A Diverse Range of Factors Affect the Nature of Neural Representations Underlying Short-Term Memory

A. Emin Orhan[1,2,†] Wei Ji Ma[3,4]

[1]Department of Neuroscience, Baylor College of Medicine
[2]Department of Electrical and Computer Engineering, Rice University
[3]Center for Neural Science, New York University
[4]Department of Psychology, New York University
[†]Corresponding author: aeminorhan@gmail.com
Code: https://github.com/eminorhan/recurrent-memory

Abstract

Sequential and attractor-based models are two prominent models of short-term memory in neural circuits. In attractor-based 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 pattern of activity across the population. Experimental evidence for both types of model in the brain 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 fixed delay durations, tasks with higher temporal complexity, strong network coupling and motion-related dynamic inputs favor sequential solutions, whereas variable delay durations, tasks with low temporal complexity, weak network coupling and symmetric Hebbian short-term synaptic plasticity favor more persistent solutions. Our results clarify some seemingly contradictory experimental results on the existence of sequential vs. attractor-like memory mechanisms in the brain.

Introduction

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 an 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, while the population as a whole maintains the memory through a dynamically changing activity pattern across time.

Experimental evidence for both types of model has been reported previously: for example, [6–10] observed persistent or nearly persistent activity during the delay period of short-term memory tasks, whereas [11–16] 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 are relevant for the observed differences in the 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, here we address this question by performing these experiments *in silico*. This allows 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 potential factors on the sequentiality or persistence of the emergent activity patterns: (i) the task; (ii) other experimental variables such as delay duration variability or whether the task had a navigation component; (iii) intrinsic network properties such as the intrinsic timescale of individual neurons and the strength of coupling between the neurons; and (iv) Hebbian short-term synaptic plasticity. The first two are task-related, whereas the last two are circuit-related factors.

We find that tasks with higher temporal complexity, fixed delay durations, stronger network coupling between the neurons, and task-irrelevant motion-related dynamic cues increase the sequentiality of the emergent solutions; whereas tasks with lower temporal complexity, variable delay durations, weak coupling between the neurons and symmetric Hebbian short-term synaptic plasticity reduce the sequentiality of the 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 the sequences and why the aforementioned factors have the effects that they do on the sequentiality of the responses.

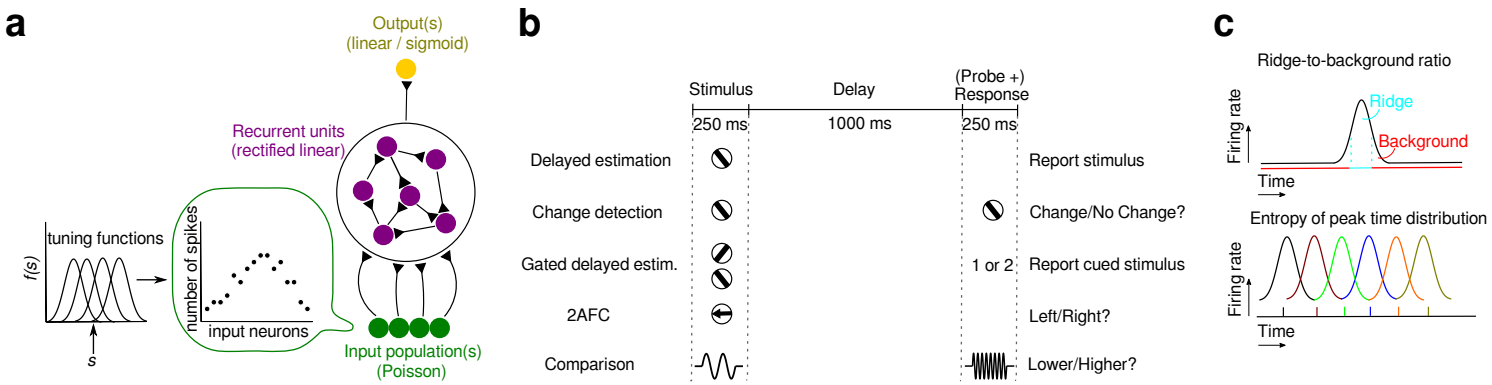

Figure 1: Experimental setup. **a** Schematic diagram of recurrent networks. **b** The experimental tasks and the common trial structure. **c** Two factors determining the sequentiality index: the ridge-to-background ratio [14] measures the temporal localization of the activity of individual units; the entropy of the peak time distribution measures the uniformity of the peak response times of the units in a given trial.

Results

Experimental setup

Networks. In our main simulations, we used vanilla recurrent neural networks (Figure 1a). We adopted a discrete-time formulation in which the network dynamics was described by:

$$\mathbf{r}_t = f(\mathbf{W}_r \mathbf{r}_{t-1} + \mathbf{W}_h \mathbf{h}_t + \mathbf{b})$$

where $\mathbf{r}_t$ and $\mathbf{h}_t$ are the responses of the recurrent and input units at time t, respectively. We interpreted each discrete time step in the simulation to correspond to 10 ms of real time in all results reported in this paper. We used linear rectification (ReLU) for the nonlinearity f. The input to the network, $\mathbf{h}_t$, was provided in the form of a population of Poisson neurons. In order to be trained successfully, the recurrent connectivity matrix $\mathbf{W}_r$ has to be initialized close to the identity [17]. We initialized $\mathbf{W}_r$ as $\lambda\mathbf{I}+\sigma\Sigma_{\text{off}}$ where λ and σ are hyperparameters controlling the amount of initial self-recurrence and recurrence from the rest of the network respectively, $\mathbf{I}$ is the identity matrix and Σ_{off} is an off-diagonal matrix whose off-diagonal elements are drawn independently from the standard normal distribution (Figure 2a). We repeated all our simulations for 100 different hyperparameter configurations drawn over a grid in the (λ, σ) space. On this grid, λ took 10 uniformly-spaced values between 0.8 and 0.98 and σ took 10 uniformly-spaced values between 0 and 0.018. This gave rise to a wide range of initial network dynamics, from quickly decaying to strongly unstable (Supplementary Figure S1). Qualitatively, increasing λ has the effect of increasing the intrinsic time constant of the individual neurons, making their activity more persistent in response to an input pulse. Increasing σ, on the other hand, introduces oscillatory components to the network response.

Tasks. In order to eliminate potential differences due to trial structure, we used a common trial structure for all our tasks (Figure 1b). Each trial started with the presentation of one or two stimuli for 250 ms. A delay period of 1000 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 (Figure 1b): (i) delayed estimation (DE), where the task was to report the stimulus or stimuli presented at the beginning of the trial; (ii) change detection (CD), where the task was to report whether the stimulus presented before the delay was the same as the stimulus presented after the delay (e.g. [18]); (iii) 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 (e.g. [18]); (iv) 2AFC, where one of two possible stimuli (e.g. left vs. right moving dots) was presented at the beginning of the trial and the task was to report which one was presented (e.g. [9, 14]); (v) 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 (e.g. [8]).

Quantifying sequentiality. Intuitively, there are two requirements for the recurrent activity of a population of neurons to be considered sequential (Figure 1c): (i) each neuron should be active only during a short interval compared to the duration of the trial; (ii) the active periods of the neurons should tile the entire duration of the trial approximately uniformly. Thus, we came up with a *sequentiality index* (SI) that takes into account both of these desiderata. The sequentiality index 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 [14]. The sequentiality index for a given experimental condition is then determined by averaging over the sequentiality indices of all trials belonging to that condition.

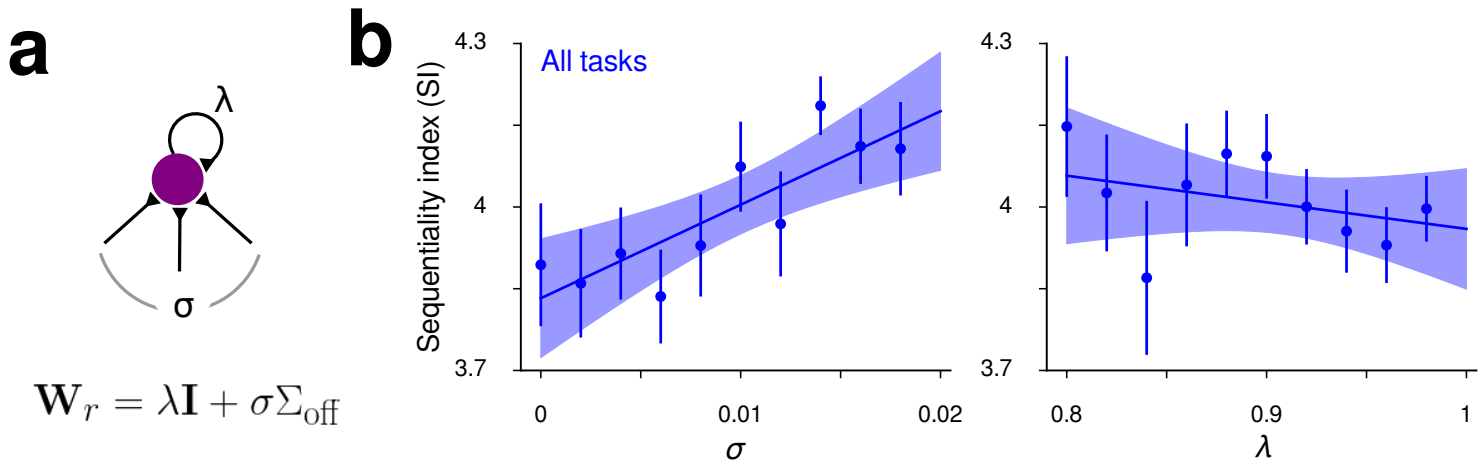

Figure 2: Initial intrinsic circuit properties and their effect on the sequentiality of the recurrent activity in trained networks. **a** The recurrent connectivity matrix was initialized as $\mathbf{W}_r = \lambda\mathbf{I} + \sigma\Sigma_{\text{off}}$ where λ controls the initial intrinsic timescale of individual units and σ controls the initial coupling between the units. All our experiments were repeated with 100 different λ, σ values on a 10×10 grid over the two-dimensional hyper-parameter space (λ, σ). **b** The sequentiality index increased with σ (linear regression slope: 17.16 ± 4.81), but did not change significantly with λ (linear regression slope: -0.49 ± 0.53). Error bars represent standard errors. Solid lines are the linear fits and shaded regions are 95% confidence intervals for the linear regression.

Factors affecting the sequentiality of the responses

Intrinsic circuit properties affect sequentiality. In successfully trained networks, the sequentiality of the recurrent activity increased with the initial network coupling σ, while the initial intrinsic timescale of the individual units, λ, did not have a significant effect on sequentiality over the range of values tested here (Figure 2b). Larger σ values introduce higher frequency oscillatory dynamics in the initial network, which promotes the emergence of high frequency sequential structure in the trained networks.

Temporal complexity of tasks affects sequentiality. There was significant variability in sequentiality index among the tasks (Figure 3a; see Supplementary Figures S2-S3 for example trials from all tasks). 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 responses. We hypothesized that this variability was related to the temporal complexity of the target functions that need to be learned in different tasks, where the target function complexity can be formalized as the mean frequency of the target function (as a function of time) [19]. 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. In delayed estimation and 2AFC tasks, on the other hand, no probe is presented after the delay and the target response does not depend on what happens after the delay. Therefore, 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. The target function thus had the following form: $\sin(2\pi f t/T_{\text{resp}})$, where $0 \leq t \leq T_{\text{resp}}$ 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 (Figure 3b). Secondly, we introduced a "tethering" manipulation in our experimental design that increased the temporal complexity of the tasks. In tethered conditions, we put a strong penalty on recurrent responses deviating from 0 during the last 50 ms of the trial (Figure 3c). Tethering increases the temporal complexity of the task, because it forces the network's output to sharply change from whatever roughly constant value it takes before the onset of tethering. We thus expect 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 (Figure 3c-d). Importantly, in many cases, tethering changed the dynamics throughout the entire trial duration and not just toward the end of the trial (see the example pair of trials shown at the bottom right in Figure 3c).

Hebbian short-term synaptic plasticity affects sequentiality. To investigate the effect of short-term synaptic plasticity on the sequentiality of the recurrent activity, we added a symmetric Hebbian short-term synaptic plasticity term to the recurrent weights (see *Methods*). This Hebbian contribution to the recurrent weights is also known as "fast weights" in machine learning [20]. Symmetric Hebbian short-term synaptic plasticity decreased the sequentiality of the recurrent activity in the trained networks (Figure 4a). Any 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 the symmetry of the short-term synaptic plasticity rule considered in this work, since asymmetric associative rules (e.g. spike-timing-dependent plasticity) usually have opposite effects, as has been demonstrated in earlier studies [21–23].

Delay duration variability affects sequentiality. To test the effect of delay duration variability, we designed versions

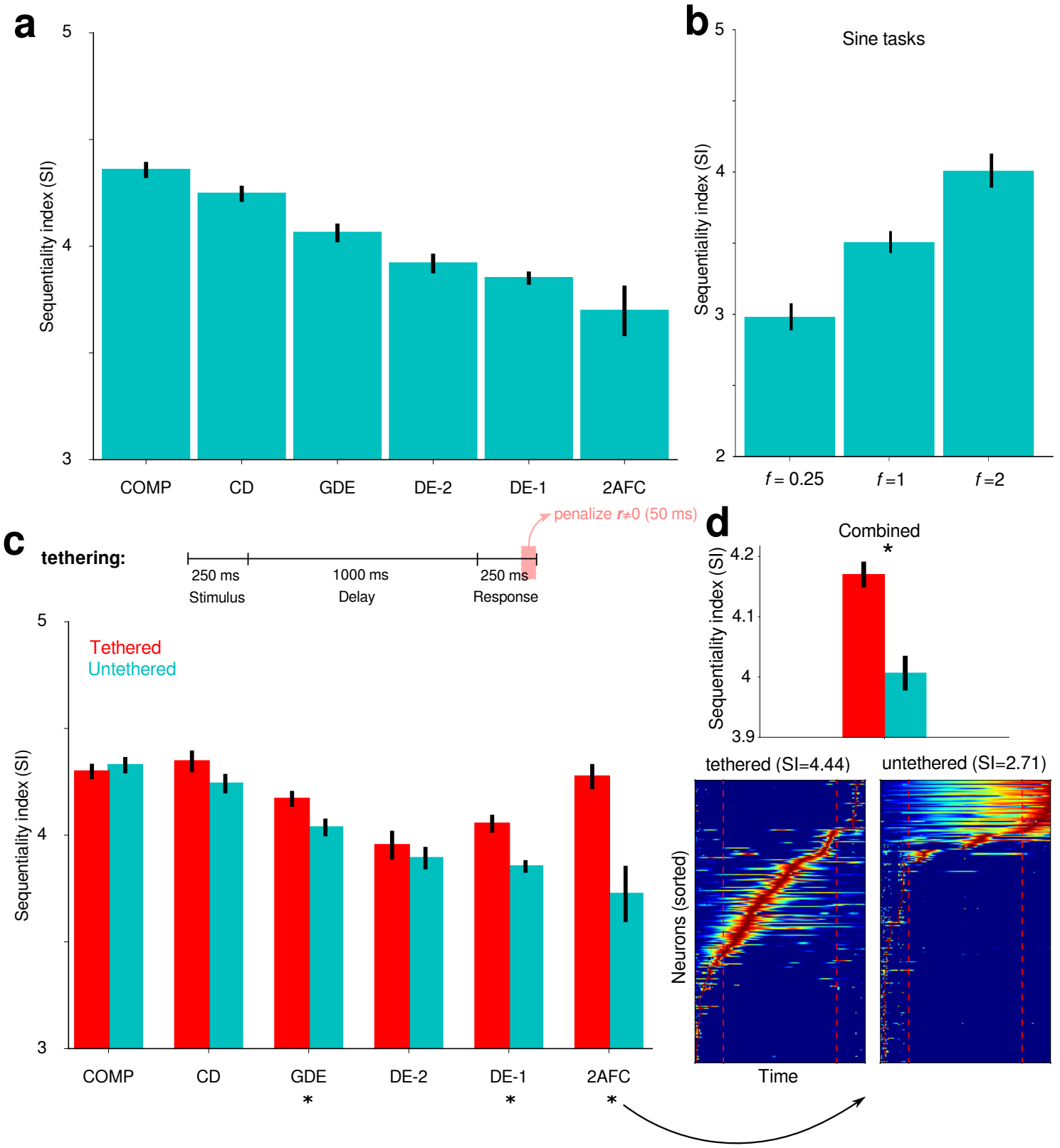

Figure 3: Temporal complexity of tasks increases the sequentiality of the recurrent activity in trained networks. **a** Sequentiality index in different tasks. DE-1 and DE-2 refer to delayed estimation tasks with one stimulus and with two stimuli, respectively. Error bars represent standard errors across different initializations. **b** Sequentiality index in the sine task. In this task, the network was trained to output a sine function with temporal frequency f during the response period. Higher frequency target functions (larger f) led to more sequential responses. **c** Tethering manipulation and its effect on the sequentiality index of different tasks. Asterisk (*) indicates a significant difference at the $p < .05$ level (Welch's t-test). The heat maps at the bottom right show the normalized responses of recurrent units in a pair of trials from the tethered and untethered versions of the 2AFC task, respectively. **d** Sequentiality index in the tethered vs. untethered conditions, combined across all tasks in **c**.

of each of our tasks with delay duration variability. In these versions, the delay duration was one of 100 ms, 400 ms, 700 ms, 1000 ms, chosen randomly on each trial. Comparing the sequentiality indices of the tasks with fixed delay duration to the sequentiality indices of the same tasks with delay duration variability, we observe that variability in delay durations significantly decreases the sequentiality of the recurrent activity (Figure 4b). In sequential solutions, the representations of task-relevant variables change over time. Therefore, these representations cannot be decoded with a fixed decoder across time. However, the variable delay duration experiments require that the representations be decoded with a fixed decoder at different delay durations, hence force the network to learn more stable representations across time.

Task-irrelevant structured dynamic inputs affect sequentiality. Motion-related signals animals receive during navigation-type experiments have previously been hypothesized to be crucial for the generation of sequential neural activity observed in rodent experiments [24]. Our results from experiments without such motion-related signals clearly show 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 might facilitate the generation of sequential activity in the network. To test this hypothesis, we designed navigation versions of our experiments where, in each trial, the network was assumed to be navigating 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. Note that the location information is completely irrelevant for performing the tasks, hence the network can safely ignore this information. Comparing the sequentiality indices of the tasks without dynamic, motion-related signals to the sequentiality indices of the tasks with such motion-related signals, we observe that the motion-related, task-irrelevant location signals increase the sequentiality of the recurrent activity (Figure 4c), suggesting that the network does not completely suppress these signals despite the fact that they are irrelevant for the tasks.

Circuit mechanism that generates the sequential activity

To probe the circuit mechanism generating sequential activity in trained networks, we performed an analysis proposed by [25]. In this analysis, we first ordered the recurrent neurons in the network by their time of peak activity. We then measured the mean and standard deviation of the recurrent weights, W_{ij}, as a function of the order difference between two neurons, $i - j$. In trained networks, connections from earlier peaking neurons to later peaking neurons had, on average, larger weights than connections from later peaking neurons to earlier peaking neurons. The mean connection weight was an approximately monotonically increasing function of $i - j$ (Figure 5a-b). This particular asymmetric structure generates sequential activity in the network with increasingly prolonged responses in later neurons in the sequence (Figure 5e). However, in trained networks with high sequentiality index (SI > 5), a prominent asymmetric peak appeared in the connection weight profile (inset in Figure 5a). 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, reduces the temporal smearing of the responses that takes place in networks with low sequentiality index (SI < 2.5), which did not display such a peak in their connection weight profile (Figure 5b).

Discussion

We have identified a diverse range of circuit-related 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, and motion-related dynamic cues promote more sequential activity; whereas 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 the sequences. 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 Figure S4 for Schur decompositions of trained recurrent connectivity matrices), with increasingly prolonged responses as the activity travels along the sequence. In networks with more sequential activity, however, this temporal smearing is reduced 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 (Figure 5).

A previous work [25] also investigated the circuit mechanism underlying the generation of sequential activity in recurrent neural networks. However, that work 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 here facilitate the emergence of such sequential activity.

Rajan et al. [25] 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. The particular asymmetry we found, however, is qualitatively different from the one found in their work. This difference is largely due to the difference in the training signals (our networks were trained on actual short-term memory tasks without constraining the dynamics, theirs were trained to generate sequential activity). Training the networks to explicitly generate sequential activity constrains the recurrent connectivity more

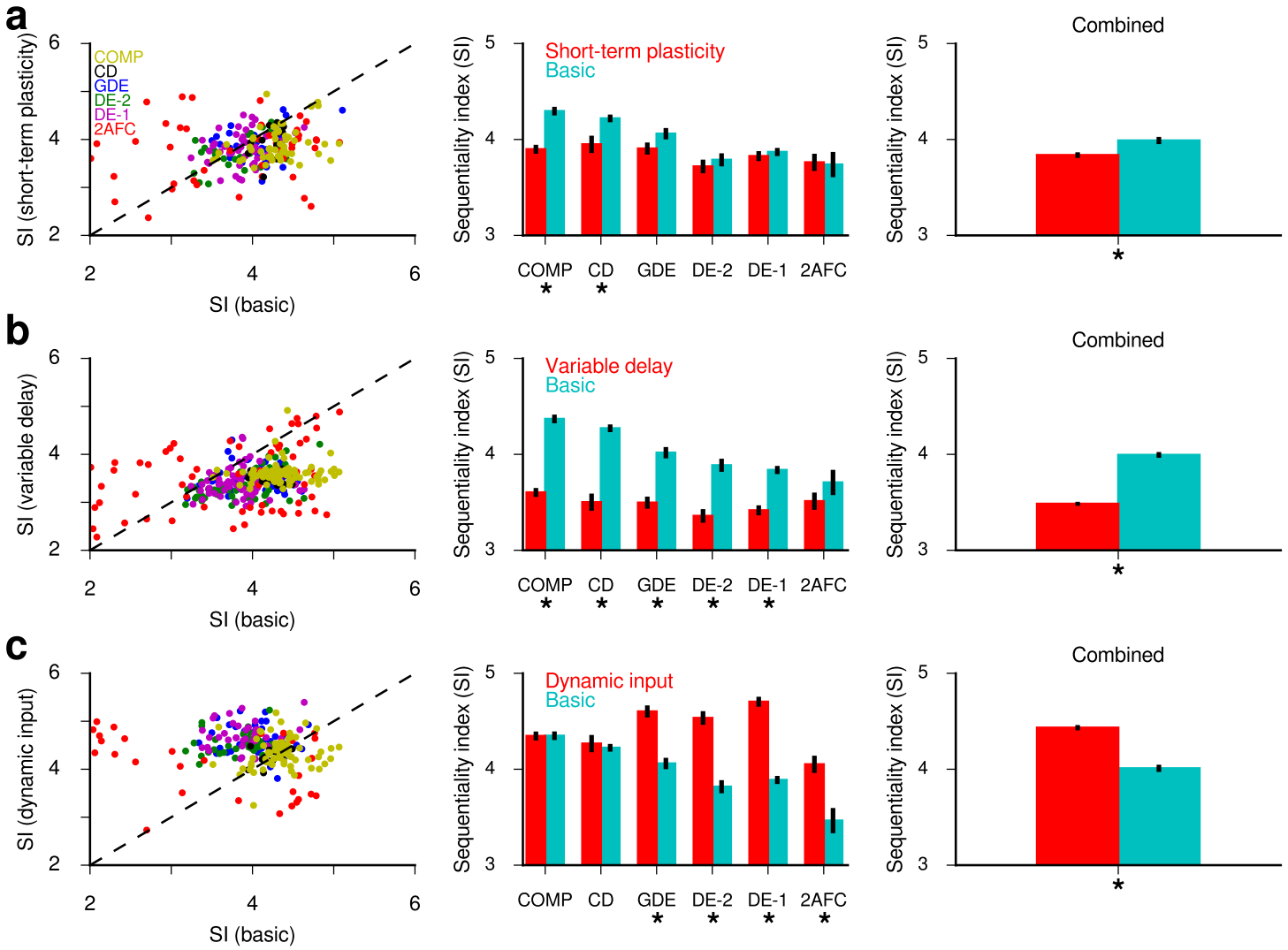

Figure 4: Hebbian short-term synaptic plasticity, delay duration variability and structured dynamic inputs affect the sequentiality of the recurrent activity in trained networks. **a** The effect of Hebbian short-term synaptic plasticity on the SI. The leftmost column shows a scatter-plot of the SI in the basic condition vs. 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 rightmost column collapses the data further across tasks and compares the SI in the basic vs. short-term plasticity conditions for the combined data. Asterisk (*) indicates a significant difference at the $p < .05$ level (Welch's t-test). **b** The effect of delay duration variability on the sequentiality index. **c** The effect of structured dynamic input on the sequentiality index.

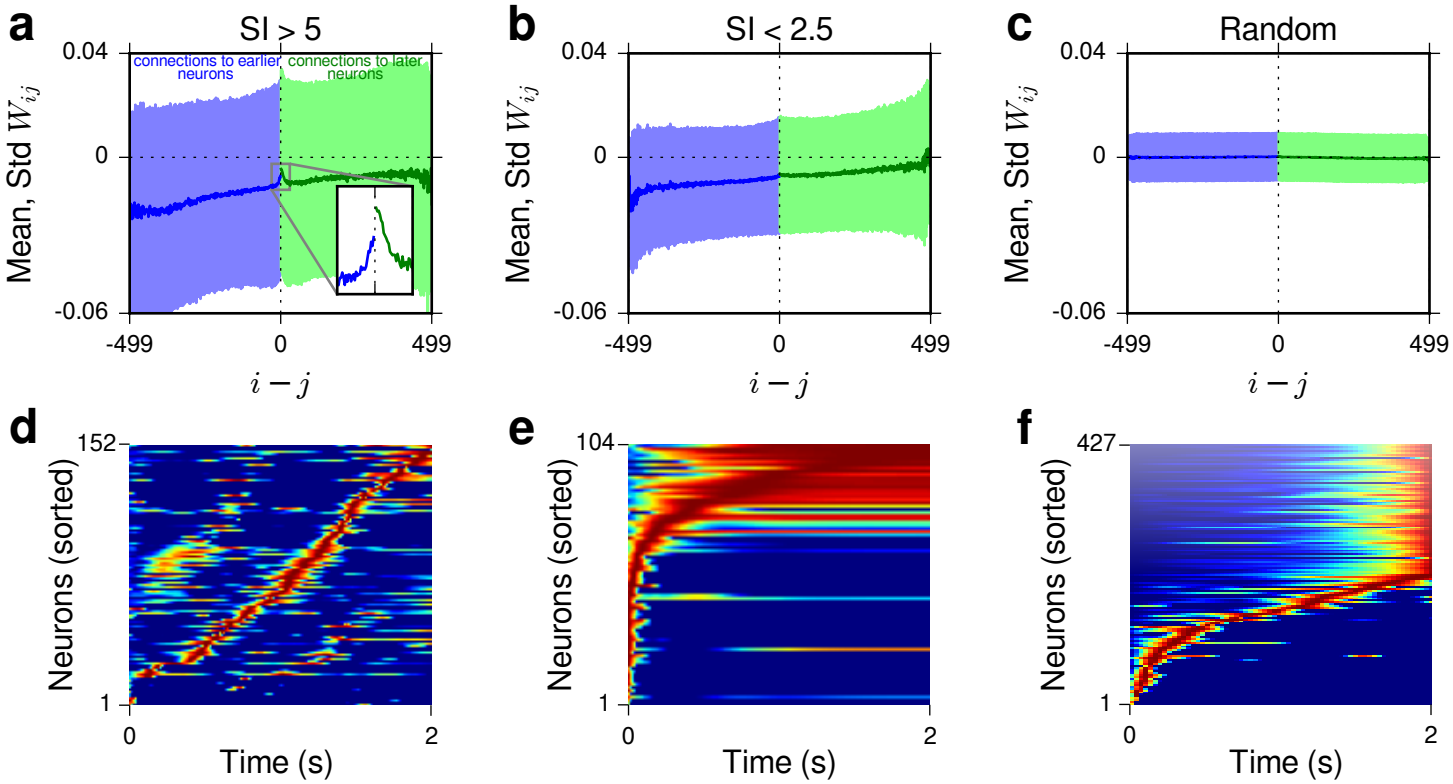

Figure 5: Circuit mechanism that generates sequences. **a, b, c** Neurons are first sorted by the time of their peak activity. We then plot the mean and standard deviation of the recurrent weights, W_{ij}, as a function of the difference between the orders of the neurons in the sequence, $i-j$. A positive $i-j$ value (green) indicates a connection from an earlier peaking neuron to a later peaking neuron. A negative $i-j$ value (blue) indicates a connection from a later peaking neuron to an earlier peaking neuron. Solid lines represent means and shaded regions represent standard deviations. **a** shows the results for all trained networks with SI > 5, **b** shows the results for all trained networks with SI < 2.5 and **c** shows the results for untrained random networks. The self-recurrence term corresponding to $i-j=0$ is not shown for clarity. **d, e, f** show normalized responses of neurons in example trials simulated with the connectivity matrices shown in **a, b, c**, respectively (see *Methods* for details). Only the active neurons are shown in these plots.

strongly and results in more structured weight profiles, especially with the tanh nonlinearity used in [25] (Supplementary Figure S5).

In addition to the difference in training signals, there are two other differences between [25] and our work. First, they used tanh units, whereas we used ReLUs in our networks. With the particular initialization we used, we were not able to successfully train networks of tanh units in any of our tasks. However, we replicated our results with two other activation functions (exponential linear [27] and softplus [28]) and found asymmetries in the trained recurrent connectivity matrices that were qualitatively similar to those observed in our ReLU networks (Supplementary Figure S6). Second, Rajan et al.'s networks 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*). Hence their simulations were more similar to our dynamic, motion-related input condition than to our basic condition. Together, Rajan et al. [25] 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 lab. There is already experimental evidence confirming the effects of some of these factors. For instance, [9] observed more persistent responses in mouse posterior parietal cortex than [14] did in the same area while animals in both studies were performing visual short-term memory tasks. There were, however, crucial differences between the experimental designs in these studies: in [9], the task was not a navigation-type task and there was significant delay duration variability, whereas in [14], 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 in tasks with structured, dynamic, motion-related inputs.

Methods

Network details. 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:

$$\mathbf{r}_t = f([\mathbf{W}_r + \sum_{\tau=1}^{T} \gamma^{\tau} \mathbf{r}_{t-\tau-1} \mathbf{r}_{t-\tau-1}^{\top}]\mathbf{r}_{t-1} + \mathbf{W}_h \mathbf{h}_t + \mathbf{b})$$

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

$$\mathbf{r}_t = f(\mathbf{W}_r \mathbf{r}_{t-1} + \gamma \kappa(\mathbf{r}_{t-2}^{\top} \mathbf{r}_{t-1}) \mathbf{r}_{t-2} + \mathbf{W}_h \mathbf{h}_t + \mathbf{b})$$

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 10 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 standard deviation 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 standard deviation of 10. The stimuli were drawn uniformly between -40 and 40. 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 throughout the delay period. In all tasks, each trial took 150 simulation steps (25 steps for the stimulus period, 100 steps for the delay period and 25 steps for the response period) and we interpreted each simulation step to correspond to 10 ms in real time, giving us trials lasting a total of 1.5 seconds each.

Training details. The networks were trained with the Adam stochastic gradient descent algorithm [26] with learning rate 0.0005 and using the appropriate cost function for each task: mean squared error for continuous output tasks and cross-entropy for categorical tasks. For all tasks, we put an additional l_2-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 25000 iterations and tested on 300 new trials. All analyses were performed on these test trials.

Analysis details. In all the analyses presented in this paper, we only considered networks that achieved a test set performance that was at least within 50% of the optimal performance. In calculating the sequentiality index (SI) for a given trial, we only included the recurrent neurons that had an average response of at least 0.1 during that trial. 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 pseudo-count of 0.1 was added to each bin before calculating the entropy.

In the simulated trials shown in Figure 5d-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.

Acknowledgments

This work was supported by Grant R01EY020958 from the National Eye Institute. We thank the staff at the High Performance Computing cluster at NYU, especially Shenglong Wang, for their excellent maintenance of this important resource and for their help with troubleshooting.

References

[1] Fuster JM, Alexander GE (1971) Neuron activity related to short-term memory. Science 173(3997):652-4.

[2] Wang XJ (2001) Synaptic reverberation underlying mnemonic persistent activity. Trends Neurosci 24(8):455-63.

[3] Goldman MS (2009) Memory without feedback in a neural network. Neuron 61(4):623-34.

[4] Druckmann S, Chklovskii DB (2012) Neural circuits underlying persistent representations despite time varying activity. Curr Biol 22(22):2095-2103.

[5] Murray JD, Bernacchia A, Roy NA, Constantinidis C, Romo R, Wang XJ (2017) Stable population coding for working memory coexists with heterogeneous neural dynamics in prefrontal cortex. PNAS 111(2):394-9.

[6] Funahashi S, Bruce CJ, Goldman-Rakic PS (1989) Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. J Neurophysiol 61(2):331-49.

[7] Miller EK, Erickson CA, Desimone R (1996) Neural mechanisms of visual working memory in prefrontal cortex of the macaque. J Neurosci 16(16):5154-5167.

[8] Romo R, Brody CD, Hernandez A, Lemus L (1999) Neural correlates of parametric working memory in the prefrontal cortex. Nature 399(6735):470-3.

[9] Goard MJ, Pho GN, Woodson J, Sur M (2016) Distinct roles of visual, parietal, and frontal motor cortices in memory-guided sensorimotor decisions. eLife 5:e13764.

[10] Guo ZV, Inagaki HK, Daie K, Druckmann S, Gerfen CR, Svoboda K (2017) Maintenance of persistent activity in a frontal thalamocortical loop. Nature 545:181-186.

[11] Baeg EH, Kim YB, Huh K, Mook-Jung I, Kim HT, Jung MW (2003) Dynamics of population code for working memory in the prefrontal cortex. Neuron 40(1):177-88.

[12] Fujisawa S, Amarasingham A, Harrison MT, Buzsaki G (2008) Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. Nat Neurosci 11(7):823-33.

[13] MacDonald CJ, Lepage KQ, Eden UT, Eichenbaum H (2011) Hippocampal "time cells" bridge the gap in memory for discontiguous events. Neuron 71(4):737-49.

[14] Harvey CD, Coen P, Tank DW (2012) Choice-specific sequences in parietal cortex during a virtual-navigation decision task. Nature 484:62-68.

[15] Schmitt LI, Wimmer RD, Nakajima M, Happ M, Mofakham S, Halassa MM (2017) Thalamic amplification of cortical connectivity sustains attentional control. Nature 545(7653):219-223.

[16] Scott BB, Constantinople CM, Akrami A, Hanks TD, Brody CD, Tank DW (2017) Fronto-parietal cortical circuits encode accumulated evidence with a diversity of timescales. Neuron 95(2):385-98.

[17] Le QV, Jaitly N, Hinton GE (2015) A simple way to initialize recurrent networks of rectified linear units. `https://arxiv.org/abs/1504.00941`.

[18] Wilken P, Ma WJ (2004) A detection theory account of change detection. J Vis 4(12):1120-35.

[19] Barron AR (1993) Universal approximation bounds for superpositions of a sigmoidal function. IEEE Trans Inf Theory 39(3):930-45.

[20] Hinton GE, Plaut DC (1987) Using fast weights to deblur old memories. In Proceedings of the 9th Annual Conference of the Cognitive Science Society, pp. 177–186. Hillsdale, NJ: Erlbaum.

[21] Sompolinsky H, Kanter I (1986) Temporal association in asymmetric neural networks. Phys Rev Lett 57(22):2861–4.

[22] Fiete IR, Senn W, Wang CZH, Hahnloser RHR (2010) Spike-time-dependent plasticity and heterosynaptic competition organize networks to produce long scale-free sequences of neural activity. Neuron 65(4):563–76.

[23] Klampfl S, Maass W (2013) Emergence of dynamic memory traces in cortical microcircuit models through STDP. J Neurosci 33(28):11515–29.

[24] Krumin M, Harris KD, Carandini M (2017) Decision and navigation in mouse parietal cortex. `https://www.biorxiv.org/content/early/2017/07/21/166413`.

[25] Rajan K, Harvey CD, Tank DW (2016) Recurrent network models of sequence generation and memory. Neuron 90:1-15.

[26] Kingma DP, Ba JL (2014) Adam: a method for stochastic optimization. https://arxiv.org/abs/1412.6980.

[27] Clevert DA, Unterthiner T, Hochreiter S (2015) Fast and accurate deep network learning by exponential linear units (ELUs). https://arxiv.org/abs/1511.07289.

[28] Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics.

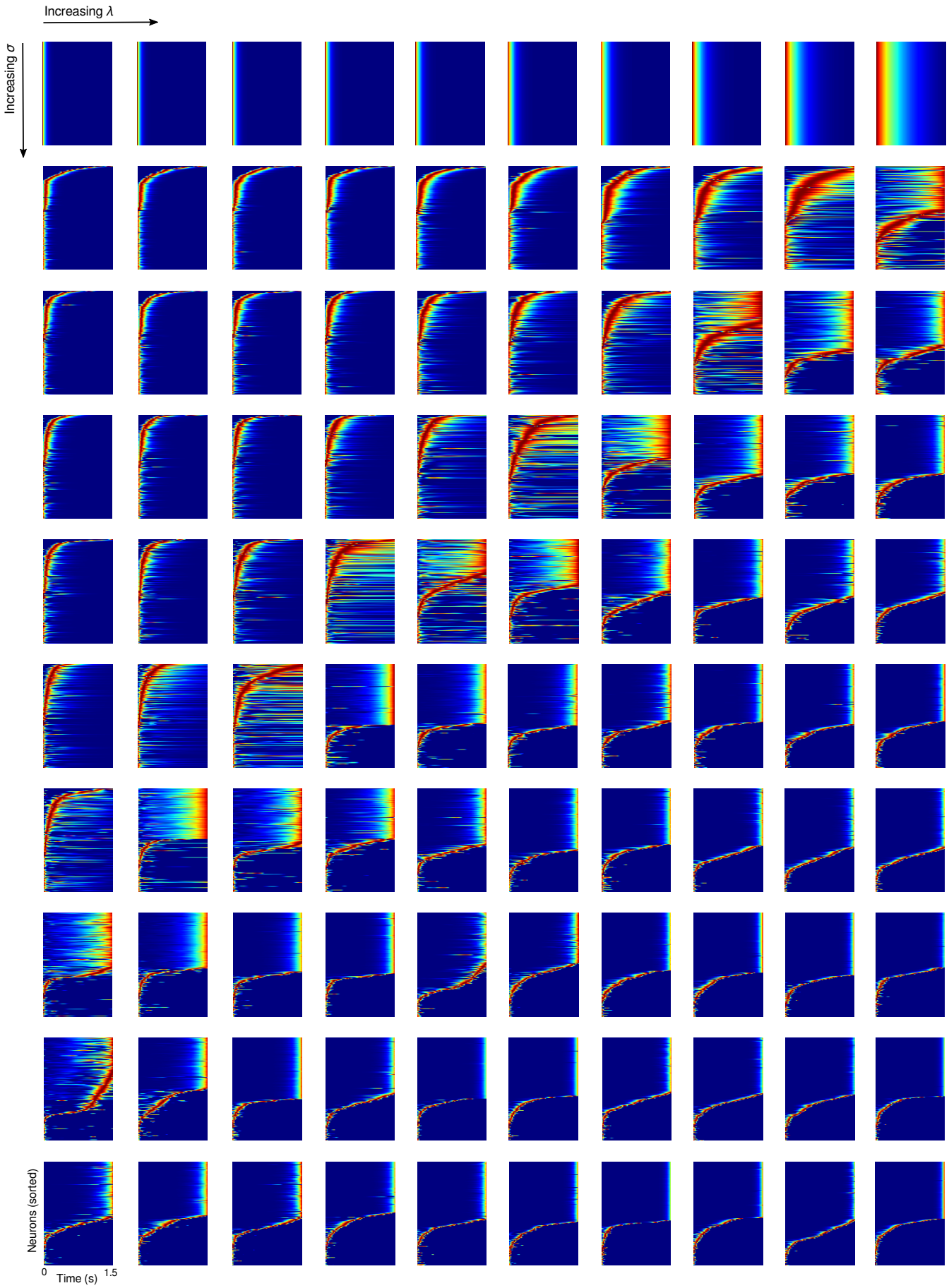

Figure S1: Initial, untrained network dynamics for different (λ, σ) values. The heat maps show the normalized responses of the recurrent units to a unit pulse delivered at time $t = 0$ to all units. Here, λ takes 10 uniformly-spaced values between 0.8 and 0.98 (columns) and σ takes 10 uniformly-spaced values between 0 and 0.018 (rows).

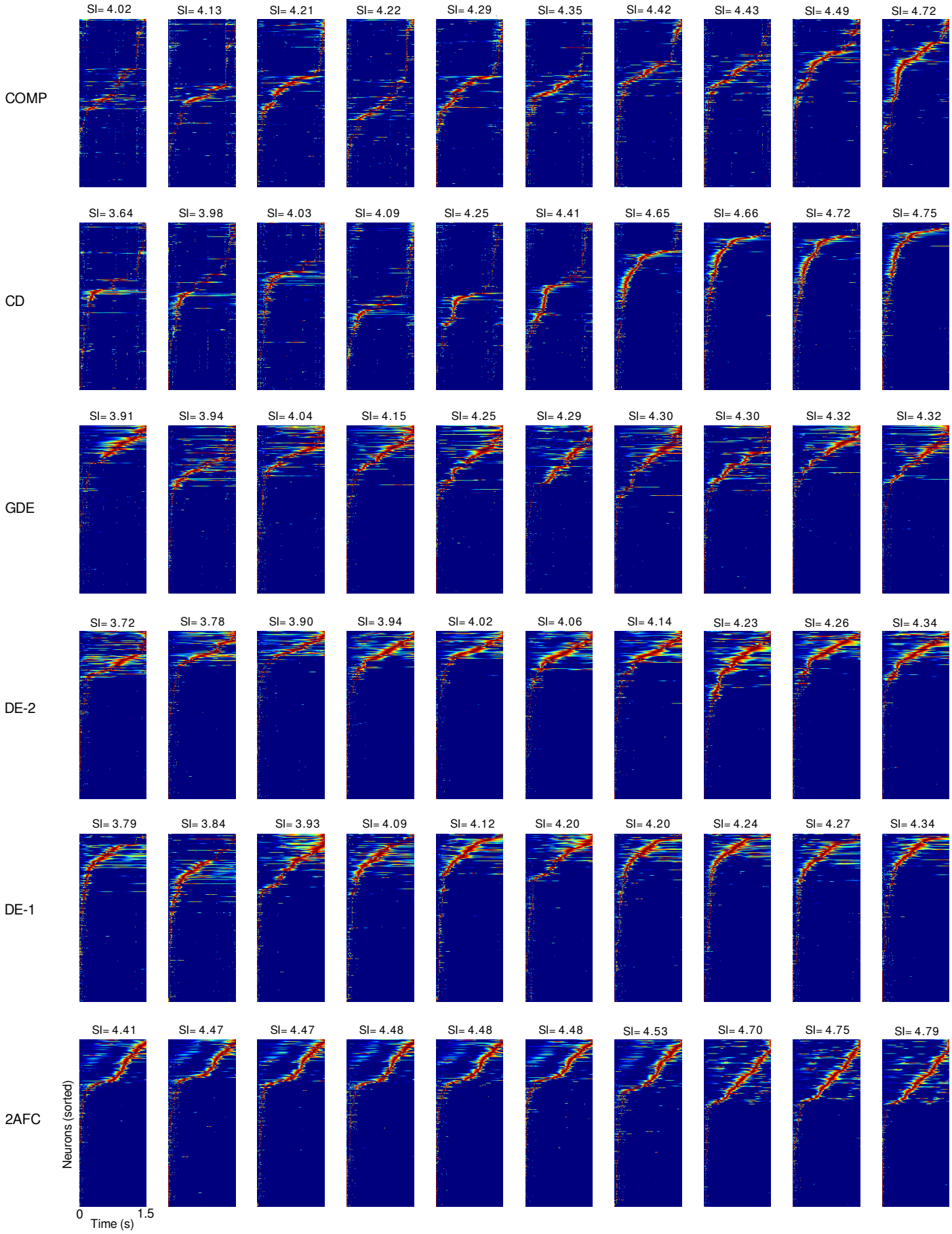

Figure S2: Example trials from the six tasks (basic condition). Sequentiality indices (SI) of the trials are indicated at the top of the plots. Trials are ordered by increasing SI from left to right. All trials shown here are from networks trained with $\lambda = 0.96$, $\sigma = 0.014$. After training, all networks shown here achieved a test set performance within 25% of the optimal performance.

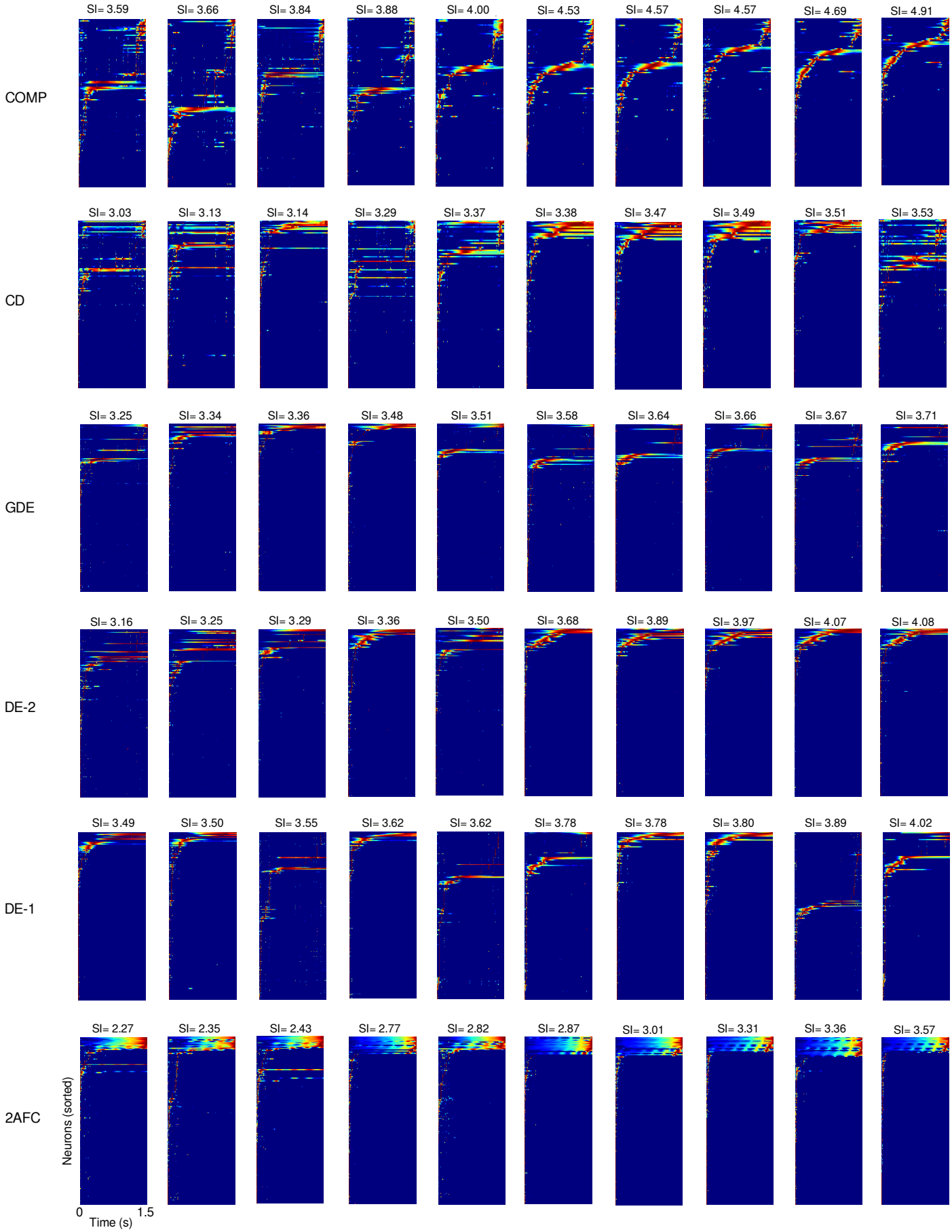

Figure S3: More example trials from the six tasks (basic condition). Sequentiality indices (SI) of the trials are indicated at the top of the plots. Trials are ordered by increasing SI from left to right. All trials shown here are from networks trained with $\lambda = 0.96$, $\sigma = 0.006$. After training, all networks shown here achieved a test set performance within 50% of the optimal performance.

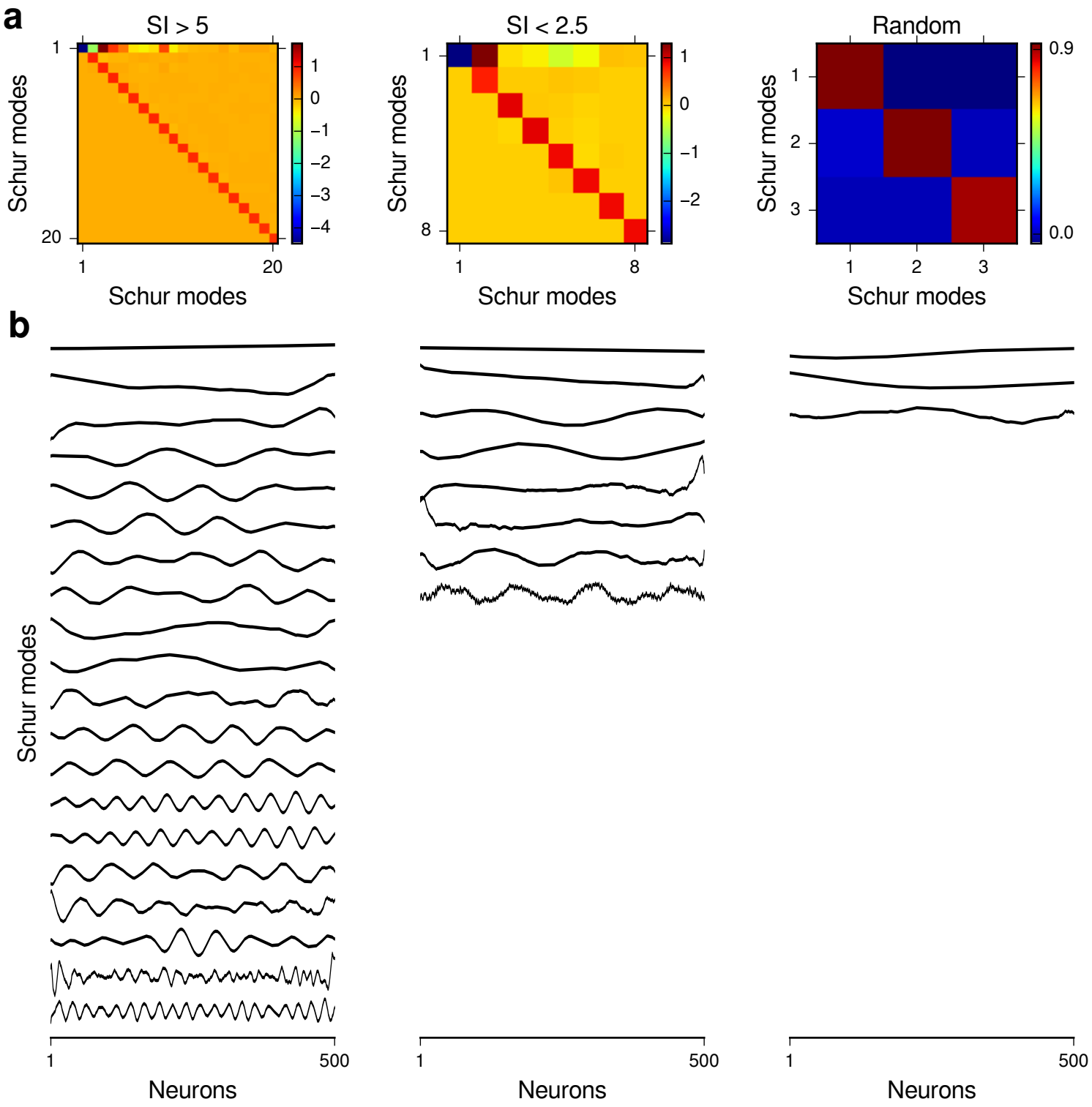

Figure S4: Schur decomposition of trained and random connectivity matrices. **a** Schur mode interaction matrices for the mean recurrent connectivity patterns shown in Figure 5a-c. Only significant Schur modes with at least one interaction of magnitude greater than 0.04 with another Schur mode are shown here. **b** The corresponding significant Schur modes. Networks with more sequential activity (SI $>$ 5) have more high-frequency Schur modes than networks with less sequential activity (SI $<$ 2.5). The random networks are close to normal.

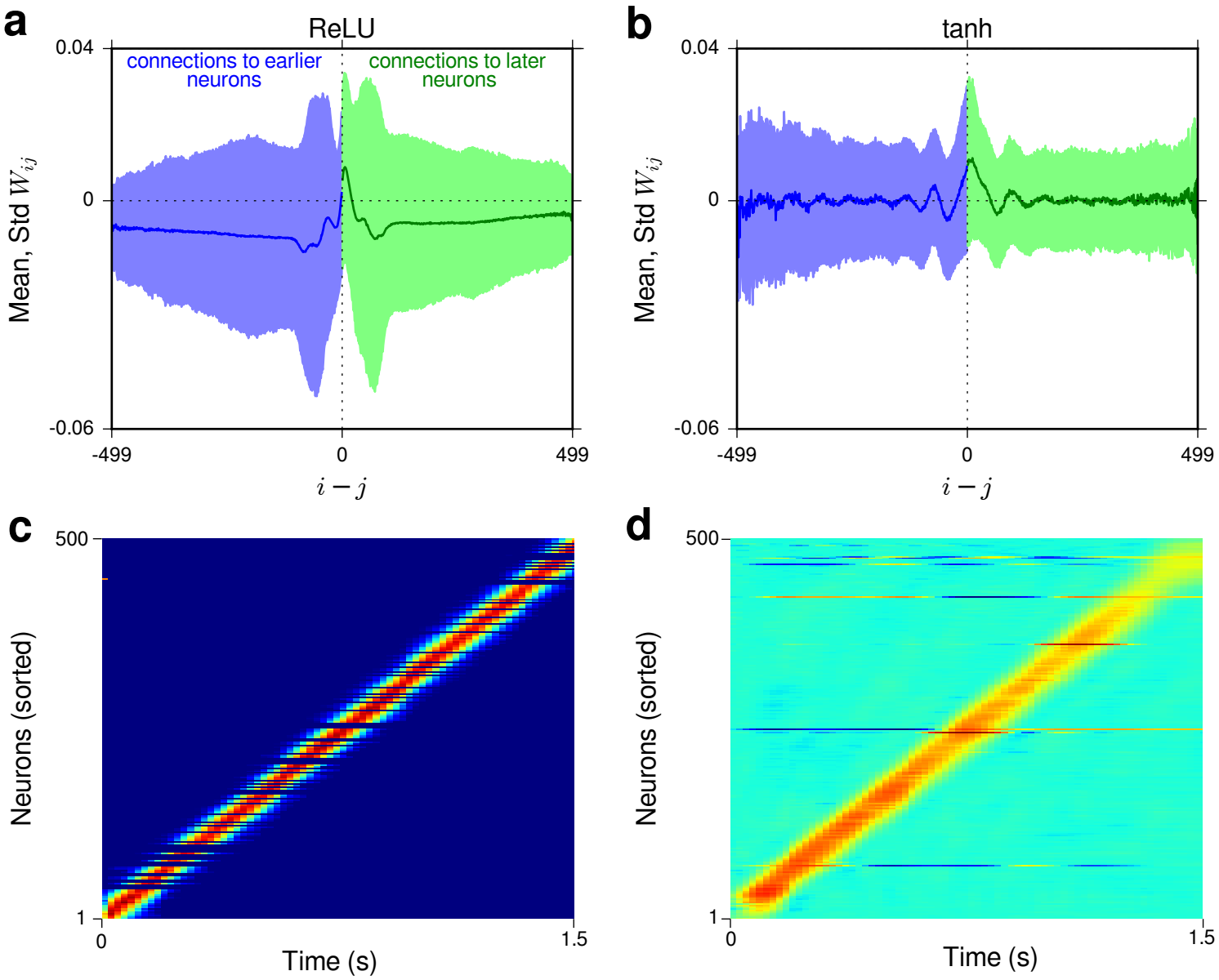

Figure S5: Results from networks explicitly trained to generate sequential activity as in [25]. **a-b** are analogous to Figure 5a-b and show the weight profiles obtained in trained networks with ReLU and tanh nonlinearities, respectively. **c-d** show example trials for the corresponding networks (trained with the same initial condition). Only networks with sequentiality index larger than 5.45 were included in the results shown here.

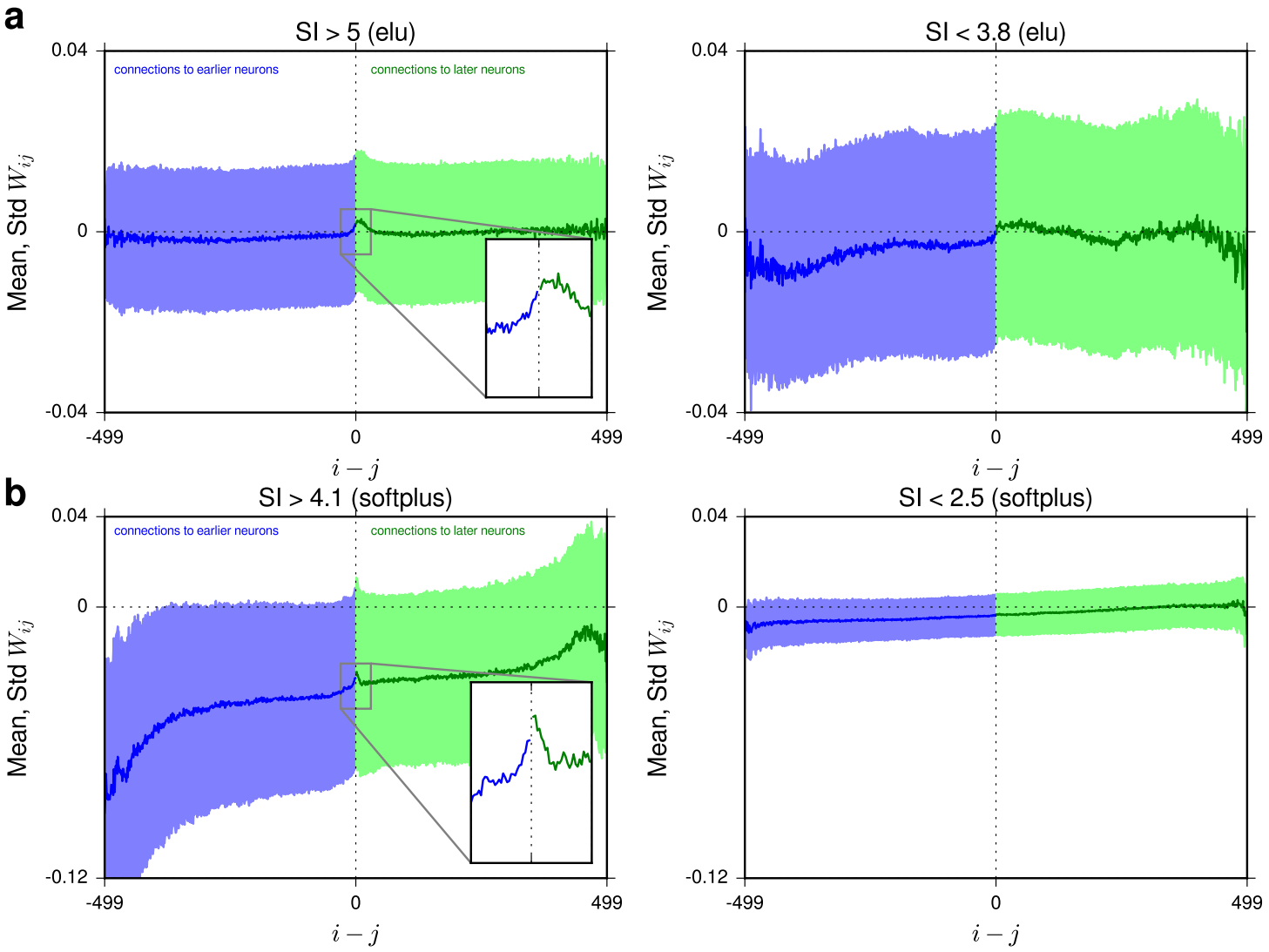

Figure S6: Circuit mechanism that generates sequential activity in networks with alternative activation functions. This figure is analogous to Figure 5a-b, but the results shown are for networks with the exponential linear (elu) activation function (**a**) and networks with the softplus activation function (**b**). Note that the elu activation function typically produced larger SIs than softplus, hence slightly different SI thresholds were used in the two cases to determine low and high SI networks.