nature neuroscience

ARTICLES

https://doi.org/10.1038/s41593-018-0314-y

A diverse range of factors affect the nature of neural representations underlying short-term memory

A. Emin Orhan[1*] and Wei Ji Ma[2,3]

Sequential and persistent activity models are two prominent models of short-term memory in neural circuits. In persistent activity models, memories are represented in persistent or nearly persistent activity patterns across a population of neurons, whereas in sequential models, memories are represented dynamically by a sequential activity pattern across the population. Experimental evidence for both models has been reported previously. However, it has been unclear under what conditions these two qualitatively different types of solutions emerge in neural circuits. Here, we address this question by training recurrent neural networks on several short-term memory tasks under a wide range of circuit and task manipulations. We show that both sequential and nearly persistent solutions are part of a spectrum that emerges naturally in trained networks under different conditions. Our results help to clarify some seemingly contradictory experimental results on the existence of sequential versus persistent activity-based short-term memory mechanisms in the brain.

Short-term memory is a fundamental cognitive function for both humans and other animals. Despite its importance, its neural basis largely remains an open problem. The classical view of how a short-term memory might be implemented in the brain relies on the idea of a fixed point attractor[1,2]. In this view, a memory is maintained via persistent activity of individual neurons. By virtue of their persistent activity, those neurons continue to represent information in the absence of any sensory stimulation. However, persistent activity of individual neurons is not necessary for maintaining information in short-term memory; dynamic activity patterns can also maintain short-term memories[3–5]. According to this alternative view, individual neurons can be active only transiently, whereas the population as a whole maintains the memory through a dynamically changing activity pattern across time.

It has been an ongoing debate as to whether one of these alternative pictures provides a more accurate representation of the neural mechanism (or mechanisms) underlying short-term memory than the other [6,7]. Experimental evidence for both alternatives has been reported previously; for example, some studies[8–12] observed persistent or nearly persistent activity during the delay period of short-term memory tasks, whereas other studies[13–18] observed sequential or dynamic activity patterns. These studies used different tasks, different stimuli, different experimental designs, and sometimes recorded from different areas or even from different species. It is difficult to know which of these differences might be relevant for the observed differences in mnemonic activity patterns. Although this question can, in principle, be addressed experimentally by running many experiments, systematically varying each experimental factor or neural circuit property that could conceivably have an effect on the observed differences, this would be too costly. Instead, we addressed this question by performing these experiments in silico. This allowed us to not only identify the relevant factors, but also understand mechanistically why those factors have the effects that they do.

More specifically, we trained recurrent neural networks on a range of short-term memory tasks and investigated the effects of a diverse array of task- and circuit-related factors on the sequentiality or persistence of the emergent activity patterns: (1) the task; (2) other experimental variables such as delay duration variability or whether the task had a navigation component; (3) whether the network was previously trained on another task; (4) intrinsic network properties such as the intrinsic timescale of individual neurons and the strength of coupling between the neurons; and (5) Hebbian short-term synaptic plasticity.

We found that both sequential and nearly persistent solutions are part of a spectrum that emerges naturally in trained networks under different conditions. Tasks with higher temporal complexity, fixed delay durations, stronger network coupling between neurons, prior training in another task, and task-irrelevant, motion-related dynamic cues that arise in navigation-like tasks all increased the sequentiality of the emergent solutions. On the other hand, tasks with lower temporal complexity, variable delay durations, weak coupling between neurons, and symmetric Hebbian short-term synaptic plasticity reduced the sequentiality of the emergent solutions. Furthermore, having complete access to the networks and their behavior allowed us to develop a detailed mechanistic understanding of the circuit mechanism that generates sequential versus persistent mnemonic activity and why the aforementioned factors have the effects that they do on the sequentiality or persistence of the neural responses.

Results

Experimental setup. *Networks.* In our main simulations, we used vanilla recurrent neural networks with rectified linear recurrent units (Fig. 1a; see Methods). The input to the network was provided in the form of a population of Poisson neurons, emitting independent Poisson counts at each time step of the simulation. Experimental evidence suggests that both the intrinsic time constants of individual neurons and the overall coupling strength

[1]Department of Neuroscience, Baylor College of Medicine, Houston, TX, USA. [2]Center for Neural Science, New York University, New York, NY, USA. [3]Department of Psychology, New York University, New York, NY, USA. *e-mail: aeminorhan@gmail.com

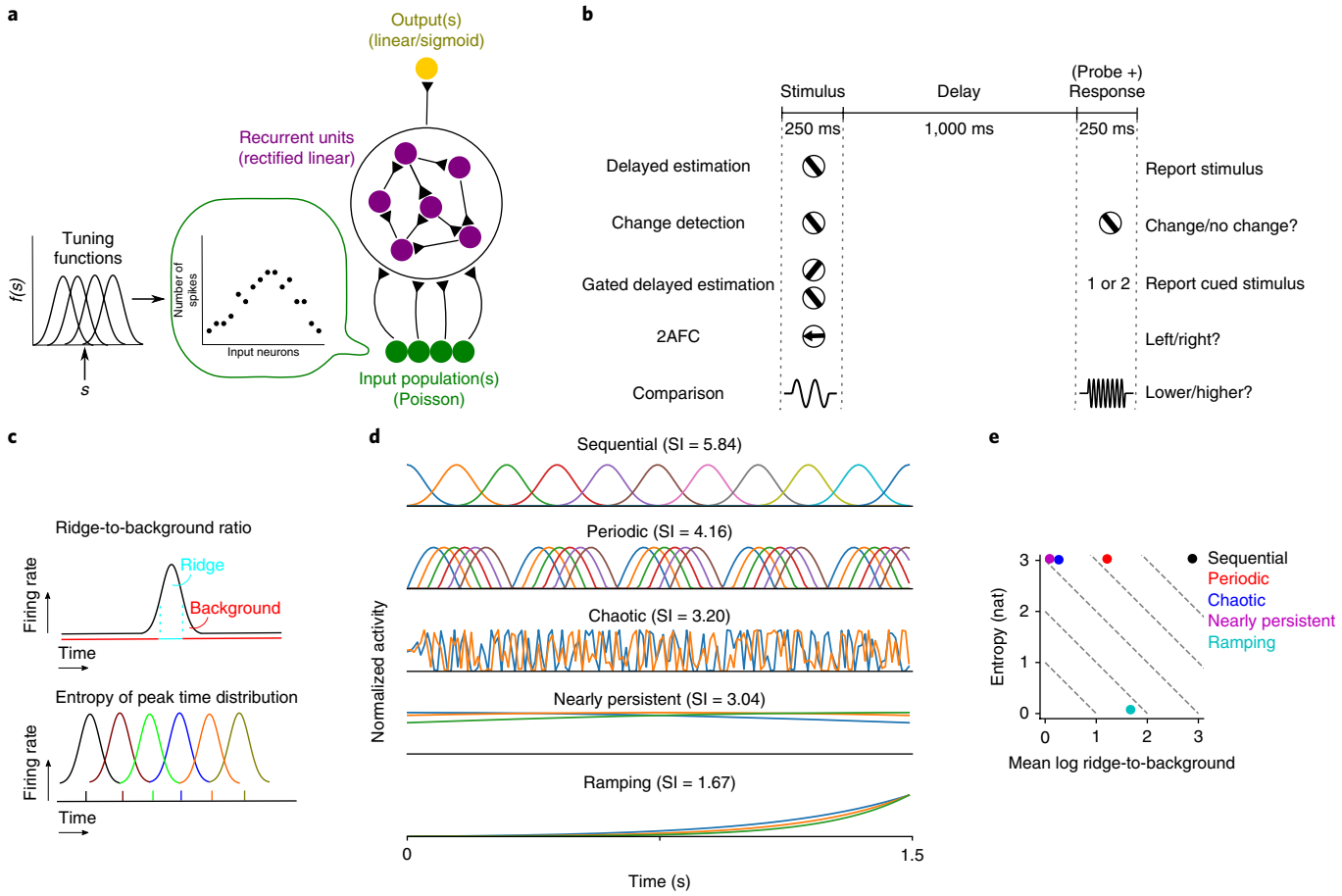


Fig. 1 | Experimental setup. a, Schematic diagram of recurrent networks. The input neurons are Poisson neurons providing noisy information about the stimulus or stimuli. These neurons project onto the recurrent neurons, which are modeled as ReLUs. Recurrent neurons in turn project onto the output unit or units, which are either linear or sigmoidal in different tasks. **b**, The five main experimental tasks and the common trial structure. **c**, Two factors determining the SI: the ridge-to-background ratio[16] measures the temporal localization of the activity of individual units; the entropy of the peak time distribution measures the uniformity of the peak response times of the units in a given trial. The SI for a given trial is then given by the sum of the mean log ridge-to-background ratio of the recurrent units and the entropy of the peak time distribution. **d**, Example idealized single-trial activity patterns with the corresponding SIs indicated at the top of each panel. The different colors represent the temporal activity patterns of a subset of individual units. These example trials were generated with the same number of recurrent units and time steps as in the simulations in the rest of the article. Hence, the SI values shown in the figure are directly comparable to the SI values reported elsewhere in the article. A small amount of noise, independent across neurons and time, was added to the responses of all neurons to break possible ties in determining peak response times. **e**, How the example trials shown in **d** score along each of the two dimensions defining the SI. The dashed lines represent several iso-SI contours. All examples except for the ramping one score close to maximum on the entropy dimension, hence their SIs are largely distinguished by the mean ridge-to-background ratio. Note that the nearly persistent example was generated by broadening the temporal activity profiles in the sequential example. Thus, it has the same peak time entropy as the sequential example, but has a much smaller mean ridge-to-background ratio. The ramping example, on the other hand, has minimal peak time entropy and a medium mean ridge-to-background ratio.

between them can vary significantly across the cortex[19,20]. To tease apart the potential effects of these two factors, we initialized the recurrent connectivity matrix as $\lambda_0 I + \sigma_0 \Sigma_{\text{off}}$, where λ_0 and σ_0 are hyperparameters controlling the amount of initial self-recurrence and recurrence from the rest of the network respectively, I is the identity matrix, and Σ_{off} is an off-diagonal matrix whose off-diagonal entries are drawn independently from a zero-mean normal distribution with standard deviation $1/\sqrt{n}$, where n is the number of recurrent units in the network. Given that regularization of the network parameters (or the recurrent activity) can sometimes significantly impact the nature of the emergent solutions[21–23], we also placed an L2 norm regularizer on the network parameters throughout training and controlled its strength through another hyperparameter, ρ. We repeated our main simulations for 800 different hyperparameter configurations drawn over a grid in the (λ_0, σ_0, ρ) space. On this grid, λ_0 took ten uniformly spaced values between 0.8 and 0.98, σ_0 took ten uniformly spaced values between 0 and 0.4025, and ρ took seven logarithmically spaced values between 10^{-6} and 10^{-3}, as well as $\rho = 0$. In general, we chose these ranges to be as large as possible, subject to the trainability of the networks, such that values outside of these ranges generally significantly impeded the trainability of the networks. These choices still gave rise to a wide range of initial network dynamics, from quickly decaying to strongly unstable (Supplementary Fig. 1). Qualitatively, increasing λ_0 has the effect of increasing the intrinsic time constant of the individual neurons, making their activity more persistent in response to an input pulse. On the other hand, increasing σ_0 introduces oscillatory components to the network response.

Tasks. To eliminate potential differences resulting from trial structure, we used a common trial structure for all our tasks (Fig. 1b). Each trial started with the presentation of one or two stimuli for

250 ms. A delay period of 1,000 ms then followed. After the delay, there was a response period of 250 ms. In some tasks, a second stimulus or a cue appeared during the response period, in which case the target response depended on this second stimulus or cue.

We considered five main tasks in our experiments (Fig. 1b; see Methods for task details): (1) delayed estimation with one (DE-1) or two stimuli (DE-2), where the task was to report the stimulus or stimuli presented at the beginning of the trial; (2) change detection, where the task was to report whether the stimulus presented before the delay was the same as the stimulus presented after the delay (for example, see Wilken and Ma[24]); (3) gated delayed estimation (GDE), where two stimuli were presented simultaneously at the beginning of the trial and the task was to report the cued one after the delay (for example, see Wilken and Ma[24]); (4) two-alternative forced choice (2AFC), where one of two possible stimuli (for example, left versus right moving dots) was presented at the beginning of the trial and the task was to report which one was presented (for example, see Goard et al.[11], Harvey et al.[16]); and (5) comparison (COMP), where the task was to report whether the stimulus presented before the delay was smaller or larger than the one presented after the delay (for example, Romo et al.[10]).

Quantifying sequentiality. Intuitively, there are two requirements for the recurrent activity of a population of neurons to be considered sequential (Fig. 1c): (1) each neuron should be active only during a short interval compared with the duration of the trial, and (2) the active periods of the neurons should tile the entire duration of the trial approximately uniformly. Thus, we designed a sequentiality index (SI) that takes into account both of these requirements. The SI for a given trial is defined as the sum of the entropy of the peak response time distribution of the recurrent neurons and the mean log ridge-to-background ratio of the neurons, where the ridge-to-background ratio for a given neuron is defined as the mean activity of the neuron inside a small window around its peak response time divided by its mean activity outside this window[16]. The SI for a given experimental condition is then determined by averaging over the SIs of all trials belonging to that condition. Fig. 1d shows some idealized single-trial temporal activity patterns and the corresponding SIs. These examples were generated using the same number of recurrent neurons and time steps as in other simulations in this study; hence, the SI values reported in Fig. 1d are directly comparable to those reported elsewhere in the article. Fig. 1e shows how these example trials score along each of the two dimensions defining the SI, namely the mean log ridge-to-background ratio and the entropy of the peak response time distribution.

Factors affecting the sequentiality of the responses. *Intrinsic circuit properties affect sequentiality.* Fig. 2a schematically illustrates the three main intrinsic circuit properties considered: the initial network coupling, σ_0; the initial intrinsic timescale of individual units, λ_0; and the regularization coefficient ρ. In successfully trained networks, the sequentiality of the recurrent activity increased with σ_0 (Fig. 2b); it did not change significantly with λ_0 (Fig. 2c) and it slightly but significantly decreased with ρ (Fig. 2d). Larger σ_0 values introduce higher-frequency oscillatory dynamics in the initial network, which promotes the emergence of a high-frequency sequential structure in the trained networks. Larger ρ values, on the other hand, have the opposite effect.

The temporal complexity of tasks affects sequentiality. There was significant variability in SI among the tasks (Fig. 3a; see Supplementary Figs. 2–6 for example trials from all tasks under different hyperparameter settings). Indeed, task was the most predictive variable in a linear regression analysis of the SI that included the task variable (coded ordinally) and the three hyperparameters σ_0, λ_0, and ρ: task alone yielded $R^2 = 0.20$ compared with $R^2 = 0.08$ for the next most predictive variable, σ_0. Some tasks, such as comparison or change detection, led to highly sequential responses, whereas other tasks, such as the basic 2AFC task, led to less sequential and more persistent responses (Fig. 3b). We hypothesized that this variability was related to the temporal complexity of the target functions that need to be learned in different tasks, where target function complexity can be formalized as the mean temporal frequency of the target function[25], for example. In change detection, gated delayed estimation, and comparison tasks, the target function depends on the probe (or cue) stimulus presented after the delay period. Thus, these tasks have higher temporal complexity. On the other hand, in delayed estimation and 2AFC tasks no probe is presented after the delay and the target response does not depend on what happens after the delay. Thus, these tasks have lower temporal complexity. Implementing temporally more complex target functions requires higher-frequency temporal basis functions; and sequential activity in the recurrent population provides such a high-frequency temporal basis.

To test this hypothesis more directly, we conducted two simple experiments. First, we trained networks to output sine functions with different temporal frequencies during the response period (upper panel in Fig. 3c). Thus, the target function had the following form: $\sin(2\pi f t/T_{\mathrm{resp}})$, where $0 \le t \le T_{\mathrm{resp}}$, and T_{resp} denotes the duration of the response period. The networks received one-dimensional random input throughout the trial in these tasks. According to our hypothesis, target functions with higher temporal frequency (larger f) should lead to more sequential responses. We observed that this was indeed the case (Fig. 3c): the linear regression of SI on f yielded a slope of 0.60 ± 0.10 ($R^2 = 0.43$, two-sided Wald test, $P < 10^{-7}$).

Second, we introduced a 'tethering' manipulation in our experimental design that increased the temporal complexity of the tasks. Under tethered conditions, we put a strong penalty on recurrent responses deviating from 0 during the last 50 ms of the trial (upper panel in Fig. 3d). An analogous tethering manipulation can be induced experimentally, for example, by optogenetic silencing of a relevant neural circuit toward the end of the trial. Tethering increases the temporal complexity of the task because it forces the network's output to sharply change from the roughly constant value it takes before the onset of tethering. Thus, we expected this manipulation to increase the sequentiality of the responses in successfully trained networks. Tethering indeed led to an overall increase in the sequentiality of the responses (Fig. 3d,e). Notably, in many cases, tethering changed the dynamics throughout the entire trial duration and not just toward the end of the trial (for example, see the representative pair of trials in Fig. 3f).

Hebbian short-term synaptic plasticity affects sequentiality. Short-term synaptic plasticity is a ubiquitous feature of synapses in real neural circuits[26]. A number of theoretical and experimental studies have suggested that short-term synaptic plasticity might be involved in short-term memory by storing information in an 'activity-silent' format in synapses[27–29]. To investigate the effect of short-term synaptic plasticity on the sequentiality of the recurrent activity in our networks, we added a simple symmetric Hebbian short-term synaptic plasticity term to the recurrent weights (see Methods). This Hebbian contribution to the recurrent weights is sometimes known as 'fast weights' in the machine learning literature[30].

Symmetric Hebbian short-term synaptic plasticity decreased the sequentiality of the recurrent activity in trained networks (Fig. 4a). A symmetric contribution to the recurrent connectivity matrix reduces the high-frequency oscillatory dynamics in the network, which in turn reduces the sequentiality of the recurrent activity. We emphasize again the symmetry of the short-term synaptic plasticity rule considered in this study, as asymmetric associative rules (for example, spike-timing-dependent plasticity) can often have opposite effects, as demonstrated in earlier studies[31–33]. We tried several

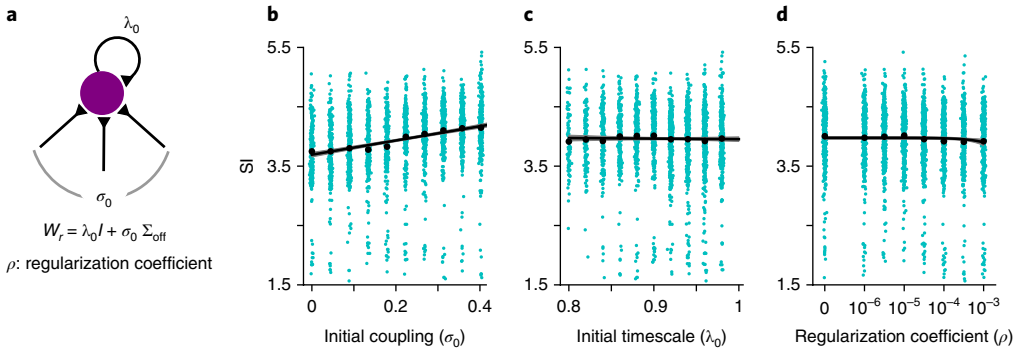


Fig. 2 | Intrinsic circuit properties and their effect on the sequentiality of recurrent activity in trained networks. a, The recurrent connectivity matrix was initialized as $W_r = \lambda_0 I + \sigma_0 \Sigma_{off}$, where λ_0 controls the initial intrinsic timescale of individual units and σ_0 controls the size of the initial coupling between the units. We also varied the strength of the L2 norm regularization on the parameters, controlled by the coefficient ρ. Our basic experiments were repeated with 800 different λ_0, σ_0, and ρ values on a $10 \times 10 \times 8$ grid over the three-dimensional hyperparameter space (λ_0, σ_0, ρ). **b**, The SI increased significantly with σ_0. Linear regression slope: 1.20 ± 0.08, $R^2 = 0.08$ (two-sided Wald test, $n = 2{,}905$ experimental conditions, $P = 0.000$). **c**, The SI did not change significantly with λ_0 (two-sided Wald test, $n = 2{,}905$ experimental conditions, $P = 0.639$). **d**, The SI slightly decreased with ρ. Linear regression slope: -76 ± 31, $R^2 = 0.002$ (two-sided Wald test, $n = 2{,}905$ experimental conditions, $P = 0.015$). Each cyan dot in **b**–**d** corresponds to the mean SI for a particular hyperparameter setting and a particular task. Black dots represent the means. Solid black lines are the linear fits and shaded regions are 95% confidence intervals for the linear regression (confidence intervals are usually too small to be clearly noticeable on the plotted scale).

asymmetric variants of our Hebbian short-term synaptic plasticity rule, but we found these rules to be quite unstable in general and we could not train our networks successfully with these kinds of rules.

Delay duration variability affects sequentiality. Thus far, our simulations assumed a fixed delay duration of 1,000 ms. However, experimenters sometimes use variable delay durations in short-term memory experiments. To test the effect of delay duration variability, we designed versions of each of our tasks with delay duration variability. In these versions, delay duration was one of 100, 400, 700, and 1,000 ms, chosen randomly on each trial. Variability in delay duration significantly decreased the sequentiality of the recurrent activity in successfully trained networks (Fig. 4b). In sequential solutions, the representations of task-relevant variables change over time. Thus, these representations cannot be decoded with a fixed decoder across time. However, the delay duration variability experiments demand that the learned representations be decodable with a fixed decoder at different delay durations, thereby forcing the network to learn more stable representations across time.

Task-irrelevant structured dynamic inputs affect sequentiality. Motion-related signals that animals receive during navigation-type experiments have previously been argued to be crucial for the generation of sequential neural activity observed in rodent experiments[34]. Our results from experiments without such motion-related signals clearly demonstrate that such signals are not necessary for the generation of sequential activity. However, it is still possible that because such signals already have a sequential structure, they may facilitate the generation of sequential activity in the network. To test this hypothesis, we designed navigation versions of our main experiments where, in each trial, the network was assumed to navigate through a linear track at constant speed. The network received noisy population-coded information about its hypothetical location in the linear track, in addition to the task-relevant inputs it received (see Methods). The location information was irrelevant for performing the tasks, hence the network could safely ignore this information. These motion-related, task-irrelevant location signals significantly increased the sequentiality of the recurrent activity in successfully trained networks (Fig. 4c), suggesting that the networks did not completely suppress these signals despite the fact that they were irrelevant to the tasks the networks were trained on.

Learning multiple tasks in sequence affects sequentiality. Our simulations so far assumed that each network is trained on a single task. However, a common situation that arises in many animal experiments is that the same animal may be trained on multiple tasks, usually sequentially. This can happen, for example, when an animal takes part in several different experiments throughout its lifetime, or when it learns to perform different tasks as part of a curriculum strategy for learning a more complex task. To investigate the effects of such sequential multitask learning, we considered networks that learned a pair of tasks sequentially. We only considered the 2AFC–COMP and 2AFC-CD task pairs, trained in either order because (1) these task pairs have the same number and type of inputs and outputs, hence they do not require any changes in the network architecture, and (2) they have maximally different SIs when trained in isolation: the COMP and CD tasks have the largest SIs and the 2AFC task has the smallest SI among all tasks (Fig. 3a). We then compared the SI in the second task of the pair with the SI of the same task when it was trained in isolation. Sequential multitask training led to an overall increase in the SIs compared with the corresponding single-task training conditions (Fig. 5a,b). This might be expected in cases where the network was first trained on a high SI task and then on a low SI task (that is, COMP → 2AFC and CD → 2AFC, although the effect was not significant in the latter case). More surprisingly, however, a significant increase in SI was also observed in the other direction, that is, training in 2AFC → COMP produced a higher SI than training in COMP alone; similarly, training in 2AFC → CD led to a higher SI than training in CD alone. We observed that this was because training a network in any task, including in low SI tasks such as 2AFC, consistently decreased the mean self-recurrence of the units, $\lambda \equiv \langle W_{ii} \rangle$, and increased the size of the fluctuations in the strength of recurrent coupling to the rest of the network, $\sigma \equiv \mathrm{std}(W_{ij,i \neq j})$, compared with the initial weights (Fig. 5c). Thus, for the second task in the pair, the effect of prior training in another task is analogous to an increase in the hyperparameter σ_0, which was shown to increase the SI (Fig. 2b).

Circuit mechanism that generates sequential versus persistent activity. To probe the circuit mechanism generating sequential versus persistent activity in trained networks, we performed an analysis proposed by Rajan et al.[35]. In this analysis, we first ordered the recurrent neurons in the network by their time of peak activity. We then

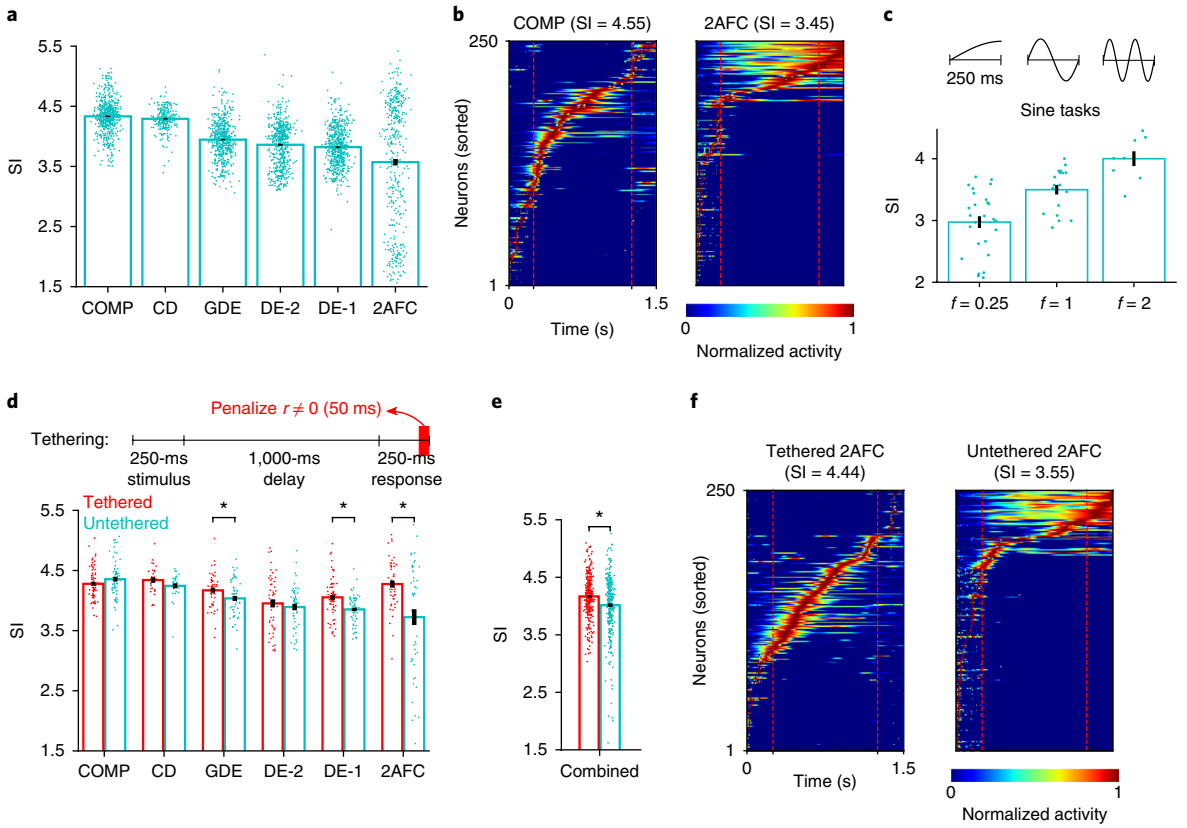


Fig. 3 | The temporal complexity of the task increases the sequentiality of the recurrent activity in trained networks. a, SI in different tasks. DE-1 and DE-2 refer to delayed estimation tasks with one and two stimuli, respectively. Each dot corresponds to the mean SI for a particular setting of the hyperparameters. The error bars represent the mean ± s.e.m. across different hyperparameter settings. **b**, Normalized responses of recurrent units in a pair of example trials from the COMP and 2AFC tasks, respectively, trained under the same hyperparameter setting. The SI values of the trials are indicated at the top of the corresponding panels. We chose representative trials with SI values close to the mean SI for the two tasks. Only the responses of the most active 250 units are shown. The actual networks always consisted of 500 recurrent units. The remaining units were mostly or completely silent throughout the trial. **c**, SI in the sine tasks. In these tasks, the network was trained to output a sine function with temporal frequency f during the response period (target functions are shown in the upper panel). Higher frequency target functions (larger f) led to more sequential responses: linear regression of SI on f yielded a slope of 0.60 ± 0.10 ($R^2 = 0.43$, two-sided Wald test, $n = 54$ experimental conditions, $P = 0.000$). The error bars represent the mean ± s.e.m. across different hyperparameter settings. **d**, Tethering manipulation and its effect on the SI of different tasks. The asterisk indicates a significant difference at $P < 0.05$ (two-sided Welch's t test). The error bars represent the mean ± s.e.m. across different hyperparameter settings. **e**, SI in the tethered versus untethered conditions, combined across all tasks in **d**. The error bars represent the mean ± s.e.m. across different hyperparameter settings and different tasks. Exact sample sizes and P values for any statistical test in **a** and **c**–**e** are reported in Supplementary Table 1. **f**, Normalized responses of recurrent units in a pair of example trials from the tethered and untethered versions of the 2AFC task, respectively, trained under the same hyperparameter setting. We again chose representative trials with SI values close to the mean SIs of the two conditions.

measured the mean and s.d. of the recurrent weights (W_{ij}) as a function of the order difference between two neurons, $i-j$. In trained networks, connections from earlier to later peaking neurons had, on average, larger weights than connections from later to earlier peaking neurons. The mean connection weight was an approximately monotonically increasing function of $i-j$ (Fig. 6a,b). This particular asymmetric structure was absent in untrained random networks (Fig. 6c); and it generated sequential activity in trained networks with increasingly prolonged responses in later peaking neurons in the sequence (Fig. 6e). However, in trained networks with a high SI (SI > 5), a prominent asymmetric peak appeared in the connection weight profile (inset in Fig. 6a). This asymmetric peak corresponds to strengthened connections between temporally close neurons in the sequence at the expense of weakened connections between temporally distant neurons, with connections in the 'forward' direction being strengthened more than those in the opposite direction. This, in turn, led to more strongly sequential responses in the network (Fig. 6d) by reducing the temporal smearing of the responses that took place in networks with a low SI (SI < 2.5), which did not display such a peak in their connection weight profile (Fig. 6b). A simplified model that only incorporated the non-linearity and idealized versions of the mean connection weight profiles shown in Fig. 6a,b captured the essential aspects of the difference between the two cases (Supplementary Fig. 7).

Notably, the preceding analysis suggests that both sequential and persistent activity patterns underlying short-term memory under different conditions emerge as two ends of a spectrum in trained networks, rather than being categorically different solutions.

Robustness of the results to variations in some architectural and experimental choices. In our simulations thus far, we have used recurrent networks of rectified linear units (ReLUs). This particular non-linearity is unbounded on one side; thus, it may be considered biologically unrealistic, even though in trained networks the recurrent units typically did not attain unrealistically large values. Thus, it is important to check whether our main results still hold for a

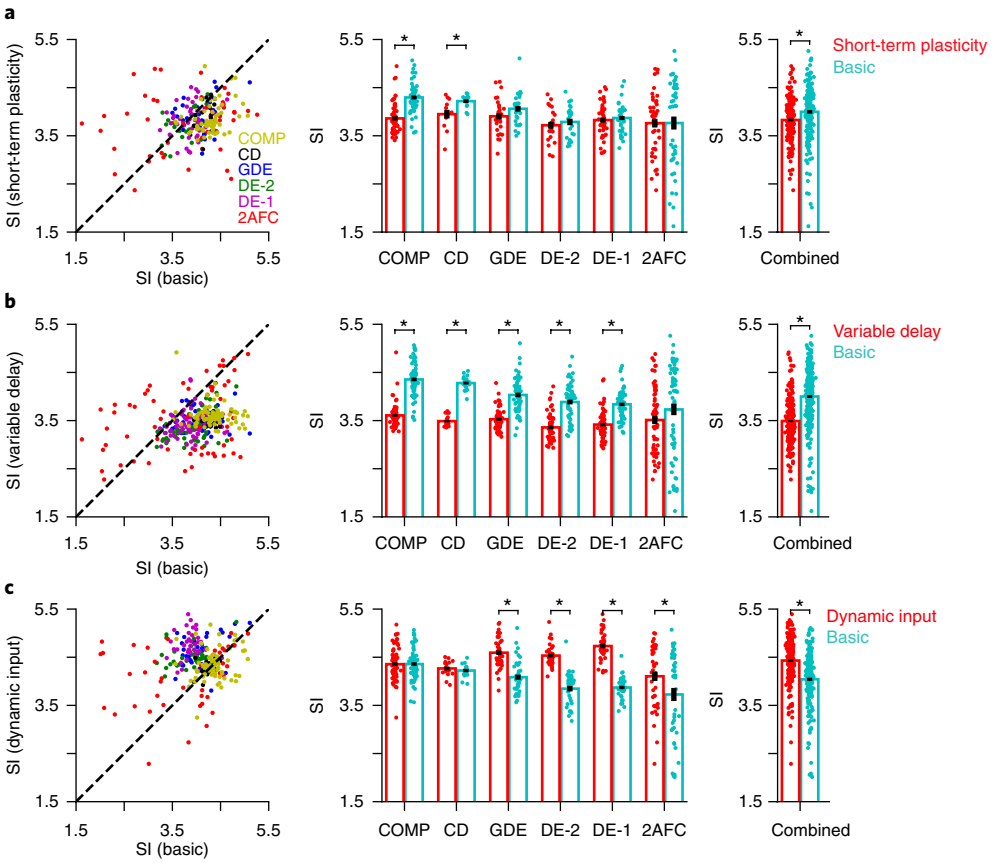


Fig. 4 | Hebbian short-term synaptic plasticity, delay duration variability, and structured dynamic inputs affect the sequentiality of the recurrent activity in trained networks. a, The effect of Hebbian short-term synaptic plasticity on the SI. The leftmost column shows a scatterplot of the SI in the basic condition versus the SI in the short-term plasticity condition. Each dot corresponds to a different initial condition and different colors represent different tasks. The middle column collapses the data across different initial conditions and compares the SI for each task. The error bars in the middle column represent the mean ± s.e.m. across different initial conditions. The rightmost column collapses the data further across tasks and compares the SI in the basic versus short-term plasticity conditions for the combined data. The error bars in the rightmost column represent the mean ± s.e.m. across different initial conditions and different tasks. The asterisk indicates a significant difference at $P < 0.05$ (two-sided Welch's t test). Hebbian short-term synaptic plasticity decreased the SI. **b**, The effect of delay duration variability on the SI. Delay duration variability decreased the SI. **c**, The effect of structured dynamic input on the SI. Structured dynamic input increased the SI. The exact sample sizes and P values for all statistical tests in the middle and rightmost columns are reported in Supplementary Table 1.

non-linearity saturating on both sides. For this purpose, we reproduced our main experiments with a simple modification to the networks; namely, we replaced the ReLU non-linearity with a clipped version of it that was bounded above by a maximum value, which we chose to be 100. Overall, the results from these simulations were qualitatively in agreement with the results from the ReLU networks. In particular, the hyperparameters σ_0 and ρ (but not λ_0) had similar effects on the SI, the ordering of the tasks by SI was similar, and the underlying mechanism that generated more sequential versus more persistent activity in different conditions was also similar in the clipped ReLU networks (see Supplementary Fig. 8).

Second, in our simulations, we chose the input noise levels to be roughly consistent with those used in Orhan and Ma[36], where generic neural networks were trained on tasks similar to those considered in this study in psychophysically realistic input noise regimes. To investigate the sensitivity of our results to the amount of input noise, we reran our main experiments with up to 2.5 times lower and up to 2 times higher levels of input noise. Increasing the input noise slightly increased the SI (Supplementary Fig. 9c). Importantly, even when we restricted the analysis to the lowest and the highest levels of input noise, we observed qualitatively very similar results to those reported for our main experiments; that is, the hyperparameters σ_0 and λ_0 had similar effects on SI, the ordering of the tasks by SI was similar, and the circuit mechanism generating more sequential versus more persistent solutions under different conditions was also similar (Supplementary Figs. 10 and 11).

Discussion

We have identified a diverse range of circuit- and task-related factors affecting the sequentiality or persistence of recurrent neural activity underlying short-term memory maintenance. Tasks with higher temporal complexity, fixed delay durations, stronger network coupling between neurons, motion-related dynamic cues, and prior training in other tasks promote more sequential activity in trained networks; tasks with lower temporal complexity, variable delay durations, weak coupling between neurons, and symmetric short-term synaptic plasticity promote more persistent activity.

We have also developed a detailed mechanistic understanding of the circuit mechanism that generates sequential versus persistent activity. In all trained networks, the basic mechanism implementing short-term memory maintenance is sequential recurrent activity generated by a non-normal recurrent connectivity matrix (see Supplementary Fig. 12 for Schur decompositions of trained recurrent connectivity matrices), with increasingly prolonged responses as the activity travels along the sequence. However, in networks with more sequential activity, this temporal smearing is reduced

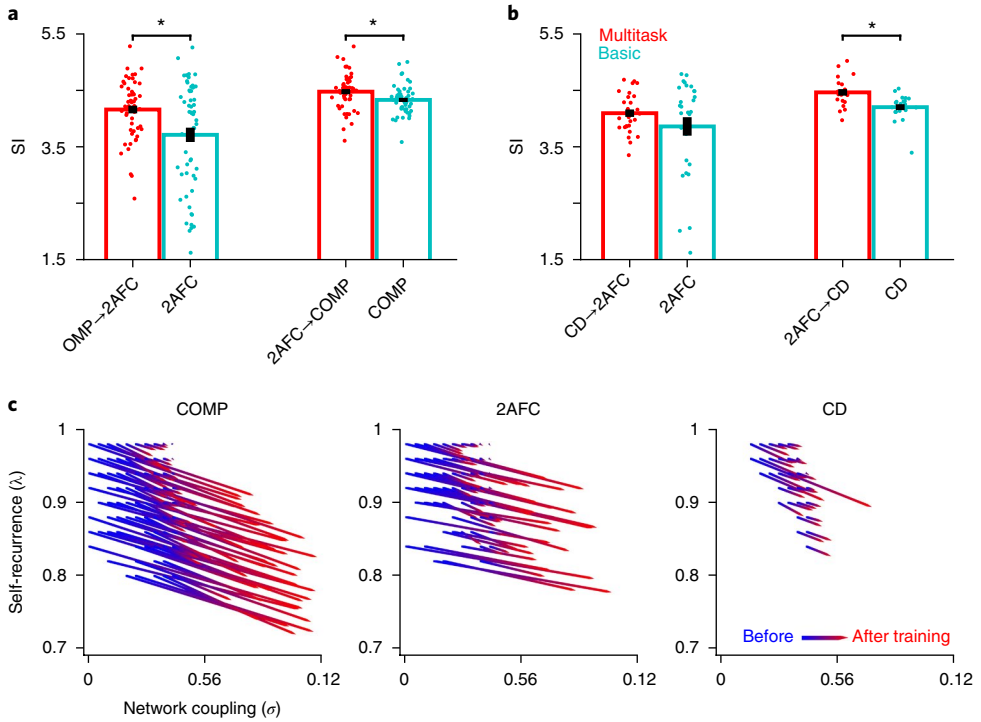


Fig. 5 | Multitask learning experiments. a, Results for the 2AFC-COMP task pair. **b,** Results for the 2AFC-CD task pair. The red bars show the results for the multitask training conditions and the cyan bars show the results for the corresponding single-task training conditions. The right arrow indicates the order of training; for example, COMP → 2AFC means the network was first trained on the COMP task and then on the 2AFC task. The error bars represent the mean ± s.e.m. across different hyperparameter settings. The asterisk indicates a significant difference at $P < 0.05$ (two-sided Welch's t test). The exact sample sizes and P values for all the statistical tests in **a** and **b** are reported in Supplementary Table 1. **c**, Training in a task consistently reduces the mean self-recurrence, $\lambda \equiv \langle W_{ii} \rangle$, and increases the fluctuations in the strength of recurrent coupling to the rest of the network, $\sigma \equiv \mathrm{std}(W_{ij,i \neq j})$ (std, standard deviation). Note that $\lambda = \lambda_0$ and $\sigma = \sigma_0$ (as defined in Fig. 2a) before training.

by a characteristic asymmetric peak in the weight profile that corresponds to strengthened connections between temporally close neurons in the sequence (at the expense of weakened connections between temporally distant neurons) with connections in the forward direction being preferentially strengthened (Fig. 6).

An important question to consider is why trained networks develop a short-term memory maintenance mechanism that relies on non-normal recurrent dynamics, even when the recurrent connectivity is initialized close to a normal matrix. For linear networks, it has been previously shown by Ganguli et al.[37] that any dynamical system with optimal memory properties must be non-normal; and a feed-forward chain is one of the simplest examples of such a non-normal dynamical system[37]. However, there are important differences between our networks and the simplified setup studied in Ganguli et al.[37]. Therefore, it remains to be seen whether this previous work can explain the emergence of non-normal structures in our trained networks. Another possibility is that non-normal solutions may just be more generic than normal solutions so that a randomly initialized network is more likely to converge to a non-normal solution.

A previous study (Rajan et al.[35]) also investigated the circuit mechanism underlying the generation of sequential activity in recurrent neural networks. However, that study did not train the networks to perform any short-term memory task, but rather trained them explicitly to generate sequential activity. Our work, on the other hand, shows that sequential activity emerges naturally in networks trained to perform short-term memory tasks; and certain factors identified in our study facilitate the emergence of such sequential activity.

Rajan et al.[35] discovered qualitatively different mechanisms generating sequential activity as the fraction of trainable connections was varied in their networks. When only a small fraction of the connections were trainable, they found an input-dependent mechanism for the generation of sequences that is different from the mechanism uncovered in this work. Our mechanism relies on an asymmetric recurrent connectivity matrix and is conceptually similar to the sequence generation mechanism they found in networks where all connections were trainable. However, the particular asymmetry we found is qualitatively different from the one found in their work. This difference is largely a result of the difference in the training signals: our networks were trained on actual short-term memory tasks without constraining the dynamics, whereas theirs were trained to generate sequential activity. Training the networks to explicitly generate sequential activity constrains the recurrent connectivity more strongly and results in more structured weight profiles, especially with the tanh non-linearity used in Rajan et al.[35] (Supplementary Fig. 13).

In addition to the difference in training signals, there are two other differences between Rajan et al.[35] and our work. First, they used tanh units, whereas we used ReLUs in our networks. We could not successfully train networks of tanh units in any of our tasks, neither with the particular initialization we used, nor with more standard initializations. However, we reproduced our experiments with two other activation functions, exponential linear[38] and softplus[39] (in addition to the double-sided saturating, clipped ReLU non-linearity discussed earlier) and found asymmetries in the trained recurrent connectivity matrices that were qualitatively similar to those observed in our ReLU networks (Supplementary Fig. 14). Second, the networks used by Rajan et al.[35] always received dynamic inputs, whereas in our basic condition, the networks did not receive any input during the delay, except for a very small amount of spontaneous input due to the stochasticity of input units (see Methods).

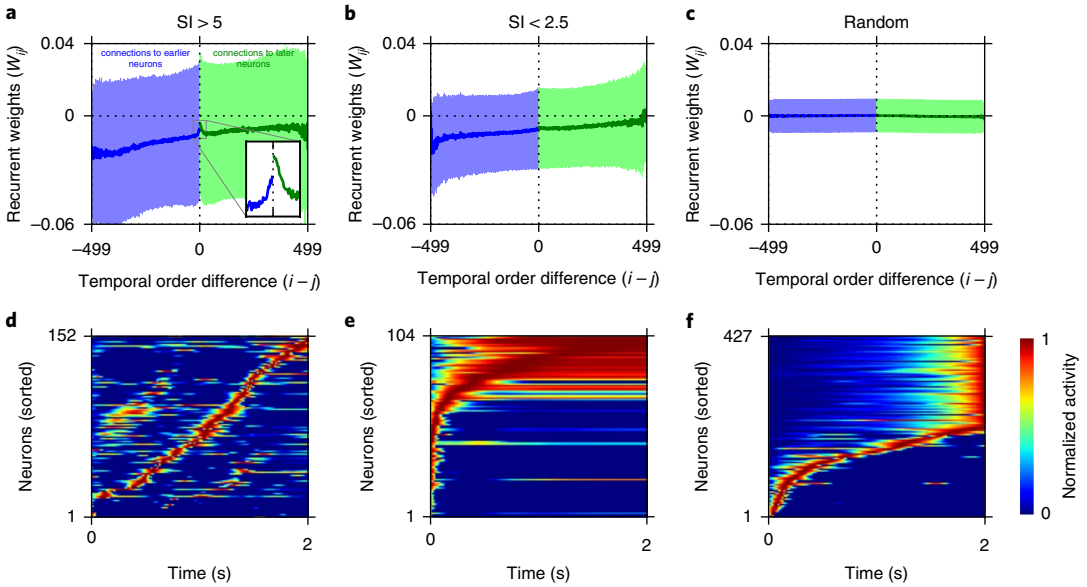


Fig. 6 | Circuit mechanism that generates sequential versus persistent activity. a–c, Neurons were first sorted by the time of their peak activity. We then plotted the mean and s.d. of the recurrent weights (W_{ij}) as a function of the difference between the orders of the neurons in the sequence $i-j$. A positive $i-j$ value (green) indicates a connection from an earlier to a later peaking neuron. A negative $i-j$ value (blue) indicates a connection from a later to an earlier peaking neuron. The solid lines represent the mean and the shaded regions represent the s.d. Both mean and s.d. are averages over multiple networks. **a**, Results for all trained networks with an SI > 5. **b**, Results for all trained networks with an SI < 2.5. **c**, Results for untrained random networks. The self-recurrence term corresponding to $i-j=0$ is not shown for clarity. **d–f**, show normalized responses of neurons in example trials simulated with connectivity matrices drawn from the profiles shown in **a–c**, respectively (see Methods for details). Only the active neurons are shown in these plots.

Hence, their simulations were more similar to our dynamic, motion-related input condition than to our basic condition. Together, Rajan et al.[35] and this study demonstrate a multiplicity of ways in which sequential activity can be generated in neural circuits.

Our results concerning the various factors affecting the sequentiality or persistence of neural activity underlying short-term memory immediately lead to experimental predictions that can be tested in the laboratory. There is already experimental evidence confirming the effects of some of these factors. For instance, Goard et al.[11] observed more persistent responses in the mouse posterior parietal cortex than Harvey et al.[16] did in the same area when animals in both studies were performing visual short-term memory tasks. However, there were crucial differences between the experimental designs in these studies: in the study by Goard et al.[11], the task was not a navigation-type task and there was significant delay duration variability, whereas in the study by Harvey et al.[16], the task was a navigation task in a simulated linear track and the delay duration variability was much smaller. Consistent with these results, we found more persistent responses in tasks with significant delay duration variability and more sequential responses in tasks with dynamic, motion-related inputs.

Our networks and learning paradigm had a number of biologically unrealistic features. Our networks consisted of simple generic rate neurons, whereas real neurons communicate via spikes and display a wide range of morphological and functional diversity. Moreover, our networks were trained with the biologically unrealistic backpropagation algorithm. However, a growing body of research demonstrates that task-trained generic neural networks like the ones we used in our simulations can capture many, sometimes surprisingly subtle, aspects of real biological circuits performing the same tasks[23,40–42], implying that one may not always need highly biologically realistic architectures or learning rules to explain the behavior of complex neural circuits performing complex tasks. Our results contribute to this literature by showing that both the sequential and nearly persistent stable activity patterns experimentally observed in short-term memory studies are part of a spectrum that emerges naturally in generic neural networks trained on short-term memory tasks under different conditions.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, statements of data availability and associated accession codes are available at https://doi.org/10.1038/s41593-018-0314-y.

Received: 9 February 2018; Accepted: 4 December 2018;
Published online: 21 January 2019

References

1. Fuster, J. M. & Alexander, G. E. Neuron activity related to short-term memory. *Science* **173**, 652–654 (1971).
2. Wang, X. J. Synaptic reverberation underlying mnemonic persistent activity. *Trends Neurosci.* **24**, 455–463 (2001).
3. Goldman, M. S. Memory without feedback in a neural network. *Neuron* **61**, 621–634 (2009).
4. Druckmann, S. & Chklovskii, D. B. Neural circuits underlying persistent representations despite time varying activity. *Curr. Biol.* **22**, 2095–2103 (2012).
5. Murray, J. D. et al. Stable population coding for working memory coexists with heterogeneous neural dynamics in prefrontal cortex. *Proc. Natl Acad. Sci. USA* **114**, 394–399 (2017).
6. Lundqvist, M., Herman, P. & Miller, E. K. Working memory: delay activity, yes! Persistent activity? Maybe not. *J. Neurosci.* **38**, 7013–7019 (2018).
7. Constantinidis, C. et al. Persistent spiking activity underlies working memory. *J. Neurosci.* **38**, 7020–7028 (2018).
8. Funahashi, S., Bruce, C. J. & Goldman-Rakic, P. S. Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. *J. Neurophysiol.* **61**, 331–349 (1989).
9. Miller, E. K., Erickson, C. A. & Desimone, R. Neural mechanisms of visual working memory in prefrontal cortex of the macaque. *J. Neurosci.* **16**, 5154–5167 (1996).
10. Romo, R., Brody, C. D., Hernández, A. & Lemus, L. Neural correlates of parametric working memory in the prefrontal cortex. *Nature* **399**, 470–473 (1999).

11. Goard, M. J., Pho, G. N., Woodson, J. & Sur, M. Distinct roles of visual, parietal, and frontal motor cortices in memory-guided sensorimotor decisions. *eLife* **5**, e13764 (2016).
12. Guo, Z. V. et al. Maintenance of persistent activity in a frontal thalamocortical loop. *Nature* **545**, 181–186 (2017).
13. Baeg, E. H. et al. Dynamics of population code for working memory in the prefrontal cortex. *Neuron* **40**, 177–188 (2003).
14. Fujisawa, S., Amarasingham, A., Harrison, M. T. & Buzsáki, G. Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. *Nat. Neurosci.* **11**, 823–833 (2008).
15. MacDonald, C. J., Lepage, K. Q., Eden, U. T. & Eichenbaum, H. Hippocampal 'time cells' bridge the gap in memory for discontiguous events. *Neuron* **71**, 737–749 (2011).
16. Harvey, C. D., Coen, P. & Tank, D. W. Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature* **484**, 62–68 (2012).
17. Schmitt, L. I. et al. Thalamic amplification of cortical connectivity sustains attentional control. *Nature* **545**, 219–223 (2017).
18. Scott, B. B. et al. Fronto-parietal cortical circuits encode accumulated evidence with a diversity of timescales. *Neuron* **95**, 385–398 (2017).
19. Murray, J. D. et al. A hierarchy of intrinsic timescales across cortex. *Nat. Neurosci.* **17**, 1661–1663 (2014).
20. Runyan, C. A., Piasini, E., Panzeri, S. & Harvey, C. D. Distinct timescales of population coding across cortex. *Nature* **548**, 92–96 (2017).
21. Sussillo, D., Churchland, M. M., Kaufman, M. T. & Shenoy, K. V. A neural network that finds a naturalistic solution for the production of muscle activity. *Nat. Neurosci.* **18**, 1025–1033 (2015).
22. Cueva, C. J. & Wei, X. X. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. Preprint at https://arxiv.org/abs/1803.07770 (2018).
23. Banino, A. et al. Vector-based navigation using grid-like representations in artificial agents. *Nature* **557**, 429–433 (2018).
24. Wilken, P. & Ma, W. J. A detection theory account of change detection. *J. Vis.* **4**, 1120–1135 (2004).
25. Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theory* **39**, 930–945 (1993).
26. Zucker, R. S. & Regehr, W. G. Short-term synaptic plasticity. *Annu. Rev. Physiol.* **64**, 355–405 (2002).
27. Mongillo, G., Barak, O. & Tsodyks, M. Synaptic theory of working memory. *Science* **319**, 1543–1546 (2008).
28. Rose, N. S. et al. Reactivation of latent working memories with transcranial magnetic stimulation. *Science* **354**, 1136–1139 (2016).
29. Wolff, M. J., Jochim, J., Akyürek, E. G. & Stokes, M. G. Dynamic hidden states underlying working-memory-guided behavior. *Nat. Neurosci.* **20**, 864–871 (2017).
30. Hinton, G. E. & Plaut, D. C. Using fast weights to deblur old memories. *Proc. 9th Annual Conference of the Cognitive Science Society*, 177–186 (Erlbaum, 1987).
31. Sompolinsky, H. & Kanter, I. Temporal association in asymmetric neural networks. *Phys. Rev. Lett.* **57**, 2861–2864 (1986).
32. Fiete, I. R., Senn, W., Wang, C. Z. H. & Hahnloser, R. H. R. Spike-time-dependent plasticity and heterosynaptic competition organize networks to produce long scale-free sequences of neural activity. *Neuron* **65**, 563–576 (2010).
33. Klampfl, S. & Maass, W. Emergence of dynamic memory traces in cortical microcircuit models through STDP. *J. Neurosci.* **33**, 11515–11529 (2013).
34. Krumin, M., Lee, J. J., Harris, K. D. & Carandini, M. Decision and navigation in mouse parietal cortex. *ELife* **7**, e42583 (2018).
35. Rajan, K., Harvey, C. D. & Tank, D. W. Recurrent network models of sequence generation and memory. *Neuron* **90**, 128–142 (2016).
36. Orhan, A. E. & Ma, W. J. Efficient probabilistic inference in generic neural networks trained with non-probabilistic feedback. *Nat. Commun.* **8**, 138 (2017).
37. Ganguli, S., Huh, D. & Sompolinsky, H. Memory traces in dynamical systems. *Proc. Natl Acad. Sci. USA* **105**, 18970–18975 (2008).
38. Clevert, D. A., Unterthiner, T., Hochreiter, S. Fast and accurate deep network learning by exponential linear units (ELUs). Preprint at https://arxiv.org/abs/1511.07289 (2016).
39. Glorot X., Bordes A., Bengio Y. Deep sparse rectifier neural networks. In *Proc. 14th International Conference on Artificial Intelligence and Statistics* (2011).
40. Mante, V., Sussillo, D., Shenoy, K. V. & Newsome, W. T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503**, 78–84 (2013).
41. Yamins, D. L. & DiCarlo, J. J. Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* **19**, 356–365 (2016).
42. Wang, J., Narain, D., Hosseini, E. A. & Jazayeri, M. Flexible timing by temporal scaling of cortical responses. *Nat. Neurosci.* **21**, 102–110 (2018).

Acknowledgements

This work was supported by grant no. R01EY020958 from the National Eye Institute. We thank the staff at the High-Performance Computing Cluster at New York University, especially S. Wang, for their help with troubleshooting.

Author contributions

A.E.O. conceived the study and developed the research plan with input from W.J.M. In several iterations, A.E.O. performed the experiments and the analyses. A.E.O. and W.J.M. then discussed the results, which helped refine the experiments and the analyses. A.E.O. wrote the initial draft of the paper. A.E.O. and W.J.M. reviewed and edited later iterations of the paper.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41593-018-0314-y.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to A.E.O.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2019

Methods

Network details. We adopted a discrete-time formulation in which the network dynamics was described by

$$\mathbf{r}_t = f\left(W_r \mathbf{r}_{t-1} + W_h \mathbf{h}_t + \mathbf{b}\right) \quad (1)$$

where $\mathbf{r}_t$ and $\mathbf{h}_t$ are the responses of the recurrent and input units at time t, respectively. Note that some previous studies start with a continuous-time formulation and obtain a discrete-time version through the Euler method. This yields an equation with the following form:

$$\mathbf{r}_t = (1-\alpha)\mathbf{r}_{t-1} + \alpha f\left(W_r \mathbf{r}_{t-1} + W_h \mathbf{h}_t + \mathbf{b}\right)$$

where $\alpha \equiv \Delta t/\tau$ describes the time step of the simulation in units of the intrinsic time scale of individual units. Typically, α is chosen to be small (for example, $\alpha = 0.05$–0.1), which is equivalent to assuming a long time constant for individual units. In contrast, our formulation (equation (1)) corresponds to choosing $\alpha = 1$; this does not assume a long time constant, but note that we increase the effective time constant of individual units through our initialization of W_r instead. More specifically, the hyperparameter λ_0 controls the initial effective time constant of the units in our formulation. We set $\Delta t = 10$ ms in all results reported in this article.

For the main experiments, we used linear rectification (ReLU) for the non-linearity f. All networks had 50 Poisson neurons in each input population and 500 recurrent neurons with ReLU activation. In networks with Hebbian synaptic plasticity, the general equation describing the network dynamics can be expressed as

$$\mathbf{r}_t = f\left(\left[W_r + \sum_{\tau=1}^{T} \gamma^{\tau} \mathbf{r}_{t-\tau-1}\, \mathbf{r}_{t-\tau-1}^{\top}\right]\mathbf{r}_{t-1} + W_h\, \mathbf{h}_t + \mathbf{b}\right)$$

In practice, however, we found networks with $T > 1$ to be very unstable and difficult to train; hence, we set $T = 1$, which yields the following equation:

$$\mathbf{r}_t = f\left(W_r \mathbf{r}_{t-1} + \gamma\kappa\left(\mathbf{r}_{t-2}^{\top}\mathbf{r}_{t-1}\right)\mathbf{r}_{t-2} + W_h \mathbf{h}_t + \mathbf{b}\right)$$

where $\kappa(\cdot)$ is a clipping function that clips its input between 0 and 100 to ensure stability and γ controls the strength of the Hebbian contribution; γ was set to 0.0005 in the change detection task and to 0.0007 in all other tasks. These values were the largest γ values that allowed the network to train successfully starting from at least ten different initial conditions.

Task details. In change detection, delayed estimation, and gated delayed estimation tasks, we used circular stimulus spaces, which can be thought of as orientation, for example. The input neurons had von Mises tuning functions with circular means uniformly spaced between 0 and π and a constant concentration parameter $\kappa = 2$. The stimuli were drawn uniformly between 0 and π. In the 2AFC and comparison tasks, linear stimulus spaces were used. In the 2AFC task, the input neurons had Gaussian tuning functions with centers uniformly spaced between −40 and 40, and a constant s.d. of 10. The stimuli presented were either −15 or 15 (randomly chosen in each trial) corresponding to the left and right choices, respectively. In the comparison task, the input neurons had Gaussian tuning functions with centers uniformly spaced between −50 and 50, and a constant s.d. of 10. The stimuli were drawn uniformly between −40 and 40. In the dynamic, motion-related input conditions, noisy, task-irrelevant pseudo-location information was provided by 50 additional Poisson neurons with Gaussian tuning functions uniformly tiling the stimulus range for each task. The s.d. of the tuning functions was 0.2236 and the network was assumed to cover the entire stimulus range at constant speed over the trial duration. In all tasks, during the stimulation periods, the gains of the input neurons were set to $1/T_{\text{stim}}$ at each time step (where T_{stim} denotes the duration of the stimulation period), yielding a cumulative gain of 1 for each input neuron throughout the stimulation period. All input neurons also had a stimulus-independent, uniform spontaneous gain of $0.1/T_{\text{delay}}$ at each time point during the delay, yielding a cumulative spontaneous firing rate of 0.1 spikes s^{-1} throughout the delay period. In all tasks, each trial took 1,500 ms (150 simulation steps): 250 ms (25 simulation steps) for the stimulus period; 1,000 ms (100 simulation steps) for the delay period; and 250 ms (25 simulation steps) for the response period.

Training details. The networks were trained with the Adam stochastic gradient descent algorithm[43] with a learning rate of 0.0005 and using the appropriate cost function for each task—mean squared error for continuous output tasks and mean cross-entropy error for categorical tasks. For all tasks, we put an additional L2 norm regularizer (with coefficient 0.0001) on the mean activity of all recurrent units in the last 50 ms of each trial. In the tethering tasks, the coefficient of this regularizer was increased to 0.1. Batch size was 50 trials in all experiments. The networks were trained for 25,000 iterations and tested on 300 new trials. All analyses were performed on these test trials.

Analysis details. Ideal observer models for each task were derived based on earlier work (for example, see Orhan and Ma[36] and Keshvari et al.[44]) and the optimal performance was calculated from these ideal observers. As in Orhan and Ma[36], for the categorical tasks (COMP, CD, 2AFC), we measured performance in terms of the fractional information loss, which is defined as the average Kullback–Leibler divergence between the actual posterior and the network's output normalized by the mutual information between the class labels and the neural responses. For the continuous output tasks (GDE, DE-1, DE-2), performance was measured in terms of the fractional root mean squared error (RMSE), which is defined as $100 \times (\text{RMSE}_{\text{netw}} - \text{RMSE}_{\text{opt}})/\text{RMSE}_{\text{opt}}$, where $\text{RMSE}_{\text{netw}}$ is the RMSE of the network and RMSE_{opt} is the RMSE of the ideal observer. In all the analyses presented in this study, we only considered networks that had at most 50% information loss or fractional RMSE on the test set to ensure that only sufficiently well-trained networks were included.

In calculating the SI for a given trial, we only included the recurrent neurons that had an average response of at least 0.1 during that trial. The remaining neurons did not contribute significantly to task performance and caused numerical instabilities in calculating the SI. In addition, the entropy of the peak time distribution, which is one of the determinants of the SI, was calculated by dividing the total trial duration into 20 bins and calculating the Shannon entropy of the resulting count distribution. A pseudocount of 0.1 was added to each bin before calculating the entropy.

In the simulated trials shown in Fig. 6d–f, a randomly selected set of 100 recurrent units (out of 500 units) received unit inputs for the entire duration of the trial, while the remaining units did not receive any direct input.

Statistical analysis. To test for the significance of mean SI differences between two conditions, we used a two-sided Welch's t test throughout the study. This test does not assume equal variances for the two conditions, but it assumes that the groups are normally distributed. We did not formally test the normality assumption. In linear regression analyses, we used the two-sided Wald test with the null hypothesis that the slope is zero. This test assumes that the estimate of the slope is normally distributed. The normality assumption was not formally tested.

No statistical methods were used to predetermine sample sizes. Since this work is a purely computational study, our sample sizes were effectively determined by the computational resources available to us in our High-Performance Computing Cluster. We used the maximum sample size that was practically feasible for all experiments. Our sample sizes are larger than those reported in comparable studies (for example, see Sussillo et al.[21], Rajan et al.[35], and Mante et al.[40]) because these earlier studies typically did not perform an extensive exploration of the hyperparameter and task space, which was one of the main goals of our study.

Since this work is a simulation study, differences between experimental conditions were precisely controlled (including random seeds). Randomization and blinding are thus not relevant for this study. Each experiment was run with a fixed and known random seed, hence all results reported here are precisely reproducible. Data collection and analysis were not performed blind to the conditions of the experiments.

Reporting Summary. Further information on experimental design is available in the Nature Research Reporting Summary linked to this article.

Code availability

The code for reproducing the experiments and analyses reported in this article is available at https://github.com/eminorhan/recurrent-memory.

Data availability

The raw simulation data used for generating each figure are available upon request.

References

43. Kingma, D. P. & Ba, J. L. Adam: a method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2015).
44. Keshvari, S., van den Berg, R. & Ma, W. J. No evidence for an item limit in change detection. *PLoS Comput. Biol.* **9**, e1002927 (2013).

nature research

Corresponding author(s): Emin Orhan

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

Statistical parameters

When statistical analyses are reported, confirm that the following items are present in the relevant location (e.g. figure legend, table legend, main text, or Methods section).

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	An indication of whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☐	☒	A description of all covariates tested
☐	☒	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistics including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated
☐	☒	Clearly defined error bars *State explicitly what error bars represent (e.g. SD, SE, CI)*

Our web collection on statistics for biologists may be useful.

Software and code

Policy information about availability of computer code

Data collection	Theano (0.8.2) and Lasagne (0.2.dev1). The code for reproducing the experiments reported in the paper is available at: https://github.com/eminorhan/recurrent-memory
Data analysis	Data analysis is performed with custom Python 3 code. The code is available at: https://github.com/eminorhan/recurrent-memory

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers upon request. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

The raw simulation data used for generating each figure are available upon request.

Field-specific reporting

Please select the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/authors/policies/ReportingSummary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	No statistical methods were used to pre-determine sample sizes. Since this work is a purely computational study, our sample sizes were effectively determined by the computational resources available to us in our High Performance Computing cluster. We used the maximum sample size that was practically feasible for all experiments. Our sample sizes are larger than those reported in comparable earlier studies, because earlier studies typically did not perform an extensive exploration of the hyperparameter and task space, which was one of the main goals of our paper.
Data exclusions	We restricted all analyses to networks that achieved a test set performance that was within 50% of the optimal performance. Some such restriction has to be made, since the remaining networks do not learn the task successfully. In calculating the sequentiality index (SI) for a trial, we also excluded neurons with an average response below 0.1 during the trial. Such neurons did not contribute significantly to task performance and caused numerical instabilities in calculating the SI.
Replication	Since this work is a simulation study, differences between experimental conditions were precisely controlled (including random seeds). Each experiment was run with a fixed and known random seed, hence all results reported here are precisely reproducible.
Randomization	Since this work is a simulation study, differences between experimental conditions were precisely controlled (including random seeds). Randomization is thus not relevant for this study.
Blinding	Data collection and analysis were not performed blind to the conditions of the experiments.

Reporting for specific materials, systems and methods

Materials & experimental systems

n/a	Involved in the study
☒	☐ Unique biological materials
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology
☒	☐ Animals and other organisms
☒	☐ Human research participants

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging