SETTING UP FOR FAILURE: AUTOMATIC DISCOVERY OF THE NEURAL MECHANISMS OF COGNITIVE ERRORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Discovering the neural mechanisms underpinning cognition is one of the grand challenges of neuroscience. However, previous approaches for building models of recurrent neural network (RNN) dynamics that explain behaviour required iterative refinement of architectures and/or optimization objectives, resulting in a piecemeal, and mostly heuristic, human-in-the-loop process. Here, we offer an alternative approach that automates the discovery of viable RNN mechanisms by explicitly training RNNs to reproduce behaviour, including the same characteristic errors and suboptimalities, that humans and animals produce in a cognitive task. Achieving this required two main innovations. First, as the amount of behavioural data that can be collected in experiments is often too limited to train RNNs, we use a non-parametric generative model of behavioural responses to produce surrogate data for training RNNs. Second, to capture all relevant statistical aspects of the data, rather than a limited number of hand-picked low-order moments as in previous moment-matching-based approaches, we developed a novel diffusion model-based approach for training RNNs. To showcase the potential of our approach, we chose a visual working memory task as our test-bed, as behaviour in this task is well known to produce response distributions that are patently multimodal (due to so-called *swap errors*). The resulting network dynamics correctly predicted previously reported qualitative features of neural data recorded in macaques. Importantly, these results were not possible to obtain with more traditional approaches, i.e., when only a limited set of behavioural signatures (rather than the full richness of behavioural response distributions) were fitted, or when RNNs were trained for task optimality (instead of reproducing behaviour). Our approach also yields novel predictions about the mechanism of swap errors, which can be readily tested in experiments. These results suggest that fitting RNNs to rich patterns of behaviour provides a powerful way to automatically discover the neural network dynamics supporting important cognitive functions.

1 INTRODUCTION

An important goal for computational neuroscience is to use behavioural data to generate viable and testable hypotheses about the neural network mechanisms that underpin cognition. To achieve this goal, the following requirements need to be met: (1) hypotheses should be formulated as recurrent neural networks (RNNs) so that dynamical neural mechanisms can be studied (2) models should be quantitatively fit to behaviour, such that they capture as many statistical properties of the data as possible, as important cognitive mechanisms have been shown to only reveal themselves in detailed patterns of response variability (rather than in averages); (3) model fitting should be automated, avoiding subjective, piecemeal iterative model construction, so neural network mechanisms can be accelerated at scale.

Previous approaches have fallen short of these desiderata. For example, cognitive models – including drift-diffusion (Resulaj et al., 2009; Pardo-Vazquez et al., 2019), Bayesian (Heald et al., 2021), symbolic (Castro et al., 2025), and large language model-based models (Binz et al., 2024) – have been successfully fit to detailed behaviour with impressive predictive power, but their architectures remained too abstract and divorced from RNNs to be useful as hypotheses about neural network dynamics (thus failing Requirement 1). Neural population coding models have suggested causal connections between neural response properties and particular patterns in behaviour (Matthey et al.,

2015; Schneegans and Bays, 2017; McMaster et al., 2022), but remained mute about the neural network dynamics that give rise to the hypothesised neural responses in the first place (failing Requirement 1), have been rarely fit to behaviour quantitatively (failing Requirement 2), and their construction was not automated (failing Requirement 3). RNN models – either with hand-crafted (Edin et al., 2009; Bouchacourt and Buschman, 2019), or with task-optimized architectures and parameters (Mante et al., 2013; Stroud et al., 2023) – have been highly successful at suggesting testable hypotheses about neural dynamics underlying cognitive performance, but they rarely capture behaviour beyond generically competent performance, let alone being quantitatively fit to detailed behavioural data (failing Requirements 2, and for hand-crafted networks also Requirement 3). In some cases, non-trivial aspects of behaviour were successfully captured by such models, but at the expense of including purpose-built design choices or ablations (Xie et al., 2023), thus making these models essentially hand-crafted in this sense (failing Requirement 3).

In this work, we introduce an approach that satisfies all three requirements. We extend automated neural mechanism discovery by directly training biologically plausible RNNs to exhibit behavioural suboptimalities seen in experimental data. We tackle two major challenges for doing so. First, RNNs are data hungry, and behavioural data is scarce. To make training feasible, we generate synthetic data using a descriptive generative model, and use this to train our RNN (Figure 1). The second challenge is defining a suitable training objective for the RNN so that they can generate complex (for example, multimodal (Bays et al., 2009) or skewed (Fritsche et al., 2020; Resulaj et al., 2009)) continuous response distributions often found in experiments. Previously used methods for training RNNs are not appropriate for this. Task optimization of RNNs typically used simple optimisation criteria, such as mean squared error or cross entropy (Yang et al., 2019), which encourage unimodal, typically normal response distributions. Even when training RNNs explicitly to generate specific distributions of responses, moment- or score matching-based criteria were used (Echeveste et al., 2020; Chen et al., 2023), which also do not scale well to capturing complex distributions (and require arbitrary choices as to which moments or statistics of the target distribution are important). Instead, by framing elicitation of a behavioural output as generating samples in Euclidean space, we train the RNN using a training criterion inspired by diffusion models (Ho et al., 2020), a state-of-the-art class of generative neural models for complex, continuous distributions. We adapt the training procedure used for diffusion models to work within an RNN, harnessing their flexibility in sampling from complex distributions. This enables us to flexibly fit to continuous response data, unlike prior methods of fitting directly to behaviour (Ji-An et al., 2025), which are limited to categorical distributions (§5).

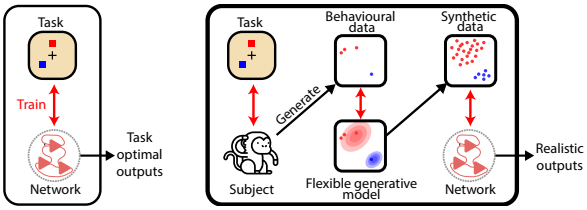

Figure 1: Left: typical procedure involves training an RNN to perform optimally in a task, without relating to real behaviour. Right: our novel method can replicate subject behaviour, by training on surrogate training data.

We demonstrate the power of our approach for automatic mechanism discovery on one of the most studied cognitive paradigms whose neural underpinning is still poorly understood: visual working memory (VWM). While there are many neural circuit models of working memory, they typically do not address the more challenging (and ecologically more relevant) situation when multiple pieces of information ('items') need to be maintained in memory Zhang (1996). Even RNNs that do perform multi-item VWM tasks fail to capture the characteristic patterns of behaviour that are specific to the multi-item situation (Yang et al., 2019; Driscoll et al., 2022), or only capture some select behavioural biases by using hand-crafted networks with purpose-built architectural motifs with limited biological plausibility (Bouchacourt and Buschman, 2019). In particular, so-called *swap errors* are a major source of errors that arise when participants recall the wrong item from their working memory, instead of the item that has been cued – and thereby create strongly multimodal response distributions (Bays et al., 2009). Current RNN models do not capture swap errors, and the resulting multimodal response distributions, at all. Conversely, there exist population coding models that do capture swap errors, but do not make predictions about neural circuit mechanisms; Schneegans and Bays (2017)). We address this gap by showing that our approach produces neural circuits that generate rich, realistic response distributions by directly training networks to perform swap errors. We use our trained networks to generate hypotheses about the biological implementation of observer models from the psychophysics literature (Schneegans and Bays (2016); McMaster et al. (2022); §4), previously inaccessible to RNNs trained for task performance.

2 BACKGROUND

RNNs in neuroscience Neural models of biologically plausible cortical circuits are described by their continuous-time "leaky" dynamics:

$$\Lambda \dot{\boldsymbol{r}} = -\boldsymbol{r} + \mathcal{F}(\boldsymbol{r}, \boldsymbol{s}, \boldsymbol{\eta}; \theta_r) \tag{1}$$

where $\boldsymbol{r}$ is the vector of neural firing rates (hidden RNN activations), $\boldsymbol{s}$ is the input from an external population (e.g. sensory input into the population), $\boldsymbol{\eta}$ is the population's biological process noise, and Λ is the diagonal matrix of neurons' membrane time constants (typically constant across cells, i.e. $\Lambda = \lambda I$). The non-linear computation $\mathcal{F}$, parameterised by θ_r, models the recurrent communication between and information integration within neurons. The network output is generically $\boldsymbol{x} = \mathcal{G}(\boldsymbol{r}; \theta_o)$, often interpreted as a behavioural response. The exact form of $\mathcal{F}$ and $\mathcal{G}$ vary across studies; in this work, we use a dendritic tree architecture for each neuron (Appendix B; Lyo and Savin (2024)), and interpret the output of the network - a simple linear projection of the firing rates - as a colour estimate on the colour circle. More details are provided in §3.

Previous attempts to train RNNs that generate samples typically train RNNs on task-optimality or moment-matching criteria (Echeveste et al., 2020). This is insufficient for us - as outlined in §1, we are interested in reproducing behaviour that takes on a distinctly multimodal distribution. To do so, we employ methods from the diffusion model literature. We briefly introduce diffusion models here, but defer many details to previous foundational work (Ho et al., 2020).

Denoising diffusion probabilistic models (DDPMs) DDPMs are a class of generative models that involve a neural network learning to undo a fixed noising process applied to data samples. During training, some $\boldsymbol{x}_T$ is drawn from data, and is iteratively corrupted:

$$q(\boldsymbol{x}_{\tau-1}|\boldsymbol{x}_\tau) = \mathcal{N}(\boldsymbol{x}_\tau; \sqrt{1-\beta_\tau}\boldsymbol{x}_\tau, \beta_\tau I) \tag{2}$$

which admits a closed form posterior $q(\boldsymbol{x}_{\tau+1}|\boldsymbol{x}_\tau, \boldsymbol{x}_T)$ with mean $\boldsymbol{\mu}_q(\boldsymbol{x}_\tau, \boldsymbol{x}_T)$. The DDPM describes a non-stationary transition kernel $p_\phi(\boldsymbol{x}_{\tau+1}|\boldsymbol{x}_\tau, \tau)$, and is trained by maximising a hierarchical evidence lower bound (ELBO) of the one-step denoising of the corrupted data. By parameterising the transition kernel by its mean $\hat{\boldsymbol{\mu}}_\phi(\boldsymbol{x}_\tau, \tau)$ only, then adding isotropic Gaussian noise with the same variance as the noising posterior, this ELBO can be expressed as a sequence of square distances, rather than KL divergences. We equate timesteps within the trial to timesteps in the DDPM denoising process, and use this objective to train the RNN to generate realistic samples from the response distribution. Training RNN-like diffusion models to perform cognitive tasks further requires a form of teacher forcing (Appendix C).

Synthetic data Neural networks are data-hungry. This is not an obstacle training for task-optimality – task variables such as stimulus features and target responses can be generated procedurally – but it is when training directly on behaviour – producing human or animal behavioural datasets is prohibitively costly at the scales required for training networks. Therefore, we generate synthetic behavioural data to plug this gap. Many of the models listed in §1 which accurately describe behaviour using neural population codes (Schneegans and Bays, 2017), Bayesian cognitive (Heald et al., 2021) and descriptive (Bruijns et al., 2023) models, and/or large language models (Binz et al., 2024) can, by the same token, be used to faithfully generate synthetic behavioural data. These synthetic datasets can be used to train our dynamical neural model; sufficient flexibility in the generative model allows synthetic data that captures desired statistical facets and suboptimalities of behaviour. Many of these models, or their relatives, have also been used to fit behavioural data from working memory (WM) tasks, with varying degrees of satisfaction of our three requirements. For this reason, and because of the sophisticated response distributions elicited from even simple WM tasks, we choose this as the testbed for our method. In §5, we will return to potential extensions to other aspects of WM, and other cognitive tasks in general.

Working memory (WM) and swap errors While not consistently distinguished from short-term memory (Aben et al., 2012), WM is generally defined as the temporary maintenance, manipulation, and retrieval of sensory information over an order of seconds, for executive functions. A dominant experimental paradigm in investigating visual working memory (VWM) is the *delayed estimation* task, in which a human or animal subject is presented with sensory information and asked, after a short delay during which the stimulus is withdrawn, to reproduce some information about it. Figure 2

shows an example of a *multiitem* delay estimation task, where the subject is presented with multiple (here, 2) items, then presented with a *probe dimension* (here, location) feature value, called a *cue*, and asked to reproduce the corresponding *report dimension* (here, colour) feature value of the cued item. From here, we will interchangeably use 'location' and 'probe feature', and 'colour' and 'report feature' for ease of language, however the features used for each role is open for experimental design. Most responses fall near the cued item's report feature value, and the distribution of estimation errors along the report feature (here, the colour circle) provides key evidence for measuring the capacity of VWM (Ma et al., 2014).

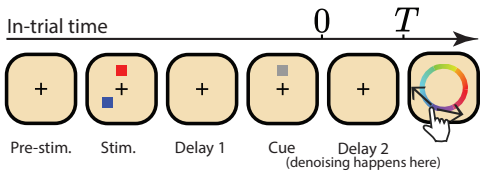

Figure 2: Two-item delayed estimation task - minimal VWM task for swap errors.

However, there is also a central tendency in responses towards the colour of *uncued items*, a.k.a. *distractors* (Bays et al., 2009; Gorgoraptis et al., 2011). This indicates the presence of *swap errors*, in which subjects mistakenly recall the colour of a distractor. Behavioural and neural evidence largely implicates *misselection* of the correct item from memory at cueing time, rather than a misbinding of feature values during memory encoding or storage – swap errors are a failure to extract the correct item from memory, rather than a misencoding into memory (but see Radmard et al. (2025)) Macaque neural signatures reflects this misselection hypothesis Alleman et al. (2024). The key evidence for this is that swap errors are more likely to happen when a distractor's probe feature value is close in feature space (here, physical space) to the cued one (Emrich and Ferber, 2012; Schneegans and Bays, 2017; McMaster et al., 2022). Describing the expected distribution of responses, including swap errors, requires a multimodal distribution, with modes placed over both target and uncued colours.

Bayesian Non-parametric model of Swap errors (BNS) We use BNS (Radmard et al., 2025) to generate synthetic multiitem delayed estimation data. This model predicts the likelihood of a swap error as a function of the distance of the distractor from the cued stimulus in each feature dimension (Appendix A). In the present work, we do not directly fit BNS to a real behavioural dataset. Instead, we opt to hand-tune the generative model to capture archetypal dependencies of swap errors. Specifically, we train RNNs with synthetic data generated by instantiations of BNS that either i) predicts no swap errors, ii) predicts the same frequency of swap errors for any distractor, and iii) predicts swap errors more frequently when its probe feature is more similar to that of the cued item, as per the extensive experimental evidence. In §3, we show that RNN representations resemble the real macaque neural geometry only in the third case, when maximal realism is captured in the training data, even in non-swap trials. We also fit BNS to the behavioural output of the trained RNNs to quantify its success in reproducing the target dependence on probe difference (§4). Again, we emphasise that the use of VWM and BNS is one instantiation of our method, and that its ethos of fitting directly to suboptimal behaviour extends beyond this domain, which we discuss briefly in §5.

3 METHODS

Network architecture Our RNN models a population of n densely connected cortical pyramidal neurons[1]. Each neuron integrates inputs from a sensory population and all other neurons in the population via a dendritic tree (Appendix B; Lyo and Savin (2024)). Overall, $\mathcal{F}$ is the somatic integration of the final, most proximal, layer of dendritic nodes with additive Gaussian noise (Appendices B, E). We train the RNN to perform the multiitem delayed estimation task (§2) with 2 items, providing the minimum viable task in which swap errors could arise. The network is first provided with two stimuli, each parameterised by two features, dubbed probe and report (visualised as location and colour), both on a circle. For *index-cued* networks, only the stimuli colours are provided as individual ordered scalars in $[-\pi, +\pi)$, and the cue as an index. This provides a direct analogy to the tasks performed by macaques (Panichello and Buschman, 2021; Alleman et al., 2024), to which we make comparison (§4). For *feature-cued* networks, the stimuli are provided to the network as activations of a palimpsest conjunctively tuned sensory neural population (Appendix B; Matthey et al. (2015); Schneegans and Bays (2017)). Importantly, in the latter case, items are order invariant - the network must store the two conjunctions between the two features. This allows us to extend to predictions about the representation of a variable probe feature. The network's firing rates vector is projected

[1]Constants used in defining the architecture/training processes are declared in Appendix E.

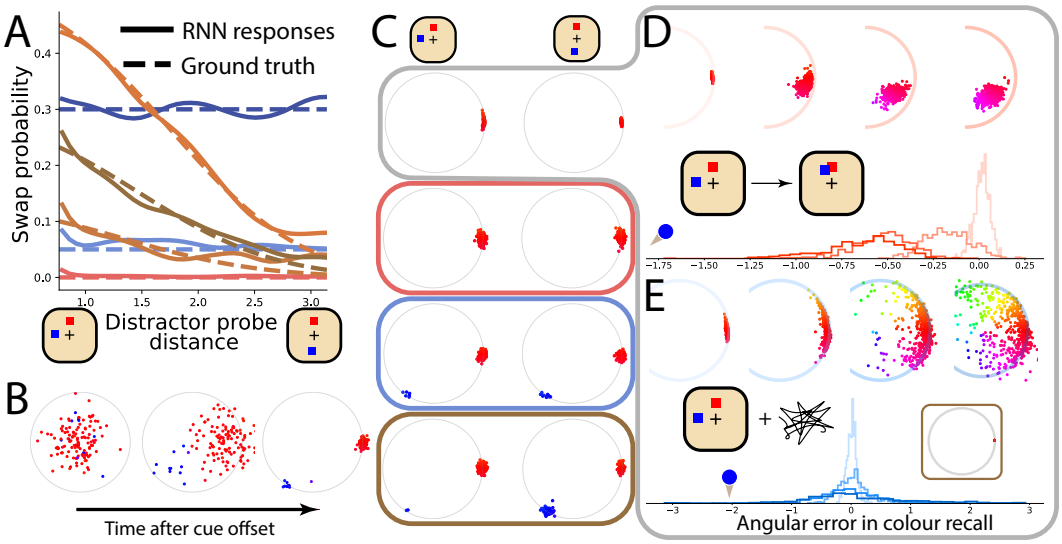

Figure 3: DDPM-style trained RNNs accurately captures swap errors in training data, unlike ablated task-optimal networks. **A** dotted lines are the target swap rate used to generate the synthetic training data for various RNNs, and solid lines are average swap rates inferred from fitting BNS to RNN-generated behaviour after they are trained on this synthetic data (Appendix A; Radmard et al. (2025)). RNNs achieve fair success in replicating training data for most trials in both probe distance-dependent and -independent cases. **B** Trajectory snapshots at different points in the second delay period, during which the network is trained to denoise behaviour. Trials are coloured by their argument at the end of denoising, which is interpreted as the colour estimation made by the network. Final samples successfully sample from their multimodal target distribution (Appendix A). **C** Typical set of final timestep samples for trials with close (left) and far (right) probe feature values. In all cases the red stimulus is cued. Borders indicate generative RNNs, referencing lines in **A**. Top row (grey border) indicates task-optimal network. **D,E** Two typical ablations that may be applied to task optimal networks to indcue swap errors: decreasing probe distance beyond the minimum margin between items seen during training to induce confusion, and increasing process noise variance beyond training conditions to increase misselection likelihood. Swap errors would be evidenced by a second mode at the colour of the distractor, indicated by a blue marker. This does not arise in either case, and further attempts would require subjective ablation design. **E** inset: Conversely, removing process noise for a network trained to perform swap errors recapitulates optimal behaviour.

to the 2D plane with a *fixed, orthonormal* output projection matrix $\boldsymbol{x} = \mathcal{G}(\boldsymbol{r}) = W_x, W_x \in \mathbb{R}^{2\times n}$ (i.e. $W_x^\intercal W_x = I_n$). Network activity is therefore split into two fixed orthogonal subspaces: the *behavioural subspace* $\boldsymbol{x} = W_x\boldsymbol{r}$, and the *behavioural nullspace* $\boldsymbol{m} = W_x^\perp \boldsymbol{r}$. We interpret the argument of this output at the final timestep of the trial, $\arctan\left(\frac{[W_x \boldsymbol{r}_T]_1}{[W_x \boldsymbol{r}_T]_2}\right)$, as the colour recalled by the network on that trial. To train for task performance, a mean squared error (MSE) loss on the discrepancy between this 2D estimate and the report feature of the cued item embedded on the circle in $\mathbb{R}^2$ suffices (Appendices A, C).

Training Instead of training for task optimality, we adapt the training procedure used for generative diffusion models (§2; Ho et al. (2020); Lyo and Savin (2024)) to train for behavioural realism, accounting for the presence of swap errors. We apply the DDPM training objective only to $\boldsymbol{x}$, not the full activity vector $\boldsymbol{r}$, equating timesteps after cue offset to timesteps in the DDPM denoising process. Besides the use of leaky dynamics (see above), this is a key difference to previous integrations of diffusion models with RNNs in cortical modeling (Lyo and Savin, 2024). Additionally, the DDPM-style criterion is only applied to the T timesteps following cue offset (Figures 2, 3B) – not repeatedly throughout the full duration of the trial – and the target distribution is trial-specific.

For each task trial, the target distribution of responses is defined by a mixture of 2 Gaussians in $\mathbb{R}^2$, with mode means determined by colours of stimuli (on the circle), and weights determined by BNS prediction (Appendix C). This final behavioural sample is then iteratively noised (equation 2) to produce the target trajectory for the second delay. The network is trained to undo this process at each

timestep of the second delay, with the transition kernel mean provided by discretising the continuous noiseless RNN dynamics (equation 1) projected to the behavioural subspace:

$$\hat{\boldsymbol{\mu}}_{\theta_r}(\boldsymbol{r}_t, t) = W_x \left[\left(1 - \frac{\mathrm{d}t}{\lambda}\right) \boldsymbol{r}_t + \frac{\mathrm{d}t}{\lambda} \mathcal{F}(\boldsymbol{r}_t, \boldsymbol{s}_t; \theta_r) \right] \tag{3}$$

Note that because diffusion only occurs during the second delay, the sensory term $\boldsymbol{s}_t$ is empty (Appendix B), and information about the full set of stimuli, and hence the target distribution, is contained in the behavioural nullspace activity $\boldsymbol{m}_t$. This dependence on previous timesteps means training requires simulation of the full trial trajectory in sequence. Because valid application of the DDPM criterion (§2) requires training examples of noised data ($\boldsymbol{x}_\tau$) to be drawn from the marginal distribution of the noising process, i.e. $q(\boldsymbol{x}_\tau|\boldsymbol{x}_T)$, we employ teacher-forcing during training to ensure that the distribution of trajectories of $\boldsymbol{x}_\tau$ stay in distribution. The training algorithm is summarised in algorithm 1, with full details provided in Appendix C.

Algorithm 1 Summary of training with teacher-forcing - full algorithm in Appendix C.

repeat
  Generate stimulus set and select cued item
  Using BNS, generate a sample $\boldsymbol{x}_T^*$ from the desired behavioural distribution in $\mathbb{R}^2$, and apply the noising process (equation 2) to generate noised data $\boldsymbol{x}_{0:T-1}^*$
  Initialise network activity from $\mathcal{N}(0, \sigma_r^2 I)$
  Run network dynamics for the prestimulus, stimulus, delay, and cue periods (Figure 2) with noisy discretised leaky dynamics (equation 3 without projection), finally arriving at network activity at cue offset $\boldsymbol{r}_0$
  for $t = 0$ **to** T (i.e. during second delay) **do**
    Apply teacher-forcing with $\boldsymbol{x}_t^*$ (Appendix C)
    Calculate mean transition kernel $\hat{\boldsymbol{\mu}}_{\theta_r}(\boldsymbol{r}_t, t)$ (equation 3)
    Run dynamics one step (equation 3) and add scheduled noise (Appendix E)
  end for
  Train on regularised DDPM criterion (Appendix C)
until converged

4 RESULTS

Swap dependence reproduction We start by summarising the success in reproducing a variety of desired behaviours, characterised by the target swap function. Figure 3A shows the ability of the RNNs to replicate the dependence of swap errors on distractor distance used to generate their synthetic training data, for a variety forms of dependence (Appendix A). Here, dotted lines show the ground truth swapping rates as a function of probe distance, and solid lines show feature-cued networks' abilities to capture them. This is inferred by fitting BNS to the estimates generated by the trained RNN. Disparities in low swap probability regions is likely due to the compounded lack of training samples for both the RNN and BNS in low probability modes. Example sample sets, coloured by the associated estimate, are shown for some of these in Figure 3C, with the multimodality achievable with our method contrasting the unimodal distribution of estimates produced by task-optimised networks (top row). Figure 3D and E summarise naïve attempts to induce this multimodality after training for task-optimality, which one might expect researchers to attempt. Failure to do so would typically set off an iterative and targetted sequence of ablations based on heuristics/prior assumptions on the origin of swap errors. We achieve swap errors by directly training for them, and find that the resulting neural signatures underlying these errors match those found in biology, as we show below.

Neural representations of memoranda We now compare the neural representations of stimuli during each delay to those of macacque lateral prefrontal cortex (LPFC) during a similar task (Panichello and Buschman, 2021). Here, two macacque monkeys performed a version of the multiitem delayed estimation task with fixed stimulus locations - one coloured square in the top and one in the bottom half of their field of vision. A binary visual cue at the point of fixation indicated which item is to be recalled. The experimentalists i) found a swap error rate of roughly 5%, and ii) identified a consistent low dimensional representation of the stimuli in memory before and after the onset of the cue during accurate trials (Figure 4B). At both points in accurate trials (i.e. no swap error, and

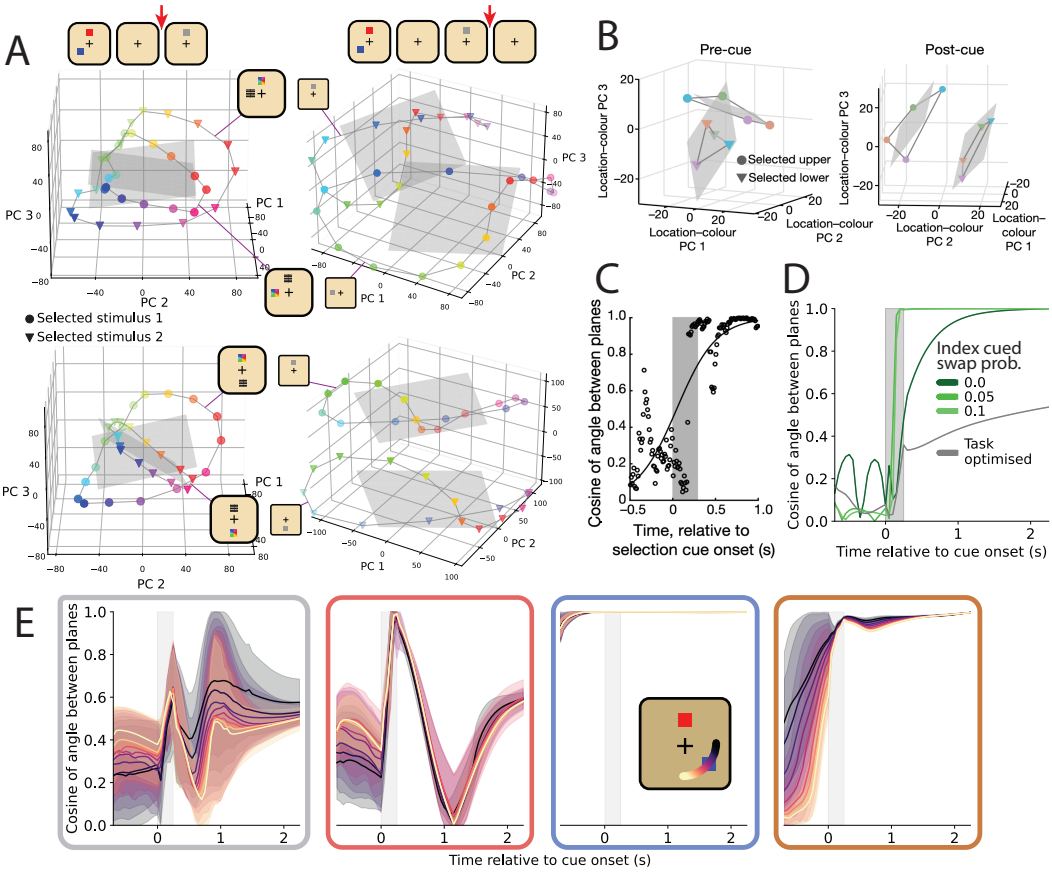

Figure 4: Behaviourally realistic, but not task optimal, networks capture neural signatures of VWM. **A** $\boldsymbol{m}_T$ averaged over many accurate trials with distractor report feature varying, for trials with close (top) and far (bottom) stimulus distance, before (left, planes misaligned) and after (right, planes aligned) cue exposure. Colours represent report feature of cued (and accurately recalled) stimulus. Representations drawn from network in **E** (right), which best matches neural data (see below). **B** This qualitatively matches equivalent representations in macaque cortex - in this case there are only two possible cue values. **C** The real data further shows that this planar alignment (quantified as cosine similarity between normals) increases before and during cue exposure (grey stripe), and remains high throughout the second delay until time of recall. **D, E** Plane alignment over time for the index-cued and feature-cued networks, respectively. Axes borders in **E** indicate the model used, as in Figure 3 .Importantly, *only the model with distance-dependent swap probability* matches the description of the biological data. A longer second delay period was chosen to encompass all experimental conditions in the original study. Error bars show mean ± std across all different spatial configurations with the denoted stimulus location distance (see inset; purple is closest distractor, yellow is furthest). ***B,C*** *adapted from Panichello and Buschman (2021).*

low angular error), the mean population response of LPFC associated with the colour of each item (upper and lower) was found by averaging neural activity across trials with that colour in that position, factoring out the colour of the distractor. These representations formed a 2D cycle when averaged across trials, reflecting the cyclicality of the behavioural responses provided by the monkey on the colour circle. Prior to cue onset, the representation of each item occupied perpendicular 2D planes (Figure 4B [left]) and after cue onset, these planes reoriented to become parallel (Figure 4B [right]), with the cosine of the angle between the planes increasing to a maximum during the cue remaining high (i.e., parallel) during the second delay (Figure 4C).

We compare these findings to representations in the behavioural nullspace activity $\boldsymbol{m}$ of our RNN, for index- and feature- cued networks, as well as task-optimised networks. For index-cued networks (Figure 4D), which most closely analogise the real task, optimisation for task performance does not capture the full alignment of the stimulus representations during and after cue presentation. The

sharp and sustained increase and alignment is only captured when using our DDPM-style training *and* including swap errors in the trial-wise target distributions. In the feature-cued case, where a continuous location is used to load and cue items from memory, this pattern is only captured when the network is trained with a location-dependent swapping rate (Figure 4E [right]). Furthermore, this network bears higher pre-cue representational alignment for distractors with closer probe feature values. This provides a prediction for the neural substrate of delay-time processes leading to memory retrieval error at cueing, previously described in the psychophysics literature as noising or drift in the abstract representations of stimuli along the probe dimension over time (Schneegans and Bays, 2017; McMaster et al., 2022). **Concretely, we predict that trial-by-trial decomposed analysis of the alignment between stimulus representations will reveal that swap errors occur more often when pre-cue representational alignment of colour is higher.** This could be due to probe similarity, as in the case presented, or other factors such as attention before cueing in the fixed item location case.

Item misselection The previous subsection established that physically closer, hence more swap-prone, distractors cause higher report feature representation alignment before cue onset (Figure 4E [right]). We now consider the neural signatures of the retrieval errors at cueing time, which lead to swap errors. For the same macaque task, experimentalists later showed that *item misselection* is the primary cause of swap errors (Figure 5A; Alleman et al. (2024)). The misselection is operationalised as follows: (i) before cue onset, items are stored as conjunctions between their probe (location - or just upper and lower in the experimental case) and report feature (colour) values (Figure 5A; 'spatial subspace'); after cue onset, items are stored as conjunctions between their *role* (target vs. distractor) and report feature value (Figure 5A; 'report subspace'); item 'misselection' is misprojection between these subspaces at cueing time. We now complement existing evidence of this neural basis of swap errors, which depends on across-trial statistical modelling of neural responses Alleman et al. (2024), with the representational geometry learned by our network.

Figure 5B shows a low-dimensional projection of $\boldsymbol{m}_T$ for a fixed set of locations and one fixed colour forming two parallel rings, coloured by the free stimulus. In the left column, when the fixed colour stimulus is cued, the bottom ring (negative PC3) is made up of the representation of swap trials, which are coloured by the uncued but *mistakenly recalled* colour, and the top ring (positive PC3) is coloured by the distractor colour when the fixed colour item is *accurately recalled*. Vice versa when the fixed colour stimulus is uncued (right column). The normal to these rings, which discriminates accurate and swap trials (Figure 5C [right]), is independent of the stimulus locations, only depending on the fixed stimulus colour considered (Figure 5C [left]). **This makes this two ring structure a candidate for the geometry of the postcue report subspace** (Alleman et al., 2024), which binds stimulus colour (around the ring) to its recalled role (choice of ring), and is independent of the original colour-location binding (Figure 5A). We have not considered exactly *how* this report subspace varies across different fixed report stimuli, just that the normals of these rings is shared whenever one of the items has a fixed colour, regardless of the item locations or whether the fixed item was cued (Figure 5C [left]). However, their alignment depends on colour similarity (Appendix D), suggesting an intuitive toroidal formation.

5 Discussion

In this work, we trained RNNs to exhibit swap errors during a multi-item delayed estimation VWM task (§3). Its success was contrasted with naïve training methods and ablations applied to task-optimal networks (Figure 3). Capturing swap errors *post hoc* would require a series of directed network modifications, likely driven by leading assumptions or heuristics as to how swap errors arise, whereas our method displays them directly. Then, we showed that this training method captured a more accurate description of neural mechanisms when trained to adhere to the empirical variability around real swap errors (Figure 4). Namely, we trained one network to produce swap errors more frequently when there is higher probe similarity between the cued item and the distractor (Figure 4E [left]), previously theorised to increase the likelihood of the item *misselection* underlying swap errors. Separating trials by probe distance, we provided a prediction for the neural representation of probe similarity before the cue. Finally, we provided a candidate representational geometry for swap errors, in line with previous work on their neural basis.

Our approach bridges the gap between flexibly modelling neural circuits and producing rich descriptions of behaviour, that were previously only achieved separately by task-optimised networks and abstract cognitive models, respectively. We did so by reversing the typical sequence of normative modelling - by directly training networks to reproduce the full distribution of behavioural responses,

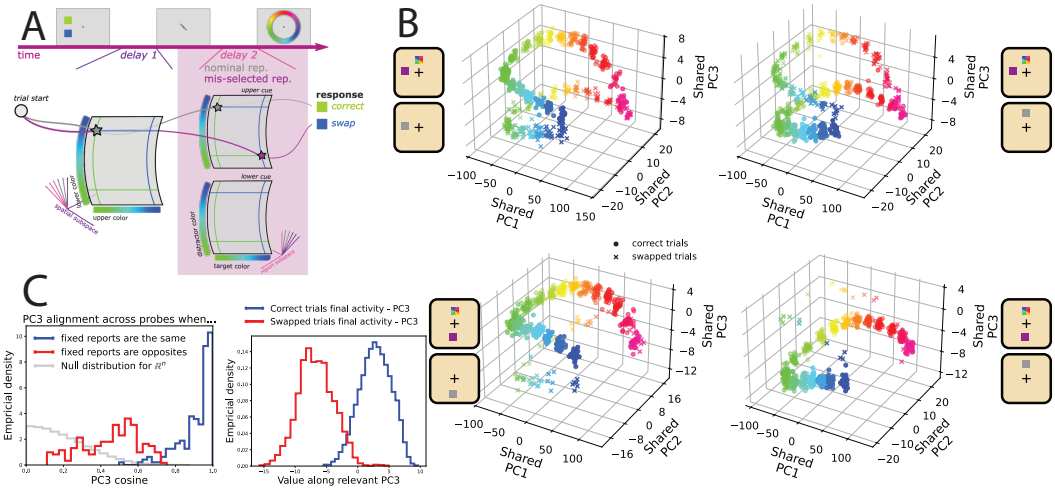

Figure 5: RNNs trained with swap errors provide a prediction about the geometry of item misselection. **A** Schematic explanation of misselection as a cause of swap errors (see main text; *reproduced from* Alleman et al. (2024)). **B** Trial-by-trial $\boldsymbol{m}_T$, projected to a shared set of principal components (PCs). One item's colour is fixed to purple, and the other item's colour is varied and used to colour the scatterplots. Left (right) axes show when the purple item is cued (a distractor), so colours indicate the uncued (target) colour. Probe features are fixed to a close (far) spatial configuration in the top (bottom) row, causing more (fewer) swap errors. Neural activities form two rings (with a gap in each due to the minimum margin between colours) - see main text. To illustrate this better, we used a network that swaps more often than seen for macaques (highest orange line in Figure 3A). **C** Left: there is high pairwise alignment in the third PC across different stimulus spatial configurations (i.e. fixed locations) if the fixed colour used to generate the two ring geometry is the same between these configurations - less so if the fixed colour is different (here, the opposite) for all pairs of spatial configurations. See Appendix D for a more graded pairwise comparison. This also applies to the first two PCs, unshown. Right: this third PC discriminates swapped versus correct trials in cases where the fixed colour item (purple here) is cued, across many different probe values.

including suboptimalities, we bypass the need for an iterative, potentially heuristic set of ablations. These principles are most tightly shared by tiny RNNs (Ji-An et al., 2025). However these small networks, like previous approaches fitting to behavioural data (Xue et al., 2024) have only been used for discrete behaviour, e.g. categorical choices, for which simple maximum likelihood fitting to behavioural datasets suffices. Our method provides a novel, principled approach to holistically fitting to all, continuous dimensions of data. The neural predictions we made by reverse engineering behaviour in this way await experimental verification, but demonstrate the value of training networks to reproduce behavioural errors rather than optimising for performance.

The present work can be extended in multiple directions. First, our analysis of the representational geometry of stimuli and responses (Figures 4 and 5) mostly considers stationary snapshots of the population activity. Besides this, Figures 4D, E aggregate information across many trials. Future work should include trial-by-trial dynamical analysis. For example, the misselection process described in Figure 5A may be due to a failure in information loading (Stroud et al., 2023) during the cue period, which is made more likely due to the higher representational alignment we uncovered here. Better interpreting why RNNs capture neural phenomena only when trained explicitly to exhibit errors, beyond the descriptive summary we have provided here, can help refine this inquiry. Looking forward, there exists a much richer set of behavioural phenomena associated with VWM which can be modelled with our method. We used this VWM task as a testbed for our novel methodology because of its inherent multimodality, but our method can be extended to any other cognitive task associated with complex behavioural suboptimalities, or other non-trivial response distributions, such as reaction times (Pardo-Vazquez et al., 2019). Further dependence of swap errors on set size (Bays et al., 2009; Emrich and Ferber, 2012) and presentation order (van Ede et al., 2021; Gorgoraptis et al., 2011), and other, unimodal, response distributions which cannot be captured by moment-matching (Fritsche et al., 2020) are all suitable next paradigms to benefit from our method.

REFERENCES

Bart Aben, Sven Stapert, and Arjan Blokland. About the distinction between working memory and short-term memory. *Frontiers in Psychology*, 3, 2012. ISSN 1664-1078. doi: 10.3389/fpsyg.2012.00301. URL https://www.frontiersin.org/articles/10.3389/fpsyg.2012.00301.

Matteo Alleman, Matthew Panichello, Timothy J. Buschman, and W. Jeffrey Johnston. The neural basis of swap errors in working memory. *Proceedings of the National Academy of Sciences*, 121 (33), August 2024. ISSN 1091-6490. doi: 10.1073/pnas.2401032121. URL http://dx.doi.org/10.1073/pnas.2401032121.

Bays, R. F. G. Catalao, and M. Husain. The precision of visual working memory is set by allocation of a shared resource. *Journal of Vision*, 9(10):7–7, September 2009. doi: 10.1167/9.10.7. URL https://doi.org/10.1167/9.10.7.

Marcel Binz, Elif Akata, Matthias Bethge, Franziska Brändle, Fred Callaway, Julian Coda-Forno, Peter Dayan, Can Demircan, Maria K. Eckstein, Noémi Éltető, Thomas L. Griffiths, Susanne Haridi, Akshay K. Jagadish, Li Ji-An, Alexander Kipnis, Sreejan Kumar, Tobias Ludwig, Marvin Mathony, Marcelo Mattar, Alireza Modirshanechi, Surabhi S. Nath, Joshua C. Peterson, Milena Rmus, Evan M. Russek, Tankred Saanum, Johannes A. Schubert, Luca M. Schulze Buschoff, Nishad Singhi, Xin Sui, Mirko Thalmann, Fabian Theis, Vuong Truong, Vishaal Udandarao, Konstantinos Voudouris, Robert Wilson, Kristin Witte, Shuchen Wu, Dirk Wulff, Huadong Xiong, and Eric Schulz. Centaur: a foundation model of human cognition, 2024. URL https://arxiv.org/abs/2410.20268.

Flora Bouchacourt and Timothy J. Buschman. A flexible model of working memory. *Neuron*, 103 (1):147–160.e8, July 2019. doi: 10.1016/j.neuron.2019.04.020. URL https://doi.org/10.1016/j.neuron.2019.04.020.

Sebastian A. Bruijns, Kcénia Bougrova, Inês C. Laranjeira, Petrina Y. P. Lau, Guido T. Meijer, Nathaniel J. Miska, Jean-Paul Noel, Alejandro Pan-Vazquez, Noam Roth, Karolina Z. Socha, Anne E. Urai, and Peter Dayan. Dissecting the complexities of learning with infinite hidden markov models. December 2023. doi: 10.1101/2023.12.22.573001. URL http://dx.doi.org/10.1101/2023.12.22.573001.

Pablo Samuel Castro, Nenad Tomasev, Ankit Anand, Navodita Sharma, Rishika Mohanta, Aparna Dev, Kuba Perlin, Siddhant Jain, Kyle Levin, Noemi Elteto, Will Dabney, Alexander Novikov, Glenn C Turner, Maria K Eckstein, Nathaniel D. Daw, Kevin J Miller, and Kim Stachenfeld. Discovering symbolic cognitive models from human and animal behavior. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=dhRXGWJ027.

Shirui Chen, Linxing Preston Jiang, Rajesh P. N. Rao, and Eric Shea-Brown. Expressive probabilistic sampling in recurrent neural networks. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Christopher J. Cueva, A. Ardalan, Misha Tsodyks, and Ning Qian. Recurrent neural network models for working memory of continuous variables: activity manifolds, connectivity patterns, and dynamic codes. *ArXiv*, abs/2111.01275, 2021. URL https://api.semanticscholar.org/CorpusID:240419711.

Laura Driscoll, Krishna Shenoy, and David Sussillo. Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. August 2022. doi: 10.1101/2022.08.15.503870. URL https://doi.org/10.1101/2022.08.15.503870.

Rodrigo Echeveste, Laurence Aitchison, Guillaume Hennequin, and Máté Lengyel. Cortical-like dynamics in recurrent circuits optimized for sampling-based probabilistic inference. *Nature Neuroscience*, 23(9):1138–1149, August 2020. doi: 10.1038/s41593-020-0671-1. URL https://doi.org/10.1038/s41593-020-0671-1.

Fredrik Edin, Torkel Klingberg, Pär Johansson, Fiona McNab, Jesper Tegnér, and Albert Compte. Mechanism for top-down control of working memory capacity. *Proceedings of the National Academy of Sciences*, 106(16):6802–6807, April 2009. doi: 10.1073/pnas.0901894106. URL https://doi.org/10.1073/pnas.0901894106.

S. M. Emrich and S. Ferber. Competition increases binding errors in visual working memory. *Journal of Vision*, 12(4):12–12, April 2012. ISSN 1534-7362. doi: 10.1167/12.4.12. URL http://dx.doi.org/10.1167/12.4.12.

Matthias Fritsche, Eelke Spaak, and Floris P de Lange. A bayesian and efficient observer model explains concurrent attractive and repulsive history biases in visual perception. *eLife*, 9, June 2020. doi: 10.7554/elife.55389. URL https://doi.org/10.7554/elife.55389.

N. Gorgoraptis, R. F. G. Catalao, P. M. Bays, and M. Husain. Dynamic updating of working memory resources for visual objects. *Journal of Neuroscience*, 31(23):8502–8511, June 2011. doi: 10.1523/jneurosci.0208-11.2011. URL https://doi.org/10.1523/jneurosci.0208-11.2011.

James B. Heald, Máté Lengyel, and Daniel M. Wolpert. Contextual inference underlies the learning of sensorimotor repertoires. *Nature*, 600(7889):489–493, November 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-04129-3. URL http://dx.doi.org/10.1038/s41586-021-04129-3.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Li Ji-An, Marcus K. Benna, and Marcelo G. Mattar. Discovering cognitive strategies with tiny recurrent neural networks. *Nature*, 644(8078):993–1001, July 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09142-4. URL http://dx.doi.org/10.1038/s41586-025-09142-4.

Benjamin S. H. Lyo and Cristina Savin. Complex priors and flexible inference in recurrent circuits with dendritic nonlinearities. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=S5aUhpuyap.

Wei Ji Ma, Masud Husain, and Paul M Bays. Changing concepts of working memory. *Nature Neuroscience*, 17(3):347–356, February 2014. ISSN 1546-1726. doi: 10.1038/nn.3655. URL http://dx.doi.org/10.1038/nn.3655.

Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, November 2013. ISSN 1476-4687. doi: 10.1038/nature12742. URL http://dx.doi.org/10.1038/nature12742.

Loic Matthey, Paul Bays, and Peter Dayan. A probabilistic palimpsest model of visual short-term memory. *PLOS Computational Biology*, 11(1):e1004003, January 2015. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004003. URL http://dx.doi.org/10.1371/journal.pcbi.1004003.

Jessica M.V. McMaster, Ivan Tomić, Sebastian Schneegans, and Paul M. Bays. Swap errors in visual working memory are fully explained by cue-feature variability. *Cognitive Psychology*, 137:101493, September 2022. doi: 10.1016/j.cogpsych.2022.101493. URL https://doi.org/10.1016/j.cogpsych.2022.101493.

Matthew F. Panichello and Timothy J. Buschman. Shared mechanisms underlie the control of working memory and attention. *Nature*, 592(7855):601–605, March 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03390-w. URL http://dx.doi.org/10.1038/s41586-021-03390-w.

Jose L. Pardo-Vazquez, Juan R. Castiñeiras-de Saa, Mafalda Valente, Iris Damião, Tiago Costa, M. Inês Vicente, André G. Mendonça, Zachary F. Mainen, and Alfonso Renart. The mechanistic foundation of weber's law. *Nature Neuroscience*, 22(9):1493–1502, August 2019. ISSN

1546-1726. doi: 10.1038/s41593-019-0439-7. URL http://dx.doi.org/10.1038/s41593-019-0439-7.

Puria Radmard, Paul M. Bays, and Máté Lengyel. A flexible bayesian non-parametric mixture model reveals multiple dependencies of swap errors in visual working memory, 2025. URL https://arxiv.org/abs/2505.01178.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, November 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL http://dx.doi.org/10.7551/mitpress/3206.001.0001.

Arbora Resulaj, Roozbeh Kiani, Daniel M. Wolpert, and Michael N. Shadlen. Changes of mind in decision-making. *Nature*, 461(7261):263–266, August 2009. ISSN 1476-4687. doi: 10.1038/nature08275. URL http://dx.doi.org/10.1038/nature08275.

Sebastian Schneegans and Paul Bays. Neural architecture for feature binding in visual working memory. *The Journal of Neuroscience*, 37(14):3913–3925, March 2017. ISSN 1529-2401. doi: 10.1523/jneurosci.3493-16.2017. URL http://dx.doi.org/10.1523/JNEUROSCI.3493-16.2017.

Sebastian Schneegans and Paul M. Bays. No fixed item limit in visuospatial working memory. *Cortex*, 83:181–193, October 2016. ISSN 0010-9452. doi: 10.1016/j.cortex.2016.07.021. URL http://dx.doi.org/10.1016/j.cortex.2016.07.021.

Wayne W.M. Soo and Máté Lengyel. Training stochastic stabilized supralinear networks by dynamics-neutral growth. October 2022. doi: 10.1101/2022.10.19.512820. URL https://doi.org/10.1101/2022.10.19.512820.

Jake P. Stroud, Kei Watanabe, Takafumi Suzuki, Mark G. Stokes, and Máté Lengyel. Optimal information loading into working memory explains dynamic coding in the prefrontal cortex. *Proceedings of the National Academy of Sciences*, 120(48), November 2023. ISSN 1091-6490. doi: 10.1073/pnas.2307991120. URL http://dx.doi.org/10.1073/pnas.2307991120.

Freek van Ede, Jovana Deden, and Anna C. Nobre. Looking ahead in working memory to guide sequential behaviour. *Current Biology*, 31(12):R779–R780, June 2021. ISSN 0960-9822. doi: 10.1016/j.cub.2021.04.063. URL http://dx.doi.org/10.1016/j.cub.2021.04.063.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Christopher Versteeg, Jonathan D. McCart, Mitchell Ostrow, David M. Zoltowski, Clayton B. Washington, Laura Driscoll, Olivier Codol, Jonathan A. Michaels, Scott W. Linderman, David Sussillo, and Chethan Pandarinath. Computation-through-dynamics benchmark: Simulated datasets and quality metrics for dynamical models of neural activity. February 2025. doi: 10.1101/2025.02.07.637062. URL http://dx.doi.org/10.1101/2025.02.07.637062.

Saurabh Vyas, Matthew D. Golub, David Sussillo, and Krishna V. Shenoy. Computation through neural population dynamics. *Annual Review of Neuroscience*, 43(1):249–275, July 2020. ISSN 1545-4126. doi: 10.1146/annurev-neuro-092619-094115. URL http://dx.doi.org/10.1146/annurev-neuro-092619-094115.

Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, June 1989. ISSN 1530-888X. doi: 10.1162/neco.1989.1.2.270. URL http://dx.doi.org/10.1162/neco.1989.1.2.270.

Yudi Xie, Yu Duan, Aohua Cheng, Pengcen Jiang, Christopher J. Cueva, and Guangyu Robert Yang. Natural constraints explain working memory capacity limitations in sensory-cognitive models, March 2023. URL http://dx.doi.org/10.1101/2023.03.30.534982.

Cheng Xue, Sol K. Markman, Ruoyi Chen, Lily E. Kramer, and Marlene R. Cohen. Task interference as a neuronal basis for the cost of cognitive flexibility. March 2024. doi: 10.1101/2024.03.04.583375. URL http://dx.doi.org/10.1101/2024.03.04.583375.

Guangyu Robert Yang, Madhura R. Joglekar, H. Francis Song, William T. Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22(2):297–306, January 2019. doi: 10.1038/s41593-018-0310-2. URL https://doi.org/10.1038/s41593-018-0310-2.

Kechen Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *Journal of Neuroscience*, 16(6):2112–2126, 1996.

A BAYESIAN NON-PARAMETRIC MODEL OF SWAP ERRORS (BNS)

Stimulus array $Z = \{\boldsymbol{z}^{(n)} : n = 1, 2\}$ is denoted by the vector of $p$robe (location) and $r$eport (colour) feature values: $\boldsymbol{z}^{(n)} = [z_p^{(n)}, z_r^{(n)}] \in S^1 \times S^1$. Item n^* is cued after the first delay. The probe-dependent BNS Radmard et al. (2025) generative process defines a mixture distribution over estimates (recalled colour, y), based on the circular distance of the distractor from the cued item (i.e. in the probe dimension). The so-called swap function f dictates the form of this dependence. For two items it is:

$$
\begin{aligned}
\boldsymbol{x}^{(n)} &= z_p^{(n)} \ominus z_p^{(n^*)}, n = 1, 2 \\
f &\sim p_f(f) \\
\tilde{\pi}^{(n)} &= \begin{cases} \tilde{\pi}_{\text{correct}} & n = n^* \\ f(z_p^{(n)} \ominus z_p^{(n^*)}) & n \neq n^* \end{cases} \\
\boldsymbol{\pi} &= \text{Softmax}(\tilde{\boldsymbol{\pi}}) \\
\beta &\sim \text{Cat}(\boldsymbol{\pi}) \\
y &= \varepsilon \oplus z_r^{(\beta)} \qquad\qquad \varepsilon \sim p_\varepsilon(\cdot; \phi)
\end{aligned}
$$

The associated graphical model is shown in Figure A.

To generate synthetic data, we fix one target swap function f^*, and perform the generative process up to component β, before embedding the samples into $\mathbb{R}^2$ and adding noise there, rather than using a circular emission distribution p_ϵ. We also generate both feature values with a minimum margin of $\pi/4$, akin to previous multiitem tasks Schneegans and Bays (2017); McMaster et al. (2022):

1. Generate Z with minimum feature value margins
2. Generate y with BNS, using target function f^*
3. Embed sample and add noise $\boldsymbol{x}_T^* \sim \mathcal{N}\left(AW_x^\intercal \begin{bmatrix}\cos(y) \\ \sin(y)\end{bmatrix}, \sigma_x^2 I\right)$

The samples of $\boldsymbol{x}_T^*$ forms the target distribution for the DDPM training procedure (Appendix C).

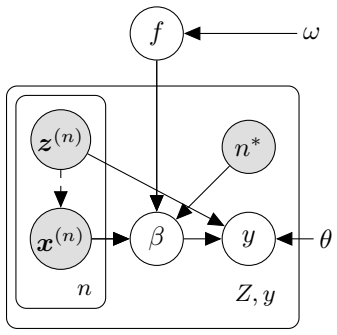

Figure 6: Graphical model of BNS, *adapted from Radmard et al. (2025)*. θ contains parameters for emission distribution p_ε. ω contains GP prior parameters and $\tilde{\pi}_{\text{correct}}$

To fit BNS to the outputs of the trained RNN, we interpret the colour estimate as the argument of the output at the final trial timestep:

$$
y = \arctan\left(\frac{[W_x \boldsymbol{r}_T]_1}{[W_x \boldsymbol{r}_T]_2}\right) \tag{4}
$$

and use variational inference to fit $q(f)$, which approximates the dataset posterior over swap function f, which is equipped with a Gaussian process Rasmussen and Williams (2005) prior p_f. We defer details to Radmard et al. (2025). We do not consider the uniform component used in the introductory work, given the rarity of RNN behavioural samples away from the target modes.

In both cases, for the single distractor case, BNS admits a tractable probability of swapping given a probe distance:

$$p(\beta \neq n^*|\Delta z_p) = \left\langle p(\beta \neq n^*|f, \Delta z_p) \right\rangle_{p(f)} = \left\langle \frac{e^{\Delta z_p}}{e^{\Delta z_p} + e^{\tilde{\pi}_{\text{correct}}}} \right\rangle_{p(f)} \tag{5}$$

For the target swap function, this can be evaluated directly ($p(f) = \delta(f^* - f)$), forming the dotted lines in Figure 3A. For the fitted behaviour, this can be estimated using i.i.d. samples from the inferred approximate posterior ($p(f) = q(f)$).

B ARCHITECTURE AND TASK DETAILS

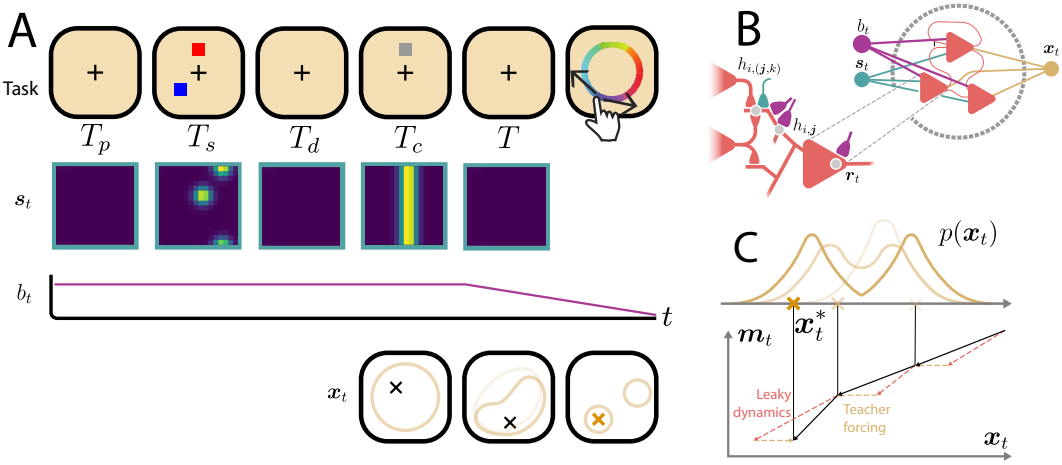

Figure 7: Schematic of architecture and training. **A** (top) task with durations; (middle) n_s by n_s sized conjunctive tuning curve representations of sensory inputs, and representation of time along trial (constant before denoising starts); (bottom) denoising to a mixture of Gaussians target distribution. **B** dendritic tree with sensory and time inputs. **C** teacher-forcing, applied to the behavioural subspace throughout the second delay period during training.

Our RNN models a population of n densely connected cortical pyramidal neurons Lyo and Savin (2024). Each neuron integrates inputs from all other neurons in the population via a dendritic tree. Distal tree nodes are provided with sensory information, and all nodes are provided with a representation of time along the trial, necessary for training it as a diffusion model.

Dendritic tree nodes are organised into layers, with each node integrating activity from nodes one layer more distal from the soma than itself. Each presynaptic neuron's axons synapse onto all leaf nodes of this tree, i.e. the layer most distal to the postsynaptic neuron, including its own. This comprises a weighted sum followed by a ReLU non-linearity (Figure 7 B):

$$h_{\boldsymbol{i}}^k = \left[\sum_j w_{(\boldsymbol{i},j)}^k h_{(\boldsymbol{i},j)}^k + b_{\boldsymbol{i}}^k(t) \right]_+ \tag{6}$$

where k indexes over neurons, $\boldsymbol{i}$ indexes a particular path of nodes in the dendritic tree, and j enumerates distal nodes connecting to $\boldsymbol{i}$. The population's n neurons[2] receive inputs from a dendritic tree with L layers, each with a branching factor of B_l from the previous layer. This results in each neuron axon synapsing onto $n \prod_{l=1}^{L} B_l$ dendritic leaf nodes. The rate vector $\boldsymbol{r}$ is projected via one set of weights $W_r \in \mathbb{R}^{nB_1 \ldots B_L \times n}$, then used as inputs to the most distal layer, instead of h.

Each node is also provided with a bias term. Before the cue offset ($t < 0$), when the network is still accumulating information, this is a constant bias b. After cue offset, when the network is denoising

[2]Undeclared numerical constants are defined and provided values in Appendix E

during the WM second delay, this becomes a smoothly changing n_t-dimensional representation of time Ho et al. (2020); Vaswani et al. (2017) projected to a scalar with a unique set of weights for each dendritic node.

To provide a sensory input to the 'index-cued' network, we separate channels for stimuli and cue:

$$\boldsymbol{s}_t = \begin{cases} [z_r^{(1)}, z_r^{(2)}, \mathbf{0}^\intercal]^\intercal & t \text{ in stim. presentation} \\ [0, 0, \tilde{\boldsymbol{c}}_1^\intercal]^\intercal & t \text{ in cue presentation}, n^* = 1 \\ [0, 0, \tilde{\boldsymbol{c}}_2^\intercal]^\intercal & t \text{ in cue presentation}, n^* = 2 \end{cases} \tag{7}$$

where $\tilde{\boldsymbol{c}}_n \in \mathbb{R}^{n_c}$ are learned embeddings of each cue case.

To provide a sensory input to the 'feature-cued' network, we represent these stimuli with activations of a palimpsest conjunctively tuned neural population (Figure 7A [second row]; Matthey et al. (2015); Schneegans and Bays (2017)), then combine the tuning curve information before projection to the neural population using two sets of weights W_p, W_r:

$$S_t \in \mathbb{R}^{n_s \times n_s} \quad S_{t,ij} = \begin{cases} \sum_{n=1}^{2} \exp\left(\cos(z_p^{(n)} - \bar{z}_p^{(i)}) + \cos(z_r^{(n)} - \bar{z}_r^{(j)})\right) & t \text{ in stim. presentation} \\ \exp\left(\cos(z_p^{(n^*)} - \bar{z}_p^{(i)})\right) & t \text{ in cue presentation} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$\boldsymbol{s}_t = \mathrm{vec}(W_p S_t W_r^t) \in \mathbb{R}^{n_i^2} \tag{9}$$

where $\bar{z}_p, \bar{z}_r$ are fixed and evenly spaced preferred stimulus values for the probe and feature dimensions. In both cases, a projection $W_s \boldsymbol{s}_t \in \mathbb{R}^{nB_1 \ldots B_L}$ is provided to the most distal dendritic nodes.

The final recurrent output of the network is the somatic activity, with added noise:

$$\mathcal{F}(\boldsymbol{r_t}, \boldsymbol{s}_t, \boldsymbol{\eta}; \theta_r) = \boldsymbol{h}_\emptyset + \sigma_t^2 \eta \qquad \boldsymbol{\eta} \sim \mathcal{N}(0, I_n) \tag{10}$$

with θ_r containing all the dendritic tree weights, sensory input parameters, time embedding projection weights, etc.. $\boldsymbol{h}_\emptyset = [h_\emptyset^1, ..., h_\emptyset^n]^\intercal$ is. the somatic activity. As with time representation b, σ_t^2 is held at a constant maximum value before cue offset ($\sigma_{<0}^2 = \sigma_t^2$), then quenched according to the diffusion noise schedule's posterior variance (Appendix E). The overall time-discretised network dynamics are therefore:

$$\boldsymbol{r}_{t+1} = \left[\left(1 - \frac{\mathrm{d}t}{\lambda}\right)\boldsymbol{r}_t + \frac{\mathrm{d}t}{\lambda}\mathcal{F}(\boldsymbol{r}_t, \boldsymbol{s}_t; \theta_r)\right] + \sigma_t \boldsymbol{\epsilon} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I) \tag{11}$$

Note that additive noise is applied to the rates, rather than to the membrane activity, meaning that rate values can fall below zero. Therefore, neural rates vector $\boldsymbol{r}$ should be interpreted as deviations of the population firing rates from some baseline firing rate.

C TRAINING DETAILS

The network trained to produce swap errors are optimised on a joint loss function. The primary terms of this loss function are the stepwise denoising errors (see below; Ho et al. (2020)), with the transition kernel mean provided by discretising the continuous RNN dynamics (equation 3) projected to the behavioural subspace. Unlike traditional DDPMs, the transition kernel takes in an n dimensional rate vector, and outputs a 2 dimensional denoised mean in the behavioural subspace. Without projection W_x, this dictates the full network dynamics throughout the diffusion period (equation 11), but is followed by a teacher forcing step in the diffusion period during training (algorithm 2). As per the advice of seminal work on DDPMs, we weight each term of this loss equally, even though this violates its original formulation as an ELBO Ho et al. (2020).

The loss function also penalises the deviation of the distribution of the behavioural subspace activity at cue offset $\boldsymbol{x}_0$ from the base (unit normal) distribution across many trials. This is a necessary starting point for the denoising process Ho et al. (2020) - because the trajectory is not teacher-forced prior to cue offset, we must explicitly regularise it. We achieve this by applying an L2/Frobenius penalty to the deviation of the first two moments of activity at the start of denoising (i.e. the first timestep of delay 2) from this target distribution Soo and Lengyel (2022). Finally, we apply an L2 regulariser on neural activity across the trial, including prior to the denoising period Yang et al. (2019); Stroud et al. (2023).

Normally, training DDPM parameters across different timesteps, i.e. $p_\phi(\cdot|\cdot', \tau)$ can be done in parallel across τ: data samples $\boldsymbol{x}_T$ are noised to a valid sample of $q(\boldsymbol{x}_\tau|\boldsymbol{x}_T)$ with the noising process accumulated over timesteps, then the DDPM objective can be calculated independently of the noised state at other timesteps. This is because DDPMs are Markovian - trained parameters ϕ can denoise a sample independently of previous timesteps. This is not possible when training RNNs to perform memory tasks. The necessary computations - memory maintenance during the delay period and retrieval during the cueing period - are implemented via non-linear *dynamical motifs* (Vyas et al., 2020; Driscoll et al., 2022; Versteeg et al., 2025) which are also driven by the remainder of the neural state space which is not projected to the response space. As such, training RNNs requires simulation of full task trials in sequence. As neural activity trajectories will likely veer out of the sequence of distributions defined by the noising process q, this sequential training risks detrimentally biasing the training distribution. To solve this, we use *teacher-forcing* to ensure the correct distribution of training targets is satisfied (Appendix C; Williams and Zipser (1989)).

C.1 CHANGING OPTIMISATION WINDOW FOR TASK-OPTIMAL NETWORKS

Task optimal networks are trained on a simple MSE loss function:

$$\sum_{\tau=0}^{T^*} \left\| AW_x^\intercal \begin{bmatrix} \cos(z_r^{(n^*)}) \\ \sin(z_r^{(n^*)}) \end{bmatrix} - W_x \boldsymbol{r}_{T-\tau} \right\|_2^2 \tag{12}$$

i.e. they are trained to minimise distance to the point on a circle in $\mathbb{R}^2$ with the same argument as the task-optimal colour estimate $z_r^{(n^*)}$. Training with a similar, output-based MSE loss function against a multimodal target distribution would lead to mode-averaging, warranting the process-based MSE loss we have developed in this work. This is illustrated in Figure 8, which shows samples generated from a network trained in this way.

Note that only the final network rate vector $\boldsymbol{r}_T$ is behaviourally relevant, as it is used to evaluate the colour estimate of the network for that trial (Appendix A). We refer to the number of timesteps T^* before this critical timestep which the optimisation kicks in as the optimisation window. We maintained the same duration and neural noise schedules of each trial period across all networks (Appendix E). For a different, simpler VWM task with a single, variable delay duration, it has previously been shown that adjusting the optimisation window alters the neural coding schemes used for VWM Stroud et al. (2023). Specifically, it affected the cross-temporal decodability of task variables in storage for the duration of the delay before the estimate is produced. In the absence of neural evidence to compare to for the neural coding scheme across this delay Panichello and Buschman (2021), we presented the extreme case of $T^* = T$ Figure 4E (left), i.e. the MSE loss function in equation 12 was applied to all timesteps of the second delay. Figure 9 shows the equivalent plot for multiple task-optimal networks with different optimisation windows. In all cases,

Algorithm 2 Training with teacher forcing. Here, $\bar{\boldsymbol{r}}$, $\boldsymbol{\Sigma}_{\boldsymbol{r}}$ are the empirical mean and covariance of $\boldsymbol{r}_0^k$ across many trials k with the same experimental conditions presented.

repeat
  Generate stimulus set and cue $(\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, n^*)$
  for each independent trial, indexed by k **do**
    From BNS, generate a sample from the desired behavioural distribution: $\boldsymbol{x}_T^{*,k}$, and apply the noising process (equation 2)[3] to generate noised data $\boldsymbol{x}_{0:T-1}^{*,k}$
    Initialise network activity $\boldsymbol{r}^k \sim \mathcal{N}(0, \sigma_r^2 I)$
    Run network dynamics for the prestimulus, stimulus, delay, and cue periods (Figure 7A, Appendix B) with noisy discretised leaky dynamics (equation 3 without projection), finally arriving at network activity at cue offset $\boldsymbol{r}_0^k$
    for $t = 0$ **to** T **do**
      Replace behavioural subspace: $\boldsymbol{r}_t^k \leftarrow \boldsymbol{r}_t^k + \boldsymbol{x}_t^* - W_x W_x^\intercal \boldsymbol{r}_t^k$
      Calculate mean transition kernel $\hat{\boldsymbol{\mu}}_{\theta_r^k}(\boldsymbol{r}_t, t)$ (equation 3) and stash
      Run dynamics one step (equation 11) and add scheduled process noise (Appendix E).
    end for
    Train on joint loss function

$$\sum_k \left[\sum_{t=0}^{T} \|\boldsymbol{\mu}_q(\boldsymbol{x}_t^{*,k}, \boldsymbol{x}_T^{*,k}) - \hat{\boldsymbol{\mu}}_{\theta_r}(\boldsymbol{r}_t^k, t)\|_2^2 + \gamma_2 \sum_{\text{all timesteps } s} \|\boldsymbol{r}_s^k\|_2^2 \right] + \gamma_1 \left(\|\bar{\boldsymbol{r}}\|_2^2 + \|\boldsymbol{\Sigma}_{\boldsymbol{r}}\|_{\text{Fr}}^2 \right)$$

  end for
until converged

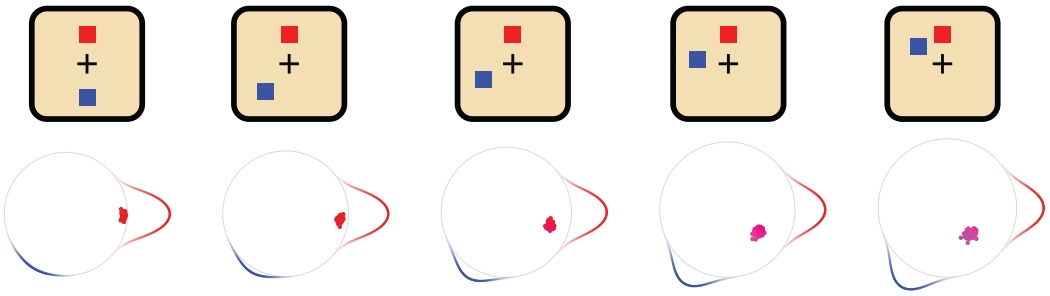

Figure 8: Network trained on a MSE loss (equation 12) against samples from a multimodal target distribution of estimates. The samples are coloured by colour estimate which they are interpreted as, and the modes of the target distribution for each trial is coloured by the nominal colour estimate at which it is centered. As the probe feature values get closer, the blue distractor is more likely to cause a swap error, as reflected in the target distributions. The target multimodality is not achieved, instead the network interpolates between the two distinct modes.

All estimates were tightly bound to the circle in the behavioural subspace, and no swap errors were observed (Figure 3C [top]). Again, we see that the hallmarks of neural data are not captured, although in networks which are afforded time before being penalised for behavioural deviation, there is a shared alignment in the target-distractor colour planes during the optimisation window. Regardless, without swap errors, it is difficult to make connections to behavioural data, as was the original aim of our novel training method. Investigating the link between this property of neural activity and optimal information loading is key to understanding both the mechanism of memory retrieval, and the dynamical source of swap errors during this time (§5), and is left to future works.

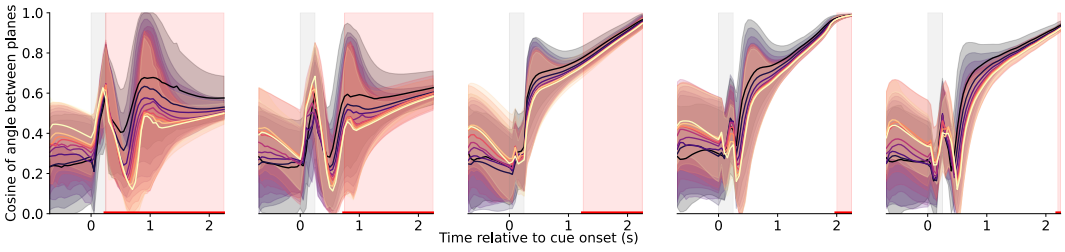

Figure 9: Plane alignment for different task-optimal networks (see appendix text). Optimisation window is highlighted in red

C.2 Training curriculum

Training feature-cued networks directly as described above, be it on task-optimality or behavioural realism, proved inefficient and time-costly. To speed up training, feature-cued networks were first trained on a much shorter version of the task, with a second delay of $T = T^* = 5$ timesteps. This ensured that networks were initialised with the ability to, at least, retrieve cued items from memory. From this checkpoint, each network type was then trained independently to prevent cross-contamination of the specialised mechanisms we sought to develop and study in each model variant.

C.3 Computational resources

All models were trained using a single NVIDIA RTX A5000 24GB GPUs at a time. All optimisation is implemented in `PyTorch`, utilising the Adam optimiser. A learning rate of 0.001 was found satisfactory across all experiments presented, and other optimiser hyperparameters were maintained at the library defaults. A batch size of 32, each with 512 independently simulated trials (indexed by k in Algorithm 2), was used throughout.

Index-cued networks trained for task optimality (behavioural realism) typically took 35000 (300000) batches at 3.0 (3.0) batches per second, resulting in 3.2 (28) hours per run. These experiments typically occupied 15650 MiB of VRAM. Feature-cued networks trained for task optimality (behavioural realism) typically took 2000[4] (300000) batches at 2.4 (2.3) batches per second, resulting in 0.23 (36) hours per run. These experiments typically occupied 15870 MiB of VRAM. The training preamble with a shortened trial duration was typically took 500000 batches at 8.6 batches per second, resulting in an additional 16 hours per initialisation; but this was inherited by multiple downstream networks with different final objectives. These experiments typically occupied 4560 MiB of VRAM. The reduction in memory is due to the shorter time per trial in this preamble.

D Neural signatures of swap errors

Figure 10 provides an extension to the alignment results of Figure 5C (left). There, the item locations are kept fixed, as well as one item's colour. The response-time hidden activity $\boldsymbol{m}_T$, when projected to its first three principal components (PCs), forms a characteristic 2 ring structure. These PCs are shared across initial item locations, as long as the fixed colour is the same (blue histogram). This is not the case when the colours are dissimilar (red histogram). Figure 10 provides the mean value of this histogram for a wider range of fixed colour pairs. We see a cyclical structure of PC3 alignment with respect to the fixed stimulus colour. This provides preliminary evidence of a toroidal structure, which would have to be verified with further topological data analysis Cueva et al. (2021).

[4]Compared to the training preamble (see later in main text) this was just a matter of extending time horizons, rather than learning a new memory retrieval mechanism from scratch

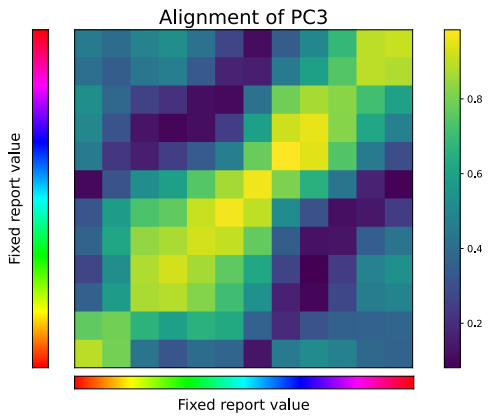

Figure 10: Continuing from Figure 5C – cosine between PC3 for response time activity geometries generated by different fixed colours, averaged over spatial configurations (spatial pairs).

E TABLE OF CONSTANTS AND NOISE SCHEDULE

Name	**Symbol**	**Value**
Minimum angular margin between item feature values	-	$\pi/4$
Radial magnitude of behavioural target	A	2.5
Variance of behavioural target models	σ_x^2	0.2^2
Variance of rate vector initialisation target models	σ_r^2	1.0^2
Network population size	n	16
Number of dendritic tree layers	L	2
Branching factor of each dendritic tree layer	B_1, B_2	10, 10
Dimensionality of global time representation	n_t	16
Shared time representation vector size	-	8
Number of tuning curves for each feature dimension	n_s	16
Feature-cued sensory projection size	n_i^2	6^2
Index-cued cue embedding size	n_c	4
Time discretisation step	$\mathrm{d}t$	0.05
Neural membrane time constant	λ	0.5
Prestimulus duration	T_p	0.15s, 3 timesteps
Stimulus exposure duration	T_s	0.25s, 5 timesteps
Delay 1 duration	T_d	0.75s, 15 timesteps
Cue exporsure	T_c	0.25s, 5 timesteps
Delay 2 (denoising) duration	T_d	2.0s, 40 timesteps
Base distribution regularisation weight	γ_1	0.01
L2 rate regularisation weight	γ_2	0.0001

Table 1: Constants and hyperparameters used in main text

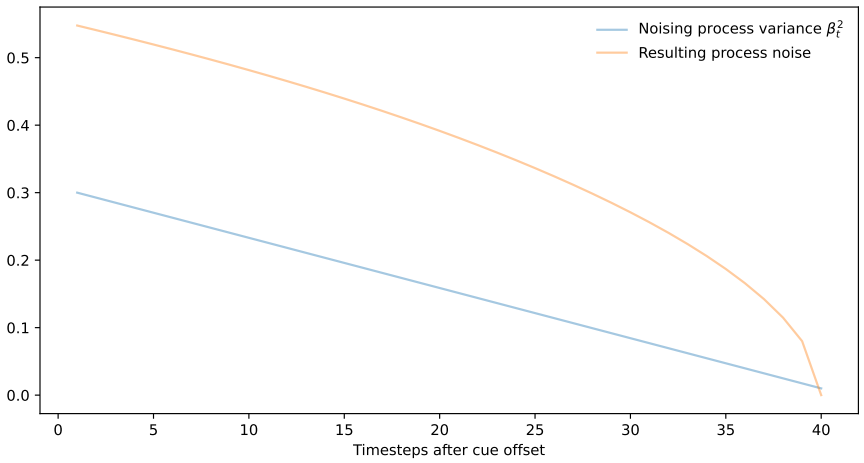

Figure 11: Schedule for corruption noise variance (β_t; equation equation 11) and corresponding generative noise standard deviation (σ_t). The former was a linear interpolation between 0.3 and 0.01 across the 40 timesteps.