2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

Synthesizing network dynamics for short-term memory of impulsive inputs

BethAnna Jones[1] and ShiNung Ching[2]

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9993238

***Abstract*— Illuminating the mechanisms that the brain uses to manage and coordinate its resources is a core question in neuroscience. In particular, circuits and networks in the brain are able to encode, store and recall large amounts of information, in the service of a wide range of functionality. How do the various dynamical mechanisms within these networks allow for such coordination? We consider the specific problem of how the dynamics of networks can enact a representation of input stimuli that is retained over time, i.e., a form of short-term memory. We utilize modeling and control-theoretic methods to approach these questions, treating the state trajectory of a dynamical system as an abstract memory trace of prior inputs. The inputs impinge on the network via a variable gain, which is to be synthesized by optimization. In order to perpetuate these memory traces of stimuli, we propose that this gain is adapted to optimize: i) the error between a ground truth representation of stimuli and the encoding of them; as well as ii) overwriting of prior information. Optimizing over these central tenets of memory, we obtain a 'policy' for adapting the input gain that is dependent on the state of the network. This derived policy yields a recurrent neural network between the policy and the neural circuits, affirming existing theories that the prefrontal cortex may hold subnetworks dedicated to working memory while actively engaging with other neural subnetworks.**

I. Introduction

Short-term memory, i.e., the storage, management, and processing of stimuli, is fundamental to how animals and humans engage with the world [1]. Such memory provides the means to maintain and utilize information for the many tasks and challenges that face us, and governs higher cognitive functions. Specific forms of short term memory, such as working memory, are central in learning and development, intelligence, problem-solving, and planning [2]–[4]. Even minor changes in its mechanisms have been shown to affect daily processing and abet impairments from Attention Deficit Disorder to debilitating conditions such as Alzheimer's [5], [6].

Understanding how networks of neurons in the brain encode short-term memory is a long-standing question in neuroscience. In particular, short-term memory inherently faces a resource allocation problem, limited by time and the amount we can remember [7]–[10]. It has been shown that shifting network activity can corrupt or alter the representation of previous encodings [11], [12]. How does the brain manage its vast number of collaborating neurons in order to encode and manage signal information efficiently, over time? Does the brain optimally reallocate its resources to accommodate new demands while sustaining ongoing memories, and if so, how and under what premises?

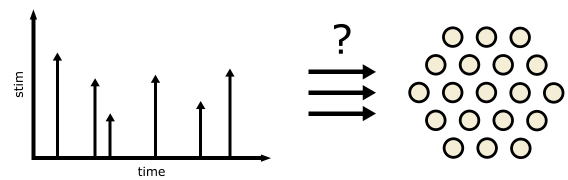

Fig. 1. We model stimuli as a sequence of impulsive inputs, which impinge on a network of dynamical units. The question considered is how to project the each successive stimulus onto these units to preserve a representation over time.

There are several prevailing theories regarding how short-term memories are represented in neuronal networks, many of which are based on dynamical systems notions. For example, it is believed that the dynamics of memory networks in the brain embed a number of asymptotically stable attractors. These attractors enable the maintenance of elevated activity patterns (known as activity 'bumps') in a stable and robust manner over time [13]–[16]. However, the underlying dynamical details in these networks of how memoranda are encoded, maintained, and later decoded are still obscure. In particular, while some attention has been directed at the dynamics of stable memory representation, less attention has been directed at how networks are able to distribute (and later untangle) sequences of incoming input stimuli to maintain sufficient representations while continuously facing the challenge of constant and unpredictable demands on its resources. That is, how are networks able to manage resources and neural activity while facing a sequential multi-item memory task?

In this paper, we approach the above questions in a simplified but intepretable control-theoretic modeling formalism. We postulate a network model consisting of a number of dynamical units [17]. Confronting this network is an input, which conveys a sequence of stimuli to be remembered. Upon this setup we adopt a 'top-down' optimization approach to construct a policy by which the input stimuli should be projected onto the network state (see Fig. 1). The problem at hand is fundamentally one of resource allocation. New stimuli continuously present new demands to the network, which must embed a 'policy' for allocating these stimuli to memory representations, i.e., resources.

This work was partially supported by grants DGE-1745038, DGE-2139839, and CMMI-1653589 from the US National Science Foundation, grant R01EB028154 from the US National Institutes of Health, and grant W911NF2110312 from the US Department of Defense.

[1]B. Jones is with the Department of Electrical & Systems Engineering, Washington Univeristy in St. Louis, MO 63130, USA bethannajones@wustl.edu

[2]S. Ching is with Faculty of Electrical & Systems Engineering, Washington Univeristy in St. Louis, MO 63130, USA shinung@wustl.edu

II. Problem Formulation

A. Multi-slot resource allocation model

We consider the retention over time (i.e., memory) of information from afferent stimuli occurring at brief, sequential moments in time. Specifically, we model stimuli as:

$$\mathbf{u}(t) = \sum_{k \in \mathbf{Z}^+} \delta(t - t_k)\beta_k, \quad (1)$$

where $\delta(\cdot)$ is a dirac delta function and t_k is the time of the k^{th} stimulus in sequence. Here, $\beta_k \in \mathbb{R}^d$ is a vector representing some information about the stimulus in d dimensions, where each dimension can be thought of as a basis of representation or aspect of the inherent information contained in the stimulus. These bases will later influence the derivation of policies regarding how to encode the stimulus.

We then formulate an abstract model for a 'multi-slot' memory network, where each slot is associated with a state variable, x_i, indicating the "activity" of that slot. Here we frame this abstract activity as a quantification of the concept of retention or fidelity of the stimuli currently encoded within that slot. We introduce a phenomenological linear dynamic model for these N variables:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}(t)\mathbf{w}_{\mathbf{x}}\mathbf{u}(t), \quad (2)$$

with given state matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ and bias parameter $\mathbf{w}_{\mathbf{x}} \in \mathbb{R}^{1 \times d}$ that determines how the network weights different bases of the stimuli when encoding. Here we assume $\mathbf{A}$ is Hurwitz. We also introduce the vector $\mathbf{b}(t) \in \mathbb{R}^N$ that acts to project the weighted input $\mathbf{w}_{\mathbf{x}}\mathbf{u}(t)$ onto the network slots. Thus, $\mathbf{b}(t)$ effectively dictates how and to what extent each stimulus β_k is encoded within and across the subnetworks. The vector $\mathbf{b}(t)$ will be the focus of our work moving forward.

B. Top-down synthesis of $\mathbf{b}(t)$

We use a top-down (or, 'normative') modeling approach to specify $\mathbf{b}(t)$ in terms of an overlying mathematical objective function. At a high-level, this objective should embed the following notions:

1) Encoding each incoming stimulus accurately.
2) Maintaining previously encoded information in the system for as long as possible.

The premise for the construction of this process is that an efficient memory system will encode stimuli accurately while also maintaining as much information as possible already stored in the network. In other words, networks should encode as much of a stimulus as possible, modulo the need to preserve previous representations.

With this in mind, we seek to embody these conflicting goals mathematically and find an optimal process $\mathbf{b}(t)$ to balance the trade-off between information accuracy and sustained retention. Importantly, we would like to perform this construction in a manner that is biologically interpretable. Networks of neurons in the brain operate in an online and decentralized fashion, without full access to the network state and without brute-force evaluation of all potential policies. We seek to understand how these determinations can be made quickly and locally as time evolves and still optimize a well-defined cost function.

a) Decoder specification: To proceed, we introduce some additional notation. We define the system's decoder as the linear mapping $\mathbf{C} \in \mathbb{R}^{d \times N}$ from the high-dimensional network space $\mathbb{R}^N$ into the low-dimensional space $\mathbb{R}^d$ of the stimuli:

$$\mathbf{z}(t) = \mathbf{C}\mathbf{x}(t). \quad (3)$$

A key assumption is that $\mathbf{C}$ is of full row rank. This specification of $\mathbf{z}(t)$ defines a representation extracted from the network activity that is directly comparable to the input stimuli.

b) Slot-restricted decoder: In particular, we consider decoding the network but restricted by the matrix $\mathbf{b}(t_k)$, i.e., the encoding at the time of the k^{th} stimulus:

$$\mathbf{z}_k(t) = \mathbf{C}\big[\mathbf{x}(t) \odot \mathbf{b}(t_k)\big], \quad (4)$$

for some $t \geq t_k$. Here the element-wise Hadamard product $\odot$ weights the value of each subnetwork at time t by the policy matrix for β_k. Since $\mathbf{b}(t_k)$ dictates how (and if) the network encodes stimulus β_k, we interpret (4) above as a characterization of the current representation of β_k maintained within the network at time t. As $t - t_k$ grows, we expect $\mathbf{z}_k(t)$ to deviate from β_k as network activity evolves and more stimuli are encountered.

c) Null encoding: In order to analyze the effect of a new encoding on a prior one, we also define the case wherein a new stimulus is not encoded at all. That is, the evolution of a prior representation within the network over time without any impingement from the current stimulus. This value is defined as:

$$\mathbf{z}_i(t)|_{\mathbf{b}(t_k)=0} \equiv \mathbf{C}\big[\mathbf{x}(t)|_{\mathbf{b}(t_k)=0} \odot \mathbf{b}(t_i)\big]$$

where $t_i < t_k$. For ease of notation going forward, we denote the above as

$$\hat{\mathbf{z}}_i(t) = \mathbf{z}_i(t)|_{\mathbf{b}(t_k)=0}.$$

In other words, $\hat{\mathbf{z}}_i(t)$ is a hypothetical decoded signal under the premise that no future stimulus impinges on the network.

C. Optimization

We are now ready to set up the optimization framework for the synthesis of $\mathbf{b}(t)$. Specifically, we can formulate the construction of $\mathbf{b}(t)$ as an optimization problem occurring at the time of each stimulus, t_k:

$$\begin{aligned} \min_{\mathbf{b}(t_k)} \quad & J(t_k) = \lambda_e \|\mathbf{z}(t_k^+) - \beta_k\|_2^2 \qquad \text{(5a)} \\ & \qquad - \lambda_{ow}\|\mathbf{z}_{k-1}(t_k^+) - \hat{\mathbf{z}}_{k-1}(t_k^+)\|_2^2 \\ \text{subject to} \quad & \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}(t)\mathbf{w}_{\mathbf{x}}\mathbf{u}(t), \\ & \mathbf{z}(t) = \mathbf{C}\mathbf{x}(t), \\ & \mathbf{z}_k(t) = \mathbf{C}\big[\mathbf{x}(t) \odot \mathbf{b}(t_k)\big], \\ & t_k^+ = t_k + \Delta t, \ \Delta t > 0 \text{ small}, \\ & \lambda_e, \lambda_{ow} \geq 0. \end{aligned}$$

Thus, the process $\mathbf{b}(t)$ is discontinuous, since it is formed through optimization at the moment of each stimulus, t_k.

a) Interpreting the objective $J(t_k)$: The first term is simply the "encoding error" between the input stimuli and the representation (3) immediately after encoding.

In the second term, $\mathbf{z}_{k-1}(t_k^+)$ is the slot-restricted decoded signal from the network shortly after t_k and the current stimulus impinges on the network. Importantly, the slot restriction here comes from $\mathbf{b}(t_{k-1})$, i.e., the projection matrix associated with the prior stimulus. Conversely, $\hat{\mathbf{z}}_{k-1}(t_k^+)$ is slot-restricted decoded signal under the hypothesis that the current stimulus is ignored. By considering the difference between these quantities, we capture a type of 'regret,' in terms of how the encoded representation of the prior stimulus would be changed through the choice of $\mathbf{b}(t_k)$ and the ensuing effect of the current stimulus. Thus, minimizing this difference effectively discourages the network from encoding successive stimuli in the same slots. Said another way, this regularizing term encourages that stimuli are 'spread out' through the network.

We note that the cost function captures the goals of II-B, insofar as the first term addresses 1), while the second addresses 2) and 3). These cost terms are weighted with λ_e, λ_{ow} to reflect the relative importance of error and overwriting, respectively. The optimization is formulated in a 'pseudo-greedy' fashion, where the construction of $\mathbf{b}(t_k)$ is made on the basis of a cost that is nominally independent of future stimuli.

III. Results

A. Gating policy

We now proceed to derive the optimal policy, stated in the following theorem:

Theorem 1. *Given the model network with N slots with dynamics specified in* (2)*, and decoders* (3)-(4)*, the policy to optimize cost function* (5a) *is given by:*

$$\mathbf{b}(t_k) = \begin{cases} \frac{\lambda_e}{(\mathbf{w_x}\beta_\mathbf{k})} e^{-\mathbf{A}\Delta t}\Phi_k^{-1}\big[\beta_k - \mathbf{C}e^{\mathbf{A}\Delta t}\mathbf{x}(t)\big], & \mathbf{w_x}\beta_k \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where

$$\Phi_k = \lambda_e\mathbf{C} + \lambda_{ow}(\mathbf{C}\mathbf{C}^T)^{-1}\mathbf{C}\theta_k^T\theta_k$$
$$\theta_k = \mathbf{C}\ \mathrm{diag}\big(\mathbf{b}(t_{k-1})\big).$$

Proof: The proof proceeds by first establishing that $J(t_k) : \mathbb{R}^N \to \mathbb{R}^+$ is a convex function and hence a minimizer $\mathbf{b}(t_k)$ of $J(t_k)$ exists. Subsequently, the gradient of the cost can be computed and the minimizer obtained analytically. The details are found in the Appendix.

We note that this policy is applicable to all β_k as long as $\mathbf{b}(t_{k-1})$ is defined. Thus, the policy requires 'initialization.' Furthermore, this policy is independent of the absolute time of each stimulus, and uses only the current state of the network and the prior encoding matrix $\mathbf{b}(t_{k-1})$. Additionally, by definition of $\mathbf{u}(t)$ as a sum of Dirac deltas, we can assume the incoming signal $\mathbf{u}(t) = 0$ between defined stimuli β_k and β_{k+1}, i.e., for $t \in (t_k, t_{k+1})$ for all k. Hence, $\mathbf{b}(\mathbf{t}) = 0$ for $t \in (t_k, t_{k+1})$ as well.

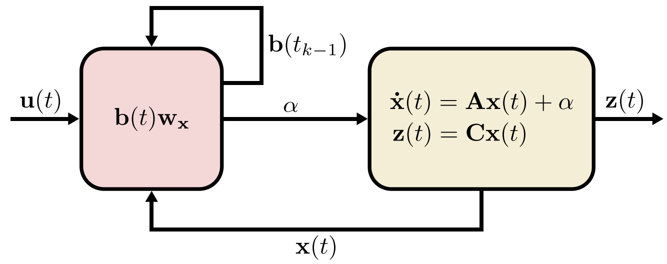

Fig. 2. The policy $\mathbf{b}(t_k)$ acts as a gating gain, weighting the input $\mathbf{u}(t)$ to be added to the network dynamics.

B. Recurrent network

The policy embedded in (6) enjoys an interpretation as adding to the overall structure and dynamics of the network. As noted, $\mathbf{b}(t)$ is dependent on the current state of the network $\mathbf{x}(t)$, and clearly has bearing on future evolution of the state. This reciprocal interaction creates a recurrence, which can form a 'closed-loop' with the network dynamics (See Fig. 2). Indeed, the network dynamics in (2) with the policy (6) can be rewritten as:

$$\dot{\mathbf{x}}(t) = \big[\mathbf{A} - \lambda_e e^{-\mathbf{A}\Delta t}\Phi_k^{-1}\mathbf{C}e^{\mathbf{A}\Delta t}\big]\mathbf{x}(t) + \lambda_e e^{-\mathbf{A}\Delta t}\Phi_k^{-1}\beta_k.$$

Consequently, the derived policy (6) describes additional dynamical mechanisms that become embedded within the network that serve to allocate its finite resources optimally to incoming stimuli, acting as a controller of sorts over the system.

It is of note that the policy is recursive within Φ_k, using the previous encoding matrix to direct the current one. Thus, not only does the policy actively interact with the state, but is itself a recurrent network operating over a nominally slower time-scale.

C. Myopic policy and state-gated gain

In order to obtain intuition about the policy, we first consider the situation when the network is only optimizing encoding accuracy, i.e., when $\lambda_{\text{ow}} = 0$. In the rest of this paper, we will refer to this situation as the myopic, or greedy, policy. In this case, Φ_k^{-1} reduces to the right-pseudoinverse of $\mathbf{C}$. Hence, the encoding error is linearly transformed in $\mathbf{b}(t_k)$ to match the dimension of the state space. As a result, $\mathbf{b}(t_k)$ acts as a 'correction' for how much the current network state is different from that which is required to encode β_k at t_k^+ with minimal error.

When $\lambda_{ow}, \lambda_e \neq 0$, we can similarly show that Φ_k^{-1} works to minimize both cost terms jointly. In this way, $\mathbf{b}(t_k)$ acts as an state-gated gain on the input stimulus. Essentially, this matrix is modulated by current network activity and projects inputs onto the state accordingly.

D. Examples

We consider a system of $N = 40$ slots evolving over a time-step $\tau = 0.01$, encountering stimuli of dimension $d = 5$. We optimize with $\mathbf{w_x}$ defined as a row vector of all ones, $\Delta t = 0.05$, $\lambda_{ow} = 1$, $\lambda_e = 0.9$, and assume a randomly generated initial condition $\mathbf{b}(t_0) \in \mathbb{R}^N$ with entries in $[0, 1]$.

We performed a Monte Carlo experiment over 500 samples, where each sample consists of a set of uniformly distributed stimuli $\beta_1, \beta_2, \ldots, \beta_{50}$ that constitute $\mathbf{u}(t)$. In other words, each sample is a different sequence of stimuli. We note that the corresponding t_k are equally spaced with $t_k \in [1, 11]$ and remain constant across samples. This assumption of equally spaced t_k can be relaxed, but allows us to directly compare how differences in the β_k affect network dynamics separate from stimuli timing. As illustrated in Figure 3, the policies perform as expected. At the time of encoding, the myopic (greedy) policy achieves near-perfect positive correlation, as it optimizes accuracy of the decoded representation independent of prior stimuli (i.e., $r \approx 1$ at time $t = t_k^+$). However, as new stimuli are encountered, the correlation rapidly decreases, so that the representation is effectively destroyed as soon as a subsequent stimulus occurs. Meanwhile, the optimal policy sustains significant positive correlation for at least two successive stimuli, thus retaining a representation of a given stimulus even after a subsequent one occurs. Clearly, both policies outperform a null (random) policy, where $\mathbf{b}(t)$ is set randomly.

IV. Conclusion

A. Summary

We considered a theoretical problem: how should a sequence of stimuli be projected onto a network of dynamical units, so as to retain stimulus representations over time. The problem is motivated by computational neuroscience questions pertaining to how networks in the brain might efficiently allocate circuit resources in the service of short-term memory. Using an abstraction of the problem, involving linear dynamical networks and impulsive stimulus sequences, we derived a 'closed-loop' policy wherein the determination of how to successive stimuli should impinge on the network is made through an analytical relation depending on the current state of the network itself. Specifically, starting with a standard linear dynamic system, we derived an optimal encoding policy that balances working memory objectives. The dynamics that emerge from this optimal encoding policy $\mathbf{b}(t)$ are inherently recurrent and project stimuli onto network slots in an online fashion. The results developed here are seen as a first step in a broader theoretical study of network dynamics and memory function, with several assumptions warranting further attention, particularly in the eventual reconciliation of this theory with neurobiology.

V. Future work

a) The decoder: The policy is based on the assumption of a linearly decoded output $\mathbf{z}(t)$, which is central in determining the current representations within the network. That is, this decoder acts as a reference and allows us to assess the performance of $\mathbf{b}(t)$. The form and parameters of the decoder are assumed *a priori* and are not optimized themselves. Certainly, since its form does not change over time, the decoder's appraisal of the network is consistent. However, the characteristics of this decoder may influence secondary details of our policy which may become relevant when attempting to reconcile it with particularities in different memory contexts, i.e., visual-spatial, verbal, or attentional working memory.

In fact, even simple differences in the parameterizations on the transformation matrix $\mathbf{C}$ can determine if the activity of a given slot even contributes to the decoded representation at all. There is room in future work to further consider these factors and their interpretibility.

b) The 'retention' term Φ_k: The term Φ_k is recalculated at every new stimulus and is based on the previous matrix $\mathbf{b}(t)$. This is a global step that would imply centralized 'executive control,' in the neurobiological context. Such a requirement is likely implausible and represents the major limitation from a biological interpretability standpoint. Future work will need to explore relaxation or approximation of this step.

c) Sustaining stimulus representations: While our optimization's current cost function encourages network activity to maintain previous representations of stimuli when facing a subsequent demand on its resources, the 'regret' term inherently only considers the following stimulus, allowing future demands to overwrite the original stimulus. In future work, we hope to generalize this objective to better encapsulate our objectives in II-B. Specifically, we hope to extend this sense of network regret to maintain previously encoded information in the system over longer horizons.

A. Future validation of the model

To validate this model and its results, we must compare it to physiological observations in relevant brain areas. Specifically, the dorsolateral prefrontal cortex (DLPFC) is the brain region primarily recognized as most correlated with the control and implementation of executive functions including working memory. As such, the DLPFC has been the focus of an extensive number of experimental studies, and the resulting data and models form a large literature base. In a retrospective approach, we can compare emergent time course activity as well as the hierarchy and connectivity between neural circuits in these relevant regions—the regions' dynamics and architectures, respectively—to the same characteristics from our model, and assess the extent to which they are compatible. We can then adjust the parameterization of the model commensurately. This process will assuredly involve some sensitivity analysis and consideration of the assumptions on $\mathbf{A}$ and $\mathbf{C}$ that govern the network dynamics.

After such a reconciliation process, we can then employ the model more insightfully, emulating the stimuli seen in canonical working memory tasks. We will analyze the resulting time course data and to what extent the model is

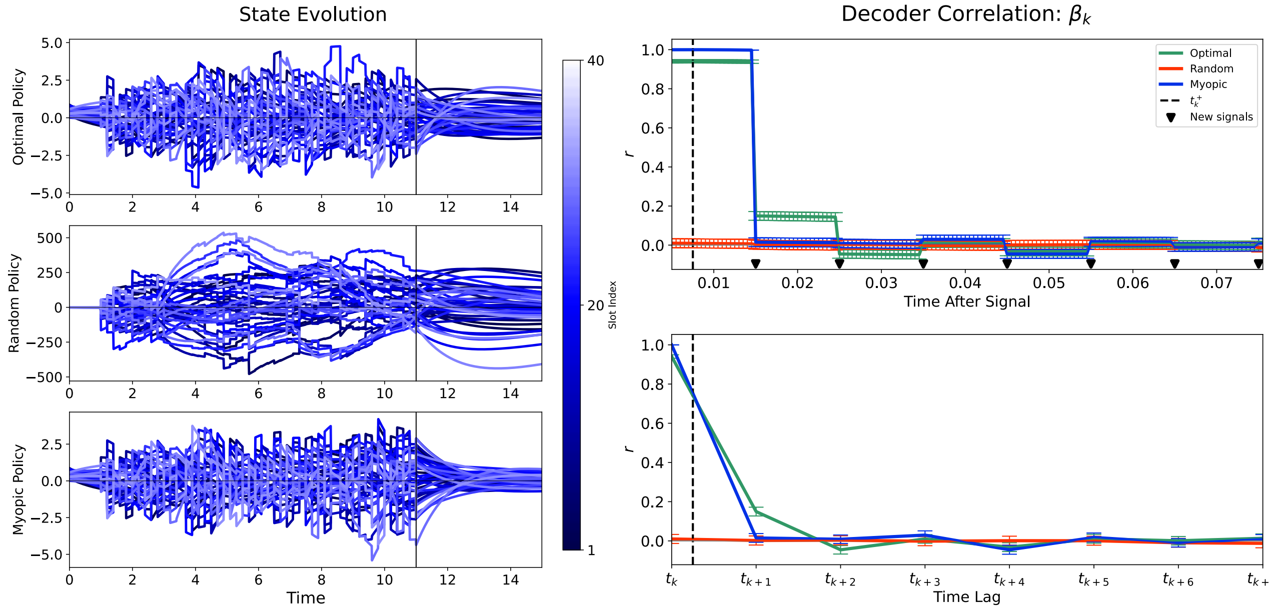

Fig. 3. **(Left)** Example of network activity for a particular $\mathbf{u}(t)$ consisting of 50 successive stimuli. Shown from top to bottom are the activity in all $N = 40$ slots for the optimal, random (null), and myopic policies. **(Right)** The correlation coefficient, r, between the ground truth stimulus β_k and the corresponding decoded representation $\mathbf{z}(t_j)$ for $j \geq k$. Correlation values have been averaged across Monte Carlo samples for a given β_k.

explanatory or predictive of actual brain dynamics, concurrently evaluating again how the assumptions on $\mathbf{A}$ and $\mathbf{C}$ not only affect these predictions, but what the structure of $\mathbf{A}$ and $\mathbf{C}$ may reveal about the underlying mechanisms within neural circuits.

APPENDIX

A. Convexity & Existence

We now consider the function $J(t_k)$. Let $J(t_k) = \lambda_e f(t_k) + \lambda_{ow} g(t_k)$ where

$$\begin{aligned} f(t_k) &= \|\mathbf{z}(t_k^+) - \beta_k\|_2^2 \\ &= \|\Gamma \mathbf{b}(t_k) - \Lambda\|_2^2, \\ g(t_k) &= \|\mathbf{z}_{k-1}(t_k^+) - \hat{\mathbf{z}}_{k-1}(t_k^+)\|_2^2 \\ &= \|\Omega \mathbf{b}(t_k)\|_2^2, \end{aligned}$$

and $\Gamma, \Omega \in \mathbb{R}^{N \times N}$, $\Lambda \in \mathbb{R}^N$. Notice $\Gamma \mathbf{b}(t_k) - \Lambda$ and $\Omega \mathbf{b}(t_k)$ are both linear and clearly convex. It is well-known that the squared norm $\|\cdot\|_2^2$ is convex and non-decreasing, so the compositions $f(t_k)$ and $g(t_k)$ are also convex. It is also well-known that the sum of positively-scaled convex functions are convex. All together, $J(t_k)$ is a sum of positively-scaled nested convex functions and is therefore also convex. Hence there exists a global minimizer $\mathbf{b}(t_k) \in \mathbb{R}^N$ of $J(t_k)$. □

B. Derivation of the Gating Policy

It is well-known that the solution of (2) is given by

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{b}(\tau)\mathbf{w_x}\mathbf{u}(t)\mathrm{d}\tau. \quad (7)$$

By definition of our stimuli in (1), we note each β_k is instantaneously added to the network through $\mathbf{b}(t)$, and so the network state immediately after doing so is given by

$$\begin{aligned} \mathbf{x}(t_k^+) &= e^{\mathbf{A}(t_k^+ - t_k)}\mathbf{x}(t_k) + \int_{t_k}^{t_k^+} e^{\mathbf{A}(t_k^+ - \tau)}\mathbf{b}(\tau)\mathbf{w_x}\mathbf{u}(\tau)\ \mathrm{d}\tau \\ &= e^{\mathbf{A}\Delta t}\mathbf{x}(t_k) + e^{\mathbf{A}\Delta t}\mathbf{b}(t_k)\mathbf{w_x}\beta_k, \end{aligned}$$

where $t_k^+ = t_k + \Delta t$ and Δt is arbitrarily small. Then, the error of encoding β_k is given by

$$\begin{aligned} \mathbf{z}(t_k^+) - \beta_k &= \mathbf{C}\mathbf{x}(t_k^+) - \beta_k \\ &= \mathbf{C}e^{\mathbf{A}\Delta t}\mathbf{x}(t_k) + \mathbf{C}e^{\mathbf{A}\Delta t}\mathbf{b}(t_k)(\mathbf{w_x}\beta_k) - \beta_k. \end{aligned}$$

Similarly, the information loss of β_{k-1} from encoding β_k simplifies to

$$\begin{aligned} \mathbf{z}_{k-1}(t_k^+) - \hat{\mathbf{z}}_{k-1}(t_k^+) &= \mathbf{C}\mathbf{x}(t_k^+) \odot \mathbf{b}(t_{k-1}) - \\ &\qquad \mathbf{C}\mathbf{x}(t_k^+)|_{\mathbf{b}(t_k)=0} \odot \mathbf{b}(t_{k-1}) \\ &= \mathbf{C}e^{\mathbf{A}\Delta t}\mathbf{b}(t_k)\mathbf{w_x}\beta_k \odot \mathbf{b}(t_{k-1}) \\ &= \mathbf{C}\mathrm{diag}\big(\mathbf{b}(t_{k-1})\big)e^{\mathbf{A}\Delta t}\mathbf{b}(t_k)\mathbf{w_x}\beta_k \end{aligned}$$

Since we have shown $J(t_k)$ to be a convex function, the gating policy that optimally minimizes $J(t_k)$ is given by $\nabla J(t_k) = 0$. That is,

$$(\mathbf{w_x}\beta_k)\mathbf{b}(t_k)^T e^{\mathbf{A}^T\Delta t}\Phi_k^T\mathbf{C} = \lambda_e\big[\beta_k - \mathbf{C}e^{\mathbf{A}\Delta t}\mathbf{x}(t)\big]^T\mathbf{C}.$$

After assuming $\mathbf{C}$ is full rank and carefully defining the right-sided psuedoinverses,

$$\begin{aligned} \mathbf{C}^\dagger &= \mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1}, \\ \Phi_k^\dagger &= \Phi_k^T(\Phi_k\Phi_k^T)^{-1}, \end{aligned}$$

we obtain:

$$\mathbf{b}(t_k) = \frac{\lambda_e}{(\mathbf{w_x}\beta_\mathbf{k})} e^{-\mathbf{A}\Delta t}\Phi_k^{-1}\big[\beta_k - \mathbf{C}e^{\mathbf{A}\Delta t}\mathbf{x}(t)\big].$$

when $\mathbf{w_x}\beta_k \neq 0$. We can easily additionally verify that $\mathbf{w_x}\mathbf{u}(t) = 0$ whenever $\mathbf{w_x}\beta_k = 0$ and so setting $\mathbf{b}(t_k) = 0$ in this instance agrees with (2). And we have consequently arrived at the policy (6). □

C. The Recurrent Network

By definition,

$$\mathbf{u}(t) = \begin{cases} \beta_k, & t = t_k \\ 0, & \text{otherwise} \end{cases}$$

and so the policy in (6) can be written as

$$\mathbf{b}(t) = \begin{cases} \frac{\lambda_e}{(\mathbf{w_x}\mathbf{u(t)})} e^{-\mathbf{A}\Delta t}\Phi_k^{-1}\big[\beta_k - \mathbf{C}e^{\mathbf{A}\Delta t}\mathbf{x}(t)\big], & t_k = t \\ 0, & \text{otherwise.} \end{cases}$$

Substituting this policy into the network dynamics,

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \lambda_e e^{-\mathbf{A}\Delta t}\Phi_k^{-1}\big[\beta_k - \mathbf{C}e^{\mathbf{A}\Delta t}\mathbf{x}(t)\big] \\ &= \big[\mathbf{A} - \lambda_e e^{-\mathbf{A}\Delta t}\Phi_k^{-1}\mathbf{C}e^{\mathbf{A}\Delta t}\big]\mathbf{x}(t) + \lambda_e e^{-\mathbf{A}\Delta t}\Phi_k^{-1}\beta_k,\end{aligned}$$

and we obtain the optimal network dynamics. □

REFERENCES

[1] Alan Baddeley. Working memory. *Science*, 255:556–559, 1992.

[2] Nelson Cowan. What are the differences between long-term, short-term, and working memory? *Progress in brain research*, 169:323, 2008.

[3] Nelson Cowan. Working memory underpins cognitive development, learning, and education. *Educational psychology review*, 26:197, 2014.

[4] Anne S. Berry, Theodore P. Zanto, Aaron M. Rutman, Wesley C. Clapp, and Adam Gazzaley. Practice-related improvement in working memory is modulated by changes in processing external interference. *Journal of Neurophysiology*, 102:1779–1789, 9 2009.

[5] Jutta S. Mayer, Robert A. Bittner, Danko Nikolić, Christoph Bledowski, Rainer Goebel, and David E.J. Linden. Common neural substrates for visual working memory and attention. *NeuroImage*, 36:441–453, 6 2007.

[6] Rodrigo Ortega, Vladimir López, Ximena Carrasco, María Josefina Escobar, Adolfo M. García, Mario A. Parra, and Francisco Aboitiz. Neurocognitive mechanisms underlying working memory encoding and retrieval in attention-deficit/hyperactivity disorder. *Scientific Reports 2020 10:1*, 10:1–13, 5 2020.

[7] Weiwei Zhang and Steven J. Luck. Sudden death and gradual decay in visual working memory: Research report. *Psychological Science*, 20:423–428, 4 2009.

[8] Weiwei Zhang and Steven J. Luck. The number and quality of representations in working memory. *Psychological science*, 22:1434, 2011.

[9] Diego Pinal, Montserrat Zurrón, and Fernando Díaz. Effects of load and maintenance duration on the time course of information encoding and retrieval in working memory: From perceptual analysis to post-categorization processes. *Frontiers in Human Neuroscience*, 8:165, 4 2014.

[10] Yoni Pertzov, Sanjay Manohar, and Masud Husain. Rapid forgetting results from competition over time between items in visual working memory. *Journal of experimental psychology. Learning, memory, and cognition*, 43:528–536, 4 2017.

[11] Albert Compte, Nicolas Brunel, Patricia S. Goldman-Rakic, and Xiao Jing Wang. Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cerebral Cortex*, 10:910–923, 9 2000.

[12] Klaus Wimmer, Duane Q. Nykamp, Christos Constantinidis, and Albert Compte. Bump attractor dynamics in prefrontal cortex explains behavioral precision in spatial working memory. *Nature Neuroscience 2014 17:3*, 17:431–439, 2 2014.

[13] Elham Ghazizadeh and ShiNung Ching. Slow manifolds within network dynamics encode working memory efficiently and robustly. *PLOS Computational Biology*, 17:e1009366, 9 2021.

[14] Srinivas G. Rao, Graham V. Williams, and Patricia S. Goldman-Rakic. Destruction and creation of spatial tuning by disinhibition: Gabaa blockade of prefrontal cortical neurons engaged by working memory. *Journal of Neuroscience*, 20:485–494, 1 2000.

[15] E. H. Baeg, Y. B. Kim, K. Huh, I. Mook-Jung, H. T. Kim, and M. W. Jung. Dynamics of population code for working memory in the prefrontal cortex. *Neuron*, 40:177–188, 9 2003.

[16] X. J. Wang. Attractor network models. *Encyclopedia of Neuroscience*, pages 667–679, 2009.

[17] Ziqiang Wei, Xiao Jing Wang, and Da Hui Wang. From distributed resources to limited slots in multiple-item working memory: A spiking network model with normalization. *Journal of Neuroscience*, 32:11228–11240, 8 2012.