eLife

Reviewed Preprint
v1 • July 7, 2026
Not revised

For correspondence:
angus.chadwick@ed.ac.uk

Competing interests: No competing interests declared

Funding: See page 47

Reviewing editor: Sukbin Lim, New York University Shanghai, China

Efficient Working Memory Maintenance via High-Dimensional Rotational Dynamics

Laura Ritter, Angus Chadwick
Institute for Adaptive and Neural Computation, University of Edinburgh, Edinburgh, United Kingdom

eLife Assessment

This **useful** study investigates noise-robust and energy-efficient circuit mechanisms for working memory by optimizing connectivity and reports that the resulting networks exhibit rotational dynamics and better match aspects of PFC population recording. However, the supporting evidence remains **incomplete**, given the restricted linear, task-specific training and analysis, and limited comparisons with other prominent models. The manuscript would be strengthened by extending the analysis to nonlinear dynamics, providing more rigorous comparisons with alternative models, and establishing a stronger link to prior theoretical and experimental work.

https://doi.org/10.7554/eLife.111445.1.sa3

Abstract

Working memory (WM) is fundamental to higher-order cognition, yet the circuit mechanisms through which memoranda are maintained in neural activity after removal of sensory input remain subject to vigorous debate. Prominent theories propose that stimuli are encoded in either stable and persistent activity patterns configured through attractor mechanisms or dynamic and time-varying activity patterns brought about through functionally-feedforward network architectures. However, cortical circuits exhibit heterogeneous responses during WM tasks that are challenging to reconcile with either hypothesis. We hypothesised that these complex response dynamics could emerge from an optimally noise-robust and energetically efficient solution to WM tasks. We show that, in contrast to previous theories, networks optimised for efficient WM encoding exhibit high-dimensional rotational dynamics. We find direct evidence for these rotational dynamics in large-scale recordings from monkey prefrontal cortex. Our findings suggest that the complex and dynamic response properties of WM circuits emerge from efficient coding principles.

Introduction

Working memory (WM) is a fundamental building block of intelligence (Baddeley, 1992). Storing items temporarily in WM enables animals to decouple motor actions from direct sensory input, facilitating flexible and adaptive behaviour and the learning of associations between temporally discontiguous events (Miller and Cohen, 2001). The neural basis of WM has long been thought to rely on persistent activity, with neurons firing both stably and selectively to specific memoranda throughout the delay between sensory input and motor response (Fuster and Alexander, 1971; Fuster, 1973; Funahashi et al., 1989; Goldman-Rakic, 1995). However, recent studies focusing on population-level neural representations during WM tasks have instead found highly dynamic neural activity patterns throughout the delay (Stokes et al., 2013; Stokes, 2015; Sreenivasan et al., 2014; Lundqvist et al., 2018; Stroud et al., 2024a; Daie et al., 2023). These findings suggest that the persistent activity hypothesis, and the mechanistic models inspired by it, are in need of revision.

Classic attractor models of WM dynamics, inspired by the persistent activity hypothesis, postulate that WM contents are stored in stable attractor states of population activity throughout the delay (Seung, 1996; Wang, 2001). In contrast, functionally-feedforward models propose that WM contents are stored in sequences of neural activity spanning the delay (Ganguli et al., 2008; Goldman, 2009; Harvey et al., 2012; Daie et al., 2023). However, neither attractor nor feedforward models can account for dynamic coding properties of WM circuits (Stroud et al., 2024a). While a hybrid model with transient feedforward amplification of external input into a stable attractor accounts for many aspects of dynamic coding (Stroud et al., 2023), attractor mechanisms are highly suboptimal in the presence of noise; recall precision deteriorates rapidly after stimulus offset due to noisy drift of the internal stimulus representation (Ganguli et al., 2008; Lim and Goldman, 2012; Burak and Fiete, 2012).

We asked whether the complex and dynamic activity patterns observed during WM tasks could be explained by a recurrent network architecture optimised for 1) robustness to neuronal noise and 2) energetic efficiency. In particular, noise introduces errors in WM tasks, and accurate WM performance therefore requires noise-resilient dynamics (Orhan and Ma, 2023; Stroud et al., 2024b; Burak and Fiete, 2012; Panichello et al., 2019). Moreover, there is considerable pressure to minimise the energic cost of neural circuit computations (Laughlin, 2001; Padamsey et al., 2022). We therefore developed a novel method to identify solutions to WM tasks that perform optimally in the presence of ongoing noise fluctuations and with minimal metabolic cost.

Surprisingly, the solutions that emerged from this method departed substantially from all previous models (Seung, 1996; Ganguli et al., 2008; Goldman, 2009; Stroud et al., 2023). In particular, we found that rotational dynamics, which are widely observed in a variety of non-WM tasks (Churchland et al., 2012; Aoi et al., 2020; Libby and Buschman, 2021), substantially outperformed the attractor and feedforward WM solutions in terms of both noise-robustness and energetic cost. These dynamics were integrated into high-dimensional feedforward-rotational motifs that closely resembled, but substantially outperformed, state-of-the-art State Space Models (SSMs) recently developed for analysis of time series data (Voelker et al., 2019; Gu et al., 2020, 2023). Moreover, these solutions recapitulated a wide range of functional properties of WM circuits, including dynamic coding at the single-neuron and population level (Spaak et al., 2017), and were present in large-scale recordings from prefrontal cortex of monkeys performing a classic spatial WM task (Panichello et al., 2024).

Our findings suggest that normative pressures on noise-robustness and energetic cost shape the complex dynamics of WM circuits. Given their similarity to SSMs, we suggest that the feedforward-rotational dynamics employed by WM circuits forms a high-dimensional substrate for flexible computation on temporally distributed data (Voelker et al., 2019; Gu et al., 2020, 2023). Beyond WM tasks, the method we develop to optimise noisy RNNs enables integration of the computation-through-dynamics framework with efficient coding theories (Averbeck et al., 2006; Sussillo, 2014; Vyas et al., 2020), allowing rigorous investigation of noise-robust and metabolically efficient dynamics across diverse cognitive tasks.

Results

Optimal Dynamics for Working Memory Tasks

We sought to determine the optimal dynamics for a WM task in which one of several possible input stimuli must be decoded from the response of a network after a delay, with ongoing noise inputs corrupting the stimulus representation over time (Figure 1, Figure 2A,B). For linear networks driven by Gaussian noise, the performance of an optimal decoder can be quantified as a signal-to-noise ratio (SNR, see Methods). We derived this SNR for a set of canonical linear WM models both analytically and numerically (Supplementary Figure S1, Supplementary Material). To test the optimality of these linear networks, we derived a Bayesian upper bound that holds for arbitrary nonlinear networks (Supplementary Figure S2, Supplementary Material).

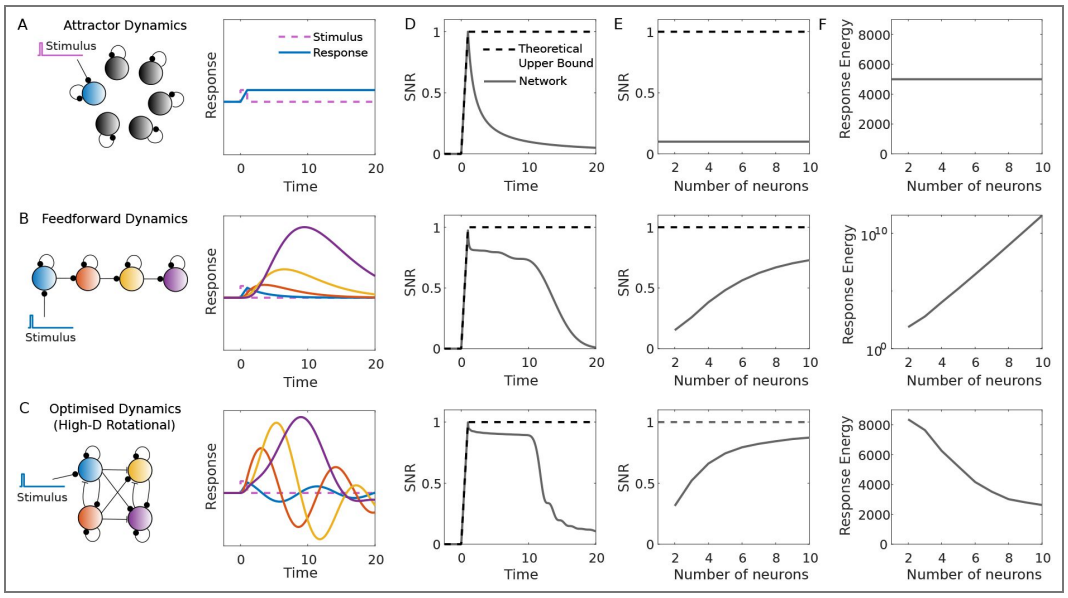

Figure 1. Models for working memory dynamics.

A: Attractor networks generate a set of slow, non-interacting dynamical modes that store stimuli in a stable code after stimulus removal. B: Feedforward dynamics involve directed interactions over a sequence of dynamical modes, and store stimuli in a dynamic sequence of neural activity. C: Optimised networks exhibit complex, high-dimensional rotational dynamics. D: Information maintanence in each network in the presence of noise (with 10 neurons, optimised for stimulus readout at $t = 10$). Attractor networks rapidly lose information after stimulus offset due to accumulation of noise (top). Feedforward and optimised networks can achieve close to theoretically optimal decoding performance up to the optimised readout time. E: Stimulus encoding improves as the number of neurons (or dynamical modes) increases in feedforward and optimised networks, but not in attractor networks. F: The energetic cost (measured as the squared magnitude of the network response integrated over time) increases exponentially with the number of neurons for feedforward networks (note log scale on y axis), is independent of the number of neurons for attractor networks, and decreases with the number of neurons for optimised networks.

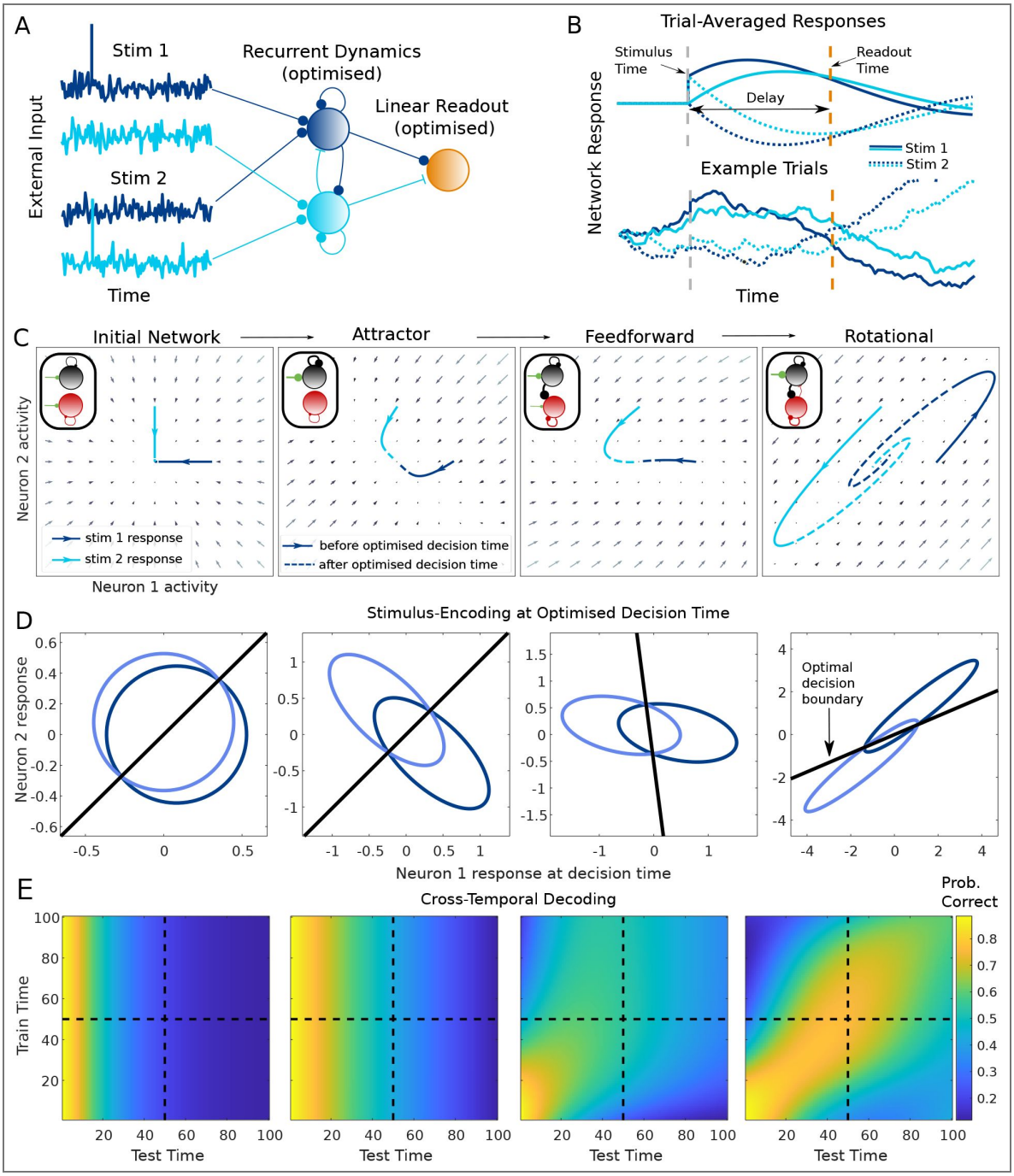

Figure 2. Optimisation of a two-neuron network for a WM task.

A: Networks were optimised to maximise the discrimination of two noisy input stimuli after a delay. B: Responses of the optimised network to each stimulus. C: Dynamics of the network at four points during optimisation: the initial network (left), the final network before feedforward dynamics emerged (middle left), the final network before rotational dynamics emerged (middle right) and the network at convergence (right). Trajectories show the mean network response for each stimulus, with solid lines up to the optimised decision time and dashed lines thereafter. Insets show the decomposition of the network into dynamical modes (Schur decomposition). D: Response distributions under each stimulus at the decision time. E: Performance of a decoder optimised for time t and tested on time t'. Dashed lines show the optimised decision time.

We first considered the performance of two classic models for WM tasks - attractor and feedforward networks (Seung, 1996; Ganguli et al., 2008; Goldman, 2009). Attractor networks generate a set of slow, non-interacting dynamical modes that can be used to load and store information over the delay (Figure 1A, Supplementary Figure S1A). We found analytically that attractor networks are highly susceptible to noise (Figure 1D top, Supplementary Figure S1B-D), with the signal-to-noise ratio for discrimination of different input stimuli decaying exponentially after stimulus offset (Supplementary Material). In contrast, feedforward networks utilise a directed chain of dynamical modes to store a stimulus in an activity sequence that spans the delay (Figure 1B, Supplementary Figure S1E-G) (Ganguli et al., 2008; Goldman, 2009). We found that, in contrast to attractor networks, feedforward networks maintain a near-optimal SNR for a period of time after stimulus removal (Figure 1D middle, Supplementary Figure S1I-K). Moreover, for feedforward networks, but not attractor networks, the SNR increases as the number of neurons is increased (Figure 1E top vs middle, Supplementary Figure S1J). However, feedforward networks require strong, amplifying feedforward connectivity and an exponential increase in firing rates to achieve these SNR improvements (Figure 1F top vs middle, Supplementary Figure S1F-H). Thus, attractor networks are energetically cheap but highly susceptible to noise, while feedforward networks are energetically costly but substantially more noise-robust.

We asked whether solutions exist which outperform these hand-crafted attractor and feedforward networks in terms of noise-robustness and energetic cost. To this end, we optimised the recurrent connectivity of networks to achieve maximal response SNR while imposing a penalty on firing rates, using a novel method that performs gradient descent on these quantities directly (see Methods). Surprisingly, the optimised networks departed significantly from classic attractor and feedforward networks. Instead, they exhibited complex oscillatory dynamics distributed over a range of frequencies (Figure 1C). Remarkably, these networks simultaneously outperformed feedforward networks in terms of SNR and attractor networks in terms of energetic cost. As in feedforward networks, the SNR increased with the number of neurons in the network. However, unlike attractor or feedforward networks, their energetic cost simultaneously decreased (Figure 1F bottom). Moreover, the SNR of the optimised networks approached the theoretical upper bound for any (potentially nonlinear) network, suggesting that the assumption of linear dynamics was not a limiting factor.

Taken together, we find that networks optimised for WM performance exhibit a novel solution that is both noise-robust and energetically efficient, substantially outperforming classic attractor and feedforward models. We next sought to understand the dynamical mechanisms employed by these optimised networks.

Efficient Working Memory Encoding via Amplifying Rotational Dynamics

To gain insight into the dynamical solutions employed by the optimised networks, we considered a minimal model of the WM task comprising a network of two neurons receiving input from one of two stimuli presented as an instantaneous pulse (Figure 2A,B). The network was driven to stationary state by Gaussian noise before stimulus onset and optimised to maximise the SNR for discrimination of the two input stimuli after a delay, with or without a penalty on energetic cost (see Methods).

Three distinct phases were observed as the network performance progressively converged toward an optimised solution (Figure 2C, Supplementary Figure S3). In the first phase of optimisation, an approximate line attractor was formed (Figure 2C second panel). During the second phase there was a transition from attractor to feedforward dynamics (Figure 2C third panel). The final phase showed a transition to rotational dynamics (Figure 2C final panel). Similar dynamics emerged in networks with and without a penalty on energetic cost (Supplementary Figure S3).

We asked whether this rotational solution was a specific consequence of our choice of loss function, optimisation procedure, and task setup, or a general feature of optimal dynamics for WM tasks. To test whether these findings were sensitive to the loss function and optimisation method, we investigated the solutions learned by backpropagation-through-time (BPTT) with a squared error loss, using an analytical solution in the limit of large batch size (see Methods). Remarkably, this method produced near-identical weight trajectories to those produced by optimisation of the SNR (Supplementary Figure S4). Moreover, we showed that optimising the SNR is mathematically equivalent to optimising the probability of correct choice, suggesting that incremental learning rules based on trial-to-trial feedback should lead to similar learning trajectories (Supplementary Material). To exclude other factors that may bias the networks towards rotational solutions, we investigated the influence of 1) the range of delays the network was optimised for 2) the initial network weights 3) the initial state covariance at the time of stimulus onset 4) the length of the stimulus period relative to the delay. Rotational dynamics emerged in all settings (Supplementary Figures S4-S7), including the limiting case where the stimulus was presented continuously until the decision time, i.e. an evidence integration task (Supplementary Figure S7).

The robust emergence of rotational dynamics suggests that these solutions are more noise-resilient than attractor or feedforward mechanisms. We therefore investigated how networks integrate noisy sensory input to form an efficient output population code. We plotted the distribution of noisy network responses to each stimulus at the optimised decision time (Figure 2D). Response SNR is high if these response distributions have minimal overlap (Averbeck et al., 2006). At all stages of optimisation, recurrent integration and amplification of sensory input generated elliptical response distributions with low and high variance dimensions (Figure 2D, 2nd-4th panels). Moreover, for attractor and feedforward networks, common amplification of signal and noise caused the stimulus-coding direction to be aligned to the principal noise direction (Figure 2D, 2nd and 3rd panels). Thus, attractor and feedforward networks had strong information-limiting correlations (Moreno-Bote et al., 2014). In contrast, alignment between signal and noise was reduced in the rotational network (Figure 2D, rightmost panel). Thus, the rotational network acted to both amplify the input signal and minimise corruption of the signal by amplified noise (Supplementary Figure S8).

To determine how the optimal decoder varied over the delay in each network, we implemented a cross-temporal decoding analysis (Figure 2E). For attractor networks, information was encoded along a stable coding direction that was independent of delay time. In contrast, the feedforward and rotational networks exhibited signatures of dynamic coding (Stokes et al., 2013) - decoding performance depended on the disparity between the times the decoder was trained and tested on. Thus, rotational dynamics achieves optimally noise-robust WM performance through a dynamic population code.

Higher-Dimensional Networks Employ Feedforward-Rotational Sequences

Having characterised the optimal WM dynamics for two-dimensional networks, we next investigated the properties of higher-dimensional networks optimised to solve the task. Classic work has shown that feedforward networks are optimal for WM tasks in discrete-time networks, and can also outperform attractor networks in continuous-time networks (Ganguli et al., 2008; Goldman, 2009; Lim and Goldman, 2012). More recently, continuous-time State Space Models (SSMs) such as the Legendre Memory Unit (LMU) have been developed, which compute an online approximation of their recent input history in terms of a set of temporal basis functions and achieve state-of-the-art performance on memory tasks (Voelker et al., 2019; Gu et al., 2020, 2023). However, the optimal solution for WM tasks has not been resolved.

Numerically optimised networks exhibited a combination of feedforward and rotational dynamics. For optimised *N-dimensi*onal networks, all eigenvalues were complex, meaning that the network contained *N/2 rotati*onal planes (Figure 3A,D, Supplementary Figure S9). Although distributed across neurons, task-relevant input targeted a single dynamical mode, which elicited a

feedforward cascade of activity through these rotational planes (Figure 3A,D). The LMU was based on a fundamentally different architecture, but exhibited rotational structure similar to the optimised network (Figure 3B, Supplementary Figure S9). Both the LMU and optimised networks departed substantially from classic feedforward models (Figure 3C). However, despite the similarity of the optimised networks to the LMU, the optimised networks substantially outperformed both feedforward networks and LMUs on the task (Figure 3E).

To understand the relative performance of these networks, we compared them to the theoretically optimal ideal observer solution (Supplementary Figure S2). This optimal solution requires the network-readout cascade to implement a delay filter, assigning a positive weight to inputs at the time of stimulus presentation and zero weight to inputs at all other times to avoid integration of pre- and post-stimulus noise. We found that only the LMU and optimised networks could successfully approximate this optimal filter, with attractor networks weighting all times equally and feedforward networks assigning positive weights to a broad range of times before and after stimulus presentation (Figure 3F). Moreover, the optimised network better approximated this ideal observer solution than the LMU, with a more sharply peaked filter around the time of stimulus presentation that enabled reduced weighting of input noise onto the readout. This sharply peaked filter was formed as a weighted combination of oscillatory modes, which provide a more flexible basis for approximating complex input-output functions than purely decaying modes.

These findings suggest that a novel dynamical mechanism combining feedforward and rotational motifs enables optimally noise-robust WM maintenance. In particular, the optimised networks compute an online approximation of their recent input history in terms of a class of oscillatory basis functions, enabling accurate approximation of a wide range of input-output functions by varying the readout weights (Voelker et al., 2019; Gu et al., 2020, 2023).

Multiple Stimuli are Encoded in Orthogonal Rotational Subspaces

So far we have considered the optimal linear dynamics for a two-stimulus WM task (Voitov and Mrsic-Flogel, 2022; Daie et al., 2023). We next investigated WM tasks with arbitrary numbers of stimuli. As in classic delayed match to sample WM tasks, we considered a circular variable $\theta \in [0, 2\pi)$, *for* example the spatial location of a stimulus in the visual field or the colour of stimulus (Goldman-Rakic, 1995; Panichello et al., 2019). We optimised a minimal four-neuron network with cosine-tuned stimulus input (Figure 4A) for maximal response SNR after the delay.

Both the stimulus inputs and network responses could be viewed within a four-dimensional state space, with the stimulus inputs constrained to a two-dimensional subspace (Figure 4B). Thus, each stimulus input could be described as a weighted sum of two orthogonal basis vectors (Figure 4B, coloured arrows). The optimised network comprised two orthogonal, non-interacting rotational subspaces (Figure 4C). Each subspace selected the input projection onto one of the two basis vectors and fed this into a private recurrent dimension via an amplifying rotational dynamics identical to that of networks optimised for two-stimulus discrimination (Figure 4D). Thus, each of the two orthogonal subspaces selectively maintained information about the magnitude of input along one of the two input dimensions, and a linear readout could reconstruct the stimulus by linearly weighting the responses within the two subspaces (see Supplementary Material).

Remarkably, this simple four-dimensional network exhibited a range of properties observed in prefrontal cortex and other brain areas during WM tasks. For example, a subset of neurons exhibited “switching selectivity”, whereby their tuning preference dynamically changed over the delay period towards their anti-preferred stimulus (Figure 4E) (Spaak et al., 2017; Libby and Buschman, 2021; Cavanagh et al., 2018). Moreover, cross-temporal pattern similarity analysis revealed dynamic coding early in the delay but stable coding throughout the remainder of the delay (Figure 4F) (Spaak et al., 2017; Stroud et al., 2024a). Finally, this dynamic code unfolded within a stable representational geometry, as estimated by the pairwise distances between population responses to different stimuli (Figure 4G)(Spaak et al., 2017).

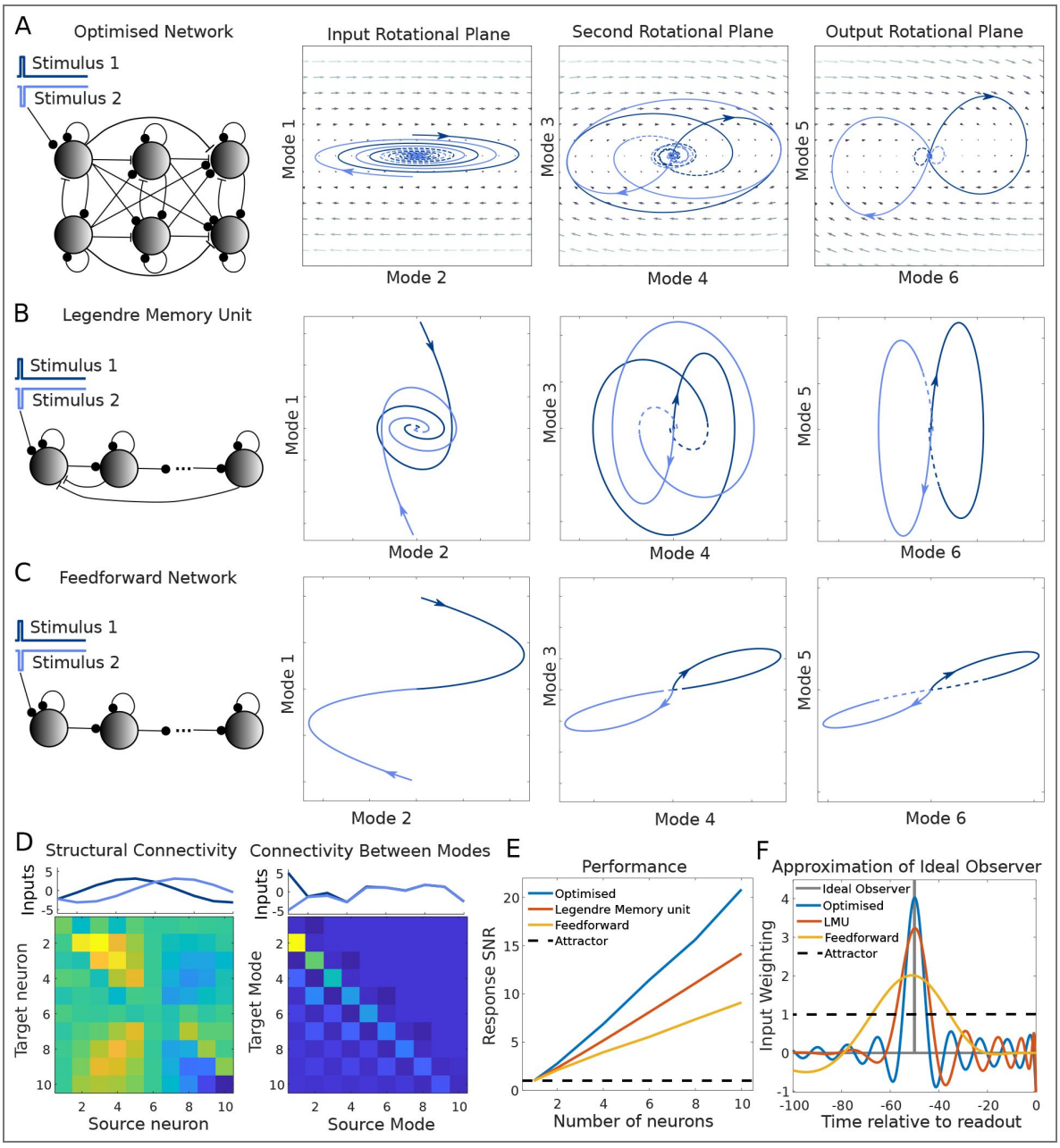

Figure 3. Numerically optimised higher-dimensional networks.

A-C: Left: Functional connectivity between dynamical modes of an optimised 6-dimensional network (A), Legendre Memory Unit (B) and feedforward network (C). In each network, the stimulus input targets a single source mode and is cascaded through the other modes. Right: The response of each mode to two input stimuli. D: Left: The inputs (top) and weight matrix (bottom) of an optimised 10-dimensional network. Right: The Schur basis of the weight matrix, which represents interactions between the dynamical modes (as in A). E: The SNR of network output at the optimised decision time, compared to that of an optimal feedforward network, a Legendre Memory Unit and an attractor network. F: Comparison of the filter applied by the network to task inputs vs that of the Bayesian ideal observer solution. The ideal observer applies a delay filter which assigns a non-zero weight only to inputs at the time of stimulus presentation while filtering out pre- and post-stimulus noise input. The LMU and and optimised networks approximate this filter using a set of oscillatory basis functions. Attractor networks can only implement constant or exponentially decaying filters.

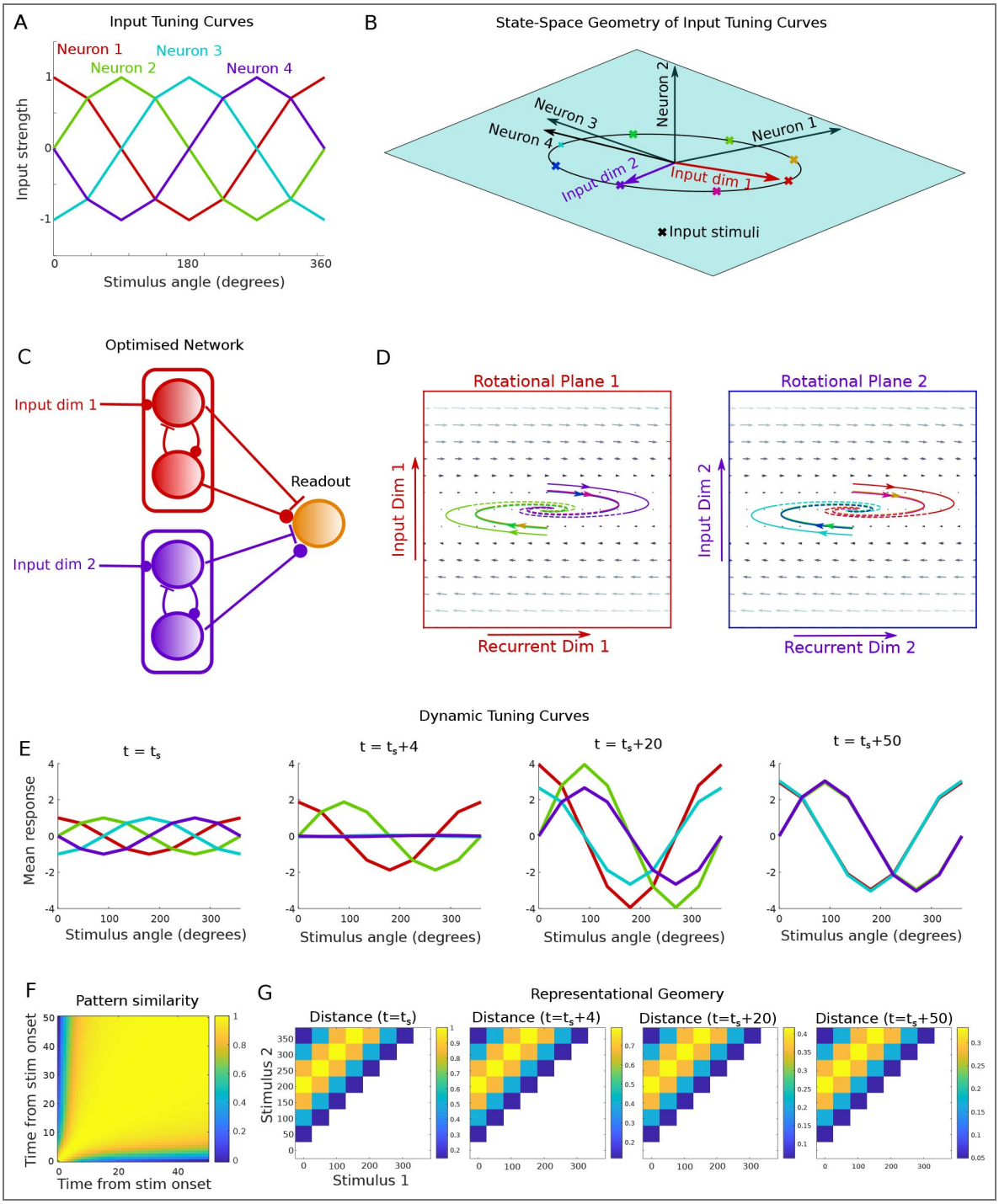

Figure 4. Working memory for multiple stimuli.

A: Tuning curves of feedforward input to 4 neurons for a set of 8 stimuli. B: The inputs are constrained to a circle within a two-dimensional plane spanned by two basis vectors. C: The optimised network comprised two orthogonal, non-interacting subspaces, each of which integrated inputs along separate basis vector. D: Each plane exhibited rotational dynamics identical to that of networks optimised to discriminate two stimuli. Note that overlapping traces were shifted slightly for visualisation purposes. E: The response tuning curves changed dynamically over the delay, with two neurons switching to their anti-preferred stimulus. F: Cross-temporal pattern similarity analysis revealed dynamic coding early in the delay and stable coding throughout the rest of the delay. G: The representional geometry of the population response remained stable throughout the delay.

Thus, a simple four-dimensional network optimised to perform a multi-stimulus WM task replicates wide range of properties of WM circuits via rotational dynamics in a set of orthogonal, non-interacting subspaces.

Prefrontal Cortex Exhibits Multi-Dimensional Rotational Dynamics in a Working Memory Task

The above findings suggest that rotational dynamics is optimal for WM tasks and accounts for features of dynamic coding found in the prefrontal cortex (PFC) and other brain regions. We next asked whether signatures of rotational dynamics could be found directly in large-scale recordings of neural circuit activity during WM tasks. To this end, we analysed neuropixel recordings from PFC of monkeys performing a classic spatial WM task (Panichello et al., 2024). On each trial of the experiment, a brief (50 ms) visual stimulus was presented at one of eight spatial locations. After a delay of 1.4-1.6 seconds, the monkeys reported the remembered stimulus via an eye movement to the corresponding location.

To uncover task-related neural dynamics in the data, we implemented a novel variant of Targeted Dimensionality Reduction (TDR) (Mante et al., 2013; Aoi et al., 2020). Specifically, we modelled activity of neuron i at time t on trial k as

$$r_i^{(k)}(t) = w_{1,i}(t) f\left(\theta^{(k)}\right) + w_{2,i}(t) g\left(\theta^{(k)}\right) + w_{0,i}(t)$$

where $\theta^{(k)}$ was the stimulus presented on trial k. The stimulus-dependent factors $f(\theta)$, $g(\theta)$ and temporal factors $w_{p,i}(t)$ were fitted directly to the observed neural responses using least squares regression (Methods). The resulting model offered a compact description of the neural population responses in terms of a set of stimulus-dependent factors shared across neurons and temporal factors shared across stimuli.

To quantify the quality of model fit, we compared the post-stimulus time histograms (PSTHs) of each neuron under the TDR model to their true PSTHs. Despite being heavily underparameterised (37.5% as many TDR parameters as PSTH data points), the TDR model accurately fit the diverse PSTHs of neurons in the data (R^2 = 0.73 ± 0.06, mean±std over 25 recording sessions, Figure 5A). Moreover, using cross-validation to predict PSTHs on held out data, we found that a TDR model with two stimulus-dependent subspaces ($f(\theta)$ and $g(\theta)$) was optimal (Figure 5B red bars), and performed substantially better than a baseline model in which the PSTHs formed on the training data were used to predict the PSTHs on the validation data (Figure 5B, grey bar). This suggested that the structure of the TDR model successfully captured shared stimulus- and time-dependent population structure in the data.

Given that the TDR model provided an excellent fit to the data, we next sought to compare the population structure it identified to that expected under competing models for WM dynamics. We considered classic attractor models (Seung, 1996), feedforward models (Ganguli et al., 2008; Goldman, 2009), and task-optimised models with rotational dynamics (Figures 1-4). A TDR model with two stimulus subspaces and no stimulus-independent subspace ($w_{0,i}(t) = 0$) yielded an exact fit to the PSTHs generated by each WM model (R^2 = 1). The stimulus factors $f(\theta)$, $g(\theta)$ were identical across all WM models considered, reflecting the circular structure of the stimulus set used in the task (Figure 5C). The temporal factors $w_{p,i}(t)$ evolved in low-dimensional subspaces captured by a small number of principal components (PCs) (Supplementary Figure S10B left). TDR applied to neural data revealed a similar low-dimensional encoding (Figure 5D, Supplementary Figure S10B right), albeit with an additional stimulus-independent subspace $w_{0,i}(t)$ not present in classic WM models (Machens et al., 2010; Cueva et al., 2020).

To investigate the structure of population activity in the low-dimensional stimulus-dependent subspaces, we plotted the top PCs of the temporal factors $w_{p,i}(t)$, which can be understood as low-dimensional "population PSTHs" (Figure 5E). The neural data exhibited highly rotational trajectories, which closely matched those of the rotational model and departed substantially from the attractor and feedforward models (Figure 5F). To quantify dynamic coding across models

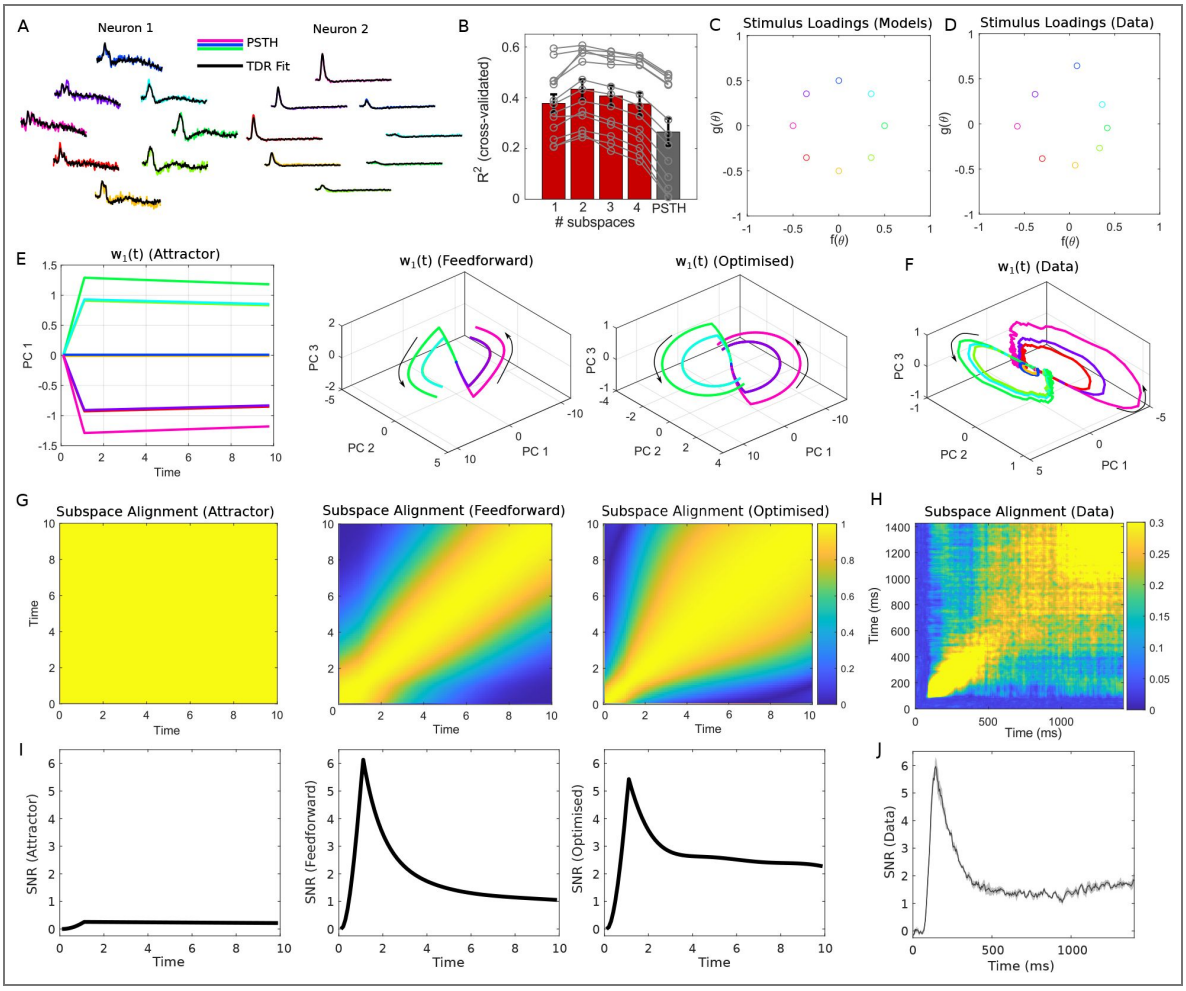

Figure 5. Rotational dynamics in monkey PFC during a spatial WM task.

A: PSTHs of two neurons for each of the 8 stimuli in the task (coloured lines). The TDR fit is shown in black. B: Cross-validation performance of the TDR model with different numbers of stimulus components (red bars). A baseline model in which the PSTH formed on the training data is used to predict the PSTH on the test data is shown for comparison (grey bar). Error bars show mean ± SEM over 12 sessions (sessions were excluded if the R^2 for the baseline model was negative). Grey traces show individual sessions (note that the optimal number of subspaces was 2 for all included sessions). C: The stimulus factors $f(\theta)$, $g(\theta)$ learned from a TDR fit to simulated data under the models. D: The stimulus-loadings of a TDR fit to experimental data (a single recording session with 586 neurons). E: The population PSTHs (principal components of $\mathbf{w}_1(t)$) for each of the models. Note that the population PSTHs of $\mathbf{w}_2(t)$ were identical to those of $\mathbf{w}_1(t)$ for all models. F. As in E, but for experimental data. The second subspace $\mathbf{w}_2(t)$ is shown in Supplementary Figure S10C. G: The alignment between the subspaces spanned by $\mathbf{w}_1(t)$, $\mathbf{w}_2(t)$ and $\mathbf{w}_1(t')$, $\mathbf{w}_2(t')$ for each of the models. H: Left: As in G, but for experimental data. I: The SNR of responses of the three models (computed on antipodal stimulus pairs). SNRs were computed assuming stationary state covariance, in contrast to Figure 1 which assumed a zero variance initial state. J: The cross-validated SNR of experimentally-recorded responses projected onto the TDR Principal Component subspace (mean±SEM over 4 antipodal stimulus pairs).

and data, we computed the cross-temporal alignment between the planes spanned by $\mathbf{w}_1(t)$, $\mathbf{w}_2(t)$ and $\mathbf{w}_1(t')$, $\mathbf{w}_2(t')$ (Figure 5G,H, Supplementary Figure S10D,E). When applied to the optimised rotational model and experi-mental data, this revealed classic signatures of dynamic coding, with the coding subspace at stimulus time and early delay period being orthogonal to that of the late delay. Attractor and feedforward models failed to replicate this dynamic code. While a recently proposed hybrid feedforward-attractor model accounted for the cross-temporal subspace alignment (Stroud et al., 2023), it exhibited lower-dimensional dynamics and population PSTHs that did not match those of the data (Supplementary Figure S10B-E).

Finally, we computed the response SNR for discrimination of antipodal stimulus pairs at each time point (Figure 5I,J). When computing the SNR we assumed that that the networks were driven to stationary state by noise input before stimulus onset. As a consequence, the SNR of an attractor network was low at all time points due to excessive integration of pre-stimulus noise along the slow attractor mode. In constrast, feedforward and optimised networks achieved substantially higher SNR, with an initial decrease after stimulus onset followed by a stable plateau (Figure 5I). Computing the cross-validated SNR of experimentally measured responses, projected into the six-dimensional TDR PC subspace, revealed a remarkably similar time course to that of the optimised model (Figure 5J).

Taken together, only the optimised model could account for the population PSTHs, dynamic coding properties, and SNR dynamics of the experimental data. Thus, neural activity in PFC during WM tasks is inconsistent with previous models involving attractor or feedforward dynamics. In contrast, rotational dynamics, emerging from directly optimised models, accurately accounts for the joint stimulus- and time-dependent neural population structure observed of PFC circuits during WM tasks.

Discussion

We developed a novel method for optimisation of continuous-time RNNs driven by noisy inputs to solve cognitive tasks and leveraged this framework to find the optimally noise-robust and energetically efficient dynamics for working memory (WM). The solutions we identified combine functionally-feedforward and rotational dynamics, and substantially outperform all previously proposed linear models of WM (Seung, 1996; Ganguli et al., 2008; Stroud et al., 2023; Voelker et al., 2019). Moreover, these solutions provide a parsimonious explanation of a wide range of functional properties of WM circuits, including the widely observed phenomenon of dynamic coding (Stokes et al., 2013; Stokes, 2015; Spaak et al., 2017; Stroud et al., 2024a; Spaak et al., 2017; Cavanagh et al., 2018; Libby and Buschman, 2021). Thus, we suggest that WM circuits are adapted for efficient computation.

Although our theoretical analysis was restricted to linear networks, several lines of argument suggest that our conclusions apply to nonlinear networks. First, the nonlinear dynamics of PFC during simple tasks has been shown to be well approximated by a linear dynamical system (Machens et al., 2010; Soldado-Magraner et al., 2024), and the dynamical solutions of task-trained nonlinear RNNs in WM and other tasks can often be qualitatively captured by linear approximations (Mante et al., 2013; Stroud et al., 2023). Second, we derived a non-dynamical Bayesian ideal observer analysis that places strong constraints on the optimal dynamics of any (linear or nonlinear) dynamical solution to the WM task, and showed that optimal linear networks achieve this upper bound. Third, we analysed large-scale recordings from monkey PFC during a WM task using a non-dynamical TDR analysis, and showed that the population structure was consistent with optimal rotational dynamics but not classic attractor or feedforward models (Seung, 1996; Ganguli et al., 2008; Goldman, 2009) or recently proposed hybrid feedforward-attractor models (Stroud et al., 2023).

Although rotational dynamics have not previously been considered for WM tasks, they have been observed across a wide range of brain areas and tasks, including in PFC in contextual decision-making tasks (Aoi et al., 2020) and in auditory cortex in implicit short-term memory tasks (Libby and Buschman, 2021). Moreover, high-dimsensional dynamics distributed across multiple brain regions have been observed in WM tasks (Voitov and Mrsic-Flogel, 2022; Daie et al., 2023).

Indeed, although we analysed experimental data collected from to PFC, we speculate that the theoretically optimal rotational dynamics we identify may be distributed across a wide network of regions of the brain that participate in the task (Christophel et al., 2017; Voitov and Mrsic-Flogel, 2022).

While, to our knowledge, feedforward-rotational dynamics have not been previously considered in the neuroscience literature, their properties can be understood in light of State Space Models (SSMs) recently developed for machine learning applications. SSMs such as the Legendre Memory Unit, MAMBA and HiPPO are built upon linear dynamical systems which are mathematically optimised to compute an online approximation of their recent input history in terms of a set of orthogonal polynomials (Voelker et al., 2019; Gu et al., 2020, 2023). We found that networks optimised for WM tasks exhibit qualitatively similar properties to these SSMs, but substanitally outperform them in the WM tasks we consider. This suggests that WM circuits may compute an online approximation of their input history in terms of an orthogonal basis of oscillatory functions. Given that SSMs now achieve state-of-the-art performance on a plethora of time-series tasks, the high-dimensional rotational solutions implemented by WM circuits may be suitable for a wide range of complex tasks beyond simple delayed recall paradigms typically studied experimentally. In particular, arbitrary functions of an SSM's recent input history can be reconstructed through simple linear readouts, providing a flexible substrate for memory-based computation (Gu et al., 2023). This use of flexible high-dimensional representations mirrors those used by PFC for complex cognitive tasks (Fusi et al., 2016), but extends them to the time domain. Thus, we suggest that PFC dynamically computes a high-dimensional, optimally compressed representation of its input history for general-purpose computation on temporally distributed data.

Methods

Task Setup

We considered continuous-time recurrent linear networks driven by noisy inputs:

$$\frac{dx_i}{dt} = \sum_{j=1}^{N} A_{ij}x_j + u_i(s,t)$$

The input comprised a stimulus at time $t = 0$ and ongoing Gaussian noise $u_i(s, t) = \tilde{u}(t)u_i(s) + n_i(t)$ where $n_i(t) \sim N(0, \sigma_n^2)$. The stimulus timecourse was either a delta pulse $\tilde{u}(t) = \delta(t)$ (for Figures 2-4, S1-S6, S8) or a boxcar function $\tilde{u}(t) = \Theta(t) - \Theta(t - T_{cue})$ where $\Theta(t)$ is the Heaviside step function and T_{cue} is the duration of stimulus presentation (for Figures 1, 5, S2, S7, S10).

For all networks, the input noise was set to $\sigma_n = 1$. The inputs were modelled as $u_i(s_j) = u_0 \cos(\hat{s}_i - s_j)$, where the stimuli were $s_j \in \{0, \pi/2\}$ for two-stimulus tasks and $s_j = 2\pi(j-1)/N_{stim}$ for multi-stimulus tasks, and the stimulus preferences were $\hat{s}_i \in \{0, \pi/2\}$ or $\hat{s}_i = 2\pi(j-1)/N$. We set $u_0 = \sigma_n / \|\mathbf{u}(s_1) - \mathbf{u}(s_2)\|$ for two-stimulus tasks and $u_0 = \sigma_n / \|\mathbf{u}(s_1) - \mathbf{u}(s_{N/2})\|$ for multi-stimulus tasks in order to ensure that the input signal-to-noise ratio was normalised to 1. For multi-stimulus tasks, the inputs could be decomposed into basis vectors $\mathbf{u}(s) = c_1(s)\mathbf{b}_1 + c_2(s)\mathbf{b}_2$, where $\mathbf{b}_1 = \cos \hat{\mathbf{s}}$, $\mathbf{b}_2 = \sin \hat{\mathbf{s}}$ spanned a two-dimensional input space embedded in N-dimensional state space.

We analysed the SNR of an optimal linear readout of the stimulus, given by $\mathrm{SNR}(s_i, s_j, t) = (\overline{\mathbf{x}}(s_i,t) - \overline{\mathbf{x}}(s_j,t))^\top \Sigma_{\mathbf{x}}^{-1} (\overline{\mathbf{x}}(s_i,t) - \overline{\mathbf{x}}(s_j,t))$ where $\overline{\mathbf{x}}(s,t)$ is the mean response (over trials) of the network to stimulus s at time t and $\Sigma_{\mathbf{x}}$ is the covariance of $\mathbf{x}(s, t)$ (which was independent of s). Note that, for the linear-Gaussian networks we considered, this linear readout is a Bayes optimal decoder of the stimulus from $\mathbf{x}(s, t)$ and the SNR is monotonically related to the probability of correct choice as $\mathrm{p(correct)} = \Phi(\sqrt{\mathrm{SNR}}/2)$, where Φ is the cumulative function of the standard normal distribution. Moreover, for continuous estimation tasks (rather than discrete classification tasks) the SNR is closely related to the Fisher information for estimation of the stimulus s from the response $\mathbf{x}$. In such tasks, the SNR places a lower bound on the mean squared decoding error.

The SNR was computed directly, without sampling noisy trajectories of the system, using a Lyapunov equation to compute the covariance matrix and matrix exponentials to compute the mean response to each stimulus (see Supplementary Material). We also analysed the response energy of the network, defined as $E = \sum_s \int_0^\infty \|\overline{\mathbf{x}}(s,t)\|^2 dt$, which can be computed directly in a similar manner to the SNR.

Attractor and Feedforward Network Models

Hand-crafted networks were constructed via their Schur form $A = QTQ^\top$ where Q is a unitary matrix and T is a lower triangular matrix. For two-stimulus tasks, Q was constructed such that $Q_{:,1} = \mathbf{b}$ and all other columns of Q were orthogonal, with $\mathbf{b} = (\mathbf{u}(s_2) - \mathbf{u}(s_1))/\|\mathbf{u}(s_2) - \mathbf{u}(s_1)\|$ the input linear discriminant. For multi-stimulus tasks, Q contained the two orthogonal basis vectors spanning the input space $\mathbf{b}_1, \mathbf{b}_2$. Specifically, we set $Q_{:,1} = \mathbf{b}_1$ and $Q_{:,N/2+1} = \mathbf{b}_2$, with the remaining columns orthogonal. In this case, T was divided into four equal-sized blocks, $T_{\mathrm{MultiStim}} = [T, 0; 0, T]$, with T equal to that of the two-stimulus model (described for each network below). Responses of networks were often plotted in the Schur basis $\mathbf{x}_{\mathrm{Schur}}(t) = Q^\top \mathbf{x}(t)$.

For attractor networks, T was diagonal, with $T_{ii} = \lambda_i = -1/\tau_i$ where λ_i are the eigenvalues and τ_i are the decay time constants of the dynamical modes. The network had a slow dynamical mode $\tau_1 = \tau_{\mathrm{slow}}$ and fast dynamical modes $\tau_{i>1} = \tau_{\mathrm{fast}}$, which with the above choice of Q ensured that the input linear discriminant was loaded onto the slowest dynamical mode (Chadwick et al., 2023).

Feedforward networks were constructed as $T_{ij} = \tau^{-1}\delta_{ij} + \omega\delta_{i,j+1}$, where τ is the time constant of each mode and ω is the feedforward weight from mode $i - 1$ to mode i. This ensured that the input linear discriminant was loaded onto the source mode of T.

The hybrid feedforward-attractor model in Supplementary Figure S10 differed from the feedforward model in two ways. First, we included only one feedforward connection. Second, this feedforward connection fed from a rapidly decaying mode into a slow attractor mode (Stroud et al., 2023). In particular, we set $T_{22} = -1/\tau_{\mathrm{slow}}$, with $T_{ii} = -1/\tau_{\mathrm{fast}}$ for $i \neq 2$. The feedforward connection was set as $T_{21} = \omega$.

For Figure 1, we considered a two-stimulus task. Networks had a fixed initial state $\mathbf{x}(s_i, 0) = 0$ (i.e., with zero initial state covariance) and a stimulus was presented for $T_{\mathrm{cue}} = 1$ (in arbitrary units of time). The SNR for stimulus discrimination was calculated at the decision time $t_d = 10$. Attractor networks had $\tau_{\mathrm{slow}} = 10^4$. Feedforward networks had $\omega = 5$ and τ selected to maximise SNR.

For Figure 3, we considered a two-stimulus task in which networks were driven to stationary state by noise input before stimulus onset and received a delta pulse input, with $t_d = 50$. The SNR of attractor and feedforward networks was computed analytically using their optimal parameters (see Supplementary Material). However, because the optimal feedforward network has infinitely strong feedforward weights, we instead plotted the trajectories of this network with $\tau = 6.25$, $\omega = 0.5$ (where τ was selected as the optimal value in the limit $\omega \to \infty$).

For Figures 5 and S10, we considered an eight-stimulus task, with $T_{\mathrm{cue}} = 1$ and $t_d = 10$. Attractor networks had $N = 6$ neurons, with slow time constants $\tau_{\mathrm{slow}} = 100$ and fast time constants $\tau_{\mathrm{fast}} = 1$. Feedforward networks had $N = 12$ neurons with $\tau = 5$ and $\omega = 0.5$. Hybrid feedforward-attractor models had $N = 6$ neurons with $\tau_{\mathrm{fast}} = 1$, $\tau_{\mathrm{slow}} = 100$ and $\omega = 1$. Note that the results we present are independent of N for attractor and hybrid feedforward-attractor models (if $N \geq 2$ and $N \geq 4$ respectively), but not feedforward models.

Optimised Networks

We minimised the following loss function:

$$\mathscr{L} = \int_0^{t_d} l(t)L(t)dt + \beta E$$

where the first term quantifies the decoder performance $L(t) = \Sigma_{i>j}$ 1/SNR(s_i, s_j, t) weighted over delays by $l(t)$ and the second term quantifies the energy of responses weighted by the regularisation parameter β. The weighting function was set as $l(t) = \delta(t - t_d)$ (where t_d is the decision time), with the exception of Supplementary Figure S4 where we set $l(t) = 1$ for $0 \leq t \leq t_d$ and 0 otherwise. We set $\beta = 10^{-6}$ for Figures 1 and S3 and $\beta = 0$ otherwise. For delta input tasks we set $t_d = 50$, and for finite cue tasks, we set $T_{\text{cue}} = 1$, $t_d = 10$ (with the exception of Supplementary Figure S7, where T_{cue} was systematically varied).

We computed the gradient $\frac{d\mathscr{L}}{dA}$ analytically and performed gradient descent $L \to L - \eta\frac{d\mathscr{L}}{dA}$ numerically with a fixed step size η that was set manually for each network. Network weights were initialised as $A_{ij} = \delta_{ij}0.05(1 - 0.005i)$, except for Supplementary Figure S5 where A was intialised randomly using the eigendecomposition $A = V\,DV^{-1}$ with $V = [\cos\theta_1, \cos\theta_2; \sin\theta_1, \sin\theta_2]$ with $\theta_i \sim \text{Unif}[-\pi, \pi]$ and $D = [\lambda_1, 0; 0, \lambda_2]$ with $\lambda_i \sim \text{Unif}[-0.2, 0]$, i.e. with random (real) eigenvectors and eigenvalues.

To transform the optimised networks from neuron space to the space of dynamical modes, we performed a real Schur decomposition on the weight matrix $A_{\text{Schur}} = Q^{\mathsf{T}}AQ$, $\mathbf{u}_{\text{Schur}} = Q^{\mathsf{T}}\mathbf{u}$ where Q is an orthogonal matrix and A_{Schur} is a block upper-triangular matrix. Given that the Schur decomposition is non-unique, we exploited a set of invariances to transform the Schur decomposition into a standardised form (see Supplementary Material for details).

Analysis of Learning Dynamics

We selected networks at four stages of optimisation to plot the flow fields, Schur modes, and responses:

1) the initial unoptimised network 2) the initial attractor network, before non-normality had increased 3) the final network before eigenvalues became complex 4) the network at convergence.

For each iteration of optimisation, we computed the SNR (defined above) and the projection of the mean and covariance of responses onto the unit length readout vector $\mathbf{w} \propto \Sigma_{\mathbf{x}}^{-1}(\overline{\mathbf{x}}(s_2, t) - \overline{\mathbf{x}}(s_1, t))$, given by $\mathbf{w}^{\top}(\overline{\mathbf{x}}(s_2, t) - \overline{\mathbf{x}}(s_1, t))$ and $\mathbf{w}^{\mathsf{T}}\Sigma_{\mathbf{x}}\mathbf{w}$ respectively (e.g., Supplementary Figure S3B,C).

The decay time constants and rotational periods of dynamical modes were computed as as $\tau = -1/\mathscr{R}(\lambda)$ and $\nu = 2\pi/\mathscr{I}(\lambda)$ respectively, where $\lambda_i = \mathscr{R}(\lambda) \pm i\mathscr{I}(\lambda)$ were the eigenvalues of A. The alignment of modes with the input discriminant $\mathbf{b}$ were computed using either eigendecomposition or Schur decomposition. For two-neuron networks with real eigenvalues, we computed the dot product between each unit length left eigenvector of A (defined as $A^{\mathsf{T}}\mathbf{v}_i = \lambda\mathbf{v}_i$) and the input linear discriminant $\mathbf{b}$. For higher-dimensional networks or networks with complex eigenvalues, we computed the dot product between the Schur vectors (columns of Q) and $\mathbf{b}$. The non-normality of the dynamics was computed as $H = \sqrt{\|A\|_F^2 - \sum_i |\lambda_i|^2}/N$, where $0 \leq H \leq 1$ and $H = 0$ for a normal (attractor) network.

To optimise networks via backpropagation through time, we used analytically derived gradients in the limit of infinite batch size (Supplementary Figure S4, see Supplementary Material for derivation). We used a squared error loss $\mathscr{L} = \Sigma_s\langle\|\mathbf{w} \cdot \mathbf{x}(s, t) + c - s\|^2\rangle$ (where the expecation was taken over trials) and optimised A using gradient descent $A \to A - \eta\frac{d\mathscr{L}}{dA}$ with a fixed step size η. The readout parameters $\mathbf{w}$, c were obtained analytically at each iteration by setting $\frac{d\mathscr{L}}{d\mathbf{w}} = 0$ and $\frac{d\mathscr{L}}{dc} = 0$.

Analysis of Dynamic Coding

In Figure 2D, we plotted ellipses of one standard deviation of the response distribution $\mathbf{x}(s,t_d) \sim N(\overline{\mathbf{x}}(s,t_d), \Sigma_\mathbf{x})$ for networks at each of the four stages of optimisation we considered.

For Figure 2E, we performed a cross-temporal decoding analysis on networks at different stages of optimisation using standard signal detection theory measures. Labelling $s_1 = 0$, $s_2 = 1$, the optimal decoder uses a binarised linear readout $\hat{s}(c, \mathbf{w}, t) = \Theta(\mathbf{w}(t)^\top \mathbf{x}(s,t) + c(t))$, with $\mathbf{w}(t) = \Sigma_\mathbf{x}^{-1}(\overline{\mathbf{x}}(s_2,t) - \overline{\mathbf{x}}(s_1,t))$ and $c(t) = -\frac{1}{2}\mathbf{w}(t)(\overline{\mathbf{x}}(s_1,t) + \overline{\mathbf{x}}(s_2,t))$. Using an arbitrary (suboptimal) $\mathbf{w}$ and c results in the probability of correct choice for each stimulus $p(\text{correct} \mid s_1) = \Phi\left(-\frac{\mathbf{w}^\top\overline{\mathbf{x}}(s_1,t)+c}{\sqrt{\mathbf{w}^\top\Sigma_\mathbf{x}\mathbf{w}}}\right)$ and $p(\text{correct} \mid s_2) = 1 - \Phi\left(-\frac{\mathbf{w}^\top\overline{\mathbf{x}}(s_2,t)+c}{\sqrt{\mathbf{w}^\top\Sigma_\mathbf{x}\mathbf{w}}}\right)$ where Φ is the cumulative distribution function of the standard normal distribution. Assuming $p(s_1) = p(s_2)$, the overall probability of correct choice is then $\mid s_2))$. Using these formulae, we computed the probability of correct choice using the optimal decoder for time t on responses at time t'.

For Supplementary Figure S8, the signal, noise, and signal-noise-alignment were computed for networks at each stage of optimisation as signal $= \|\overline{\mathbf{x}}(s_2,t) - \overline{\mathbf{x}}(s_1,t)\|^2$, noise = Trace($\Sigma_\mathbf{x}$), $= (\overline{\mathbf{x}}(s_2,t) - \overline{\mathbf{x}}(s_1,t))^\top \Sigma_\mathbf{x} (\overline{\mathbf{x}}(s_2,t) - \overline{\mathbf{x}}(s_1,t))$ / signal. The SNR was computed for optimised networks numerically (using a Lyapunov equation to compute the covariance) and the optimal SNRs of the attractor, feedforward and rotational networks were plotted using analytically-derived expressions (see Supplementary Material).

To replicate the analyses performed on experimental data by (Spaak et al., 2017), we computed the cross-temporal pattern similarity $S(t,t') = \frac{\overline{\mathbf{x}}(s,t)\cdot\overline{\mathbf{x}}(s,t')}{\|\overline{\mathbf{x}}(s,t)\|\|\overline{\mathbf{x}}(s,t')\|}$ (the result was independent of the stimulus s for the models we considered). In addition, we computed the representational dissimilarity matrices R_{ij} = SNR(s_i, s_j, t) (Kriegeskorte and Wei, 2021).

Legendre Memory Unit

The Legendre Memory Unit is governed by the equations

$$\tau \frac{dx_i}{dt} = \sum_{j=1}^{N} A_{ij} x_j + b_i u(t)$$
$$y(t) = \sum_{i=1}^{N} w_i x_i(t),$$

where $u(t)$ is a scalar input time series, τ is a global network time constant, A_{ij} are the recurrent network weights, b_i is an input loading vector and w_i are the readout weights (Voelker et al., 2019; Voelker, 2019). Classic results from control theory provide methods to optimise the parameters of the above system such that the output $y(t)$ is related to the input $u(t)$ by a desired transfer function f, i.e. $y \approx (u \star f)(t)$ where $\star$ is convolution (Partington, 2004; Brockett, 2015). Voelker (2019) considered the case of a delay transfer function $y(t) \approx (u \star \delta_{-t'})(t) = u(t - t')$, which we show in the Supplementary Material is a critical component of the Bayesian ideal observer solution to the workingmemory task we consider (Supplementary Figure S2). In approximating the delay transfer function using the above SSM, it was further shown that the solution can be understood through the lens of function approximation theory - each element of the network state $x_i(t)$ in the optimised system computes the coefficient of a Legendre polynomial approximation to the input time series $u(t')$ over an interval $t - \tau \le t' \le t$, and subsequent work has shown how this approach can be extended to a wide range of approximating function classes (Gu et al., 2020, 2023). As a consequence, it is possible to reconstruct arbitrary functions of the input time series over this interval through an appropriate choice of readout weights w_i, as described below.

The parameters of the LMU are found by forming a Padé approximation of the desired transfer function in the Laplace domain (Partington, 2004; Brockett, 2015; Voelker et al., 2019). The resulting system can be written in multiple equivalent bases, all of which implement an identical transfer function. In the standard Legendre basis of the LMU, the recurrent weight matrix is given by:

$$A_{ij} = 2i\begin{cases}-1 & i < j \\ (-1)^{(i-j+1)} & i \geq j\end{cases}$$

and the input vector is given by $b_i = (2i)(-1)^{i-1}$. In this basis, the activations $x_i(t)$ represent the delayed inputs as $u\left(t-t^{'}\right) \approx \sum_{i=1}^{N} \mathscr{P}_{i-1}\left(\frac{t^{'}}{\tau}\right) x_i(t) = \mathbf{p}\left(t^{'}\right) \cdot \mathbf{x}(t)$ where $\mathscr{P}_i\left(\frac{t^{'}}{\tau}\right)$ is the ith shifted Legendre polynomial defined on the interval $0 \leq \frac{t^{'}}{\tau} \leq 1$ (Voelker et al., 2019). As a consequence, an arbitrary filtering of the input time series can be approximately recovered by setting the readout weights as $\int_0^\tau f\left(t^{'}\right) u\left(t-t^{'}\right) dt^{'} \approx \int_0^\tau f\left(t^{'}\right) \mathbf{p}\left(t^{'}\right) \cdot \mathbf{x}(t) dt^{'} = \mathbf{w} \cdot \mathbf{x}(t)$ where $\mathbf{w} = \int_0^\tau f\left(t^{'}\right) \mathbf{p}\left(t^{'}\right) dt^{'}$.

The controllable canonical form provides an alternative representation of the LMU, where the weight matrix is given by $A_{i1} = -\nu_1$, $A_{i,i-1} = \nu_i$, with $A_{ij} = 0$ otherwise, and the inputs are given by $b_i = [\nu_1, 0, \ldots, 0]^\intercal$ (where $\nu_i = (N + i - 1)(N - i + 1)/i$ for $i = 1 \ldots N$). In this basis, the inputs target only the first neuron of the network, consistent with the Schur basis for the optimised and hand-crafted networks. We therefore presented results in controllable canonical form in the main text, and compared to results in the Legendre basis in the Supplementary Material. However, we note that the two forms of the LMU are fully equivalent, with identical eigenvalues and SNR.

We simulated responses of the LMU for varying numbers of units N and varying time constant τ. Inputs comprised a delta pulse with ongoing Gaussian noise $u(s, t) = \delta(t)u(s) + n(t)$. We set $n(t) \sim N(0, 1)$ and $u(s) = \pm u_0$, with u_0 chosen to normalise the input SNR to 1. We computed the mean and stationary state covariance of responses $\overline{\mathbf{x}}(s,t) = e^{At}\mathbf{b}u(s), \Sigma_{\mathbf{x}} = \mathrm{lyap}\left(A, \mathbf{b}\mathbf{b}^\top\right)$ and the SNR $= \left(\overline{\mathbf{x}}\left(s_2,t\right) - \overline{\mathbf{x}}\left(s_1,t\right)\right)^\top \Sigma_{\mathbf{x}}^{-1} \left(\overline{\mathbf{x}}\left(s_2,t\right) - \overline{\mathbf{x}}\left(s_1,t\right)\right)$. For each N, we chose the value of τ that maximised this SNR. We note that the task faced by the LMU is simpler than that faced by the other models we considered; for the LMU, input noise targets only the first unit, since $\mathbf{b} \propto [1, 0, \ldots, 0]^\intercal$. Indeed, when simulating LMUs with noise targetting all units (as was the case for all other networks), performance was substantially degraded and decreased with N (not shown).

Analysis of Electrophysiological Data

We analysed publicly available data from (Panichello et al., 2024). Stimuli were presented for 50 ms, followed by a delay of 1400-1600 ms after which monkeys were cued to report the remembered stimulus. We analysed data during the period from stimulus onset (0 ms) to the end of the shortest delay (1400 ms). As preprocessing steps, the raw spiking data were converted into spike counts within 25 ms bins using a sliding window of step size 5 ms. We discarded error trials and subtracted the mean spike count of each neuron (averaged over both time bins and trials). All analyses were applied to individual experimental sessions and repeated over multiple sessions to verify consistency. For Figure 5F,H,J, we used session *22-10-21*. For Supplementary Figure S10, we used sessions *22-10-24*, *22-10-21* and *22-10-19*.

We fitted the targeted dimensionality reduction (TDR) model

$$r_i^{(k)}(t) = \sum_{p=0}^{P} w_{p,i}(t) f_p\left(\theta^{(k)}\right)$$

to the mean-subtracted spike counts $r_i^{(k)}(t)$ using alternating least squares, iteratively solving for $w_{p,I}$ and f_p until convergence. We set $f_0(\theta) = 1$ for the stimulus-independent subspace. In this notation, $f_1(\theta) = f(\theta)$, $f_2(\theta) = g(\theta)$, $f_0(\theta) = 1$ recovers the $P = 2$ model described in the main text. Given the 8 stimuli in the experimental task, the $f_p(\theta)$ parameters were given by 8-

dimensional vectors $f_p = [f_p(\theta_1), ..., f_p(\theta_8)]^\intercal$ that were inferred directly from the data. We initialised $f_p(\theta) \sim N(0, 1)$ (for $p > 0$) and began the optimisation algorithm by updating $w_{p,i}(t)$. For $N_{\text{stim}} = 8$ stimuli, T time points and N neurons in a given experimental session, the model had $(P + 1)NT + PN_{\text{stim}}$ parameters, with NT parameters for each of the $w_{p,i}(t)$ components and N_{stim} parameters for each of the f_p components. In contrast, the PSTHs of the data contained $N_{\text{stim}}NT$ datapoints. This reduction in parameters reflects the shared structure of the TDR model across stimuli, which incorporates the assumption that the population PSTHs for different stimuli should have a similar temporal profile.

For cross-validation, we split the data into two subsets of randomly selected trials of equal size. The TDR model was fit to each split and used to predict the remaining split. We reported the cross-validated coefficient of determination

$$R^2 = 1 - \frac{\sum_{t=1,i=1,\theta=1}^{t=T,i=N,\theta=N_{\text{Stim}}} \left(\hat{r}_i(\theta,t) - \bar{r}_i(\theta,t)\right)^2}{\sum_{t=1,i=1,\theta=1}^{t=T,i=N,\theta=N_{\text{Stim}}} \left(\bar{r}_i(\theta,t) - \bar{r}_{\text{total}}\right)^2}$$

where $\hat{r}_i(\theta,t) = \sum_{p=0}^{P} w_{p,i}(t) f_p(\theta)$ is the predicted PSTH formed by the TDR fit to the train-ing data, $\bar{r}_i(\theta,t) = \frac{1}{N_{\text{TestTrials}}(\theta)} \sum_{k \in \text{TestTrials}(\theta)} r_i^{(k)}(t)$ is the PSTH formed on the test data and $\bar{r}_{\text{total}} = \frac{1}{TNN_{\text{Stim}}} \sum_{t=1,i=1,\theta=1}^{t=T,i=N,\theta=N_{\text{Stim}}} \bar{r}_i(\theta,t)$. Given that the TDR model was underparameterised, we also reported this measure without cross-validation (where $\hat{r}, \bar{r}, \bar{r}_{\text{total}}$ were formed on the full dataset). We repeated this cross-validation for 25 random two-way splits of the data (with random TDR parameter initialisations) and averaged R^2 both over splits within each recording session (grey traces in Figure 5B) and over recording sessions (red bars in Figure 5B). We compared this to a PSTH only model in which the predicted PSTH was $\hat{r}_i(\theta,t) = \frac{1}{N_{\text{TrainTrials}}(\theta)} \sum_{k \in \text{TrainTrials}(\theta)} r_i^{(k)}(t)$.Note that a TDR model with $P = N_{\text{stim}}$ and $w_{0,i}(t) = 0$ is equivalent to this PSTH only model.

For application of TDR to classic WM models, we simulated the PSTHs (without noise) of each model under each stimulus and fit a TDR model to this set of PSTHs. We observed that a TDR model with no time component $w_{0,i}$ was sufficient to fit these simulated PSTHs exactly, and therefore omitted this term.

Having fit the TDR model to data, we computed the Principal Component (PC) subspaces by taking the singular value decomposition of the matrix $(W_p)_{i,t} = w_{p,i}(t)$. Writing $W_p = USV^\intercal$, where S is a diagonal matrix containing the singular values in descending order, the columns of U contained the N-dimensional basis vectors in neural state space and the rows of $SV^\intercal$ contained the population PSTHs. Before computing these subspaces, we first exploited several invariances of the model to transform the fitted parameters $w_{p,i}(t)$ and $f_p(\theta)$ into a standard form (Supplementary Material, cf Aoi et al. (2020)). We plotted these population PSTHs within the top three PCs of W_1 and W_2, and plotted the singular value spectra (diagonal of S) to assess the dimensionality of each subspace $p \in \{0, 1, 2\}$.

Subspace alignments were computed using random two-way splits over trials, as for the cross-validation analysis. Given the vectors $\mathbf{w}_p^{(s)}(t) = \left(W_p^{(s)}\right)_{:,t}$ obtained from the TDR model fit to split s, we computed the subspace angle between the plane spanned by $\mathbf{w}_1^{(s)}(t), \mathbf{w}_2^{(s)}(t)$ and the plane spanned by $\mathbf{w}_1^{(s')}(t'), \mathbf{w}_2^{(s')}(t')$ using the Matlab function *subspace*. We plotted the cosine of this subspace angle, averaged over 25 random two-way splits of the data, as a function of t and t'. The cosine of the angle between $\mathbf{w}_1^{(s)}(t)$ and $\mathbf{w}_2^{(s')}(t')$ was computed in a similar fashion.

To compute the SNR of the data in the TDR Principal Component Subspace, we used a similar two-fold cross-validation split as above. First, a TDR model was fitted to the full dataset (without cross-validation). Given this fit, we formed the matrix $W_{\text{concat}} = [W_1, W_2]$ to obtain an $N \times 2T$ matrix describing the set of neural activity patterns generated over stimuli and time points. We per-

formed a singular value decomposition $W_{\text{concat}} = USV^{\top}$ and projected the binned spiking data onto the top six PCs as $\mathbf{x}^{(k)}(t) = U_{:,1:6}^{\top}\mathbf{r}^{(k)}(t)$. We then computed the mean and covariance of $\mathbf{x}^{(k)}(t)$ over trials separately within each split of the data, conditioned on stimulus and time bin. This gave the set of linear discriminant vectors computed on split s, $\mathbf{w}^{(s)}(\theta_1, \theta_2, t) = \left[0.5\left(\Sigma_{\mathbf{x}}^{(s)}(\theta_1, t) + \Sigma_{\mathbf{x}}^{(s)}(\theta_2, t)\right)\right]^{-1}\left(\overline{\mathbf{x}}^{(s)}(\theta_1, t) - \overline{\mathbf{x}}^{(s)}(\theta_2, t)\right)$. The cross-validated SNR was computed by applying the de-coder trained on one split of the data to the responses on the second split, i.e. $\text{SNR}(\theta_1, \theta_2, t) = \frac{\mathbf{w}^{(s)}(\theta_1,\theta_2,t)\cdot\left[\overline{\mathbf{x}}^{(s')}(\theta_1,t) - \overline{\mathbf{x}}^{(s')}(\theta_2,t)\right]}{\sqrt{\mathbf{w}^{(s)}(\theta_1,\theta_2,t)\cdot\left[0.5\left(\Sigma_{\mathbf{x}}^{(s')}(\theta_1,t)+\Sigma_{\mathbf{x}}^{(s')}(\theta_2,t)\right)\right]\mathbf{w}^{(s)}(\theta_1,\theta_2,t)}}$. We then averaged the resulting SNR over 100 ran-dom two-way splits.

Data availability

The current manuscript involves only computational modelling and previously published data. Modelling code are available on GitHub and previously published data are available on Dryad (*http://doi.org/10.5061/dryad.kkwh70sct*).

Supplementary Material

A Network Model

We considered a linear dynamical system governing the evolution of the network state $\mathbf{x}(t)$ via the differential equation

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{u}(s,t), \tag{1}$$

which had as parameters the connectivity matrix A and the stimulus-dependent input $\mathbf{u}(s, t)$ where the stimulus was labelled s. The input

$$\mathbf{u}(s,t) = \mathbf{u}(s)\delta(t - t_s) + \mathbf{n}(t) \tag{2}$$

consisted of a stimulus-dependent vector $\mathbf{u}(s)$ indicating the direction in state space along which the network was driven by the stimulus s at the time of stimulus presentation t_s, and temporally uncorrelated Gaussian white noise $\mathbf{n}(t) \sim \mathcal{N}(0, \Sigma_{\mathbf{n}})$, $\langle \mathbf{n}(t)\mathbf{n}^{\top}(t')\rangle = \Sigma_{\mathbf{n}}\delta(t - t')$. We also considered extensions to temporally extended inputs $\mathbf{u}(s, t) = \mathbf{u}(s)\tilde{u}(t) + \mathbf{n}(t)$ where $\tilde{u}(t)$ was a boxcar function (as in Figure 1).

Given the initial condition $\mathbf{x}_0 = \mathbf{x}(t_0)$, Equation 1 has the general solution for $t \geq t_0$

$$\mathbf{x}(t) = e^{A(t-t_0)}\mathbf{x}_0 + \int_{t_0}^{t} e^{A(t-\tau)}\mathbf{u}(s,\tau)d\tau. \tag{3}$$

Inserting Equation 2 into Equation 3 gives (for t, $t_s \geq t_0$)

$$\mathbf{x}(t) = e^{A(t-t_0)}\mathbf{x}_0 + e^{A(t-t_s)}\mathbf{u}(s)\Theta(t - t_s) + \int_{t_0}^{t} e^{A(t-\tau)}\mathbf{n}(\tau)d\tau,$$

where $\Theta(t)$ is the Heaviside step function. Using the shorthand for the stimulus-conditioned and marginal means of a variable $\mathbf{a}$ as $\overline{\mathbf{a}}(s) := \int_{-\infty}^{\infty} p(\mathbf{a} \mid s)\mathbf{a}d\mathbf{a}$ and $\overline{\mathbf{a}} := \int_{-\infty}^{\infty} p(\mathbf{a})\mathbf{a}d\mathbf{a}$, the stimulus-conditioned mean and covariance of $\mathbf{x}(t)$ are

$$\overline{\mathbf{x}}(s,t) = e^{A(t-t_0)}\overline{\mathbf{x}}_0 + e^{A(t-t_s)}\mathbf{u}(s)\Theta(t - t_s), \tag{4}$$

and

$$\begin{aligned}\Sigma_{\mathbf{x}(t)} &= \left\langle (\mathbf{x}(t) - \overline{\mathbf{x}}(s,t))(\mathbf{x}(t) - \overline{\mathbf{x}}(s,t))^\top \right\rangle \\ &= e^{A(t-t_0)}\Sigma_{\mathbf{x}_0}e^{A^\top(t-t_0)} + \int_{t_0}^{t} e^{A(t-\tau)}\Sigma_{\mathbf{n}}e^{A^\top(t-\tau)}d\tau \\ &= \underbrace{e^{A(t-t_0)}\Sigma_{\mathbf{x}_0}e^{A^\top(t-t_0)}}_{\Sigma^{\mathrm{IC}}_{\mathbf{x}(t)}} + \underbrace{\int_{0}^{t-t_0} e^{A\tau}\Sigma_{\mathbf{n}}e^{A^\top\tau}d\tau}_{\Sigma^{\mathrm{input}}_{\mathbf{x}(t)}},\end{aligned} \tag{5}$$

where we assumed that the initial condition is distributed as $\mathbf{x}_0 \sim \mathcal{N}(\overline{\mathbf{x}}_0, \Sigma_{\mathbf{x}_0})$ and is independent of the subsequent input noise $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{n}})$. The terms $\sum^{\mathrm{IC}}_{\mathbf{x}(t)}$ and $\sum^{\mathrm{input}}_{\mathbf{x}(t)}$ capture the response covariance generated by propagating the initial state and subsequent noise input through the recurrent dynamics, respectively. The response at time t under each stimulus is normally distributed over trials, with $\mathbf{x}(t) \sim N(\overline{\mathbf{x}}(s,t), \Sigma_{\mathbf{x}(t)})$, and the covariance $\Sigma_{\mathbf{x}(t)}$ is independent of the stimulus s.

B Loss Functions and Their Derivatives

We consider the case where one of M possible input stimuli $s \in \{0, \ldots, M-1\}$ is presented, and optimised the performance of an optimal decoder of s from the network response $\mathbf{x}(t)$. Since the response statistics are Gaussian with stimulus-independent covariance, the performance of an optimal decoder when $M = 2$ is given by a linear readout $\hat{s} = \Theta(\mathbf{w}_{\mathrm{LD}} \cdot \mathbf{x}(t) + c)$, where $\mathbf{w}_{\mathrm{LD}} \propto \Sigma^{-1}_{\mathbf{x}(t)}(\overline{\mathbf{x}}(s=1,t) - \overline{\mathbf{x}}(s=0,t)) = \Sigma^{-1}_{\mathbf{x}(t)}\Delta\overline{\mathbf{x}}(t)$ is the linear discriminant vector and c is a bias term (Chadwick et al., 2023). The performance of an arbitrary linear decoder $\mathbf{w}$ can be described by a signal-to-noise ratio given by:

$$\begin{aligned}\mathrm{SNR}(\mathbf{w}, t) &:= \frac{(\mathbf{w} \cdot \Delta\overline{\mathbf{x}}(t))^2}{\mathbf{w} \cdot \Sigma_{\mathbf{x}(t)}\mathbf{w}} \\ \leq \mathrm{SNR}(t) &:= \mathrm{SNR}(\mathbf{w}_{\mathrm{LD}}, t) = \Delta\overline{\mathbf{x}}(t) \cdot \Sigma^{-1}_{\mathbf{x}(t)}\Delta\overline{\mathbf{x}}(t).\end{aligned} \tag{6}$$

In particular, the probability of correct choice is $p_{\mathrm{correct}}(t) = \Phi(\sqrt{\mathrm{SNR}}/2)$, where $\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-x^2/2}dx$ is the cumulative distribution function of the standard normal distribution (Stanislaw and Todorov,1999). Although we consider discrete classification tasks, the SNR also places a lower bound on the decoding error for continuous estimation tasks. In particular, the Fisher information is given by $I_F(s) = \overline{\mathbf{x}}'(s) \cdot \Sigma^{-1}_{\mathbf{x}(t)}\overline{\mathbf{x}}'(s)$, and the Cramer-Rao lower bound is $\langle (s - \hat{s})^2 \rangle \geq 1/I_F(s)$.

For two-stimulus tasks, we therefore defined the loss function $\mathscr{L} = \int_0^\infty l(t)L(t)dt$, where $L(t) = \frac{1}{\mathrm{SNR}(t)}$ and $l(t)$ is an arbitrary weighting function. We chose to minimise the reciprocal of the SNR rather than directly maximising the SNR, as this avoids solutions where a low SNR at time point t is compensated by a high SNR at t'.

The derivative of this loss function with respect to the weights is

$$\begin{aligned}\frac{\partial L}{\partial A_{ij}} = \frac{\partial L}{\partial \Sigma_{\mathbf{x}(t)}} \cdot \frac{\partial \Sigma_{\mathbf{x}(t)}}{\partial A_{ij}} + \frac{\partial L}{\partial \Delta\overline{\mathbf{x}}(t)} \cdot \frac{\partial \Delta\overline{\mathbf{x}}(t)}{\partial A_{ij}} = -\frac{1}{\left(\Delta\overline{\mathbf{x}}^\top(t)\Sigma^{-1}_{\mathbf{x}(t)}\Delta\overline{\mathbf{x}}(t)\right)^2} \\ \left(\frac{\partial \Delta\overline{\mathbf{x}}^\top(t)\Sigma^{-1}_{\mathbf{x}(t)}\Delta\overline{\mathbf{x}}(t)}{\partial \Sigma_{\mathbf{x}(t)}} \cdot \frac{\partial \Sigma_{\mathbf{x}(t)}}{\partial A_{ij}} + \frac{\partial \Delta\overline{\mathbf{x}}^\top(t)\Sigma^{-1}_{\mathbf{x}(t)}\Delta\overline{\mathbf{x}}(t)}{\partial \Delta\overline{\mathbf{x}}(t)} \cdot \frac{\partial \Delta\overline{\mathbf{x}}(t)}{\partial A_{ij}}\right),\end{aligned}$$

where $M \cdot N = \mathrm{Trace}(MN^\top) = \Sigma_{i,j}M_{ij}N_{ij}$ denotes a sum over elements. Given the derivatives $\frac{\partial \Delta\overline{\mathbf{x}}(t)}{\partial A_{ij}}$ and $\frac{\partial \Sigma_{\mathbf{x}(t)}}{\partial A_{ij}}$, the above equation allows for a direct computation of the gradient of the loss function with respect to A. We describe how these derivatives can be computed in Section C.

For the general setting of networks with N neurons, M stimuli, and a penalty on response energy, we optimised the loss

$$\mathscr{L} = \int_0^\infty l(t)L(t)dt + \beta E,$$

where $L(t) = \sum_{0\le s<s'\le M-1}\left[\left(\overline{\mathbf{x}}(s,t) - \overline{\mathbf{x}}\left(s',t\right)\right)^\top \Sigma_{\mathbf{x}(t)}^{-1}\left(\overline{\mathbf{x}}(s,t) - \overline{\mathbf{x}}\left(s',t\right)\right)\right]^{-1}$ and $E = \sum_{s=0}^{M-1}\int_0^\infty \|\overline{\mathbf{x}}(s,t)\|^2 dt$.

C Derivatives of the Network Statistics with Respect to the Weights

To perform gradient descent on this loss function, we computed the derivatives of the mean $\overline{\mathbf{x}}(s,t)$, covariance $\Sigma_{\mathbf{x}(t)}$ and energy E with respect to the weight matrix A.

Derivative of mean response

The derivative of the mean response (Equation 4) is

$$\frac{d\overline{\mathbf{x}}(s,t)}{dA_{ij}} = \frac{de^{A(t-t_0)}}{dA_{ij}}\overline{\mathbf{x}}_0 + \frac{de^{A(t-t_s)}}{dA_{ij}}\mathbf{u}(s)\Theta\left(t-t_s\right).$$

The loss functions we consider depend on the difference in mean response to the two stimuli $\Delta\overline{\mathbf{x}}(t) = \overline{\mathbf{x}}(s=1,t) - \overline{\mathbf{x}}(s=0,t)$., which has derivative:

$$\frac{d\Delta\overline{\mathbf{x}}(t)}{dA_{ij}} = \frac{de^{A(t-t_s)}}{dA_{ij}}\Delta\mathbf{u}\Theta\left(t-t_s\right)$$

The derivative of the matrix exponential can be computed numerically (Najfield and Havel, 1995; Al-Mohy and Higham, 2008), and a solver is available via the `expm_frechet` function in SciPy's Linear Algebra library.

Derivative of response covariance

The derivative of the covariance (Equation 5) requires the solution of two Lyapunov equations, with numerical solvers widely available. The first Lyapunov equation is the result of integrating $\sum_{\mathbf{x}(t)}^{\text{input}}$ by parts

$$\begin{aligned}\int_0^{t-t_0}\frac{d}{d\tau}\left[e^{A\tau}\Sigma_{\mathbf{n}}e^{A^\top\tau}\right]d\tau &= \left[e^{A\tau}\Sigma_{\mathbf{n}}e^{A^\top\tau}\right]_0^{t-t_0}\\ \implies A\Sigma_{\mathbf{x}(t)}^{\text{input}} + \Sigma_{\mathbf{x}(t)}^{\text{input}}A^\top + \Phi &= 0,\end{aligned}$$

where $\Phi = \Sigma_{\mathbf{n}} - e^{A(t-t_0)}\Sigma_{\mathbf{n}}e^{A^\top(t-t_0)}$. The second Lyapunov equation results from differentiating the first with respect to A_{ij} to obtain

$$A\frac{d\Sigma_{\mathbf{x}(t)}^{\text{input}}}{dA_{ij}} + \frac{d\Sigma_{\mathbf{x}(t)}^{\text{input}}}{dA_{ij}}A^\top + \Psi = 0,$$

where $\Psi = \frac{d\Phi}{dA_{ij}} + E^{(ij)}\Sigma_{\mathbf{x}(t)} + \Sigma_{\mathbf{x}(t)}E^{(ji)}$, with $E_{kl}^{(ij)} = \delta_{ik}\delta_{jl}$ and $\frac{d\Phi}{dA_{ij}} = -\frac{de^{A(t-t_0)}}{dA_{ij}}\Sigma_{\mathbf{n}}e^{A^\top(t-t_0)} - e^{A(t-t_0)}\Sigma_{\mathbf{n}}\frac{de^{A^\top(t-t_0)}}{dA_{ij}}$. Since this is another Lyapunov equation it can be solved in terms of A and Ψ to obtain $\frac{d\sum_{\mathbf{x}(t)}^{\text{input}}}{dA_{ij}}$. The derivative of the network covariance $\frac{d\Sigma_{\mathbf{x}(t)}}{dA_{ij}} = \frac{d\Sigma_{\mathbf{x}(t)}^{\text{IC}}}{dA_{ij}} + \frac{d\Sigma_{\mathbf{x}(t)}^{\text{input}}}{dA_{ij}}$ can then be evaluated by handling $\frac{d\sum_{\mathbf{x}(t)}^{\text{input}}}{dA_{ij}}$ as described here and $\frac{d\Sigma_{\mathbf{x}(t)}^{\text{IC}}}{dA_{ij}}$ using a numerical solver

for the derivative of the matrix exponential, as described above for the derivative of the mean (assuming that the initial state covariance $\Sigma_{\mathbf{x}_0}$ is independent of the weights A).

Limiting cases for response covariance

Two special cases allowed us to neglect the role of the initial condition-driven covariance $\sum_{\mathbf{x}(t)}^{\mathrm{IC}} \cdot 1)$ the scenario where the network was initialised at stationary state with respect to the input noise $\Sigma_{\mathbf{n}}$, which can be found by setting $t_0 \to -\infty$, so that $\Phi = \Sigma_{\mathbf{n}}$ and $\frac{d\Phi}{dA_{ij}} = 0$, and $\Sigma_{\mathbf{x}(t)}^{\mathrm{IC}} = 0, 2)$ the scenario where $\Sigma_{\mathbf{x}_0} = 0$, which corresponds to a fixed (zero-variance) initial condition. We focused on these special cases in order to reduce the space of free parameters in the model and to gain insight into the behaviour of the model in these limiting situations.

Derivative of the network energy term

The energy E and its gradient $\frac{dE}{dA_{ij}}$ could be computed directly without numerical integration using $\int_0^\infty \|\overline{\mathbf{x}}(s,t)\|^2 dt = \mathbf{u}(s)^\top \int_0^\infty e^{A^\top t} e^{At} dt \mathbf{u}(s)$, with the inte-gral and its derivative having direct solutions via Lyapunov equations, as was the case for the network covariance.

D Relationship to Backpropagation and Other Learning Rules

The loss function and optimisation procedure described above are based on the trial-averaged statistics of the network response. How does this relate to learning rules that operate based on trial-to-trial feedback, such as reward-based Hebbian learning, backpropagation through time, or reinforcement learning? To address this question, we show that 1) in the limit of slow learning rate, any learning rule that maximises the fraction of correct choices based on trial-by-trial feedback will converge to the same weight updates as our method 2) in the limit of large batch size, backpropagation through time on a squared error loss function produces similar weight updates to our method and converges to the same solution.

Relationship to learning from trial-by-trial feedback

The objective we optimise L = 1/SNR is directly related to the probability of a correct decision p_{correct} using an optimal decoder, since $p_{\mathrm{correct}} = \Phi(\sqrt{\mathrm{SNR}}/2)$ where $0.5 \le \Phi(x) \le 1$ is monotonically increasing (as described above). Thus, $\frac{dp_{\mathrm{correct}}}{dA} = \frac{1}{4\sqrt{2\pi\,\mathrm{SNR}}} e^{-\mathrm{SNR}/8} \frac{d\,\mathrm{SNR}}{dA}$ and $\frac{dL}{dA} = \eta(\mathrm{SNR}) \frac{dp_{\mathrm{correct}}}{dA}$. The factor $\eta(\mathrm{SNR}) = -4\frac{\sqrt{2\pi}}{\mathrm{SNR}^{3/2}} e^{\mathrm{SNR}/8}$ depends only on the SNR, so that the two gradients differ only up to a scaling factor, and therefore produce identical weight trajectories. In the limit of a small learning rate, any learning rule that optimises the fraction of correct choices using trial-by-trial feedback will converge to the gradient $\frac{dp_{\mathrm{correct}}}{dA}$, and will therefore follow the same learning dynamics as our method.

Relationship to backpropagation through time

Methods which do not explicitly optimise the probability of correct choice may nevertheless yield very similar learning dynamics to those produced by our method. To illustrate this finding, we derive analytical expressions for the updates generated via backpropagation through time (BPTT) and show that numerical implementation of these updates generates near-identical weight trajectories to those produced by our method. Specifically, we consider BPTT on the continuous-time linear recurrent neural network $\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{u}(s,t) =: \mathbf{f}(\mathbf{x},\mathbf{u})$ with $\mathbf{u}(s,t) = \mathbf{u}(s)\delta(t - t_s) + \mathbf{n}(t)$. On each trial, we consider a trajectory from initial time point t_0 to decision time t_d with a loss L that depends only on the final time point $\mathbf{x}(t_d)$. The continuous-time BPTT updates are then given by

$$\frac{dL}{dA} = \sum_{i,j=1}^{N} \int_{t_0}^{t_d} \frac{\partial L}{\partial x_i(t_d)} \frac{\partial x_i(t_d)}{\partial x_j(t)} \frac{\partial f_j(\mathbf{x}(t),\mathbf{u}(t))}{\partial A} dt. \tag{7}$$

We consider the squared error loss function $L(t_d) = (\mathbf{w}\cdot\mathbf{x}(t_d) + c - s)^2 = (\hat{s} - s)^2$. The terms in Equation (7) are the $\frac{\partial L}{\partial \mathbf{x}(t_d)} = 2(\mathbf{w}\cdot\mathbf{x}(t_d) + c - s)\mathbf{w}$, $\frac{\partial \mathbf{x}(t_d)}{\partial \mathbf{x}(t)} = e^{A(t_d - t)}$, $\frac{\partial f_i(\mathbf{x},\mathbf{u})}{\partial A_{jk}} = \delta_{ij} x_k$. Therefore

$$\frac{dL}{dA_{jk}} = \sum_{i=1}^{N} \int_{t_0}^{t_d} \frac{\partial L}{\partial x_i\left(t_d\right)} \left[e^{A(t_d-t)}\right]_{ij} x_k(t)dt \Longrightarrow \frac{dL}{dA} = \int_{t_0}^{t_d} e^{A^\top(t_d-t)} \frac{\partial L}{\partial \mathbf{x}\left(t_d\right)} \mathbf{x}^\top(t)dt.$$

We next consider the stationary state limit $t_0 \to -\infty$, with

$$\frac{dL}{dA} = \int_0^\infty e^{A^\top t} \frac{\partial L}{\partial \mathbf{x}\left(t_d\right)} \mathbf{x}^\top\left(t_d - t\right) dt.$$

Inserting the expressions derived above for $\frac{\partial L}{\partial \mathbf{x}(t_d)}$ and $\mathbf{x}(t) = \int_0^\infty e^{At}\mathbf{u}\left(s, t-t^{'}\right) dt^{'}$ we obtain

$$\begin{aligned}\frac{dL}{dA} &= \alpha \int_0^\infty \int_0^\infty e^{A^\top t} \mathbf{w}\mathbf{u}^\top\left(s, t_d - t - t^{'}\right) e^{A^\top t^{'}} dt dt^{'} \\ &= \alpha \int_0^\infty \\ &\int_0^\infty e^{A^\top t} \mathbf{w}\left(\mathbf{u}^\top(s)\delta\left(t_d - t - t^{'} - t_s\right) + \mathbf{n}^\top\left(t_d - t - t^{'}\right)\right) e^{A^\top t^{'}} dt^{'} dt\end{aligned}$$

where $\alpha = 2\mathbf{w}^\top \int_0^\infty e^{At^{'}} \mathbf{u}\left(s, t_d - t^{'}\right) dt^{'} + 2(c-s) = 2\mathbf{w}^\top e^{A(t_d-t_s)}\mathbf{u}(s)$ $+ 2\mathbf{w}^\top \int_0^\infty e^{At^{'}} \mathbf{n}\left(t_d - t^{'}\right) dt^{'} + 2(c-s)$.Using $\langle \mathbf{n}(t) \rangle = 0$ and $\langle \mathbf{n}(t)\mathbf{n}^\top(t^{'}) \rangle = \Sigma_{\mathbf{n}}\delta(t - t^{'})$, we then compute the expected gradient over all possible realisations of the noise term $\mathbf{n}$ (for fixed s):

$$\begin{aligned}\left\langle \frac{dL}{dA} \right\rangle &= 2\left(\mathbf{w}^\top e^{A(t_d-t_s)}\mathbf{u}(s) + c - s\right) \int_0^\infty \\ &\int_0^\infty e^{A^\top t} \mathbf{w}\mathbf{u}^\top(s) e^{A^\top t^{'}} \delta\left(t_d - t - t^{'} - t_s\right) dt dt^{'} \\ &+ 2\int_0^\infty \int_0^\infty \\ &\int_0^\infty e^{A^\top t} \mathbf{w}\mathbf{w}^\top e^{At^{''}} \left\langle \mathbf{n}\left(t_d - t^{''}\right) \mathbf{n}^\top\left(t_d - t - t^{'}\right)\right\rangle e^{A^\top t^{'}} dt dt^{'} dt^{''} \\ &= 2\left(\mathbf{w}^\top e^{A(t_d-t_s)}\mathbf{u}(s) + c - s\right) \underbrace{\int_0^{t_d-t_s} e^{A^\top t} \mathbf{w}\mathbf{u}^\top(s) e^{A^\top(t_d-t_s-t)} dt}_{I_1}\end{aligned} \tag{8}$$

$$+ 2\underbrace{\left[\int_0^\infty e^{A^\top t} \mathbf{w}\mathbf{w}^\top e^{At} dt\right]}_{I_2} \underbrace{\left[\int_0^\infty e^{At^{'}} \Sigma_{\mathbf{n}} e^{A^\top t^{'}} dt^{'}\right]}_{I_3}. \tag{9}$$

Each of the above integrals I_1, I_2, I_3 can be reformulated as a Lyapunov or Sylvester equation, with numerically efficient and stable solvers available. In particular, I_2 = lyap($A^\dagger$, $\mathbf{w}\mathbf{w}^\dagger$) and I_3 = lyap(A, $\Sigma_{\mathbf{n}}$). Integrating I_1 by parts gives $A^\top I_1 - I_1 A^\top = e^{A^\top(t_d-t_s)}\mathbf{w}\mathbf{u}^\top - \mathbf{w}\mathbf{u}^\top e^{A^\top(t_d-t_s)}$, which is a Sylvester equa-tion of the form $PI_1 + I_1 Q = C$. However, this Sylvester equation does not have a unique solution (Sylvester equations have a unique solution if P and $-Q$ do not share a common eigenvalue. Here, however, $P = -Q$, violating this condition). Instead, a unique solution for I_1 can be found by taking the eigendecomposition $e^{At} = Ve^{\Lambda t}V^{-1} = \sum_{k=1}^{N} \mathbf{v}_k \tilde{\mathbf{v}}_k^\top e^{\lambda_k t}$, which gives

$$I_1 = \sum_{ij} \tilde{\mathbf{v}}_i \mathbf{v}_j^\top \left(\mathbf{v}_i^\top \mathbf{w}\mathbf{u}^\top(s)\tilde{\mathbf{v}}_j\right) e^{\lambda_j(t_d-t_s)} \int_0^{t_d-t_s} e^{(\lambda_i-\lambda_j)t} dt = V^{-\top} Q V^\top \tag{10}$$

where $Q = (V^{\top}\mathbf{w}\mathbf{u}^{\top}V^{-\top}) \odot S$, with $S_{ii} = (t_d - t_s)\, e^{\lambda_i (t_d - t_s)}$ and $S_{ij} = \left(e^{\lambda_i (t_d - t_s)} - e^{\lambda_j (t_d - t_s)}\right) / (\lambda_i - \lambda_j)$ (for $i \neq j$), with $\odot$ the Hadamard (or element-wise) product. Thus, each update of A during the BPTT algorithm can be achieved by solving two Lyapunov equations (for I_2 and I_3) and performing an eigendecomposition on A to compute I_1.

To complete the BPTT algorithm, we require updates for the readout weights $\mathbf{w}$ and threshold c. These updates are given as follows:

$$\left\langle \frac{\partial L}{\partial \mathbf{w}} \right\rangle = 2\left[\Sigma_{\mathbf{x}(t)} + \overline{\mathbf{x}}(s,t_d)\,\overline{\mathbf{x}}^{\top}(s,t_d)\right]\mathbf{w} + 2(c-s)\overline{\mathbf{x}}(s,t_d)$$
$$\left\langle \frac{\partial L}{\partial c} \right\rangle = 2\left[\mathbf{w}^{\top}\overline{\mathbf{x}}(s,t_d) + c - s\right].$$

These readout parameters could be updated via gradient descent alongside the recurrent weights A. However, if we assume that the updates of the recurrent weights are slow relative to the readout weights, we can set

$$\sum_{s,t_d} \left\langle \frac{\partial L}{\partial c} \right\rangle = 0 \Longrightarrow c = -\sum_{s,t_d} \left(\mathbf{w}^{\top}\overline{\mathbf{x}}(s,t_d) - s\right) \tag{11}$$

$$\sum_{s,t_d} \left\langle \frac{\partial L}{\partial \mathbf{w}} \right\rangle = 0 \Longrightarrow \mathbf{w} = -\sum_{s,t_d} (c-s) M^{-1}\overline{\mathbf{x}}(s,t_d), \tag{12}$$

where $M = \sum_{s,t_d}\left[\Sigma_{\mathbf{x}} + \overline{\mathbf{x}}(s,t_d)\,\overline{\mathbf{x}}^{\top}(s,t_d)\right]$, and we included a sum over time points to handle the case of a fixed (time-independent) readout over a range of delays. Combining these two expressions gives:

$$c = \frac{\sum_{s,t_d}\left(s - \sum_{s',t_d'} s'\overline{\mathbf{x}}^{\top}(s',t_d')\, M^{-1}\overline{\mathbf{x}}(s,t_d)\right)}{\sum_{s,t_d}\left(1 - \sum_{s',t_d'} \overline{\mathbf{x}}^{\top}(s',t_d')\, M^{-1}\overline{\mathbf{x}}(s,t_d)\right)} \tag{13}$$

with $\mathbf{w}$ obtained by inserting this expression into Equation (12). Taken together, the full BPTT algorithm involves iteratively using Equations (12-13) to update readout weights followed by Equations (8-10) to update the recurrent weights (summed over stimuli and time points). An implementation of this algorithm is shown in Supplementary Figure S4.

E Optimal Solution to Task: An Ideal Observer Analysis

We consider an ideal observer of the network input $\mathbf{u}(s, t) = \mathbf{u}(s)\tilde{u}(t) + \mathbf{n}(t)$ tasked with reporting the stimulus identity $s \in \{0, 1\}$ at time t_d. Assuming that the input noise is Gaussian, stimulus-independent, and temporally uncorrelated, the optimal solution involves two steps: 1) projecting the inputs onto the linear discriminant vector $\mathbf{b} \propto \Sigma_{\mathbf{n}}^{-1}\Delta\mathbf{u}$2) applying a temporal filter f that selects and delays inputs over time. Here, we formalise this solution mathematically, showing that the optimal solution is a form of matched filtering (Turin, 1960).

We consider a linear projection and temporal filtering of the input time series $\mathbf{u}(s, t \leq t_d)$ onto a vector $\mathbf{w}$ and filter $y(s,t_d) = \int_0^{\infty} f(t,t_d)\,(\mathbf{w}\cdot\mathbf{u}(s,t_d - t))\,dt$ with

$$\begin{aligned}\mathrm{SNR}\left(t_d\right) &:= \frac{\left(\bar{y}\left(s=1,t_d\right)-\bar{y}\left(s=0,t_d\right)\right)^2}{\frac{1}{2}\sum_s\left\langle\left(y\left(s,t_d\right)-\bar{y}\left(s,t_d\right)\right)^2\right\rangle} \\ &= \frac{\left(\mathbf{w}\cdot\Delta\mathbf{u}\right)^2}{\mathbf{w}\cdot\Sigma_{\mathbf{n}}\mathbf{w}}\frac{\left[\int_0^\infty dt f\left(t,t_d\right)\tilde{u}\left(t_d-t\right)\right]^2}{\int_0^\infty dt f^2\left(t,t_d\right)} \\ &\le \underbrace{\left(\Delta\mathbf{u}^\top\Sigma_{\mathbf{n}}^{-1}\Delta\mathbf{u}\right)}_{\mathrm{SNR}_{\mathrm{input}}^0}\int_0^\infty dt\tilde{u}^2\left(t_d-t\right).\end{aligned} \tag{14}$$

In particular, the above SNR is maximised when $\mathbf{w}=\mathbf{b}\propto\Sigma_{\mathbf{n}}^{-1}\Delta\mathbf{u}$ and $f(t, t_d) \propto \tilde{u}(t_d - t)$. These filtering and projection operations can be understood as linear discriminant analysis over both neurons and time, and can also be derived as the maximum likelihood estimate of s given the full input time series up to time t_d, i.e. by maximising $p(\mathbf{u}(s, t \le t_d)\, s)$, with $\hat{s}_{\mathrm{ML}} = \Theta\left(\int_0^\infty \left(\mathbf{b}\cdot\mathbf{u}\left(s,t_d-t\right)+c\right)\tilde{u}\left(t_d-t\right)dt\right)$. Thus, Equation (14) quantifies the performance of a Bayes optimal ideal observer of the network input (assuming equal prior likelihood of each stimulus).

The above solution requires a filter that depends on the optimised readout time, also known as a time-varying linear filter. In contrast, a fixed filter optimised for a single readout time t_d will have a suboptimal output SNR for $t \neq t_d$

$$\mathrm{SNR}\left(t,t_d\right)=\mathrm{SNR}_{\mathrm{input}}^0\frac{\left[\int_0^\infty d\tau\tilde{u}\left(t_d-\tau\right)\tilde{u}(t-\tau)\right]^2}{\int_0^\infty d\tau\tilde{u}^2\left(t_d-\tau\right)} \tag{15}$$

To illustrate these solutions, we provide examples from three tasks: first, a working memory task with instantaneous inputs, $\tilde{u}_\delta(t) = \delta(t - t_s)$; second, an evidence integration task where $\tilde{u}_\Theta(t) = \Theta(t - t_s)$; third, a working memory task where the stimulus appears for a fixed interval of time, modelled as a "boxcar" function $\tilde{u}_\Pi(t) = \Theta(t - t_s) - \Theta(t - (t_s + T))$. The optimal time-varying filters for these tasks are $f_\delta(t, t_d) = \delta(t_d - t_s - t)$, $f_\Theta(t, t_d) = \Theta(t_d - t_s - t)$, $f_\Pi(t, t_d) = \Theta(t_d - t_s - t) - \Theta(t_d - t_s - T - t)$.

Following Equation (14), these filters give rise to the following SNRs:

$$\begin{aligned}\frac{\mathrm{SNR}_\delta\left(t_d\right)}{\mathrm{SNR}_{\mathrm{input}}^0} &= \delta(0)\Theta\left(t_d-t_s\right)=\begin{cases}\infty, & \text{if } t_d>t_s\\ 0, & \text{if } t_d<t_s\end{cases} \\ \frac{\mathrm{SNR}_\Theta\left(t_d\right)}{\mathrm{SNR}_{\mathrm{input}}^0} &= \left(t_d-t_s\right)\Theta\left(t_d-t_s\right)=\left[t_d-t_s\right]_+=\begin{cases}t_d-t_s, & \text{if } t_d>t_s\\ 0, & \text{if } t_d<t_s\end{cases} \\ \frac{\mathrm{SNR}_\Pi\left(t_d\right)}{\mathrm{SNR}_{\mathrm{input}}^0} &= \left[t_d-t_s\right]_+-\left[t_d-t_s-T\right]_+ \\ &= \begin{cases}T, & \text{if } t_d>t_s+T\\ t_d-t_s, & \text{if } t_s+T>t_d>t_s\\ 0, & \text{if } t_d<t_s\end{cases}\end{aligned}$$

These solutions are illustrated in Supplementary Figure S2, where we also show the performance of fixed filters optimised for a single readout time (Equation (15)).

F Analysis of Linear Dynamical System Task Performance

We next sought to determine the performance of linear network models on the working memory task. The above ideal observer analysis provides an upper bound on the SNR of network responses, and suggests that networks could be suboptimal due to two factors: 1) a failure to correctly combine inputs to different neurons via a suboptimal projection $\mathbf{w}$, or 2) a failure to correctly combine inputs across time via the wrong temporal filtering f.

We show that attractor networks (those with orthogonal eigenvectors, also called normal networks) perform poorly on working memory tasks, with their SNR decaying at best as t_d^{-1}. We then show that (non-normal) functionally-feedforward networks can approximate the ideal observer solution to an arbitrary level of precision, but require exponential increases in firing rates to achieve this. Finally, we show that networks with rotational dynamics achieve higher SNR than both attractor and feedforward networks.

In the following we consider networks receiving delta pulse inputs $\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{u}(s)\delta(t) + \mathbf{n}(t)$ (we set $t_s = 0$ for convenience). We consider the signal-to-noise ratio of network responses along a readout vector $\mathbf{w}$ defined as $\mathrm{SNR}(\mathbf{w}) = \frac{\left(\mathbf{w}^\top \Delta\overline{\mathbf{x}}\right)^2}{\mathbf{w}^\top \Sigma_{\mathbf{x}(t)} \mathbf{w}}$. We will often consider the optimal readout $\mathbf{w}_{\mathrm{LD}} \propto \Sigma_{\mathbf{x}(t)}^{-1} \Delta\overline{\mathbf{x}}$ with $\mathrm{SNR}\left(\mathbf{w}_{\mathrm{LD}}\right) = \Delta\overline{\mathbf{x}}^\top \Sigma_{\mathbf{x}(t)}^{-1} \Delta\overline{\mathbf{x}}$.

F.1 SNR Along Individual Dynamical Modes

Previous work has suggested that alignment of dynamical modes with the input linear discriminant provides a neural mechanism for working memory and evidence integration tasks (Seung, 1996; Mante et al., 2013; Chadwick et al., 2023; Stroud et al., 2023; Pagan et al., 2024). We therefore first investigated the SNR along individual dynamical modes of a network.

To decompose the dynamics into individual modes, we use the eigendecomposition $A = \sum_i \lambda_i \mathbf{v}_i \tilde{\mathbf{v}}_i^\top$, $e^{At} = \sum_i \mathbf{v}_i \tilde{\mathbf{v}}_i^\top e^{\lambda_i t}$, where $\tilde{\mathbf{v}}_i \cdot \mathbf{v}_j = \delta_{ij}$ are left and right eigenvectors chosen such that $\|\mathbf{v}\|^i = 1$. Given Equations (4) and (5), the mean and variance of network output projected onto a left eigenvector are (for $t \geq t_s = 0$ and $t_0 \leq t_s = 0$)

$$\begin{aligned}
\tilde{\mathbf{v}}_k \cdot \Delta\overline{\mathbf{x}}(t) &= \tilde{\mathbf{v}}_k \cdot e^{At} \Delta\mathbf{u} = \left(\tilde{\mathbf{v}}_k \cdot \Delta\mathbf{u}\right) e^{\lambda_k t} \\
\tilde{\mathbf{v}}_k \cdot \Sigma_{\mathbf{x}(t)} \tilde{\mathbf{v}}_k &= \tilde{\mathbf{v}}_k \cdot \left[e^{A(t-t_0)} \Sigma_{\mathbf{x}_0} e^{A^\top (t-t_0)}\right] \tilde{\mathbf{v}}_k + \tilde{\mathbf{v}}_k \\
&\quad \cdot \left[\int_0^{t-t_0} e^{A\tau} \Sigma_{\mathbf{n}} e^{A^\top \tau} d\tau\right] \tilde{\mathbf{v}}_k \\
&= \left(\tilde{\mathbf{v}}_k \cdot \Sigma_{\mathbf{x}_0} \tilde{\mathbf{v}}_k\right) e^{2\lambda_k (t-t_0)} + \left(\tilde{\mathbf{v}}_k \cdot \Sigma_{\mathbf{n}} \tilde{\mathbf{v}}_k\right) \left(2\lambda_k\right)^{-1} \left(e^{2\lambda_k (t-t_0)} - 1\right)
\end{aligned}$$

Thus,

$$\mathrm{SNR}\left(\tilde{\mathbf{v}}_k, t\right) := \frac{\left(\tilde{\mathbf{v}}_k \cdot \Delta\overline{\mathbf{x}}(t)\right)^2}{\tilde{\mathbf{v}}_k \cdot \Sigma_{\mathbf{x}(t)} \tilde{\mathbf{v}}_k} = \left[\frac{e^{-2(t-t_0)/\tau_k}}{\mathrm{SNR}_{\mathbf{x}_0}\left(\tilde{\mathbf{v}}_k\right)} - \frac{\frac{\tau_k}{2}\left(e^{-2(t-t_0)/\tau_k} - 1\right)}{\mathrm{SNR}_{\mathrm{input}}\left(\tilde{\mathbf{v}}_k\right)}\right]^{-1}, \tag{16}$$

where $\mathrm{SNR}_{\mathbf{x}_0}\left(\tilde{\mathbf{v}}_k\right) = \left(\Delta\mathbf{u} \cdot \tilde{\mathbf{v}}_k\right)^2 / \left(\tilde{\mathbf{v}}_k \cdot \Sigma_{\mathbf{x}_0} \tilde{\mathbf{v}}_k\right)$ and $\mathrm{SNR}_{\mathrm{input}}\left(\tilde{\mathbf{v}}_k\right) = \left(\Delta\mathbf{u} \cdot \tilde{\mathbf{v}}_k\right)^2 / \left(\tilde{\mathbf{v}}_k \cdot \Sigma_{\mathbf{n}} \tilde{\mathbf{v}}_k\right)$.

Optimal dynamics at stationary state

If the network is initialised at stationary state with respect to the input noise (i.e. if $t_0 \to -\infty$), the response SNR is

$$\mathrm{SNR}\left(\tilde{\mathbf{v}}_k, t\right) = \frac{2}{\tau_k} e^{-2t/\tau_k} \mathrm{SNR}_{\mathrm{input}}\left(\tilde{\mathbf{v}}_k\right)$$

This solution is optimised when the left eigenvector is aligned to the input discriminant, i.e. $\tilde{\mathbf{v}}_k = \mathbf{b}$. The SNR depends non-monotonically on the integration time constant $\tau_k = -1/\lambda_k$, and has an optimal time constant for readout time t_d given by $\tau_k = 2t_d$. Thus, a single eigenmode is optimised for a readout at time t_d if its left eigenvector is aligned to the input discriminant and its eigenvalue is $\lambda_k = -\frac{1}{2t_d}$. However, this optimal SNR decreases as a function of the optimised readout time according to $\mathrm{SNR}(\mathbf{b}, t_d) = (et_d)^{-1} \mathrm{SNR}_{\mathrm{input}}(\mathbf{b})$. This is in contrast to the optimal ideal observer solution, whose performance is independent of the readout time provided $t_d \geq t_s$.

Optimal dynamics for fixed initial state

If the network is initialised at a fixed (i.e. zero-variance) initial condition $\mathbf{x}_0$ at time $t_0 \leq t_s = 0$, the response SNR is:

$$\mathrm{SNR}\left(\tilde{\mathbf{v}}_k, t\right) = \frac{2}{\tau_k} \frac{1}{e^{2t/\tau_k} - e^{2t_0/\tau_k}} \mathrm{SNR}_{\mathrm{input}}\ \left(\tilde{\mathbf{v}}_k\right)$$

Plotting this equation numerically as a function of $\lambda_k = -1/\tau_k$ for different values of t_0 shows that stable dynamics ($\lambda_k < 0$) are optimal when $|t_0| > t$, neutrally stable dynamics ($\lambda_k = 0$) are optimal when $|t_0| = t$, and unstable dynamics ($\lambda_k > 0$) are optimal when $|t_0| < t$ (Supplementary Figure S6B-D). The optimal λ_k scales monotonically with t_0, respectively approaching infinitely strong amplification $\lambda_k \to \infty$ as $t_0 \to 0$ and the stationary state solution $\lambda_k \to -\frac{1}{2t_d}$ as $t_0 \to -\infty$. The amplifying dynamics for $|t_0| < t$ act to amplify the signal throughout the delay, while the decaying dynamics for $|t_0| > t$ are required to avoid amplification of pre-stimulus noise input above the signal.

These solutions are highly suboptimal with respect to the ideal observer solution. This suboptimality arises because an individual eigenmode implements an exponential filter with time constant $\tau_k = -1/\lambda_k$, and therefore integrates noise before and after stimulus presentation. The ideal observer solution has a narrow integration time window but a long memory time constant, and departs substantially from a simple exponential filter.

F.2 Attractor Networks

For attractor (normal) networks, the eigenvectors form an orthogonal basis with $\tilde{\mathbf{v}}_i = \mathbf{v}_i$ and $\mathbf{v}_i \cdot \mathbf{v}_j = \delta_{ij}$. As a consequence, their response SNR follows trivially from the above analysis of individual eigenmodes as we show below. The mean response of an attractor network to the inputs considered above is

$$\Delta\overline{\mathbf{x}}(t) = \sum_i \left(\mathbf{v}_i \cdot \mathbf{u}(s)\right) e^{\lambda_i t} \mathbf{v}_i$$

and the response covariance is

$$\begin{aligned}
\Sigma_{\mathbf{x}(t)} &= e^{A(t-t_0)} \Sigma_{\mathbf{x}_0} e^{A^\top (t-t_0)} + \int_0^{t-t_0} e^{A\tau} \Sigma_{\mathbf{n}} e^{A^\top \tau} d\tau \\
&= \sum_{ij} \left(\mathbf{v}_i \cdot \Sigma_{\mathbf{x}_0} \mathbf{v}_j\right) e^{(\lambda_i + \lambda_j)(t-t_0)} \mathbf{v}_i \mathbf{v}_i^\top \\
&+ \sum_{ij} \left(\mathbf{v}_i^\top \Sigma_{\mathbf{n}} \mathbf{v}_j\right) \left(\lambda_i + \lambda_j\right)^{-1} \left(e^{(\lambda_i + \lambda_j)(t-t_0)} - 1\right) \mathbf{v}_i \mathbf{v}_j^\top \\
&= \sum_i \left[\sigma^2_{\mathbf{x}_0} e^{2\lambda_i (t-t_0)} + \sigma^2_{\mathbf{n}} (2\lambda_i)^{-1} \left(e^{2\lambda_i (t-t_0)} - 1\right)\right] \mathbf{v}_i \mathbf{v}_i^\top
\end{aligned}$$

where the last step assumed that $\Sigma_{\mathbf{x}_0} = \sigma^2_{\mathbf{x}_0} I$ and $\Sigma_{\mathbf{n}} = \sigma^2_{\mathbf{n}} I$.

The SNR of network output at time t is

$$\mathrm{SNR}(t) = \Delta\overline{\mathbf{x}}(t) \cdot \Sigma^{-1}_{\mathbf{x}(t)} \Delta\overline{\mathbf{x}}(t) = \sum_k \mathrm{SNR}\left(\mathbf{v}_k, t\right)$$

where $\mathrm{SNR}(\mathbf{v}_k, t)$ is the single-mode SNR given in equation (16). The SNR of an attractor network is therefore maximised when a single eigenmode is aligned to the input discriminant, i.e. $\mathbf{v}_k = \mathbf{b}$, in which case the solution reduces to that of single modes considered above, with all other modes having zero SNR.

Thus, attractor networks perform poorly on the working memory task, with SNR decaying exponentially over time due to noise (and decaying as $1/t_d$ as a function of the optimised delay). We next show how functionally-feedforward (non-normal) and rotational dynamics can improve upon attractor mechanisms.

F.3 Preliminary: The Real and Complex Schur Decomposition

While the eigendecomposition of the matrix A is sufficient to characterise the dynamics and task-performance of a normal network, this description often obscures the transient dynamics of non-normal and rotational networks (Ganguli et al., 2008; Goldman, 2009; Murphy and Miller, 2009). Instead we turned to the Schur decomposition to characterise such networks. Here we briefly describe the real and complex forms of the Schur decomposition that we use extensively in the following sections.

Complex Schur Decomposition

Any matrix $A \in \mathbb{R}^{N\times N}$ can be expressed in Schur form as $A = ULU^\dagger$ where $U \in \mathbb{C}^{N\times N}$ is a unitary matrix, i.e. $U^{-1} = U^\dagger$ where $U^\dagger := (U^\intercal)^*$, and $L \in \mathbb{C}^{N\times N}$ is an upper-triangular matrix, i.e. $L_{i,i+k} = 0$ for all $k < 0$. Given a connectivity matrix A, the off-diagonal elements of L describe "functionally-feedforward" weights between a set of orthogonal neural activity patterns (Schur modes, i.e. columns of U). However, when A has complex eigenvalues, the elements of both U and L become complex-valued, which complicates this interpretation.

Real Schur Decomposition

When some eigenvalues are complex, a more interpretable basis in which to analyse the dynamics is given by the real Schur decomposition $A = QTQ^\intercal$, where Q and T are real-valued matrices, with $Q^{-1} = Q^\intercal$ (i.e., Q is an orthogonal matrix) and T a block upper-triangular matrix

$$T = \begin{pmatrix} B_1 & \Omega_{12} & \Omega_{13} & \dots & \Omega_{1K} \\ 0 & B_2 & \Omega_{23} & \dots & \Omega_{2K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & B_{K-1} & \Omega_{K-1,K} \\ 0 & 0 & \dots & 0 & B_K \end{pmatrix}$$

Here, K is the number of real eigenvalues plus the number of pairs of complex conjugate eigenvalues and the diagonal blocks B_i are either 1×1 submatrices corresponding to real eigenvalues of A (i.e., $B_i = \lambda_k$) or 2 × 2 submatrices corresponding to complex conjugate eigenvalue pairs (with $B_i = \begin{pmatrix} \lambda_r & a \\ b & \lambda_r \end{pmatrix}$ for some $a, b \in \mathbb{R}$ such that the eigenvalues of B_i are equal to a pair of complex-conjugate eigenvalues of A, $\lambda = \lambda_r \pm i\lambda_\omega$). Since the columns of Q are orthonormal, the real Schur decomposition provides an orthonormal basis partitioned into one-dimensional non-rotational modes and two-dimensional rotational planes, with functionally-feedforward interactions between these one- and two-dimensional subspaces given by the off-diagonal blocks Ω_{ij}. Thus, the Ω_{ij}'s are either 1 × 1, 2 × 1, 1 × 2, or 2 × 2 matrices capturing functionally-feedforward interactions between these modes.

Two-Dimensional Case

A matrix with eigendecomposition $A = \lambda_1 \mathbf{v}_1 \tilde{\mathbf{v}}_1^\top + \lambda_2 \mathbf{v}_2 \tilde{\mathbf{v}}_2^\top$ and $\lambda \in \mathbb{R}$ can be expressed in Schur form as $A = \lambda_1 \mathbf{q}_1 \mathbf{q}_1^\top + \lambda_2 \mathbf{q}_2 \mathbf{q}_2^\top + \omega \mathbf{q}_2 \mathbf{q}_1^\top$, where ω is a functionally-feedforward weight from the input Schur mode $\mathbf{q}_1 := \tilde{\mathbf{v}}_1 / \|\tilde{\mathbf{v}}_1\|$ to the output Schur mode $\mathbf{q}_2 := \mathbf{v}_2$. Thus, $Q = [\mathbf{q}_2, \mathbf{q}_1]$ and

$$T = \begin{pmatrix} \lambda_2 & \omega \\ 0 & \lambda_1 \end{pmatrix}$$

A matrix $A \in \mathbb{R}^2$ with complex eigenvalues $\lambda_r \pm i\lambda_\omega$ can be written in real Schur form as $A = QTQ^\intercal$ where

$$T = \begin{pmatrix} \lambda_r & a \\ b & \lambda_r \end{pmatrix} = \begin{pmatrix} \lambda_r & \epsilon\lambda_\omega \\ -\lambda_\omega/\epsilon & \lambda_r \end{pmatrix}$$

with $\sqrt{-ab} = \lambda_\omega$ and $Q^\intercal = Q^{-1} \in \mathbb{R}^{2\times2}$ a real orthogonal matrix. In the second equality, we introduced the parameter $\epsilon = a/\lambda_\omega$, which can be understood as the ellipticity (or eccentricity) of the rotation, and is closely related to the non-normality of the system. In this basis, the major and minor axes of the ellipse are aligned to the two basis axes (defined by the columns of Q). Thus, in real Schur form the matrix A is characterised by four parameters: the real and imaginary parts of the eigenvalues, λ_r and λ_ω, which characterise the damping rate and rotational frequency respectively; the ellipticity of the orbit ϵ; and the angle of the major axis of the ellipse in neural state space (determined by the orthogonal matrix Q). The matrix exponential of A is $e^{At} = Qe^{Tt}Q^\intercal$, and the matrix exponential of T can be computed directly as

$$e^{Tt} = e^{\lambda_r t}\begin{bmatrix}\cos\lambda_\omega t, & \epsilon\sin\lambda_\omega t\\ -\frac{1}{\epsilon}\sin\lambda_\omega t, & \cos\lambda_\omega t.\end{bmatrix}$$

Importantly, these results for two-dimensional systems also apply to any 2 × 2 submatrix B_i in the N-dimensional real Schur decomposition.

Invariances of the Real Schur Decomposition

The Schur decomposition is non-unique: each ordering of eigenvalues leads to a different Q and T. For the real Schur decomposition, this can be factored into an ordering of blocks and an ordering within 2 × 2 blocks. Within a 2 × 2 block, the ordering of eigenvalues makes only a superficial difference to the resulting decomposition. In particular, if k, $k+1$ corresponds to a rotational plane (i.e., if $T_{k:k+1,k:k+1} = B_i$ for some i), the change of Schur basis $Q = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_k, \mathbf{q}_{k+1}, ..., \mathbf{q}_N] \to [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_{k+1}, \mathbf{q}_k, ..., \mathbf{q}_N]$ and $T_{:,k} \leftrightarrow T_{:,k+1}$, $T_{k,:} \leftrightarrow T_{k+1,:}$ yields another real Schur decomposition with $\epsilon \to -1/\epsilon$. A further invariance for both real and complex Schur decompositions is to reverse the sign of a mode, i.e. $Q = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_k, ..., \mathbf{q}_N] \to [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_k, ..., \mathbf{q}_N]$, $T_{k,:} \to -T_{k,:}$, $T_{:,k} \to -T_{:,k}$ with $\epsilon \to -\epsilon$. Through this combination of reordering and sign-reversal of real Schur modes, it is possible to select a real Schur decomposition for which rotations are clockwise or anticlockwise ($\epsilon \gtrless 0$) and have their major axis aligned to the x- or y-axis ($|\epsilon| \gtrless 1$), and have either positive or negative loading onto the input discriminant ($\mathbf{q}_k \cdot \mathbf{b} \gtrless 0$). For numerically optimised networks, we utilised these invariances to define a unique Schur decomposition in which 1) the decay time constants $\tau = -1/\lambda_r$ are in descending order from input to output mode 2) elliptical orbits have their major axis aligned to the x axis ($|\epsilon| > 1$) 3) rotations in each plane are clockwise ($\epsilon > 0$) 4) the input mode to each plane has positive alignment with the input discriminant.

F.4 Two-Dimensional Feedforward Networks

We first considered the performance of feedforward (non-normal) networks with real eigenvalues on the working memory task. We restrict our analysis to the stationary state limit $t_0 \to -\infty$. Using the Schur basis discussed in Section F.3, the dynamics of a two-dimensional non-normal network with real eigenvalues decouples into two equations:

$$\begin{aligned}\mathbf{q}_1 \cdot \dot{\mathbf{x}} &= \lambda_1 \mathbf{q}_1 \cdot \mathbf{x} + \mathbf{q}_1 \cdot \mathbf{u} \Longrightarrow \mathbf{q}_1 \cdot \mathbf{x} = \int_{-\infty}^{t} e^{\lambda_1 \tau}\mathbf{q}_1 \cdot \mathbf{u}(t-\tau)d\tau\\ \mathbf{q}_2 \cdot \dot{\mathbf{x}} &= (\lambda_2\mathbf{q}_2 + \omega\mathbf{q}_1)\cdot\mathbf{x} + \mathbf{q}_2 \cdot \mathbf{u}\end{aligned}$$

The mean and variance of responses along the first mode are given by the single-mode analysis in Section F.1:

$$\begin{aligned}\mathbf{q}_1 \cdot \overline{\mathbf{x}}(s,t) &= (\mathbf{q}_1 \cdot \mathbf{u}(s))\, e^{\lambda_1 t}\\ \mathbf{q}_1 \cdot \Sigma_{\mathbf{x}(t)}\mathbf{q}_1 &= (\mathbf{q}_1 \cdot \Sigma_{\mathbf{n}}\mathbf{q}_1)\,(-2\lambda_i)^{-1}\\ \mathrm{SNR}\,(\mathbf{q}_1, t) &= \mathrm{SNR}_{\mathrm{input}}\ (\mathbf{q}_1)\, e^{2\lambda_1 t}\,(-2\lambda_1)\,.\end{aligned}$$

For the second mode these are given by

$$\begin{aligned}\mathbf{q}_2 \cdot \mathbf{x} &= \int_{-\infty}^{t} d\tau e^{\lambda_2(t-\tau)} \left[\mathbf{q}_2 \cdot \mathbf{u}(s,\tau) \right. \\ &\left. + \omega \int_{-\infty}^{\tau} d\tau' e^{\lambda_1(\tau-\tau')} \left(\mathbf{q}_1 \cdot \mathbf{u}\left(s,\tau'\right)\right)\right] \\ \implies \mathbf{q}_2 \cdot \overline{\mathbf{x}}(s,t) &= \left[e^{\lambda_2 t}\left(\mathbf{q}_2 \cdot \mathbf{u}(s)\right) + \frac{\omega}{\lambda_1 - \lambda_2}\left(e^{\lambda_1 t} - e^{\lambda_2 t}\right)\left(\mathbf{q}_1 \cdot \mathbf{u}(s)\right)\right],\end{aligned}$$

and

$$\mathbf{q}_2 \cdot \Sigma_{\mathbf{x}(t)} \mathbf{q}_2 = -\left(\mathbf{q}_2 \cdot \Sigma_{\mathbf{n}} \mathbf{q}_2\right) \frac{1}{2\lambda_2} + \mathcal{O}(\omega) - \left(\mathbf{q}_1 \cdot \Sigma_{\mathbf{n}} \mathbf{q}_1\right) \frac{\omega^2}{2\lambda_1 \lambda_2 \left(\lambda_1 + \lambda_2\right)}$$

where we omitted the explicit form of the $\mathcal{O}(\omega)$ interaction term for brevity. In the non-normal limit $\omega \to \infty$, the response SNR along the output Schur mode becomes:

$$\begin{aligned}\mathrm{SNR}\left(\mathbf{q}_2, t\right) &= \frac{\left(\mathbf{q}_2 \cdot \Delta\overline{\mathbf{x}}(t)\right)^2}{\left(\mathbf{q}_2 \cdot \Sigma_{\mathbf{x}(t)} \mathbf{q}_2\right)} \\ &\approx -\mathrm{SNR}^0_{\mathrm{input}}\left(\mathbf{q}_1\right) \frac{2\lambda_1\lambda_2\left(\lambda_1 + \lambda_2\right)}{\left(\lambda_1 - \lambda_2\right)^2}\left(e^{\lambda_1 t} - e^{\lambda_2 t}\right)^2.\end{aligned}$$

This SNR behaves non-monotonically as a function of time, reaching a peak at a finite time before decaying back to zero. This non-monotonic timecourse is due to the transfer of information from the input to output mode over time, and the fact that the readout is restricted to the output mode - the SNR of the optimal (time-varying) readout decreases monotonically due to a shift in weighting of the readout from input to output mode over time (Section F.6).

For a given choice of eigenvalues, the SNR is maximal at the delay $t_d = \log(\lambda_2/\lambda_1)/(\lambda_1 - \lambda_2)$. How should the eigenvalues then be chosen in order to optimise response SNR at a particular delay t_d? Writing $\lambda_1 = \lambda$, $\lambda_2 = \lambda - \delta\lambda$ and taking the limit $\delta\lambda \to 0$, we have $t_d = \log(1 - \delta\lambda/\lambda)/\delta\lambda \approx -1/\lambda$ and

$$\begin{aligned}\mathrm{SNR}\left(\mathbf{q}_2, t\right) &= -\mathrm{SNR}^0_{\mathrm{input}}\left(\mathbf{q}_1\right) \frac{2\lambda(\lambda - \delta\lambda)(2\lambda - \delta\lambda)}{(\delta\lambda)^2}\left(e^{\lambda t}\left(1 - e^{-\delta\lambda t}\right)\right)^2 \\ &\approx -\mathrm{SNR}^0_{\mathrm{input}}\left(\mathbf{q}_1\right) 4\lambda^3 e^{2\lambda t} t^2 \\ &\leq \mathrm{SNR}^0_{\mathrm{input}}\left(\mathbf{q}_1\right) 4e^{-2} t_d^{-1} \\ &= \mathrm{SNR}_{\mathrm{normal}}\left(t_d\right) 4e^{-1}\end{aligned}$$

where the inequality gives the behaviour of a network with λ optimised for the delay t_d, and $\mathrm{SNR}_{\mathrm{normal}}\left(t_d\right) := \mathrm{SNR}^0_{\mathrm{input}}\left(\mathbf{q}_1\right)\left[e t_d\right]^{-1}$ is the SNR of a normal network optimised for readout time t_d. Compared to the optimal solution of the normal network with stationary state covariance, the response SNR in this non-normal solution is scaled by a factor of $4e^{-1} \approx 1.47$. We note the limits $\omega \to \infty$ and $\lambda_1 \approx \lambda_2$ taken above are consistent with the observed behaviour of networks during optimisation before transition to rotational dynamics (Supplementary Figure S3E-F).

F.5 Two-Dimensional Rotational Networks

We next considered a two-dimensional rotational network for which the two eigenvalues and eigenvectors of the weight matrix A form a complex conjugate pair λ, λ^* and $\mathbf{v}$, $\mathbf{v}^*$ (where $\mathbf{x}^* = \mathrm{Re}(\mathbf{x}) - i\mathrm{Im}(\mathbf{x})$ is complex conjugation). Using the results of Section F.3, the mean and stationary state covariance of the network response can be computed in the real Schur basis as

$$\begin{aligned}
\Delta\overline{\mathbf{x}}(t) &= e^{At}\Delta\mathbf{u} = Qe^{Tt}Q^{\top}\Delta\mathbf{u} = Qe^{Tt}\Delta\tilde{\mathbf{u}} \\
&= e^{\lambda_r t}Q\begin{bmatrix}\cos(\lambda_\omega t)\,\Delta\tilde{u}_1 + \epsilon\sin(\lambda_\omega t)\,\Delta\tilde{u}_2 \\ -\frac{1}{\epsilon}\sin(\lambda_\omega t)\,\Delta\tilde{u}_1 + \cos(\lambda_\omega t)\,\Delta\tilde{u}_2\end{bmatrix}, \\
\Sigma_{\mathbf{x}(t)} &= \int_0^\infty e^{At}\Sigma_{\mathbf{n}}e^{A^{\top}t}dt \\
&= \frac{1}{2}\sigma_{\mathbf{n}}^2 Q\int_0^\infty e^{2\lambda_r t} \\
&\begin{bmatrix}(1+\epsilon^2)+(1-\epsilon^2)\cos(2\lambda_\omega t), & \left(\epsilon-\frac{1}{\epsilon}\right)\sin(2\lambda_\omega t) \\ \left(\epsilon-\frac{1}{\epsilon}\right)\sin(2\lambda_\omega t), & \left(1+\frac{1}{\epsilon^2}\right)+\left(1-\frac{1}{\epsilon^2}\right)\cos(2\lambda_\omega t)\end{bmatrix}dtQ^{\top} \\
&= \frac{1}{4}\sigma_{\mathbf{n}}^2 Q\begin{bmatrix}-\frac{(1+\epsilon^2)}{\lambda_r}-(1-\epsilon^2)\frac{\lambda_r}{\lambda_r^2+\lambda_\omega^2}, & \left(\epsilon-\frac{1}{\epsilon}\right)\frac{\lambda_\omega}{\lambda_r^2+\lambda_\omega^2} \\ \left(\epsilon-\frac{1}{\epsilon}\right)\frac{\lambda_\omega}{\lambda_r^2+\lambda_\omega^2}, & -\frac{\left(1+\frac{1}{\epsilon^2}\right)}{\lambda_r}-\left(1-\frac{1}{\epsilon^2}\right)\frac{\lambda_r}{\lambda_r^2+\lambda_\omega^2}\end{bmatrix}Q^{\top} \\
&= -\frac{\sigma_{\mathbf{n}}^2}{4\lambda_r\left(\lambda_r^2+\lambda_\omega^2\right)}Q\begin{bmatrix}2\lambda_r^2+(1+\epsilon^2)\lambda_\omega^2, & -(\epsilon-\epsilon^{-1})\lambda_r\lambda_\omega \\ -(\epsilon-\epsilon^{-1})\lambda_r\lambda_\omega, & 2\lambda_r^2+(1+\epsilon^{-2})\lambda_\omega^2\end{bmatrix}Q^{\top},
\end{aligned}$$

where we defined $\tilde{\mathbf{u}} = Q^{\top}\mathbf{u}$ and assumed that $\Sigma_{\mathbf{n}} = \sigma_{\mathbf{n}}^2 I$.

The covariance matrix can be inverted directly as

$$\begin{aligned}
\det\left(\Sigma_{\mathbf{x}(t)}\right) &= \frac{\sigma_{\mathbf{n}}^4}{16}\left[\frac{2+\epsilon^2+\epsilon^{-2}}{\lambda_r^2}+\frac{2-\epsilon^2-\epsilon^{-2}}{\lambda_r^2+\lambda_\omega^2}\right] \\
\Sigma_{\mathbf{x}(t)}^{-1} &= -\frac{4\lambda_r}{\sigma_{\mathbf{n}}^2\left(4\lambda_r^2+(2+\epsilon^2+\epsilon^{-2})\lambda_\omega^2\right)}Q \\
&\begin{bmatrix}(1+\epsilon^{-2})\lambda_\omega^2+2\lambda_r^2, & (\epsilon-\epsilon^{-1})\lambda_r\lambda_\omega \\ (\epsilon-\epsilon^{-1})\lambda_r\lambda_\omega, & (1+\epsilon^2)\lambda_\omega^2+2\lambda_r^2\end{bmatrix}Q^{\top}.
\end{aligned}$$

The SNR of the rotational network takes on a simple form when the input linear discriminant is aligned to the one of the Schur modes. In particular, setting $\Delta\tilde{u}_1 = 0$, the SNR is

$$\begin{aligned}
\mathrm{SNR}(t) =& \Delta\overline{\mathbf{x}}^{\top}\Sigma_{\mathbf{x}(t)}^{-1}\Delta\overline{\mathbf{x}} \\
=& -\frac{(\Delta\tilde{u}_2)^2}{\sigma_{\mathbf{n}}^2}\left[\lambda_r^2+\frac{1}{4}(2+\epsilon^2+\epsilon^{-2})\lambda_\omega^2\right]^{-1}\lambda_r e^{2\lambda_r t} \\
&\times\left[(1+\epsilon^2)(\lambda_\omega^2+\lambda_r^2)+(1-\epsilon^2)\lambda_r^2\cos(2\lambda_\omega t)\right. \\
&\left.-(1-\epsilon^2)\lambda_\omega\lambda_r\sin(2\lambda_\omega t)\right] \\
=& -\frac{(\Delta\tilde{u}_2)^2}{\sigma_{\mathbf{n}}^2}\left[\lambda_r^2+\frac{1}{4}(2+\epsilon^2+\epsilon^{-2})\lambda_\omega^2\right]^{-1}\lambda_r e^{2\lambda_r t} \\
&\times\left[(1+\epsilon^2)(\lambda_\omega^2+\lambda_r^2)\right. \\
&\left.+(1-\epsilon^2)\lambda_r\sqrt{\lambda_r^2+\lambda_\omega^2}\cos\left(2\lambda_\omega t+\arctan\left(\frac{\lambda_\omega}{\lambda_r}\right)\right)\right] \\
=& \mathrm{SNR}_{\mathrm{normal}}(t)\times \\
&\begin{cases}1, & \text{if } \epsilon = 1 \\ 0, & \text{if } \epsilon \to 0 \\ 2\left[1+\frac{\lambda_r^2}{\lambda_\omega^2}-\sqrt{\frac{\lambda_r^2}{\lambda_\omega^2}+\frac{\lambda_r^4}{\lambda_\omega^4}}\cos\left(2\lambda_\omega t+\arctan\left(\frac{\lambda_\omega}{\lambda_r}\right)\right)\right], & \text{if } \epsilon \to \infty\end{cases}
\end{aligned}$$

The SNR when $\epsilon = 1$ is identical to that of a normal network with real eigenvalue $\lambda = \lambda_r$ (given by $\mathrm{SNR}_{\mathrm{normal}}(t) = (-2\lambda_r)\,e^{2\lambda_r t}(\Delta\tilde{u}_2)^2/\sigma_{\mathbf{n}}^2$), independent of the rotational frequency λ_ω. When $\epsilon \to \infty$, the SNR derived above matches that observed in optimised networks (Supplementary Figure S8B). This case corresponds to a highly elliptical rotation, with the input linear discriminant aligned to the minor axis of the ellipse. Indeed, when optimising numerically (without an energy

penalty), we found that $\epsilon \to \infty$ with increasing numbers of iterations, albeit very slowly and with only marginally increasing SNR. The third limit, $\epsilon \to 0$, corresponds to a highly elliptical rotation with the input discriminant aligned to the major axis of the ellipse. In this case, noise input to the minor axis of the ellipse is strongly amplified, but the input signal is damped, leading to low SNR.

F.6 N-Dimensional Functionally-Feedforward Networks

We next considered delay line architectures, in which multiple feedforward modes are chained together in sequence (Ganguli et al., 2008; Goldman, 2009). Previous studies have shown that such architectures maximise the SNR of discrete-time linear dynamical systems, provided the number of neurons is equal to or greater than the number of time steps in the delay (Ganguli et al., 2008). However, the optimal dynamics for continuous-time networks, or for discrete-time networks with more time steps than neurons, was not determined. While Goldman (2009) considered a continuous-time network and showed that a delay line architecture can generate responses with memory time constant proportional to the number of neurons, they did not consider noise robustness or show that such architectures are optimal. We derive analytical expressions for the signal-to-noise ratio of a continuous-time delay line and show that, although SNR scales with the number of neurons, it is outperformed by the rotational solutions obtained via numerical optimisation (Supplementary Figure S1, Figure 3F).

F.6.1 Mean and Covariance of Network Response

We considered networks with $A_{ij} = \lambda\delta_{ij} + \omega\delta_{i,j-1}$ (we choose a lower-triangular rather than upper-triangular decomposition for this analysis as it leads to simpler indices in the following derivations). This network is known as a homogeneous delay line Ganguli et al. (2008). In this basis, the network is structurally feedforward with each element i corresponding to a single neuron. However, a rotation of the connectivity matrix $A \to QAQ^\top$ produces a functionally-feedforward network for which the feedforward structure acts on Schur modes rather than single neurons.

The matrix exponential for the homogeneous delay line can be computed directly from its power series, $e^{At} = \sum_{k=0}^{\infty} A^k/k! \implies \left[e^{tA}\right]_{ij} = \delta_{i\geq j}\frac{(\omega t)^{(i-j)}}{(i-j)!}e^{\lambda t}$, where $\delta_{i\geq j}$ is 1 for $i \geq j$ and zero otherwise. If the network is initialised in the state $x_i(0) = \delta_{i1}$, then in the absence of input the response propagates down the delay line as $x_i(t) = \frac{(\omega t)^{(i-1)}}{(i-1)!}e^{\lambda t}$. Thus, if the stimulus inputs $\mathbf{u}(s, t) = \mathbf{u}(s)\delta(t) + \mathbf{n}(t)$ are such that $\Delta u = \|\Delta\mathbf{u}\|\delta_{i1}$, the trial-averaged responses follow $\Delta\bar{x}_i(t) = \|\Delta\mathbf{u}\|\frac{(\omega t)^{(i-1)}}{(i-1)!}e^{\lambda t}$. The station-ary state covariance matrix for the delay line driven by Gaussian white noise input $\Sigma_{\mathbf{x}} = \int_0^\infty e^{At}\sigma_{\mathbf{n}}^2 I e^{A^\top t}dt \implies (\Sigma_{\mathbf{x}})_{ij} = \sigma_{\mathbf{n}}^2\int_0^\infty \sum_{k=1}^{\min(i,j)}\frac{(\omega t)^{i+j-2k}}{(i-k)!(j-k)!}e^{2\lambda t}dt$
$= \frac{\sigma_{\mathbf{n}}^2}{-2\lambda}\sum_{k=1}^{\min(i,j)}\frac{(i+j-2k)!}{(i-k)!(j-k)!}\left(\frac{\omega}{-2\lambda}\right)^{i+j-2k}$.

F.6.2 Response SNR of Individual Schur Modes

In this section, we place a tractable lower bound on the total response SNR by computing the SNR of the ith Schur mode. Writing $\mathrm{SNR}(i,t) := (\Delta\bar{x}_i(t))^2/(\Sigma_{\mathbf{x}})_{ii}$, we have

$$\mathrm{SNR}(i,t) = \mathrm{SNR}_{\mathrm{input}}^0\,(-2\lambda)e^{2\lambda t}\frac{(\omega t)^{2(i-1)}}{[(i-1)!]^2}\left[\sum_{k=1}^{i}\frac{(2(i-k))!}{[(i-k)!]^2}\left(\frac{\omega}{-2\lambda}\right)^{2(i-k)}\right]^{-1}$$

In the limit $\omega \to \infty$ the first term in the sum dominates, giving

$$\mathrm{SNR}(i,t) = \frac{(-2\lambda t)^{2(i-1)}}{(2(i-1))!}\underbrace{\mathrm{SNR}_{\mathrm{input}}^0\,(-2\lambda)e^{2\lambda t}}_{:=\mathrm{SNR}_{\mathrm{normal}}(t)}\overset{i!\approx\sqrt{2\pi i}(i/e)^i}{\approx}\frac{1}{2\sqrt{\pi(i-1)}}\left(\frac{-e\lambda t}{i-1}\right)^{2(i-1)}\mathrm{SNR}_{\mathrm{normal}}(t)$$

where $\mathrm{SNR}_{\mathrm{normal}}(t)$ is the SNR of a normal network with a left eigenvector aligned to the input discriminant and Stirling's approximation was used for $i \gg 1$. This SNR is maximised

for a delay t_d when $\lambda = -(i - 1/2)/t_d$, with

$$\begin{aligned}\mathrm{SNR}(i,t) &= \frac{\left(2(i-1/2)t/t_d\right)^{2(i-1)}}{(2(i-1))!}\mathrm{SNR}_{\mathrm{normal}}(t)\\ &\quad i! \approx \sqrt{2\pi i}(i/e)^i \frac{1}{2\sqrt{\pi(i-1)}}\left(\frac{i-1/2}{i-1}\frac{et}{t_d}\right)^{2(i-1)}\mathrm{SNR}_{\mathrm{normal}}(t)\\ &\overset{i>\gtrapprox 1}{\approx} \frac{1}{2\sqrt{\pi i}}\left(\frac{et}{t_d}\right)^{2(i-1)}\mathrm{SNR}_{\mathrm{normal}}(t)\\ &\overset{t=t_d}{=} \sqrt{\frac{i}{\pi}}(et_d)^{-1}\,\mathrm{SNR}^0_{\mathrm{input}}\ .\end{aligned}$$

Thus, although the optimal response SNR decays as $1/t_d$, it also grows with the number of neurons as $\sqrt{N}$, and an arbitrarily high SNR can be achieved for any delay t_d by increasing the length of the delay line. Note that the optimal time constant decreases as a function of the length of the delay line as as $\tau = -1/\lambda \sim t_d/N$.

F.6.4 Response SNR of Optimal Population Readout

The previous section considered the performance of a single-mode readout, which is in general suboptimal. We next derive a general expression for the SNR of the optimal readout of the delay line, and calculate this SNR explicitly for the two-dimensional case.

The SNR of the network response at time t along a readout $\mathbf{w}$ is

$$\begin{aligned}\mathrm{SNR}(\mathbf{w},t) &= \frac{(\mathbf{w}\cdot\Delta\overline{\mathbf{x}}(t))^2}{\mathbf{w}\cdot\Sigma_{\mathbf{x}}\mathbf{w}}\\ &= \mathrm{SNR}^0_{\mathrm{input}}\,e^{2\lambda t}(-2\lambda)\frac{\left[\sum_{i=1}^N w_i\frac{(\omega t)^{(i-1)}}{(i-1)!}\right]^2}{\sum_{i,j=1}^N w_i w_j \sum_{k=1}^{\min(i,j)}\frac{(i+j-2k)!}{(i-k)!(j-k)!}\left(\frac{\omega}{-2\lambda}\right)^{i+j-2k}}\end{aligned}$$

This can be written as:

$$\mathrm{SNR}(\mathbf{w},t) = \mathrm{SNR}^0_{\mathrm{input}}(-2\lambda)e^{2\lambda t}\frac{(\mathbf{w}\cdot\mathbf{m}(t))^2}{\mathbf{w}\cdot P\mathbf{w}}$$

where $m_i(t) = \frac{(\omega t)^{i-1}}{(i-1)!}$ and $P_{ij} = \sum_{k=1}^{\min(\mathrm{i,j})}\frac{(i+j-2k)!}{(i-k)!(j-k)!}\left(\frac{\omega}{-2\lambda}\right)^{i+j-2k}$. The SNR is maximised using the linear discriminant readout $\mathbf{w} \propto P^{-1}\mathbf{m}(t)$, which gives

$$\mathrm{SNR}(t) = \mathrm{SNR}_{\mathrm{normal}}(t)\mathbf{m}(t)\cdot P^{-1}\mathbf{m}(t)$$

Thus, the factor $\mathbf{m}(t)\cdot P^{-1}\mathbf{m}(t)$ measures the gain in performance introduced by non-normal integration.

To determine the behaviour of this SNR in the large ω limit, we define $\tilde{w}_i = w_i\left(\frac{\omega}{-2\lambda_i}\right)\frac{1}{(i-1)!}$, which gives

$$\begin{aligned}&\mathrm{SNR}(\tilde{\mathbf{w}},t)\\ &= \mathrm{SNR}_{\mathrm{normal}}(t)\frac{\left[\sum_i \tilde{w}_i(-2\lambda t)^{i-1}\right]^2}{\sum_{i,j}\tilde{w}_i\tilde{w}_j\sum_{k=1}^{\min(i,j)}\frac{(i+j-2k)!(i-1)!(j-1)!}{(i-k)!(j-k)!}\left(\frac{\omega}{-2\lambda_i}\right)^{2(1-k)}}\end{aligned}$$

In the limit $\omega \to \infty$, this becomes

$$\begin{aligned}\mathrm{SNR}(\tilde{\mathbf{w}},t) &\approx \mathrm{SNR}_{\mathrm{normal}}(t)\frac{\left[\sum_i \tilde{w}_i(-2\lambda t)^{i-1}\right]^2}{\sum_{i,j}\tilde{w}_i\tilde{w}_j(i+j-2)!}\\ &\leq \mathrm{SNR}_{\mathrm{normal}}(t)\tilde{\mathbf{m}}(t)\cdot\tilde{P}^{-1}\tilde{\mathbf{m}}(t),\end{aligned} \tag{17}$$

where $\tilde{m}_i(t) = (-2\lambda t)^{i-1}$ and $\tilde{P}_{ij} = (i+j-2)!$ and equality is achieved by setting $\tilde{\mathbf{w}} \propto \tilde{P}^{-1}\tilde{\mathbf{m}}$.

Two-Dimensional Case

For a two-dimensional network, $\tilde{P}^{-1} = [2,-1;-1,1]$, so that $\mathrm{SNR}(\tilde{\mathbf{w}},t) = \mathrm{SNR}_{\mathrm{normal}}(t)\left(2+4\lambda t+4\lambda^2t^2\right) = -t^{-1}e^{\beta}\left(2\beta+2\beta^2+\beta^3\right)$ where $\beta = 2\lambda t$ and we set $\mathrm{SNR}^0_{\mathrm{input}} = 1$. The extrema of this SNR for a given delay t_d are found by solving $\frac{d\,\mathrm{SNR}(\tilde{\mathbf{w}},t)}{d\beta} = 0$, which gives $\beta^3 + 4\beta^2 + 4\beta + 2 = 0$. This equation has three real roots at $\beta = -1, \beta = -2\pm\sqrt{2}$. The first root ($\beta = 1$) has $\mathrm{SNR}(\tilde{\mathbf{w}},t_d) = (et_d)^{-1}$, which is equal to the SNR of a normal network optimised for t_d. The second and third roots $(\beta = -2\pm\sqrt{2})$ have $\mathrm{SNR}(\tilde{\mathbf{w}},t_d) = (et_d)^{-1}e^{-1\pm\sqrt{2}}4(3\mp2\sqrt{2})$. The root $\beta = -2+\sqrt{2}$ has $\mathrm{SNR}(\tilde{\mathbf{w}},t_d) \approx 1.04(et_d)^{-1}$, while the root $\beta = -2-\sqrt{2}$ has $\mathrm{SNR}(\tilde{\mathbf{w}},t_d) \approx 2.09(et_d)^{-1}$.

We numerically plot the solution to Equation (17) for different values of N (with optimal λ) in Figure 3F), finding an approximately linear scaling of response SNR with the number of neurons N.

F.7 Encoding Multidimensional Stimulus Spaces Using Orthogonal Subspaces

For tasks in which multiple stimuli $s \in \{s_0, s_2, \ldots, s_{M-1}\}$ can be presented, we considered the SNR for each stimulus pair SNR(s_i, s_j, t). Here, we show how networks optimised for discrimination of two stimuli can naturally be composed to form networks that discriminate arbitrarily many stimuli within a P-dimensional space with SNR for any stimulus pair equal to that of an N-dimensional network optimised exclusively for that stimulus pair. The assumption that inputs span a linear subspace is true in particular for the case of cosine-tuned inputs considered in the main text (where $P = 2$), but we present results for the general case here.

– ∝ ∈

To construct such a network, we start with an N-dimensional network optimised for discrimination of two stimuli s_1, s_2, with $A = QTQ^{\intercal}$ and $\mathbf{u}(s_2)\,\mathbf{u}(s_1)\,\mathbf{b}\,\mathbb{R}^N$. We then use this network to construct a network that discriminates a set of input stimuli that span a P-dimensional subspace of a PN-dimensional space, i.e. $\tilde{\mathbf{u}}(s) = \sum_{p=0}^{P-1}\tilde{c}_p(s)\tilde{\mathbf{b}}_p$ with $\tilde{\mathbf{b}}_p \in \mathbb{R}^{PN}$. In particular, we set $\tilde{A} = \tilde{Q}\tilde{T}\tilde{Q}^{\top}$ with

$$\tilde{T} = \begin{bmatrix} T & 0 & \ldots & 0 \\ 0 & T & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & T \end{bmatrix}$$

and with $\tilde{Q} = \left[\tilde{Q}_0, \ldots, \tilde{Q}_{P-1}\right]$ with $\tilde{Q}_p^{\top}\tilde{\mathbf{b}}_q = Q^{\top}\mathbf{b}\delta_{pq}$ and $\tilde{Q}_p^{\top}\tilde{Q}_q = I\delta_{pq}$. Then we have:

$$e^{\tilde{A}t} = \tilde{Q}e^{\tilde{T}t}\tilde{Q}^{\top} = \tilde{Q}\begin{bmatrix} e^{Tt} & 0 & \cdots & 0 \\ 0 & e^{Tt} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{Tt} \end{bmatrix}\tilde{Q}^{\top} = \sum_{p=0}^{P-1}\tilde{Q}_p e^{Tt}\tilde{Q}_p^{\top}$$

We can then compute the signal-to-noise ratio for any stimulus pair

$$
\begin{aligned}
\mathrm{SNR}\left(s_i, s_j, t\right) &= \left(\tilde{\mathbf{u}}\left(s_i\right) - \tilde{\mathbf{u}}\left(s_j\right)\right)^{\top} e^{\tilde{A}^{\top} t} \left[\sigma_{\mathbf{n}}^2 \int_0^{\infty} e^{\tilde{A}\tau} e^{\tilde{A}^{\top}\tau} d\tau\right]^{-1} e^{\tilde{A}t} \\
&\quad \left(\tilde{\mathbf{u}}\left(s_i\right) - \tilde{\mathbf{u}}\left(s_j\right)\right) \\
&= \sigma_{\mathbf{n}}^{-2} \left(\tilde{\mathbf{u}}\left(s_i\right) - \tilde{\mathbf{u}}\left(s_j\right)\right)^{\top} \tilde{Q} e^{\tilde{T}^{\top} t} \left[\int_0^{\infty} e^{\tilde{T}\tau} e^{\tilde{T}^{\top}\tau} d\tau\right]^{-1} e^{\tilde{T}t} \tilde{Q}^{\top} \\
&\quad \left(\tilde{\mathbf{u}}\left(s_i\right) - \tilde{\mathbf{u}}\left(s_j\right)\right) \\
&= \sum_{p=0}^{P-1} \left[\left(\tilde{\mathbf{u}}\left(s_i\right) - \tilde{\mathbf{u}}\left(s_j\right)\right) \cdot \tilde{\mathbf{b}}_p\right]^2 \underbrace{\mathrm{SNR}_{2-\mathrm{Stim}}\left(s_i, s_j\right)}_{\mathbf{b}^{\top} Q e^{T^{\top} t} \left[\sigma_{\mathbf{n}}^2 \int_0^{\infty} e^{T\tau} e^{T^{\top}\tau} d\tau\right]^{-1} e^{Tt} Q \mathbf{b}} \\
&= \mathrm{SNR}_{2-\mathrm{Stim}}\left(s_i, s_j\right) \frac{\sum_{p=0}^{P-1} \left(\left(\tilde{\mathbf{u}}\left(s_i\right) - \tilde{\mathbf{u}}\left(s_j\right)\right) \cdot \tilde{\mathbf{b}}_p\right)^2}{\left\|\tilde{\mathbf{u}}\left(s_i\right) - \tilde{\mathbf{u}}\left(s_j\right)\right)\right\|^2} \\
&= \mathrm{SNR}_{2-\mathrm{Stim}}\left(s_i, s_j\right),
\end{aligned}
$$

where $\mathrm{SNR}_{2\text{-}\mathrm{Stim}}(s_i, s_j)$ is the SNR of the original two-stimulus network A for discrimination of a pair of inputs with $\mathbf{u}\left(s_2\right) - \mathbf{u}\left(s_1\right) = \left\|\tilde{\mathbf{u}}\left(s_i\right) - \tilde{\mathbf{u}}\left(s_j\right)\right)\right\| \mathbf{b}$.

Thus, by aligning an orthogonal subspace to each stimulus dimension $\tilde{\mathbf{b}}_p$, networks optimised for discrimination of a single stimulus pair can be naturally composed to form networks that discriminate arbitrarily many stimuli drawn from a P-dimensional space, with the number of neurons scaling only linearly with P.

G Targeted Dimensionality Reduction

Given a set of measured responses $\mathbf{x}^{(k)}(t)$ of N neurons, over K trials with stimuli $\mathrm{stim}(k) = \theta^{(k)}$, and T time points per trial, we fit the model:

$$
\mathbf{x}^{(k)}(t) = \sum_{p=1}^{P} \mathbf{w}_p(t) f_p\left(\theta^{(k)}\right) + \mathbf{w}_0(t)
$$

where $\mathbf{w}_p(t)$ are neuron- and time-dependent factors and $f_p(\theta^{(k)})$ are stimulus-dependent factors. This equation can be written in matrix form as:

$$
X = WF
$$

where

$$
X = \begin{bmatrix}
\mathbf{x}^{(1)}(1) & \mathbf{x}^{(2)}(1) & \ldots & \mathbf{x}^{(K)}(1) \\
\mathbf{x}^{(1)}(2) & \mathbf{x}^{(2)}(2) & \ldots & \mathbf{x}^{(K)}(2) \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{x}^{(1)}(T) & \mathbf{x}^{(2)}(T) & \ldots & \mathbf{x}^{(K)}(T)
\end{bmatrix},
$$

$$
F = \begin{bmatrix}
f_1\left(\theta^{(1)}\right) & f_1\left(\theta^{(2)}\right) & \ldots & f_1\left(\theta^{(K)}\right) \\
f_2\left(\theta^{(1)}\right) & f_2\left(\theta^{(2)}\right) & \ldots & f_2\left(\theta^{(K)}\right) \\
\vdots & \vdots & \ddots & \vdots \\
f_P\left(\theta^{(1)}\right) & f_P\left(\theta^{(2)}\right) & \ldots & f_P\left(\theta^{(K)}\right) \\
1 & 1 & \ldots & 1
\end{bmatrix},
$$

$$
W = \begin{bmatrix}
\mathbf{w}_1(1) & \mathbf{w}_2(1) & \ldots & \mathbf{w}_P(1) & \mathbf{w}_0(1) \\
\mathbf{w}_1(2) & \mathbf{w}_2(2) & \ldots & \mathbf{w}_P(2) & \mathbf{w}_0(2) \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{w}_1(T) & \mathbf{w}_2(T) & \ldots & \mathbf{w}_P(T) & \mathbf{w}_0(T)
\end{bmatrix}
$$

Thus, given an estimate of the set of P vectors $\mathbf{f}_p \in \mathbb{R}^{N_{\text{stim}}}$, a least squares estimate of the factors $\mathbf{w}_p(t)$ can be obtained by solving the above linear equation via the Moore-Penrose pseudoinverse $W = XF^{+}$.

To solve for $f_p(\theta)$ given estimates of $\mathbf{w}_p(t)$, note that:

$$\bar{X} = W_{:,1:P}\bar{F}_{1:P,:} + W_0\mathbf{1}^\top$$

where $\bar{X} \in \mathbb{R}^{NT\times N_{\text{stim}}}$ contains the PSTHs $\left\langle \mathbf{x}^{(k)}(t)\right\rangle_{\text{stim}(k)=\theta}$, $\bar{F}_{p,\theta} = f_p(\theta)$ and $\mathbf{1} \in \mathbb{R}^{N_{\text{stim}}}$ is a vector of ones (with $1 \le \theta \le N_{\text{stim}}$ the stimulus labels). Thus, we obtain $\bar{F}_{1:P,:} = W^{+}\left(\bar{X} - W_0\mathbf{1}^\top\right)$. To fit the TDR model to data, we iterated between these two updates until convergence, initialising with $f_p(\theta) \sim N(0, 1)$ (for $1 \le p \le P$ and $1 \le \theta \le N_{\text{stim}}$).

The TDR model has a number of invariances that we exploited to convert the model into a standard form. First, the stimulus factors can be written as $f_p(\theta) = \bar{f}_p + \delta f_p$ where $\bar{f}_p = \frac{1}{N_{\text{stim}}}\sum_{\theta=1}^{N_{\text{stim}}} f_p(\theta)$ is the mean over stimuli. We removed the mean component for all p by setting $\mathbf{w}_0(t) \to \mathbf{w}_0(t) + \sum_{p=1}^{P}\mathbf{w}_p(t)\bar{f}_p$.Second, there exists an invariance $W_{:,1:P} \to W_{:,1:P}M$, $\bar{F}_{1:P,:} \to M^{-1}\bar{F}_{1:P,:}$ for arbitrary invertible $M \in \mathbb{R}^{P\times P}$. We selected a unique basis by taking the singular value decomposition $\bar{F}_{1:P,:} = USV^\top$ and redefining $\bar{F}_{1:P,:} \to V^\top$ and $W_{:,1:P} \to W_{:,1:P}US$ (note that this was called the Principal Component Basis by Aoi et al. (2020)). An advantage of this choice of basis is that the vectors $\mathbf{f}_p$ are orthonormal, and the stimulus-independent component $\mathbf{f}_0 = \mathbf{1}$ is also orthogonal (since all other vectors have zero mean, and the dot product of a constant vector with a zero mean vector is zero). Finally, for the circular stimulus set in the working memory task, $F_{1:P,:}$ had two leading principal components of roughly equal magnitude (the two basis vectors that spanned the circle), so that the Principal Component Basis was approximately invariant to a further rotation in this two-dimensional subspace. To ensure that all models (with $P = 2$) were plotted and compared in a comparable basis, we therefore performed an Orthogonal Procrustes transformation to obtain a 2×2 orthogonal matrix $R = \underset{R^\top R=I}{\operatorname{argmin}} \left\| R\bar{F}_{1:2,:} - [\cos\boldsymbol{\theta}, \sin\boldsymbol{\theta}]^\top\right\|^2$ and set $W_{1:2,:} \to W_{1:2,:}R^\top$ and $W_{1:2,:} \to W_{1:2,:}R^\top$.

Supplementary Figures

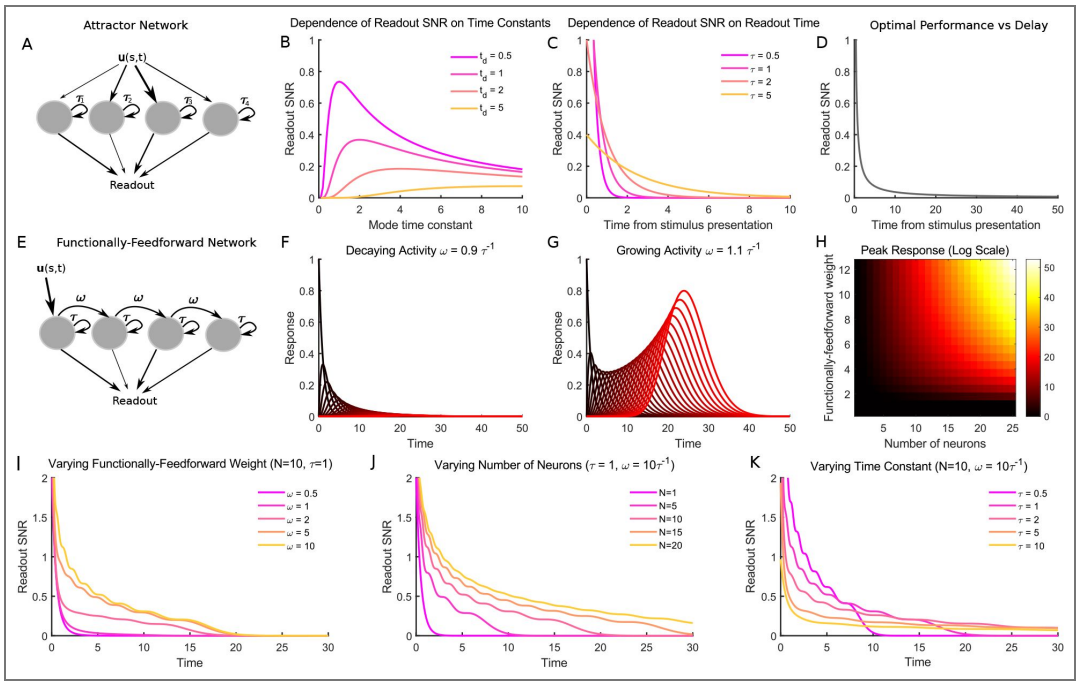

Figure S1. Behaviour of attractor (normal) and functionally-feedforward (non-normal) networks on the WM task with delta pulse stimulus input and stationary state covariance. A: Normal networks can be characterised by a set of independent eigenmodes. Each eigenmode applies an independent filter e^{-t/τ_i} to the projection of network input onto its left eigenvector $\tilde{\mathbf{v}}_i$. A downstream readout could in principle combine information from multiple such eigenmodes to obtain information about different projections of the recent history of sensory input over different timescales. B: A single eigenmode can be optimised for readout at a specific delay time t_d by setting its time constant as $\tau_k = 2t_d$. C: Readout performance along a single eigenmode decays exponentially after stimulus presentation. D: Even when optimised for a given delay, the readout performance decays as $(et_d)^{-1}$ for attractor networks. E: Functionally-feedforward networks are better characterised by a Schur decomposition, in which each mode has a decay time constant τ and a feedforward weight onto the next mode ω (also called a homogeneous delay line). F, G: Activity propagates along the delay line, and depending on the value of $\omega\tau$, can either decay ($\omega\tau < 1$, F) or grow ($\omega\tau > 1$, G) exponentially along the delay line. H: Despite being asymptotically stable when $\tau > 0$, activity can grow to arbitrarily large values as ω and N are increased. I-K: The performance of the functionally-feedforward network grows as ω and N are increased and τ is decreased, with SNR $\to \infty$ when $N, \omega \to \infty$ and $\tau \to 0$. Thus, the delay line can substantially outperform normal networks (A-D), but these improvements require an exponential increase in the firing rate of neurons to achieve (H).

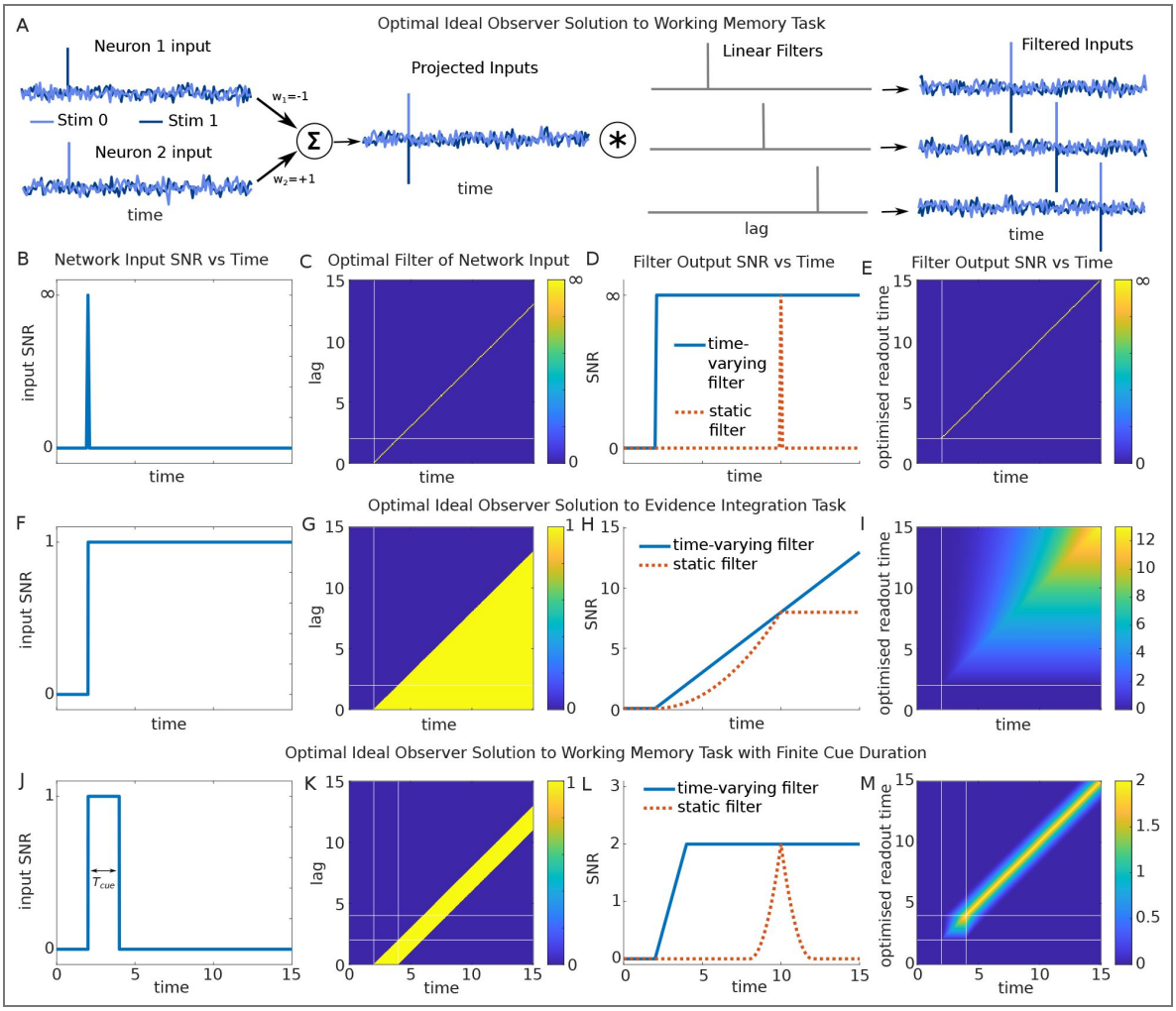

Figure S2. Bayesian deal observer solutions to WM and perceptual decision-making tasks.

A: Stimuli can be optimally discriminated based on network input by first projecting these inputs onto their linear discriminant, then convolving them with a delay filter matched to the readout time t_d and applying a threshold operation (not shown). B: Signal-to-noise ratio of network input vs time for the WM task with delta pulse stimulus input. C: A linear time-varying linear filter $f(t_d, \tau) = \delta(t_d - t_s - \tau)$ achieves the optimal response SNR. D: The SNR of filter output is infinite at the optimised readout time t_d. The optimal time-varying filter achieves perfect performance at all times $t > t_s$, while a static filter can achieve optimal performance only at the optimised delay t_d. E: The SNR of filter output vs time for static filters optimised for each readout time. F: In a perceptual decision-making task in which the stimulus is presented continuously, the SNR of network input is zero before stimulus onset and constant after stimulus onset, requiring temporal integration to achieve a high output SNR. G: The optimal time-varying filter for this task integrates the history of network input for all times $t > t_s$, but assigns zero weight to inputs at times $t < t_s$. H: Response SNR for the optimal time-varying filter grows linearly after stimulus onset, while an optimal static filter has bounded performance. J-M: When the stimulus appears for a fixed interval of time T_{cue}, the optimal filter integrates network input during the stimulus period (with an integration window T_{cue}) and then delays this input for a period $t_d - t_s - T_{cue}$.

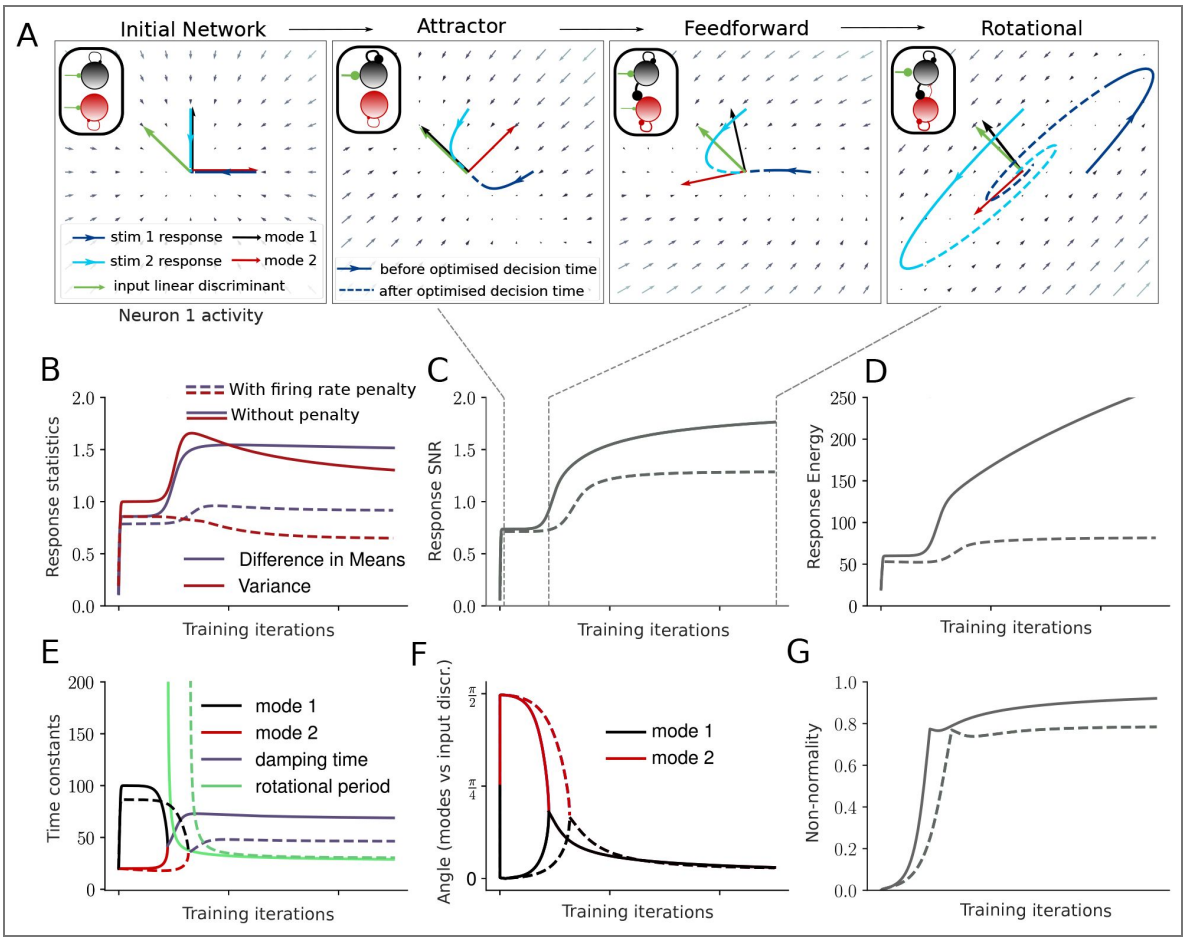

Figure S3. Influence of energetic penalty on learned dynamics.

A: As in Figure 2C, but showing the linear discriminant and two dynamical modes (Schur modes) at each stage of optimisation (without energy penalty). B: The mean separation and variance of responses along the readout vector at the decision time, at each iteration of the optimisation process for networks with and without a penalty on energetic cost. C: The SNR of responses for the two networks. D: The total energy of the network response (time-integrated squared norm of firing rate vector). E: The time constants of the two dynamical modes. Before the transition to rotational dynamics, the time constants are determined by the (real) eigenvalues. After the transition, they are given by the real and imaginary parts of a complex-conjugate pair of eigenvalues. F: The angle between each input mode (left eigenvector) and the input linear discriminant. After the transition to rotational dynamics the input Schur mode is shown. G: Non-normality of the dynamics matrix *A* (normalised between 0 and 1, see Methods).

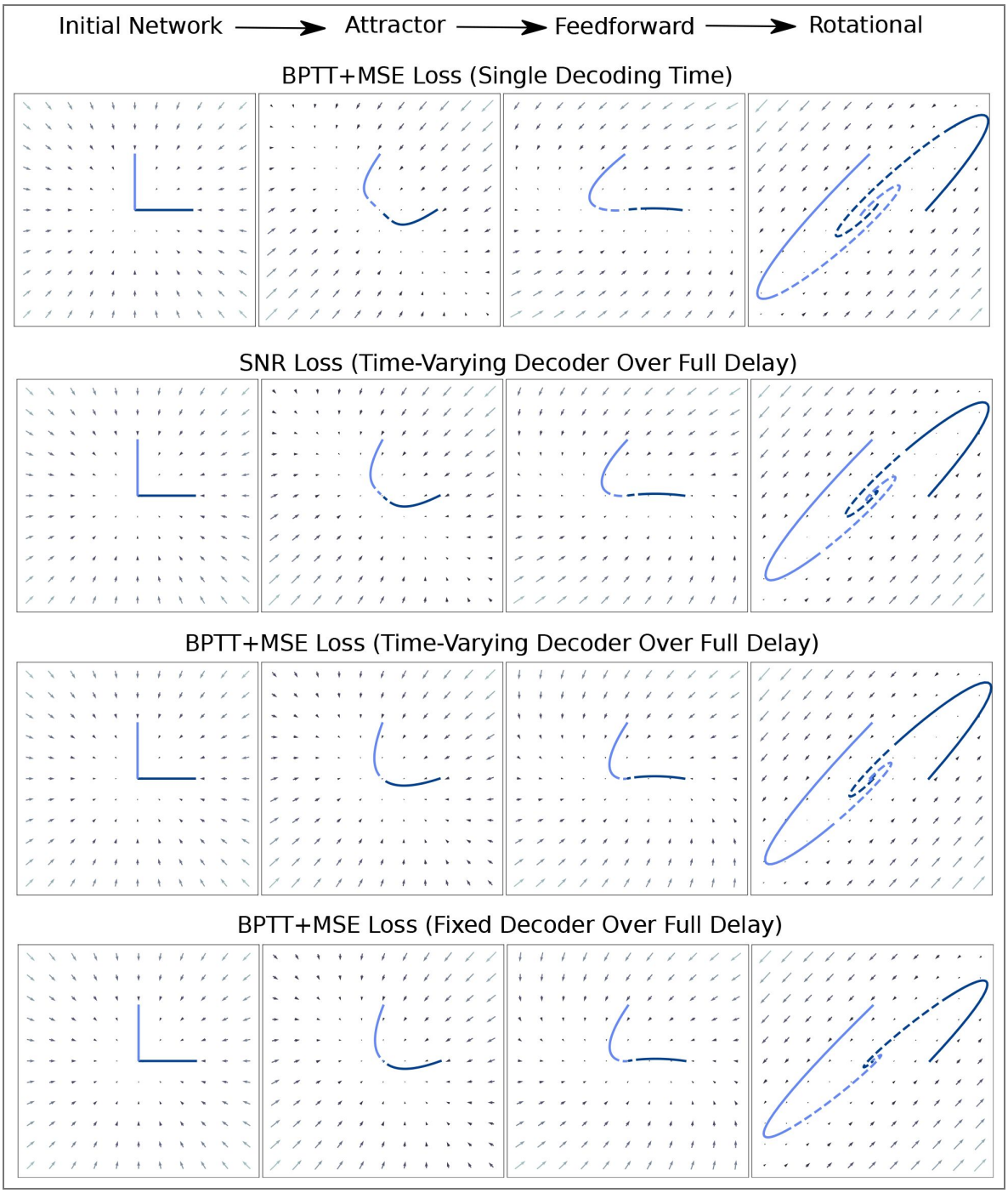

Figure S4. Comparison of networks optimised using 1) SNR loss vs backpropagation through time (BPTT) with mean squared error (MSE) loss, 2) loss evaluated at a single time point vs all time points over the delay, and 3) a fixed decoder vs a time-varying decoder.

Top row: Network optimised using BPTT with MSE loss at a single time point $t_d = 50$. Second row: Optimisation with the SNR loss, weighted equally at all delays $\left(L = \int_0^{50} 1/\,\mathrm{SNR}(t)dt\right)$. Third row: optimisation using BPTT+MSE with a separate linear decoder fit to each time point $t_d \in [0, 50]$. Bottom row: Optimisation using BPTT+MSE with a single decoder fit to minimise MSE across all time points $t_d \in [0, 50]$.

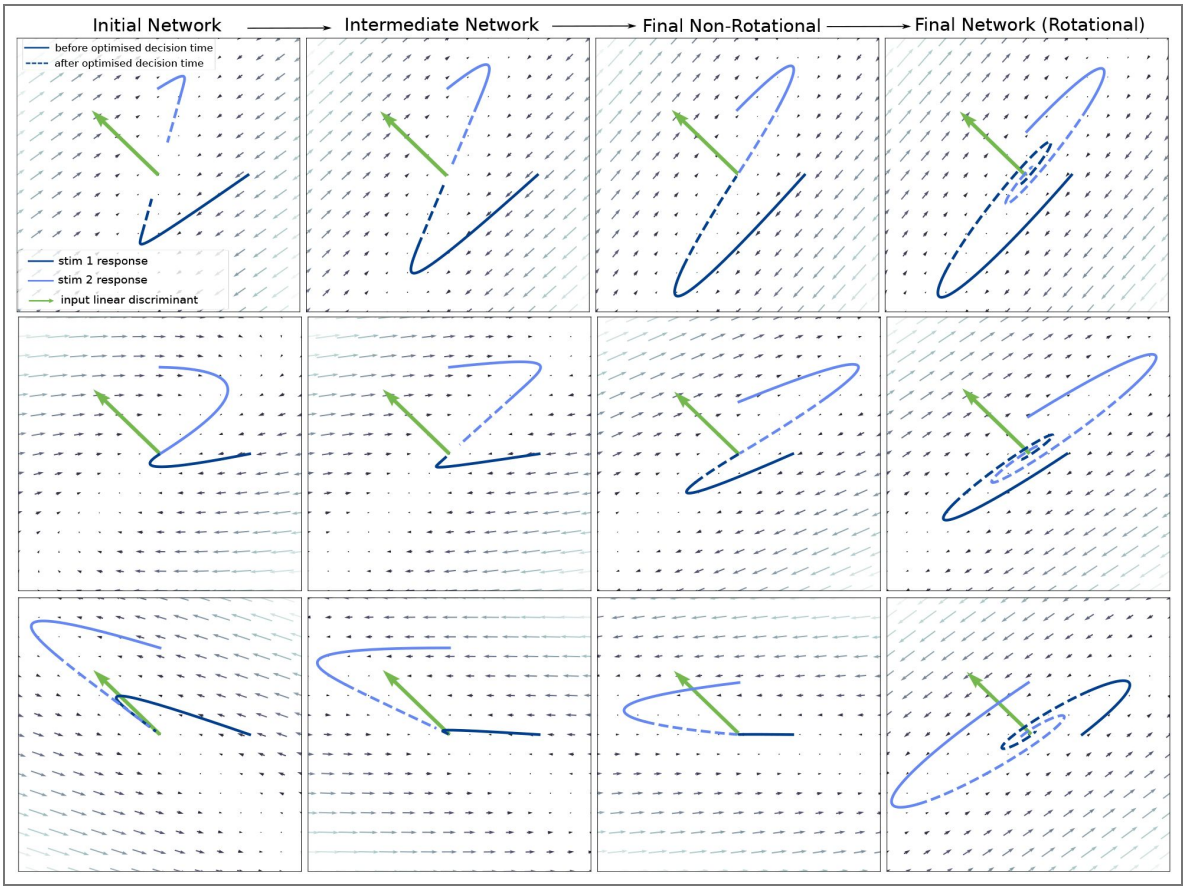

Figure S5. Rotational solutions are insensitive to initial network weights.

Each row shows four phases of the optimisation of a network with different random weight initialisation.

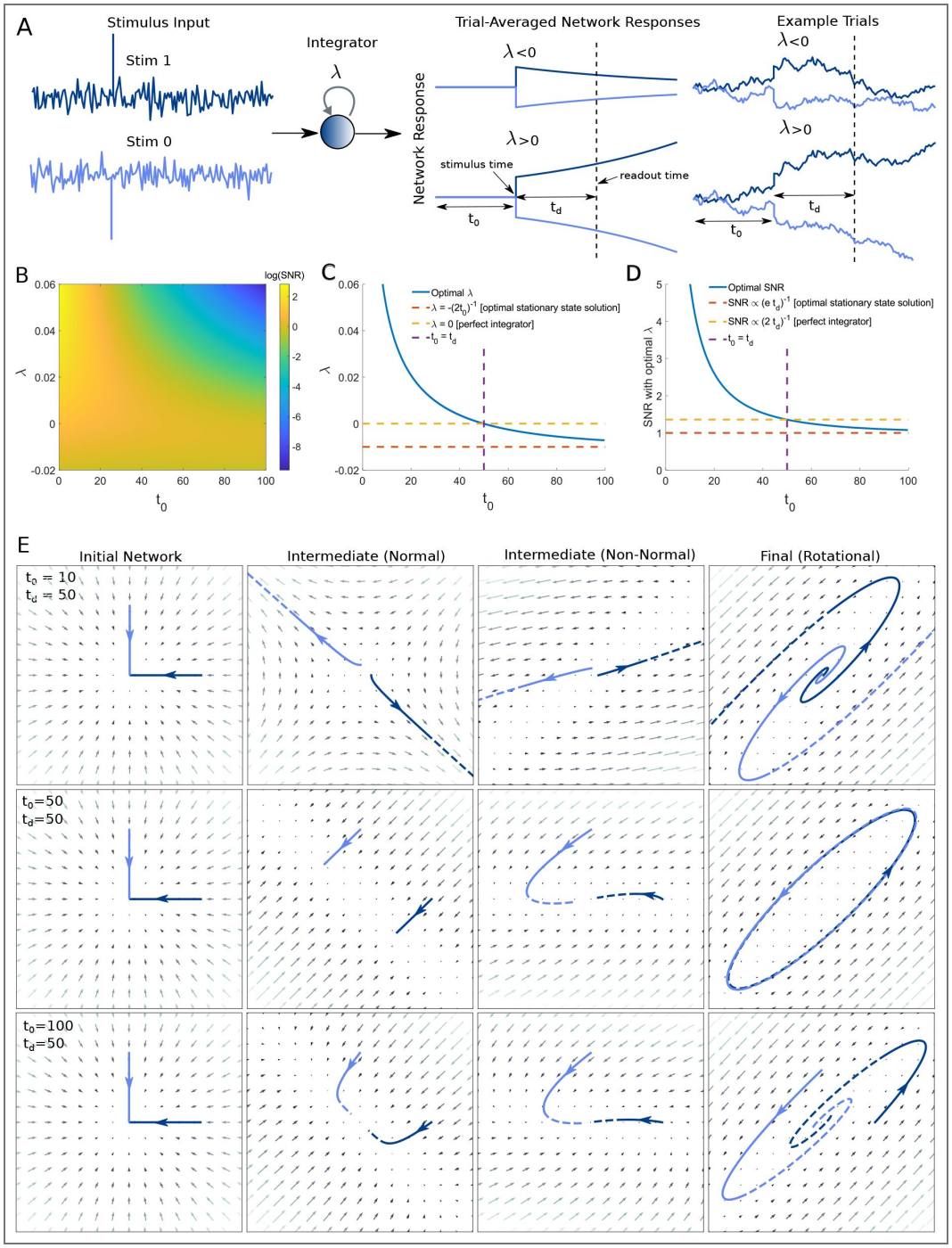

Figure S6. Effect of initial state covariance on optimal dynamics.

A: We derived analytical solutions for a one-dimensional integrator with recurrent weight λ receiving inputs from one of two stimuli given by $u(s_0, t) = -\delta(t - t_s) + n(t)$ and $u(s_1, t) = \delta(t - t_s) + n(t)$. The integrator is inititialised at a fixed (zero variance) initial condition at a time $t_s - t_0$ and the readout occurs at time $t_s + t_d$, where t_s is the stimulus time. Thus, noise accumulates for a time window t_0 before stimulus onset. B: The response SNR at the readout time depends on the values of t_0 and λ. C, D: The optimal value of λ is positive for $t_0 < t_d$ and negative for $t_0 > t_d$, reflecting a transition from leaky integrator to unstable amplifier as the pre-stimulus noise integration time window t_0 is decreased. E: Numerically optimised networks exhibit rotational dynamics, but the damping rate exhibits the predicted transition from stable and

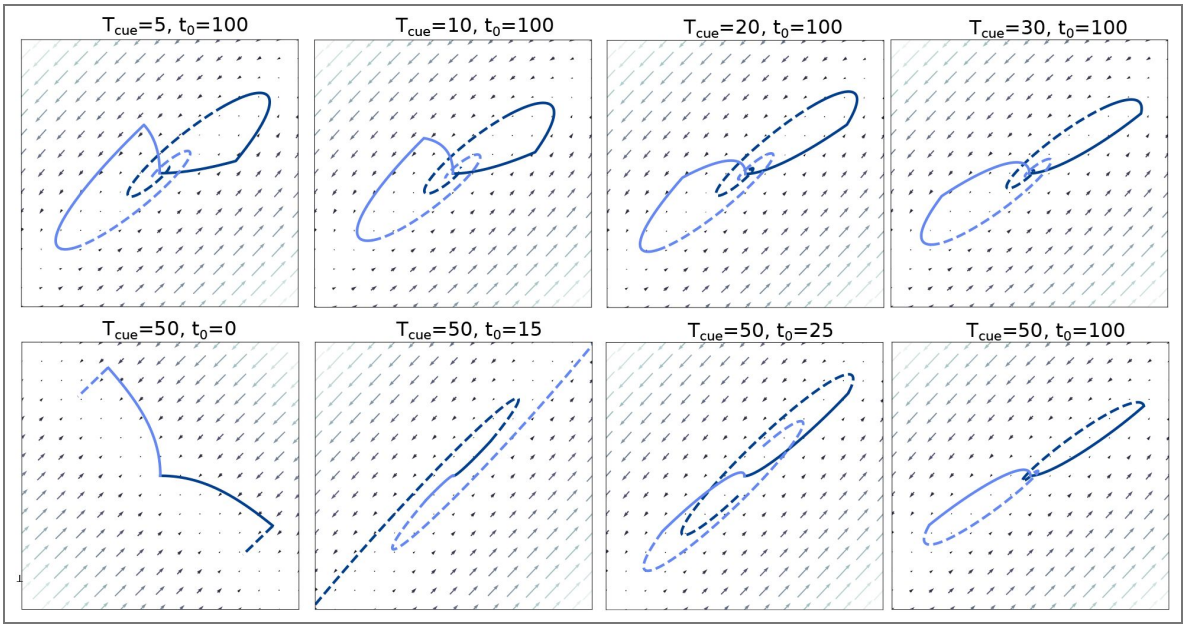

Figure S7. Effect of length of stimulus period and initial state covariance on optimal dynamics.

Top row shows effect of changing the length of the stimulus period T_{cue}. The readout time was $t_d = 50$ and networks were driven by noise for a period $t_0 = 100$ before stimulus onset (at $t_s = 0$). All optimised networks had complex eigenvalues, and the differences in trajectories were primarily due different input durations T_{cue} rather than changes in flow fields. Bottom row shows effect of varying the duration of pre-stimulus noise accumulation t_0 for networks with the stimulus presented continuously from $t = 0$ to $t = t_d = 50$. The network optimised with $t_0 = 0$ learned a classic line attractor solution to the task, which is known to be optimal in this setting (Gold and Shadlen, 2007). In contrast, for $t_0 > 0$ a line attractor would generate substantial pre-stimulus variability, and so is no longer optimal. In this case, networks must trade off integration during stimulus presentation against avoidance of integration of pre-stimulus noise (Supplementary Figure S2F-I). All networks optimised with $t_0 > 0$ had complex eigenvalues, suggesting that rotational dynamics are optimal for evidence integration tasks in which pre-stimulus noise influences task performance. Note that, although the trajectories during the stimulus presentation (solid lines) use the linear part of the curved rotational trajectories, noise inputs before stimulus onset would have been integrated for a longer period of time and therefore would be rotated through the curved part of the trajectories (dashed lines), thereby reducing their influence on the decoder.

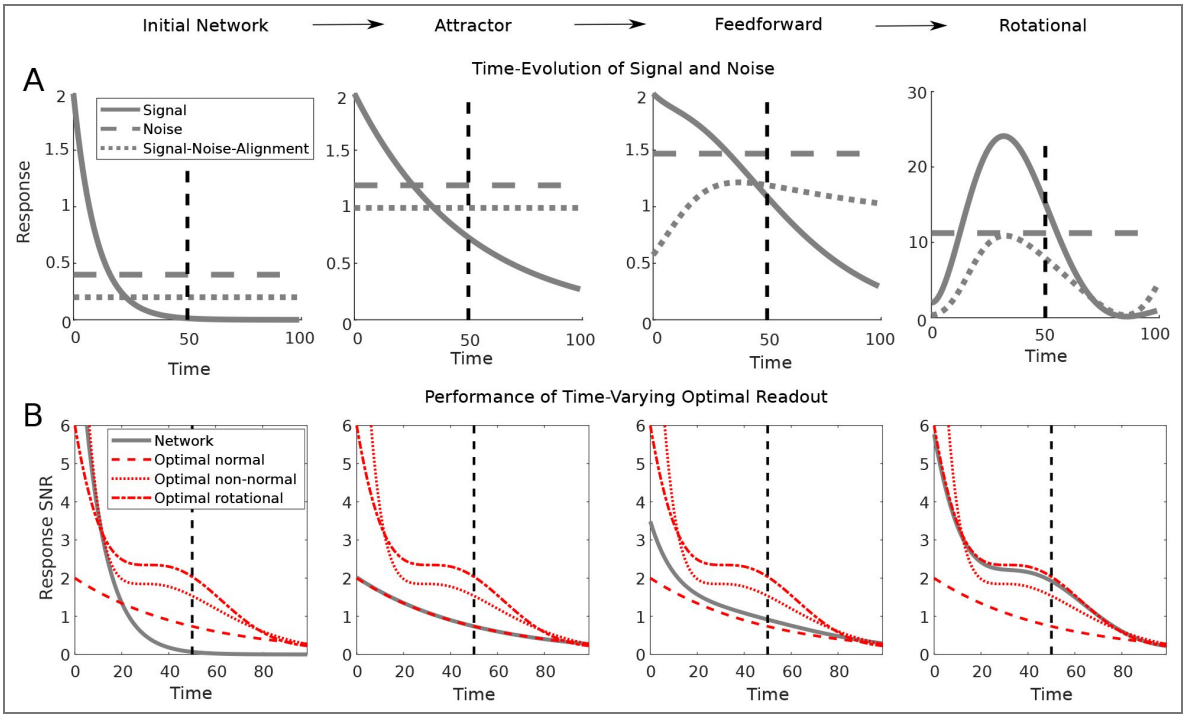

Figure S8.

A: The signal (squared norm of vector separating two stimulus-evoked mean responses), noise (total variance of responses to either stimulus) and signal-noise-alignment (see Methods) for each network as a function of time during the delay. Vertical dashed black line shows optimised decision time t_d. Note that networks were initialised at stationary state, so that noise statistics do not vary during the delay. B: The performance of the optimal readout at each time following stimulus onset (grey line). Red lines show the analytically computed optimal normal and two-dimensional non-normal or rotational networks.

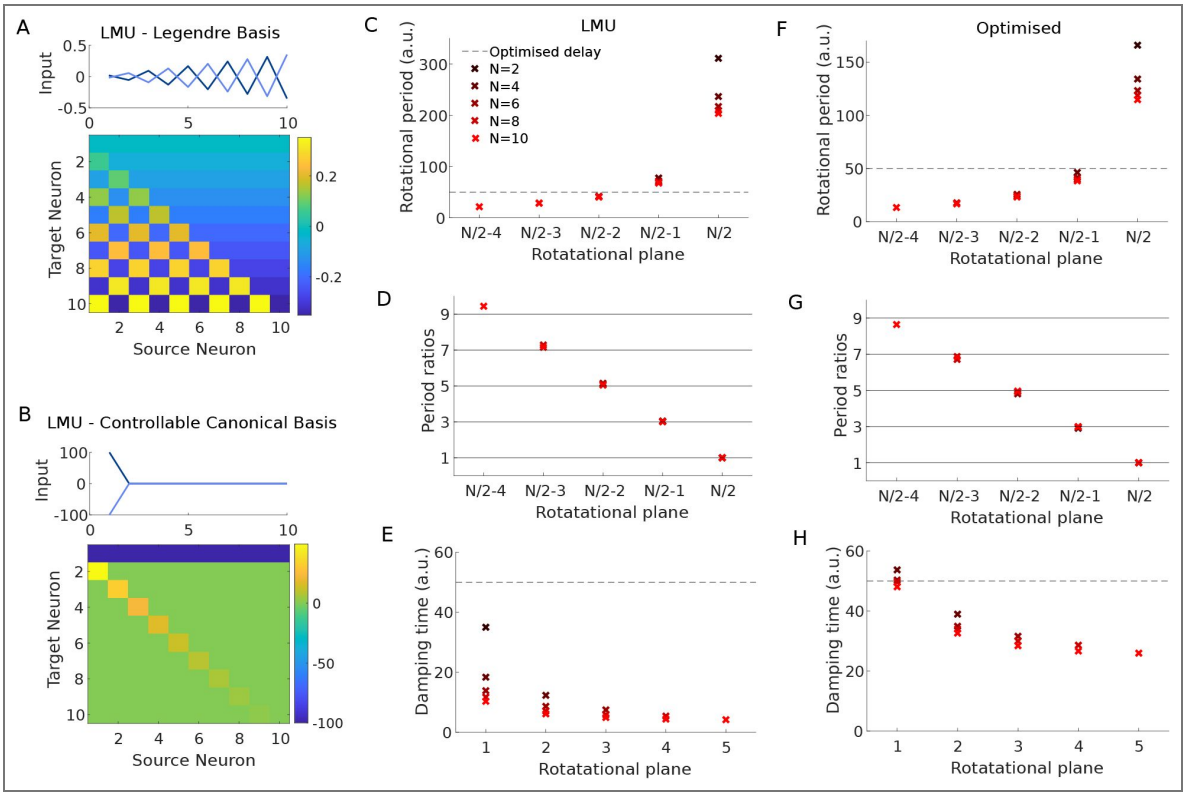

Figure S9. Rotational structure in optimised networks vs the Legendre Memory Unit (LMU).

A: Representation of the LMU in the Legendre basis. B: Equivalent representation of the LMU in the Controllable Canonical Basis. C: The rotational periods (given by imaginary parts of eigenvalues) of the LMU. Note that the eigenvalues are the same in the two bases in A and B. D: The ratio of each rotational period with that of the output rotational plane. E: The damping times (given by real parts of eigenvalues) of the LMU. F-H: As in C-E, but for the numerically optimised networks.

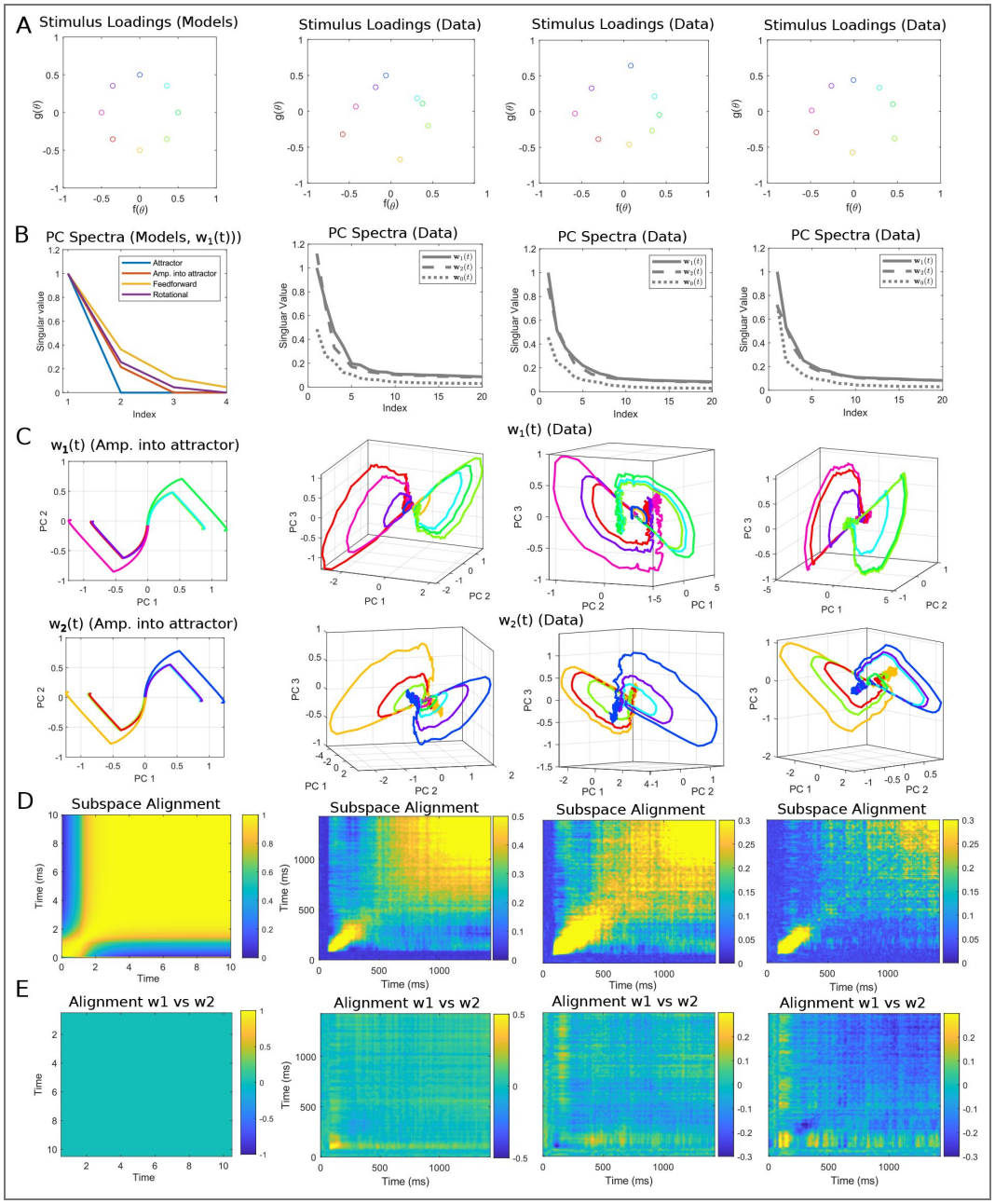

Figure S10.

A: Stimulus loadings $f(\theta)$, $g(\theta)$ in TDR models fit to data simulated from each WM model (left panel, all models yield identical results), and three experimental recording sessions (right three panels). B: Principal component spectrum of $\mathbf{w}_p(t)$ matrices of TDR models fit to simulated data (left panel) and three experimental sessions (right three panels). For the simulated data, only $\mathbf{w}_1(t)$ is shown, but $\mathbf{w}_2(t)$ has an identical spectrum and $\mathbf{w}_0(t) = 0$. For the experimental recordings, all three matrices are shown, with singular values normalised by the maximum singular value of $\mathbf{w}_1(t)$. C: Responses to each stimulus along the top principal components of $\mathbf{w}_1(t)$ and $\mathbf{w}_2(t)$ for the amplification into attractor model (left) and three recording sessions. Other models are shown in Figure 5. D: The alignment of the plane spanned by $\mathbf{w}_1(t)$, $\mathbf{w}_2(t)$ and the plane spanned by $\mathbf{w}_1(t')$, $\mathbf{w}_2(t')$ (left: amplification into attractor model, right: data). E: Alignment of $\mathbf{w}_1(t)$ and $\mathbf{w}_1(t')$ (left: amplification into attractor model, right: data).

Acknowledgements

We thank Alex Cayco-Gajic, Arthur Pellegrino, Sina Tootoonian, Matt Nolan and Matt Panichello for valuable feedback. We are particularly grateful to Matt Panichello and Tirin Moore for provision of experimental data. This work was supported in part by the Biotechnology and Biological Sciences Research Council [BB/Y513957/1].

Additional information

Funding

Funder	Grant reference number	Author
UKRI | Biotechnology and Biological Sciences Research Council (AFRC)	BB/Y513957/1	Angus Chadwick

Author ORCID iDs

Angus Chadwick: https://orcid.org/0000-0003-2664-0746

References

Al-Mohy A, Higham N (2008) Computing the fréchet derivative of the matrix exponential, with an application to condition number estimation. *SIAM J. Matrix Analysis Applications* **30**:1639-1657 https://doi.org/10.1137/080716426

Aoi M, Mante V, Pillow J (2020) Prefrontal cortex exhibits multidimensional dynamic encoding during decision-making. *Nature Neuroscience* https://doi.org/10.1038/s41593-020-0696-5 | PubMed

Averbeck B, Latham P, Pouget A (2006) Neural correlations, population coding and computation. *Nat Rev Neurosci* https://doi.org/10.1038/nrn1888 | PubMed

Baddeley A (1992) Working memory. *Science* **255**:556-559 https://doi.org/10.1126/science.1736359 | PubMed

Brockett RW (2015) *Finite Dimensional Linear Systems* Philadelphia, PA: Society for Industrial and Applied Mathematics.

Burak Y, Fiete I (2012) Fundamental limits on persistent activity in networks of noisy neurons. *Proceedings of the National Academy of Sciences* https://doi.org/10.1073/pnas.1117386109 | PubMed

Cavanagh SE, Towers J, Wallis JD, Hunt LT, Kennerley SW (2018) Reconciling persistent and dynamic hypotheses of working memory coding in prefrontal cortex. *Nature Communications* **9** https://doi.org/10.1038/s41467-018-05873-3 | PubMed

Chadwick A, Khan A, Poort J, Blot A, Hofer S, Mrsic-Flogel T, Sahani M (2023) Learning shapes cortical dynamics to enhance integration of relevant sensory input. *Neuron* https://doi.org/10.1016/j.neuron.2022.10.001 | PubMed

Christophel TB, Klink PC, Spitzer B, Roelfsema PR, Haynes J.-D (2017) The distributed nature of working memory. *Trends in Cognitive Sciences* **21**:111-124 https://doi.org/10.1016/j.tics.2016.12.007 | PubMed

Churchland MM, Cunningham JP, Kaufman MT, Foster JD, Nuyujukian P, Ryu SI, Shenoy KV (2012) Neural population dynamics during reaching. *Nature* https://doi.org/10.1038/nature11129 | PubMed

Cueva CJ, Saez A, Marcos E, Genovesio A, Jazayeri M, Romo R, Salzman CD, Shadlen MN, Fusi S (2020) Low-dimensional dynamics for working memory and time encoding. *Proceedings of the National Academy of Sciences* **117**:23021-23032 https://doi.org/10.1073/pnas.1915984117 | PubMed

Daie K, Fontolan L, Druckmann S, Svoboda K (2023) Feedforward amplification in recurrent networks underlies paradoxical neural coding. *biorxiv* https://doi.org/10.1101/2023.08.04.552026 | PubMed

Funahashi S, Bruce CJ, Goldman-Rakic PS (1989) Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. *Journal of Neurophysiology* **61**:331-349 https://doi.org/10.1152/jn.1989.61.2.331 | PubMed

Fusi S, Miller EK, Rigotti M (2016) Why neurons mix: high dimensionality for higher cognition. *Current Opinion in Neurobiology* **37**:66-74 https://doi.org/10.1016/j.conb.2016.01.010 | PubMed

Fuster J (1973) Unit activity in prefrontal cortex during delayed-response performance: neuronal correlates of transient memory. *Journal of Neurophysiology* **36**:61-78 https://doi.org/10.1152/jn.1973.36.1.61 | PubMed

Fuster JM, Alexander GE (1971) Neuron activity related to short-term memory. *Science* **173**:652-654 https://doi.org/10.1126/science.173.3997.652 | PubMed

Ganguli S, Huh D, Sompolinsky H (2008) Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences* **105**:18970-18975 https://doi.org/10.1073/pnas.0804451105 | PubMed

Gold JI, Shadlen MN (2007) The neural basis of decision making. *Annual Review of Neuroscience* **30**:535-574 https://doi.org/10.1146/annurev.neuro.29.051605.113038 | PubMed

Goldman MS (2009) Memory without feedback in a neural network. *Neuron* **61**:621-634 https://doi.org/10.1016/j.neuron.2008.12.012 | PubMed

Goldman-Rakic P (1995) Cellular basis of working memory. *Neuron* **14**:477-485 https://doi.org/10.1016/0896-6273(95)90304-6 | PubMed

Gu A, Dao T, Ermon S, Rudra A, Ré C (2020) Hippo: Recurrent memory with optimal polynomial projections. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H (Eds). *Advances in Neural Information Processing Systems* **33** Curran Associates, Inc. pp. 1474-1487

Gu A, Johnson I, Timalsina A, Rudra A, Re C (2023) How to train your HIPPO: State space models with generalized orthogonal basis projections. In: International Conference on Learning Representations.

Harvey C, Coen P, Tank D (2012) Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature* https://doi.org/10.1038/nature10918 | PubMed

Kriegeskorte N, Wei X (2021) Neural tuning and representational geometry. *Nat Rev Neurosci* https://doi.org/10.1038/s41583-021-00502-3 | PubMed

Laughlin SB (2001) Energy as a constraint on the coding and processing of sensory information. *Current Opinion in Neurobiology* **11**:475-480 https://doi.org/10.1016/s0959-4388(00)00237-3 | PubMed

Libby A, Buschman TJ (2021) Rotational dynamics reduce interference between sensory and memory representations. *Nature neuroscience* https://doi.org/10.1038/s41593-021-00821-9 | PubMed

Lim S, Goldman MS (2012) Noise Tolerance of Attractor and Feedforward Memory Models. *Neural Computation* **24**:332-390 https://doi.org/10.1162/neco_a_00234 | PubMed

Lundqvist M, Herman P, Miller EK (2018) Working memory: Delay activity, yes! persistent activity? maybe not. *The Journal of Neuroscience* **38**:7013-7019 https://doi.org/10.1523/jneurosci.2485-17.2018 | PubMed

Machens C, Romo R, Brody C (2010) Functional, but not anatomical, separation of "what" and "when" in prefrontal cortex. *J Neurosci* https://doi.org/10.1523/jneurosci.3276-09.2010 | PubMed

Mante V, Sussillo D, Shenoy KV, Newsome WT (2013) Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503**:78-84 https://doi.org/10.1038/nature12742 | PubMed

Miller E, Cohen J (2001) An integrative theory of prefrontal cortex function. *Annu Rev Neurosci* https://doi.org/10.1146/annurev.neuro.24.1.167 | PubMed

Moreno-Bote R, Beck J, Kanitscheider I, Pitkow X, Latham P, Pouget A (2014) Information-limiting correlations. *Nat Neurosci* https://doi.org/10.1038/nn.3807 | PubMed

Murphy BK, Miller KD (2009) Balanced amplification: A new mechanism of selective amplification of neural activity patterns. *Neuron* **61**:635-648 https://doi.org/10.1016/j.neuron.2009.02.005 | PubMed

Najfield I, Havel T (1995) Derivatives of the matrix exponential and their computation. *Advances in Applied Mathematics* https://doi.org/10.1006/aama.1995.1017

Orhan A, Ma W (2023) A diverse range of factors affect the nature of neural representations underlying short-term memory. *Nat Neursci* https://doi.org/10.1038/s41593-018-0314-y

Padamsey Z, Katsanevaki D, Dupuy N, Rochefort NL (2022) Neocortex saves energy by reducing coding precision during food scarcity. *Neuron* **110**:280-296. https://doi.org/10.1016/j.neuron.2021.10.024 | PubMed

Pagan M, Tang VD, Aoi MC, Pillow JW, Mante V, Sussillo D, Brody CD (2024) A new theoretical framework jointly explains behavioral and neural variability across subjects performing flexible decision-making. *bioRxiv* https://doi.org/10.1101/2022.11.28.518207

Panichello M, DePasquale B, Pillow J, Buschman T (2019) Error-correcting dynamics in visual working memory. *Nat Commun* https://doi.org/10.1038/s41467-019-11298-3 | PubMed

Panichello M, Jonikaitis D, Oh Y. e. a (2024) Intermittent rate coding and cue-specific ensembles support working memory. *Nature* https://doi.org/10.1038/s41586-024-08139-9 | PubMed

Partington JR (2004) *Linear Operators and Linear Systems: An Analytical Approach to Control Theory, volume 60 of London Mathematical Society Student Texts* Cambridge University Press.

Seung H (1996) How the brain keeps the eyes still. *Proceedings of the National Academy of Sciences* https://doi.org/10.1073/pnas.93.23.13339 | PubMed

Soldado-Magraner J, Mante V, Sahani M (2024) Inferring context-dependent computations through linear approximations of prefrontal cortex dynamics. *Science Advances* **10**:eadl4743 https://doi.org/10.1126/sciadv.adl4743 | PubMed

Spaak E, Watanabe K, Funahashi S, Stokes MG (2017) Stable and dynamic coding for working memory in primate prefrontal cortex. *J Neurosci* **37**:6503-6516 https://doi.org/10.1523/jneurosci.3364-16.2017 | PubMed

Sreenivasan KK, Curtis CE, D'Esposito M (2014) Revisiting the role of persistent neural activity during working memory. *Trends in cognitive sciences* https://doi.org/10.1016/j.tics.2013.12.001 | PubMed

Stanislaw H, Todorov N (1999) Calculation of signal detection theory measures. *Behav Res Methods Instrum Comput* https://doi.org/10.3758/bf03207704 | PubMed

Stokes M, Kusunoki M, Sigala N, Nili H, Gaffan D, Duncan J (2013) Dynamic coding for cognitive control in prefrontal cortex. *Neuron* https://doi.org/10.1016/j.neuron.2013.01.039 | PubMed

Stokes MG (2015) 'activity-silent' working memory in prefrontal cortex: a dynamic coding framework. *Trends in Cognitive Sciences* **19**:394-405 https://doi.org/10.1016/j.tics.2015.05.004 | PubMed

Stroud J, Duncan J, Lengyel M (2024a) The computational foundations of dynamic coding in working memory. *Trends Cogn Sci* https://doi.org/10.1016/j.tics.2024.02.011 | PubMed

Stroud J, Watanabe K, Suzuki T, Stokes M, Lengyel M (2023) Optimal information loading into working memory explains dynamic coding in the prefrontal cortex. *Proc Natl Acad Sci* https://doi.org/10.1073/pnas.2307991120 | PubMed

Stroud JP, Wójcik M, Jensen KT, Kusunoki M, Kadohisa M, Buckley MJ, Duncan J, Stokes MG, Lengyel M (2024b) Effects of noise and metabolic cost on cortical task representations. *eLife* https://doi.org/10.7554/elife.94961.1

Sussillo D (2014) Neural circuits as computational dynamical systems. *Current Opinion in Neurobiology* **25**:156-163 https://doi.org/10.1016/j.conb.2014.01.008 | PubMed

Turin G (1960) An introduction to matched filters. *IRE Transactions on Information Theory* **6**:311-329 https://doi.org/10.1109/TIT.1960.1057571

Voelker A (2019) Dynamical Systems in Spiking Neuromorphic Hardware. University of Waterloo.

Voelker A, Kajić I, Eliasmith C (2019) Legendre memory units: Continuous-time representation in recurrent neural networks. In: Advances in Neural Information Processing Systems. **32**

Voitov I, Mrsic-Flogel T (2022) Cortical feedback loops bind distributed representations of working memory. *Nature* https://doi.org/10.1038/s41586-022-05014-3 | PubMed

Vyas S, Golub MD, Sussillo D, Shenoy KV (2020) Computation through neural population dynamics. *Annual Review of Neuroscience* **43**:249-275 https://doi.org/10.1146/annurev-neuro-092619-094115 | PubMed

Wang X (2001) Synaptic reverberation underlying mnemonic persistent activity. *Trends in neurosciences* **24**:455 https://doi.org/10.1016/s0166-2236(00)01868-3 | PubMed

Panichello M, Jonikaitis D, Oh J, Zhu S, Trepka E, Moore T (2024) Intermittent rate coding and cue-specific neuronal ensembles support working memory. Dryad Digital Repository. https://doi.org/10.5061/dryad.kkwh70sct

Peer reviews

Reviewer #1 (Public review):

Summary:

In this manuscript, the authors address the question of working memory maintenance, starting from the experimental observation that recordings of neural activity during the delay period of working memory tasks are sometimes observed to be dynamic. They introduce a new combination of metrics (noise-robustness and energy efficiency) to quantify the performance of various network mechanisms of memory maintenance, in linear networks. They compared attractor networks, feed-forward networks, and networks trained with a loss that includes a robustness and an energy-efficiency component. They show, by plotting state-space trajectories, that networks optimized with this loss exhibit a form of rotational dynamics. They analyzed the data recorded during the delay of a working memory task in PFC, and observed state-space trajectories similar to those of the trained networks.

The comparison with other network mechanisms is interesting in principle, but limited by the fact that only linear networks are considered. This led to counter-intuitive and misleading statements, like the fact that attractor networks are not robust to noise, or that feed-forward networks have energy consumption that is exponential in the number of neurons.

Strengths:

(1) The idea to use both robustness to noise and energy efficiency to assess the performance of networks on working memory tasks is interesting.

(2) The manuscript is clearly written.

(3) There is an interesting combination of methodologies: theory on simple models, network training, and data analysis.

Weaknesses:

(1) Linear networks only.

The main feature of attractor networks is their robustness to noise, which is typically allowed by the non-linearity of neural responses. To fit their modeling framework, the authors focused only on continuous attractor neural networks (e.g., Seung 1996) and ignored point-attractor models such as the Hopfield model, which are typically used to model WM tasks, and which would presumably lead to very different results, e.g., in Figure 1D.

The linearity assumption is also problematic for the comparison with feed-forward models. It seems that the authors obtained runaway firing rates, explaining Figure 1F middle, which are typically prevented in non-linear networks.

The choice of parameters for the attractor network in Figure 1 is not explained. Why is t_slow = 10^4 chosen, and what does it correspond to? We expect in linear networks that activity goes back to zero or diverges as an exponential, but in principle, the time constant can be chosen to be of the same order as the time delay, with approximately linearly decreasing SNR.

Regarding the comparison of the different mechanisms, it would have been nice to better define the notion of rotational dynamics, beyond only considering state-space analysis, which is limited to providing mechanistic interpretations.

(2) Fixed duration of delay periods.

I have understood that for a given network, the duration of the delay period is fixed, as opposed to a delay duration that would fluctuate from trial to trial. This would be an important assumption to relax as well, to better match common experimental paradigms, as well as to expose a fairer comparison with other network mechanisms. See Orhan and Ma (2023) for such a discussion.

(3) Relationship with previous works

Many other works addressed the question of dynamic firing rates during maintenance periods of WM tasks; they should be discussed and compared to the mechanism proposed here. This includes: Barak et al, Progress in Neurobio. 2013, Pereira-Obilinovic, Aljadeff, Brunel, PRX 2023, Hansel, Mato, 2013, or works pertaining to the activity-silent neural states (allowed by short-term plasticity), the framework in which the data of Panichello et al are interpreted in the original publication.

https://doi.org/10.7554/eLife.111445.1.sa2

Reviewer #2 (Public review):

In this manuscript, Ritter et al. propose a model of working memory (WM) that combines feedforward and rotational dynamics. The model is discovered by optimizing a linear RNN using a loss function that encourages maximization of signal-to-noise ratio (SNR) and minimization of activation magnitude. The authors argue that the optimized model outperforms other WM models in terms of SNR and energetic efficiency, while also better replicating key features of neural responses recorded in monkey pre-frontal cortex (PFC) during a WM task. The authors also draw connections to state space models (SSM) used for other machine learning applications.

My main issue with this manuscript is that it does not appear to convincingly demonstrate that rotational dynamics offer any advantage over purely feedforward dynamics. The authors adopt three criteria according to which they compare models:

(1) SNR.

(2) Energy efficiency.

(3) Similarity to neural data.

In terms of SNR, purely feedforward models seem to perform similarly to the optimized models (Figure 1). Figure 1 does seem to show that the optimized network produces responses of smaller magnitude when the number of units is large, but the authors do not explain why adding rotational dynamics would produce such a relationship. In fact, the responses that are plotted for the feedforward network in Figures 1B, 2C, and 5E look similar, if not smaller in magnitude than those of the optimized model. Lastly, while the authors claim in the body of the text that the optimized model replicates key features of monkey PFC responses better than the purely feedforward model, this is not apparent to me from the

comparisons plotted in Figure 5E-J. The authors thus do not show strong evidence that the model they propose beats what they claim is an established baseline on any of the three criteria.

Another weakness of the manuscript is that the comparison to attractor and feedforward models seems somewhat unfair. In Figure 1, the rotational model is optimized, while the parameters for the attractor and feedforward models seem to have been at least partially chosen by hand. Figure 5C again shows the three models side by side, but the fact that it compares the same network at different stages during training complicates the comparison. Instead, one should compare the rotational solution to the optimal attractor and feedforward models, respectively (obtained by constrained optimization). From looking at the flow-fields, it seems that a feedforward network with an optimized level of amplification may work just as well. On a mechanistic level, it is unclear what computational advantage rotations offer over feedforward dynamics in the WM context.

The choice of baseline models to compare against might be questionable. The simple line attractor model by Seung et al. (1996) was initially designed to explain oculomotor integration. It is true that a line attractor has been suggested as a mechanism for working memory, e.g., in the seminal work by Machens et al (2005). However, it seems fair to say that most studies employing non-linear networks have focused on point attractors as mechanisms of working memory (e.g., Wong & Wang, 2006; Driscoll, Shenoy, Sussillo, 2024). A point attractor arguably does not suffer the SNR issues of a line attractor, because it does not lead to integration of the noise over time. However, non-trivial point attractors cannot be implemented in linear networks of the kind studied by the authors of the present study.

The authors should expand their discussion to include other, potentially closely related work proposing rotation-like dynamics in artificial neural networks during working memory. In particular, the manuscript does not discuss Sharma, Proca, et al, ICML 2026, which describes a rotational solution to a similar WM task obtained by optimizing linear RNNs (Sharma et al., 2026, Fig. 6). Notably, Sharma et al. arrive at a similar rotational (and likely also non-normal) mechanism without using either noisy inputs or a constraint on energy efficiency. The authors of the present manuscript should discuss to what extent this finding contradicts their claim that "normative pressures on noise-robustness and energetic cost shape the complex dynamics of WM circuits." (present manuscript, Introduction). Given the obvious parallels between the two studies, a comparison between the present work and Sharma et al. (2026) would add necessary context to the Discussion.

The authors should also clarify the significance of the "novel method for optimization of continuous-time RNNs driven by noisy inputs" (see Discussion) that the authors propose. This method is mentioned in the first line of the Discussion section but is barely discussed, let alone sufficiently explained, in the previous Sections. The only time a comparison to BPTT with a simple MSE loss is mentioned, it is stated that the two procedures produce the same results. The novel method appears to consist of a loss with two terms, the second of which is a well-known L2-penalty on unit activations (Sussillo et al., 2015). It is not clear that the method is either novel or necessary to obtain the reported results.

Except for the fact that higher-dimensional networks also converge on rotational solutions, Figure 3 does not add much to the reader's understanding of the optimized model (except for panel F). I find the comparison to SSMs too superficial to provide real insight.

Figure 4 claims to show that the optimized model recapitulates "a range of properties observed in prefrontal cortex and other brain areas during WM tasks" (p. 7) but does not show neural data for comparison.

https://doi.org/10.7554/eLife.111445.1.sa1

Reviewer #3 (Public review):

Summary:

The authors optimize continuous-time linear recurrent networks driven by noisy input, computing the gradient of decoding performance numerically and analytically. Optimizing for stimulus discriminability after a delay, with a penalty on firing rate, they find networks that adopt what they call high-dimensional rotational dynamics. They argue that these outperform attractor and feedforward models on noise robustness and energetic cost, and resemble state-of-the-art state-space models. They then fit a targeted dimensionality reduction model to prefrontal recordings from monkeys performing a spatial working memory task and argue that the population structure matches the rotational solution.

Strengths:

The evolution of the dynamics throughout learning is a nice observation, as are the analytical calculations, although I am not sure they are new since there is a fair share of work on the learning dynamics of linear networks.

Weakness:

I see many weaknesses. I will classify them into five groups.

(1) Strawman comparison and no clear definition of what is rotational. The paper is centered on comparing a trained model with two models meant to represent "attractor dynamics" and non-normal dynamics. Both are picked as the weakest member of their class.

I use quotation marks for "attractor dynamics" because I am not sure a linear system with an eigenvalue equal to zero is a representative model for the class. This is a particular linear instantiation of the line attractor from Seung 1996, but most attractor models are nonlinear and far more robust to noise, and they are robust through error correction that this linear model does not have. Even modern continuous attractors (Rivkind and Darshan) are very robust to noise through multiple mechanisms. So what the authors picked as an "attractor model" is a limited zero-eigenvalue case that, of course, will drift. "Attractor networks are highly susceptible to noise" is therefore true only of the toy they built, not of the class.

Second, what they call a non-normal model is in fact a feedforward chain, the extreme of non-normality. There are degrees of non-normality in any matrix, and the homogeneous delay line is the corner that requires the largest firing rates. This is not representative. See Daie et al., which has a skip and recurrent structure, or Stroud, which is not a pure chain. So the feedforward chain was also picked as a strawman, chosen so that the energetic cost they then complain about is guaranteed.

This brings me to the real problem in this section. "Rotational" is never defined. If it means complex eigenvalues, then it is a spectral property of any non-normal matrix, and "rotational versus feedforward" is not a dichotomy; it is two regions of the same continuous space of non-normal connectivity. Their own Figure 2C shows the network passing continuously through an attractor, then feedforward, then rotational during optimization. If these are points on a continuum, then "rotational dynamics is optimal" is just a statement about where the optimizer lands under this particular loss and input normalization, not the discovery of a new dynamical class. They need to define the term operationally and show the solution is qualitatively, not just quantitatively, different from non-normal feedforward. I do not think it survives that test.

This brings me to the references.

(2) The dynamical mechanisms of working memory have been studied for more than two decades, and I am surprised how much directly relevant work is missing. First, Druckmann and Chklovskii 2012, where a linear system produces stable encoding from oscillating modes. This is essentially their result more than a decade earlier, and it is not cited. They also miss Murray et al. on stable encoding and heterogeneous timescales in data. They oversimplify the attractor picture; for example, Pereira-Obilinovic et al. 2023 show you can have genuinely stable attractors. They do cite Daie et al., but they ignore its central claim, that non-normality is the underlying mechanism, which is more troubling than not citing it because it means they read it and did not engage. Overall, the references are idiosyncratic, missing relevant work, and not engaging the results of papers they cite.

This brings me to the third point.

(3) Novelty and the relationship to Stroud and Orhan. Those papers take a similar optimization approach and find that, depending on the task parameters, the optimal solution is non-normal, non-normal plus attractor, or attractor. My impression is that what this work calls rotational is just the dynamics of a strongly non-normal A, selected here by the firing-rate regularizer. They never clarify the connection with Stroud. Is the only difference the energy penalty?

The way to settle this is quantitative, and they have the handle and do not use it: report the Henrici departure-from-normality of their optimized A and place the solution inside Stroud's regime structure.

There is also a tension they leave implicit. In Stroud, the early loading direction is orthogonal to the late persistent readout, and that orthogonality is the source of dynamic coding. This paper's subspace alignment result (Figure 5G, H) shows exactly this early-to-late orthogonalization in both model and data, and then presents it as evidence for the rotational account and against Stroud's hybrid. You cannot reproduce a Strout's stim vs. decoder orthogonality and claim it against Strout's without doing more work.

(4) I did not understand the SSM section, and I think it should be cut. Is this a result? Either "SSM" just means a linear dynamical system, in which case it is trivial since every linear network here, including the LMU is an SSM, or it means the network matches a fixed-connectivity model like the LMU, which it does not seem to either. So in what sense is it a result?

(5) The data analysis is one section, and the analysis could be described as feeling somewhat like an afterthought on a very rich dataset. The coding structure they show for the rotational model also looks like the Stroud non-normal-plus-attractor model to me. They even state that the hybrid reproduces the cross-temporal subspace. What are the quantitative, cross-session metric that discriminates rotational from the non-normal-plus-attractor hybrid? Is it eyeballed trajectories?

https://doi.org/10.7554/eLife.111445.1.sa0