversal, we demonstrate the validity of this systematic approach, strongly supporting Restricted-RNN as systematic hypothesis generator.

Connection with classical biophysical models

Classical biophysical models, such as Machens model and Wong-Wang model mentioned here, provide enormous insights into the neural mechanisms underlying various cognitive processes. The model generated from Restricted-RNN provides an alternative description for these cognitive processes at an abstract level in terms of connectivity ranks, modules and pathways. Despite this abstraction, this kind of description still preserves the key ingredients closely linking with neural physiological data, as demonstrated in all of examples presented here. The exhibited parsimony of such an abstract description may provide insights into more complex cognitive process such as sequence working memory manipulation (Tian et al., 2024).

In his famous essay, Einstein said "Everything should be made as simple as possible, but not simpler". Here, by introducing a novel mathematical language in terms of connectivity ranks, modules and pathways, the proposed Restricted-RNN training framework provides a systematic approach towards identifying the circuit mechanisms for the challenging higher cognition problems with high dimensionality and high complexity in a learnable and interpretable fashion.

Reference

Badre, D., Bhandari, A., Keglovits, H., & Kikumoto, A. (2021). The dimensionality of neural representations for control. *Current Opinion in Behavioral Sciences*, *38*, 20–28. https://doi.org/10.1016/j.cobeha.2020.07.002

Barack, D. L., & Krakauer, J. W. (2021). Two views on the cognitive brain. *Nature Reviews Neuroscience*, *22*(6), 359–371. https://doi.org/10.1038/s41583-021-00448-6

Beiran, M., Dubreuil, A., Valente, A., Mastrogiuseppe, F., & Ostojic, S. (2021). Shaping Dynamics With Multiple Populations in Low-Rank Recurrent Networks. *Neural Computation*, *33*(6), 1572–1615. https://doi.org/10.1162/neco_a_01381

Chen, J., Zhang, C., Hu, P., Min, B., & Wang, L. (2024). Flexible control of sequence working memory in the macaque frontal cortex. *Neuron*, S0896627324005695. https://doi.org/10.1016/j.neuron.2024.07.024

Cohen, J. D. (2017). Cognitive Control: Core Constructs and Current Considerations. In T. Egner (Ed.), *The Wiley Handbook of Cognitive Control* (1st ed., pp. 1–28). Wiley. https://doi.org/10.1002/9781118920497.ch1

Cueva, C. J., & Wei, X.-X. (2018). *Emergence of grid-like representations by training recurrent neural networks to perform spatial localization* (No. arXiv:1803.07770). arXiv. https://doi.org/10.48550/arXiv.1803.07770

DePasquale, B., Sussillo, D., Abbott, L. F., & Churchland, M. M. (2023). The centrality of population-level factors to network computation is demonstrated by a versatile approach for training spiking networks. *Neuron*, *111*(5), 631-649.e10. https://doi.org/10.1016/j.neuron.2022.12.007

Driscoll, L. N., Shenoy, K., & Sussillo, D. (2024). Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *Nature Neuroscience*, *27*(7), 1349–1363. https://doi.org/10.1038/s41593-024-01668-6

Dubreuil, A., Valente, A., Beiran, M., Mastrogiuseppe, F., & Ostojic, S. (2022). The role of population structure in computations through neural dynamics. *Nature Neuroscience*, *25*(6), 783–794. https://doi.org/10.1038/s41593-022-01088-4

Gold, J. I., & Shadlen, M. N. (2007). The Neural Basis of Decision Making. *Annual Review of Neuroscience*, *30*(1), 535–574. https://doi.org/10.1146/annurev.neuro.29.051605.113038

Hernández, A., Salinas, E., García, R., & Romo, R. (1997). Discrimination in the Sense of Flutter: New Psychophysical Measurements in Monkeys. *The Journal of Neuroscience*, *17*(16), 6391–6400. https://doi.org/10.1523/JNEUROSCI.17-16-06391.1997

Hirokawa, J., Vaughan, A., Masset, P., Ott, T., & Kepecs, A. (2019). Frontal cortex neuron types categorically encode single decision variables. *Nature*, *576*(7787), 446–451. https://doi.org/10.1038/s41586-019-1816-9

Kingma, D. P., & Welling, M. (2022). *Auto-Encoding Variational Bayes* (No. arXiv:1312.6114). arXiv. https://doi.org/10.48550/arXiv.1312.6114

Langdon, C., & Engel, T. A. (2025). Latent circuit inference from heterogeneous neural responses during cognitive tasks. *Nature Neuroscience*, *28*(3), 665–675. https://doi.org/10.1038/s41593-025-01869-7

Machens, C. K., Romo, R., & Brody, C. D. (2005). Flexible Control of Mutual Inhibition: A Neural Model of Two-Interval Discrimination. *Science*, *307*(5712), 1121–1124. https://doi.org/10.1126/science.1104171

Mante, V., Sussillo, D., Shenoy, K. V., & Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, *503*(7474), 78–84. https://doi.org/10.1038/nature12742

Marr, D. (2010). *Vision: A computational investigation into the human representation and processing of visual information*. MIT press.

Mastrogiuseppe, F., & Ostojic, S. (2018). Linking Connectivity, Dynamics, and Computations in Low-Rank Recurrent Neural Networks. *Neuron*, *99*(3), 609-623.e29. https://doi.org/10.1016/j.neuron.2018.07.003

Miller, E. K., & Cohen, J. D. (2001). An Integrative Theory of Prefrontal Cortex Function. *Annual Review of Neuroscience*, *24*(1), 167–202. https://doi.org/10.1146/annurev.neuro.24.1.167

Mountcastle, V. B., Atluri, P. P., & Romo, R. (1992). Selective output-discriminative signals in the motor cortex of waking monkeys. *Cerebral Cortex (New York, N.Y.: 1991)*, *2*(4), 277–294. https://doi.org/10.1093/cercor/2.4.277

Mountcastle, V., Steinmetz, M., & Romo, R. (1990). Frequency discrimination in the sense of flutter: Psychophysical measurements correlated with postcentral events in behaving monkeys. *The Journal of Neuroscience*, *10*(9), 3032–3044. https://doi.org/10.1523/JNEUROSCI.10-09-03032.1990

Okazawa, G., Hatch, C. E., Mancoo, A., Machens, C. K., & Kiani, R. (2021). Representational geometry of perceptual decisions in the monkey parietal cortex. *Cell*, *184*(14), 3748-3761.e18. https://doi.org/10.1016/j.cell.2021.05.022

Pagan, M., Tang, V. D., Aoi, M. C., Pillow, J. W., Mante, V., Sussillo, D., & Brody, C. D. (2024). Individual variability of neural computations underlying flexible decisions. *Nature*. https://doi.org/10.1038/s41586-024-08433-6

Raposo, D., Kaufman, M. T., & Churchland, A. K. (2014). A category-free neural population supports evolving demands during decision-making. *Nature Neuroscience*, *17*(12), 1784–1792. https://doi.org/10.1038/nn.3865

Romo, R., Brody, C. D., Hernández, A., & Lemus, L. (1999). Neuronal correlates of parametric working memory in the prefrontal cortex. *Nature*, *399*(6735), 470–473. https://doi.org/10.1038/20939

Romo, R., Hernández, A., & Zainos, A. (2004). Neuronal Correlates of a Perceptual Decision in Ventral Premotor Cortex. *Neuron*, *41*(1), 165–173. https://doi.org/10.1016/S0896-6273(03)00817-1

Romo, R., Hernández, A., Zainos, A., Lemus, L., & Brody, C. D. (2002). Neuronal correlates of decision-making in secondary somatosensory cortex. *Nature Neuroscience*, *5*(11), 1217–1225. https://doi.org/10.1038/nn950

Romo, R., Hernández, A., Zainos, A., & Salinas, E. (1998). Somatosensory discrimination based on cortical microstimulation. *Nature*, *392*(6674), 387–390. https://doi.org/10.1038/32891

Sussillo, D., Churchland, M. M., Kaufman, M. T., & Shenoy, K. V. (2015). A neural network that finds a naturalistic solution for the production of muscle activity. *Nature Neuroscience*, *18*(7), 1025–1033. https://doi.org/10.1038/nn.4042

Tian, Z., Chen, J., Zhang, C., Min, B., Xu, B., & Wang, L. (2024). Mental programming of spatial sequences in working memory in the macaque frontal cortex. *Science*, *385*(6716), eadp6091. https://doi.org/10.1126/science.adp6091

Vyas, S., Golub, M. D., Sussillo, D., & Shenoy, K. V. (2020). Computation Through Neural Population Dynamics. *Annual Review of Neuroscience*, *43*, 249–275. https://doi.org/10.1146/annurev-neuro-092619-094115

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

Xie, Y., Hu, P., Li, J., Chen, J., Song, W., Wang, X.-J., Yang, T., Dehaene, S., Tang, S., Min, B., & Wang, L. (2022). Geometry of sequence working memory in macaque prefrontal cortex. *Science*, *375*(6581), 632–639. https://doi.org/10.1126/science.abm0204

Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T., & Wang, X.-J. (2019). Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, *22*(2), 297–306. https://doi.org/10.1038/s41593-018-0310-2

Zhang, Y., Feng, J., & Min, B. (2024). *Elucidating the Selection Mechanisms in Context-Dependent Computation through Low-Rank Neural Network Modeling*. https://doi.org/10.7554/elife.103636.1

Methods

Mean-field theory of low-rank RNNs

Understanding how a network's connectivity relates to its function is a fundamental challenge in the study of recurrent neural networks (RNNs). This problem is particularly complex for general RNNs due to their high-dimensional and nonlinear dynamics. Recently, a theoretical framework based on mean-field theory has been introduced, demonstrating that the dynamics of low-rank networks can be expressed in a solvable form when the connectivity strength of each neuron is randomly sampled from a Multivariate Gaussian Mixture Model (GMM).

Consider a rank-R low-rank RNN with N neurons, where the dynamics are described by the following equation:

$$\tau \frac{d\boldsymbol{x}(t)}{dt} = -\boldsymbol{x}(t) + J\boldsymbol{r}(t) + \sum\nolimits_{l=1}^{C} \boldsymbol{I}_l u_l\,(t), \tag{1.}$$

$$\boldsymbol{r}(t) = \phi\big(\boldsymbol{x}(t)\big) \tag{2.}$$

$$J = \frac{1}{N}\sum\nolimits_{j=1}^{R} \boldsymbol{a}_j \boldsymbol{b}_j^T\,. \tag{3.}$$

Where τ is the time-constant, $\boldsymbol{x}(t)$ is a N-dimensional vector representing the activation of each neuron at time t, $\boldsymbol{r}(t)$ is activity vector, ϕ is a non-linear activation function, and J is the connectivity matrix of the network. Here, $\boldsymbol{a}_j$ and $\boldsymbol{b}_j$ are the j-th left and right connectivity vectors, respectively. The term $\boldsymbol{I}_l$ represents the l-th input vector, and $u_l(t)$ is the corresponding time-dependent input signal. The network's output is defined as:

$$z(t) = \frac{1}{N}\boldsymbol{w}^T\phi\big(\boldsymbol{x}(t)\big), \tag{4.}$$

where $\boldsymbol{w} \in \mathbb{R}^N$ is the readout vector.

The network's activation $\boldsymbol{x}(t)$ is constrained in the subspace spanned by the left connectivity vectors $\{\boldsymbol{a}_j\}$ and input vectors $\{\boldsymbol{I}_l\}$, and can be expressed as:

$$\boldsymbol{x}(t) = \sum\nolimits_{j=1}^{R} \kappa_j(t)\boldsymbol{a}_j + \sum\nolimits_{l=1}^{C} \nu_l(t)\boldsymbol{I}_l\,. \tag{5.}$$

where $k_j(t)$ and $\nu_l(t)$ are the task and input variable associated with $\boldsymbol{a}_j$ and $\boldsymbol{I}_l$. respectively. Together, these variables are referred to as latent variables. The dynamics of these variables are termed as latent dynamics, given by:

$$\tau\frac{d\kappa_j}{dt} = -\kappa_j + \frac{1}{N} n_j \phi\left(\sum\nolimits_{i=1}^{R} \kappa_i(t)\boldsymbol{m}_i + \sum\nolimits_{l=1}^{C} \nu_l(t)\boldsymbol{I}_l\right), \tag{6.}$$

$$\tau\frac{d\nu_l}{dt} = -\nu_l + u_l(t). \tag{7.}$$

Assume that the **connectivity component vector** $\{I_{1,i}, \dots, I_{s,i}, a_{1,i}, \dots, a_{R,i}, b_{1,i}, \dots, b_{R,i}, w_i\}$, associated with the i-th neuron, is sampled from a Gaussian Mixture Model (GMM) with P modules. The probability density of the connectivity component vector is given by:

$$P(I_1, \dots, I_S, a_1, \dots, a_R, b_1, \dots, b_R, w) = \sum\nolimits_{p=1}^{P} \alpha_p \mathcal{N}(\boldsymbol{\mu}_p, \Sigma_p), \tag{8.}$$

where α_p is the weight of the p-th module, and each module is a Gaussian distribution with mean $\boldsymbol{\mu}_p$ and covariance matrix Σ_p. Under these assumptions, in the mean-field limit ($N \to \infty$), the dynamics of the task variables are described by:

$$\tau\frac{d\kappa_j}{dt} = -\kappa_j + \sum_{p=1}^{P} \alpha_p \left[\mu_{b_j}^{(p)}\langle\phi\rangle_p + \left(\sum_{i=1}^{R} \sigma_{n_j,m_i}^{(p)}\kappa_i + \sum_{l=1}^{C} \sigma_{n_j,I_l}^{(p)}\nu_l\right)\langle\phi'\rangle_p\right], \tag{9.}$$

where $\mu_{b_j}^{(p)}$ is the mean of b_r in the p-th module, and $\sigma_{b_j,a_i}^{(p)}$ ($\sigma_{b_j,I_l}^{(p)}$) is the covariance of b_j with a_i (I_l) in the p-th module, as defined in Σ_p. The term $\langle\phi\rangle_p$ and $\langle\phi'\rangle_p$ are the average activity and average **gain** of the p-th module, respectively, and are defined as:

$$\langle\phi\rangle_p = \frac{1}{2\pi}\int_{-\infty}^{+\infty} dz\, e^{-\frac{z^2}{2}}\phi\left(\Delta_p(t)z + \xi_p(t)\right), \tag{10.}$$

$$\langle\phi'\rangle_p = \frac{1}{2\pi}\int_{-\infty}^{+\infty} dz\, e^{-\frac{z^2}{2}}\phi'\left(\Delta_p(t)z + \xi_p(t)\right), \tag{11.}$$

where $\Delta_p^2(t) = \vec{\boldsymbol{\kappa}}^T\Sigma_{I,a}^{(p)}\vec{\boldsymbol{\kappa}}$, $\xi_p(t) = \vec{\boldsymbol{\kappa}}^T\mu_{I,a}^{(p)}$. Here $\vec{\boldsymbol{\kappa}} = [\nu_1, \dots, \nu_C, \kappa_1, \dots, \kappa_R]^T$ is the vector of all input and task variables, $\Sigma_{I,a}^{(p)}$ is the submatrix of Σ_p corresponding to the set of variable $\{I_1, \dots, I_C, a_1, \dots, a_R\}$. Similarly, $\mu_{I,a}^{(p)}$ is the subset of the mean vector $\boldsymbol{\mu}_p$ that corresponds to these variables. For simplicity, in rest of the paper, we further assume $\mu_{b_j}^{(p)} = 0$ for all ranks and modules, reducing the dynamics of task variables to:

$$\tau\frac{d\kappa_j}{dt} = -\kappa_j + \sum_{i=1}^{R} E_{\kappa_i\to\kappa_j}\kappa_i + \sum_{l=1}^{C} E_{\nu_l\to\kappa_j}\nu_l\,, \tag{12.}$$

where $E_{\kappa_i\to\kappa_j}$ ($E_{\nu_l\to\kappa_j}$)is the effective connectivity strength of the κ_i (ν_l) to κ_j pathway defined by:

$$E_{\kappa_i\to\kappa_j} = \sum\nolimits_{p=1}^{P} E_{\kappa_i\to\kappa_j,p}\,, \tag{13.}$$

$$E_{\nu_l\to\kappa_j} = \sum\nolimits_{p=1}^{P} E_{\nu_l\to\kappa_j,p}\,, \tag{14.}$$

$$E_{\kappa_i\to\kappa_j,p} = \alpha_p\sigma_{b_j,a_i}^{(p)}\langle\phi'\rangle_p, \tag{15.}$$

$$E_{\nu_l \to \kappa_j, p} = \alpha_p \sigma^{(p)}_{b_j, I_l} \langle \phi' \rangle_p. \qquad (16.)$$

Here each $E_{\kappa_i \to \kappa_j, p}$ ($E_{\nu_l \to \kappa_j, p}$) stands for the effective connectivity strength mediated by the p-th module.

Necessary of multiple modules. The capacity of a network with only one single module is limited. Consider a network containing two input variables ν_1 and ν_2 and one decision variable k_{dv} is designed to handle context-dependent computation. In context 1, κ_{dv} have to receive the information from ν_1, whereas in context 2, it should instead receive input from ν_2. If the network only has one module, the effective connectivity strength of $\nu_1 \to \kappa_{dv}$ and $\nu_2 \to \kappa_{dv}$ are both controlled by the same value $\langle \phi' \rangle_1$, which means that the two path are either simultaneously active or inactive. This constraint (one module) prevents the network from selectively routing different inputs in different contexts. As a conclusion, incorporating multiple modules enables the network to support more flexible computations.

Restricted-RNN training framework

Restricted-RNN Training framework: An overview

In the first section of the Methods, we introduced the mean-field theory of low-rank RNNs. Previous studies have primarily used this theory as a post hoc tool to explain trained low-rank RNNs, which were initially trained at the implementation level by adjusting neuron-to-neuron connection weights. Afterwards, mean-field theory was applied to describe the resulting latent dynamics with collective-level parameters. However, this approach fails when the trained connectivity matrix becomes too complex to be effectively modeled by a Gaussian Mixture Model (GMM) with a reasonable number of clusters, as required by the cognitive task or indicated by experimental data.

By contrast, our method adopts an alternative perspective. We directly use collective-level parameters to specify the network and then employ a generative process to construct the implementation-level RNN. For this RNN, the training objective focuses on the collective-level parameters, rather than on individual connection weights.

In this section, we will elaborate on the details of the Restricted-RNN framework, including a novel pathway-based generative model, how to get an instantiated RNN and the overall training pipeline of the framework.

Pathway-based generative model

We now consider building an RNN with P modules, R task variables and C input variables, where each module contains N units. Because there are R task variables and C input variables, the network includes $R^2 + RS$ possible pathways. The weight of each module is parameterized by:

$$\alpha_p = \frac{e^{\omega_p}}{\sum_{l=1}^{P} e^{\omega_l}}, \tag{17.}$$

where $\{\omega_1, \dots, \omega_P\}$ are unnormalized weight parameters. Each pathway $\kappa_i \to \kappa_j$ (or $\nu_s \to \kappa_j$) represents the information flow from the sender variable to the receiver variable. Here the sender variable can be an internal task variable or an input variable, whereas the receiver variable can only be an internal task variable. Specifically, a pathway from κ_i (ν_s) to κ_j is associated with $2P$ **connectivity parameters**, denoted $S^{in}_{\kappa_i \to \kappa_j, p}$ and $S^{out}_{\kappa_i \to \kappa_j, p}$ ($S^{in}_{\nu_s \to \kappa_j, p}$ and $S^{out}_{\nu_s \to \kappa_j, p}$) (where $p = 1, \dots, P$).

The **input connectivity vector** for $\kappa_i \to \kappa_j$ is a PN-dimensional vector given by:

$$\boldsymbol{\mathcal{T}}^{in}_{\kappa_i \to \kappa_j} = (\mathcal{T}^{in}_{\kappa_i \to \kappa_j, 1}, \dots, \mathcal{T}^{in}_{\kappa_i \to \kappa_j, PN})^T. \tag{18.}$$

Here, each element is generated by:

$$\mathcal{T}^{in}_{\kappa_i \to \kappa_j, n} = \epsilon_{\kappa_i \to \kappa_j, n} S^{in}_{\kappa_i \to \kappa_j, p_n} M_{\kappa_i \to \kappa_j}, \tag{19.}$$

where $p_n = \lfloor (n-1)/N \rfloor + 1$ denoting which module the n-th neuron belong to (the first to the N-th neurons belong to module 1, the $(N+1)$-th to the $2N$-th neurons belong to module 2 and so on), $\epsilon_{\kappa_i \to \kappa_j, n}$ is independently sampled from a standard normal distribution $\mathcal{N}(0, 1)$ and $M_{\kappa_i \to \kappa_j} \in \{0, 1\}$ is a **structural mask** to specify whether the $\kappa_i \to \kappa_j$ pathway is active (1) or inactive (0).

Similarly, the **output connectivity vector** for $\kappa_i \to \kappa_j$ is defined as:

$$\boldsymbol{\mathcal{T}}^{out}_{\kappa_i \to \kappa_j} = (\mathcal{T}^{out}_{\kappa_i \to \kappa_j, 1}, \dots, \mathcal{T}^{out}_{\kappa_i \to \kappa_j, PN})^T, \tag{20.}$$

with each element given by

$$\mathcal{T}^{out}_{\kappa_i \to \kappa_j, i} = P \alpha_{p_n} \epsilon_{\kappa_i \to \kappa_j, n} S^{out}_{\kappa_i \to \kappa_j, p_n} M_{\kappa_i \to \kappa_j}. \tag{21.}$$

Note that the same random variable $\epsilon_{\kappa_i \to \kappa_j, n}$ is shared between the input and output connectivity vectors. Using the same way, we can define the input and output connectivity vector for $\nu_l \to \kappa_j$ as:

$$\mathcal{T}^{in}_{\nu_l \to \kappa_j, n} = \epsilon_{\nu_l \to \kappa_j, n} S^{in}_{\nu_l \to \kappa_j, p_n} M_{\nu_l \to \kappa_j}, \tag{22.}$$

$$\mathcal{T}^{out}_{\nu_l \to \kappa_j, n} = P \alpha_{p_n} \epsilon_{\nu_l \to \kappa_j, n} S^{out}_{\nu_l \to \kappa_j, p_n} M_{\nu_l \to \kappa_j}. \tag{23.}$$

Assembling input and out connectivity vectors into an instantiated RNN

We can assemble the input and output connectivity vectors into an RNN containing PN neurons. Let $\boldsymbol{x}(t)$ is the PN-dimensional activation vector and the activity vector is given by $\boldsymbol{r}(t) = \phi(\boldsymbol{x}(t))$, as defined in equation (2).

The terms $\sum_{n=1}^{PN} r_n(t)\,\mathcal{T}_{\kappa_i\to\kappa_j,n}^{out}$ and $\sum_{n=1}^{PN} r_n(t)\,\mathcal{T}_{\nu_l\to\kappa_j,n}^{out}$ capture the information flowing from κ_i to κ_j and from ν_l to κ_j, respectively. Summing over all sender variables yields the total input to κ_r:

$$\kappa_j^{\text{rec}}(t) = \frac{1}{PN}\sum_{n=1}^{PN} r_n(t)\left(\sum_{i=1}^{R} \mathcal{T}_{\kappa_i\to\kappa_j,n}^{\text{out}} + \sum_{l=1}^{C} \mathcal{T}_{\nu_l\to\kappa_j,n}^{\text{out}}\right). \qquad (24.)$$

Next, the total input to unit n is as following:

$$\hat{x}_n(t) = \sum_{j=1}^{R} \kappa_j^{\text{rec}}(t)\left(\sum_{i=1}^{R} \mathcal{T}_{\kappa_j\to\kappa_i,n}^{in} + \mu_{\kappa_j}^{(p_n)}\right) + \sum_{l=1}^{C} u_l(t)\left(\sum_{i=1}^{R} \mathcal{T}_{\nu_l\to\kappa_i,n}^{in} + \mu_{\nu_l}^{(p_n)}\right). \qquad (25.)$$

where $\mu_{\kappa_j}^{(p_n)}$ and $\mu_{\nu_l}^{(p_n)}$ are parameters represents the modular specific mean value for the corresponding task variable. At the implementation level, the RNN dynamics follow:

$$\tau\frac{d\boldsymbol{x}}{dt} = -\boldsymbol{x} + \hat{\boldsymbol{x}}(t). \qquad (26.)$$

It can be verified that the instantiated RNN is a rank-R low-rank network described by:

$$\tau\frac{d\boldsymbol{x}}{dt} = -\boldsymbol{x} + \frac{1}{PN}\left(\sum_{j=1}^{R} \boldsymbol{a}_j\boldsymbol{b}_j^T\right)\phi(\boldsymbol{x}) + \sum_{l=1}^{C} \boldsymbol{I}_l u_l(t)\,, \qquad (27.)$$

where $\boldsymbol{a}_j$ is a (PN)-dimensional vector with element

$$a_{j,n} = \sum_{i=1}^{R} \mathcal{T}_{\kappa_j\to\kappa_i,n}^{in} + \mu_{\kappa_j}^{(p_n)}, \qquad (28.)$$

$\boldsymbol{b}_j$ is a (PN)-dimensional vector with element

$$b_{j,n} = \sum_{i=1}^{R} \mathcal{T}_{\kappa_i\to\kappa_j,n}^{out} + \sum_{l=1}^{C} \mathcal{T}_{\nu_l\to\kappa_j,n}^{out} \qquad (29.)$$

and $\boldsymbol{I}_l$ is a PN-dimensional input vector with element

$$I_{l,n} = \sum_{i=1}^{R} \mathcal{J}^{in}_{\nu_l\to\kappa_i,n} + \mu^{(p_n)}_{\nu_l}. \tag{30.}$$

The readout vector $\boldsymbol{w}$ is expressed as a linear combination of all output connectivity vectors:

$$\boldsymbol{w} = \sum_{i=1}^{R}\sum_{j=1}^{R} \gamma_{\kappa_i\to\kappa_j}\boldsymbol{\mathcal{J}}^{out}_{\kappa_i\to\kappa_j} + \sum_{l=1}^{C}\sum_{j=1}^{R} \gamma_{\nu_l\to\kappa_j}\boldsymbol{\mathcal{J}}^{out}_{\nu_l\to\kappa_j}, \tag{31.}$$

where $\gamma_{\kappa_i\to\kappa_j}$ and $\gamma_{\nu_l\to\kappa_j}$ are free parameters. The network's output is defined by:

$$z(t) = \frac{1}{NP}\boldsymbol{w}^T\phi(\boldsymbol{x}). \tag{32.}$$

Cross-level matching for the instantiated RNN

For each neuron in the p-th module in the instantiated RNN (Equation 27) is sampled in a multivariate Gaussian distribution with $S + 2 \times R + 1$. Mean of the distribution is given by:

$$\mu_p = \left[\mu^{(p)}_{\nu_1}, \dots, \mu^{p}_{\nu_C}, \mu^{(p)}_{\kappa_1}, \dots, \mu^{(p)}_{\kappa_R}, 0, \dots, 0\right]^T \tag{33.}$$

The covariance matrix can be expressed in the following form:

$$\Sigma_p = \begin{bmatrix} \Sigma^{(p)}_{II} & \Sigma^{(p)}_{Ia} & \Sigma^{(p)}_{Ib} & \Sigma^{(p)}_{Iw} \\ \Sigma^{(p)}_{aI} & \Sigma^{(p)}_{aa} & \Sigma^{(p)}_{ab} & \Sigma^{(p)}_{aw} \\ \Sigma^{(p)}_{bI} & \Sigma^{(p)}_{ba} & \Sigma^{(p)}_{bb} & \Sigma^{(p)}_{bw} \\ \Sigma^{(p)}_{wI} & \Sigma^{(p)}_{wa} & \Sigma^{(p)}_{wb} & \Sigma^{(p)}_{ww} \end{bmatrix} \tag{34.}$$

Each element is given by:

$$\sigma^{(p)}_{I_i,I_j} = \delta_{ij}\sum_{l=1}^{R} M_{\nu_i\to\kappa_l}\left(S^{in}_{\nu_i\to\kappa_l,p}\right)^2, \tag{35.}$$

$$\sigma^{(p)}_{I_l,b_j} = P\alpha_p M_{\nu_l\to\kappa_j} S^{in}_{\nu_l\to\kappa_j,p} S^{out}_{\nu_l\to\kappa_j,p}, \tag{36.}$$

$$\sigma_{I_l,w}^{(p)} = P\alpha_p \sum\nolimits_{j=1}^{R} \gamma_{\nu_l\to\kappa_j} M_{\nu_l\to\kappa_j} S_{\nu_l\to\kappa_j}^{in} S_{\nu_l\to\kappa_j}^{out}\,, \tag{37.}$$

$$\sigma_{a_i,a_j}^{(p)} = \delta_{ij} \sum\nolimits_{l=1}^{R} M_{\kappa_i\to\kappa_l} \left(S_{\kappa_i\to\kappa_l,p}^{in}\right)^2, \tag{38.}$$

$$\sigma_{a_i,b_j}^{(p)} = P\alpha_p M_{\kappa_i\to\kappa_j} S_{\kappa_i\to\kappa_j,p}^{in} S_{\kappa_i\to\kappa_j,p}^{out}, \tag{39.}$$

$$\sigma_{a_i,w}^{(p)} = P\alpha_p \sum\nolimits_{j=1}^{R} \gamma_{\kappa_i\to\kappa_j} M_{\kappa_i\to\kappa_j} S_{\kappa_i\to\kappa_j}^{in} S_{\kappa_i\to\kappa_j}^{out}\,, \tag{40.}$$

$$\sigma_{b_i,b_j}^{(p)} = \left(P\alpha_p\right)^2 \delta_{ij} \left(\sum\nolimits_{l=1}^{R} M_{\kappa_l\to\kappa_i} \left(S_{\kappa_l\to\kappa_i,p}^{out}\right)^2 + \sum\nolimits_{l=1}^{C} M_{\nu_l\to\kappa_i} \left(S_{\nu_l\to\kappa_i,p}^{out}\right)^2\right), \tag{41.}$$

$$\sigma_{b_j,w}^{(p)} = \left(P\alpha_p\right)^2 \left[\sum\nolimits_{i=1}^{R} \gamma_{\kappa_i\to\kappa_j} M_{\kappa_i\to\kappa_j} \left(S_{\kappa_i\to\kappa_j}^{out}\right)^2 + \sum\nolimits_{l=1}^{C} \gamma_{\nu_l\to\kappa_j} M_{\nu_l\to\kappa_j} \left(S_{\nu_l\to\kappa_j}^{out}\right)^2\right], \tag{42.}$$

Therefore, the instantiate RNN (Equation 27) satisfy both the low-rank structure and the modular structure. Under the mean-field limit ($N \to \infty$), the latent variable defined by Equation 5 and Equation 27 follow the dynamical equation described below:

$$\tau \frac{d\kappa_j}{dt} = -\kappa_j + \sum_{p=1}^{P} \frac{1}{P}\left[\left(\sum_{i=1}^{R} \sigma_{a_i,b_j}^{(p)} \kappa_i + \sum_{l=1}^{C} \sigma_{I_l,b_j}^{(p)} \nu_l\right) \langle \phi' \rangle_p\right], \tag{43.}$$

which, after simplification, can be expressed as

$$\tau \frac{d\kappa_j}{dt} = -\kappa_j + \sum_{i=1}^{R} E_{\kappa_i\to\kappa_j} \kappa_i + \sum_{l=1}^{C} E_{\nu_l\to\kappa_j} \nu_l \tag{44.}$$

Here, the effective coupling strengths are:

$$E_{\kappa_i\to\kappa_j} = M_{\kappa_i\to\kappa_j} \sum\nolimits_{p=1}^{P} \alpha_p \, \langle \phi' \rangle_p S_{\kappa_i\to\kappa_j,p}^{in} S_{\kappa_i\to\kappa_j,p}^{out}, \tag{45.}$$

$$E_{\nu_l \to \kappa_j} = M_{\nu_l \to \kappa_j} \sum_{p=1}^{P} \alpha_p \, \langle \phi' \rangle_p S^{in}_{\nu_l \to \kappa_j, p} S^{out}_{\nu_l \to \kappa_j, p}, \tag{46.}$$

The modular gain $\langle \phi' \rangle_p$ can be expressed as a closed form:

$$\langle \phi' \rangle_p = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{1}{2\pi} \int_{-\infty}^{+\infty} dz \, e^{-\frac{z^2}{2}} \phi' \left(\Delta_p(t) z + \xi_p(t) \right), \tag{47.}$$

where $\Delta_p(t)$ and $\xi_p(t)$ are given by:

$$\Delta_p^2(t) = \sum_{l=1}^{C} \left[\sum_{j=1}^{R} M_{\nu_l \to \kappa_j} \left(S^{in}_{\nu_l \to \kappa_j, p} \right)^2 \right] \nu_l^2 + \sum_{j=1}^{R} \left[\sum_{i=1}^{R} M_{\kappa_j \to \kappa_i} \left(S^{in}_{\kappa_j \to \kappa_i, p} \right)^2 \right] \kappa_j^2 \tag{48}$$

Pipeline of the restricted-RNN training framework

First, we propose a hypothesized information flow structure based on task-specific knowledge or preliminary analyses of neural recordings. This structure captures both neuronal connectivity and the patterns of information propagation within the network. The proposed structure serves as a structural prior for training the restricted-RNN model, where the network learns to accomplish the designated tasks under these constraints. Once the trained restricted -RNN successfully performs the target tasks, the model's internal dynamics can be examined by visualizing an information flow diagram, which reveals how various task variables interact. In addition, mean-field theory can be applied to further investigate how network functionality aligns with the underlying structural properties, thereby illuminating the model's internal mechanisms.

After gaining a thorough understanding of the restricted-RNN's operational principles, it becomes possible to formulate multi-level predictions about neuronal activity or the functional characteristics of the network. These predictions are then compared against actual neural data to assess the model's validity. If significant discrepancies arise, the structural priors are revised accordingly, and the process is repeated until a satisfactory model is achieved.