SCIENTIFIC REPORTS

OPEN

Received: 29 December 2016
Accepted: 11 April 2017
Published online: 30 May 2017

Working Memory Requires a Combination of Transient and Attractor-Dominated Dynamics to Process Unreliably Timed Inputs

Timo Nachstedt[1,2] & Christian Tetzlaff[2,3]

Working memory stores and processes information received as a stream of continuously incoming stimuli. This requires accurate sequencing and it remains puzzling how this can be reliably achieved by the neuronal system as our perceptual inputs show a high degree of temporal variability. One hypothesis is that accurate timing is achieved by purely transient neuronal dynamics; by contrast a second hypothesis states that the underlying network dynamics are dominated by attractor states. In this study, we resolve this contradiction by theoretically investigating the performance of the system using stimuli with differently accurate timing. Interestingly, only the combination of attractor and transient dynamics enables the network to perform with a low error rate. Further analysis reveals that the transient dynamics of the system are used to process information, while the attractor states store it. The interaction between both types of dynamics yields experimentally testable predictions and we show that this way the system can reliably interact with a timing-unreliable Hebbian-network representing long-term memory. Thus, this study provides a potential solution to the long-standing problem of the basic neuronal dynamics underlying working memory.

Humans and animals continuously receive information conveyed by stimuli from the environment. To survive, the brain has to store and process this stream of information which is mainly attributed to the processes of working memory (WM[1, 2]). These two distinct abilities of WM, to store and to process information, yield a debate about the underlying neuronal network dynamics[3–5]: the network dynamics might either follow (i) attractor or (ii) transient dynamics.

Attractor dynamics denotes neuronal network dynamics which is dominated by groups of neurons being persistently active. In general, such a persistent activation is related to an attractor state of the dynamics, with each attractor associated to a specific information content[3, 6–8]. Several experimental and theoretical studies hypothesize that the dynamics underlying WM are dominated by such persistent dynamics[5, 8–10]. In contrast to attractor dynamics, neuronal networks with *transient dynamics* are dominated by an attractor-less continuous flow of neuronal activity across a possibly large neuronal population[11–14]. This type of dynamics implies a high diversity and complexity which is linked by theoretical studies with a large computational capacity required to process information[15–17]. These theoretical studies as well as several pieces of experimental evidence[18–20] yield the hypothesis that the dynamics underlying WM are dominated by transient dynamics[20, 21]. Thus, although the two hypotheses – attractor or transient dynamics – seem to contradict each other, experimental and theoretical evidence supports both yielding a debate about the neuronal network dynamics underlying WM[5].

To resolve this contradiction, in this study, we consider the fact that the timing of stimuli received by the WM is highly unreliable. In other words, when interacting with the environment, the WM of humans and animals evidently cannot rely on receiving precisely timed stimuli. For instance, listening to spoken language requires the ability to deal with different and irregular speech rates. The influence of such variance in the stimuli timing on the WM operation has been mainly analyzed on the psychological level[22] using, amongst others, the so-called *N*-back task. In this task a subject is exposed to a stream of different stimuli[23, 24]. Whenever a new stimulus is presented,

[1]Third Institute of Physics, Universität Göttingen, 37077, Göttingen, Germany. [2]Bernstein Center for Computational Neuroscience, 37077, Göttingen, Germany. [3]Max Planck Institute for Dynamics and Self-Organization, 37077, Göttingen, Germany. Correspondence and requests for materials should be addressed to T.N. (email: timo.nachstedt@phys.uni-goettingen.de)

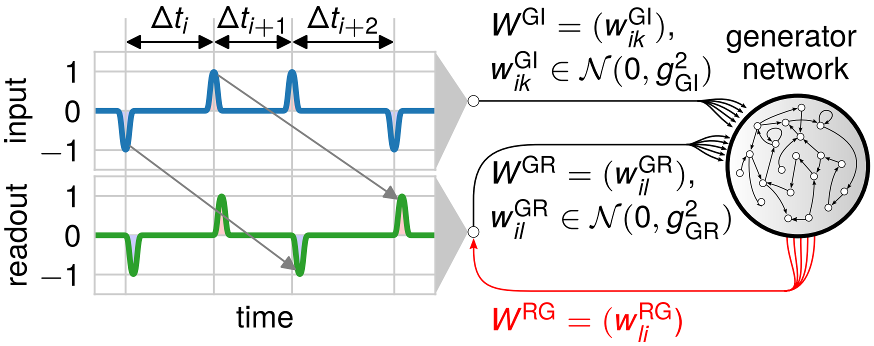
Δt_i
Δt_{i+1}
Δt_{i+2}
input
1
0
−1
readout
1
0
−1
time
$W^{\mathrm{GI}} = (w^{\mathrm{GI}}_{ik})$,
$w^{\mathrm{GI}}_{ik} \in \mathcal{N}(0, g^2_{\mathrm{GI}})$
generator network
$W^{\mathrm{GR}} = (w^{\mathrm{GR}}_{il})$,
$w^{\mathrm{GR}}_{il} \in \mathcal{N}(0, g^2_{\mathrm{GR}})$
$W^{\mathrm{RG}} = (w^{\mathrm{RG}}_{li})$

Figure 1. Setup of the benchmark *N*-back task to test the capability of transient networks to cope with variances in the input timings. The input signal is composed of smooth either positive or negative pulses separated by time intervals Δt_i drawn from a normal distribution with mean $\mu_{\Delta t}$ and variance $\sigma^2_{\Delta t}$. It is projected into the generator network via a synaptic weight matrix W^{GI} with elements w^{GI}_{ik} drawn from a normal distribution with zero mean and variance g^2_{GI}. The task is to produce an output pulse of defined shape (at the readout neurons) when a new input pulse is presented. The sign of the output pulse depends on the second last input pulse (compare arrows). The readout weight matrix W^{RG} is adapted during learning (red). The resulting readout signal is fed back into the network with a weight matrix W^{GR} with elements w^{GR}_{il} drawn from a normal distribution with zero mean and variance g^2_{GR}.

the subject has to execute an action which depends on the stimulus presented *N* stimuli before. Therefore, in order to succeed in this task, the subject has to store the information of the last *N* stimuli in its WM. Dependent on the timing of the stimuli, this information has to be continuously updated. Interestingly, whether the stimuli are presented with exact inter-stimulus timing or with unreliable timing *does not* influence the subject's performance of solving the *N*-back task[22]. This result indicates that the mechanisms implementing WM are robust against variance in the timing of the input stimuli. Based on this experimentally found property of WM, in this study, we investigate under which conditions the dynamics of the neuronal networks underlying WM is able to perform an *N*-back task with the same robustness with respect to variances in the stimuli timing.

First, we investigate a theoretical neuronal network model of WM showing purely transient dynamics – a so called reservoir network[25,26] – and test its performance on the *N*-back task. Interestingly, with small variations of the timing of the inputs, such a purely transient system exhibits a very poor performance (Figs 1 and 2). In the next step, we show that the performance of the network increases significantly if the system is directly trained in a supervised manner to maintain the relevant information (Figs 3 and 4). A further analysis reveals that the underlying neuronal dynamics of the trained system are dominated by attractor states which are interlinked by regions of transient dynamics. By comparing these combined dynamics with the dynamics of the purely transient system during performing the *N*-back task, we demonstrate that only this combination of attractor and transient dynamics allows the execution of the task robust against variances in stimuli timings (Fig. 5). In addition, we show that, in general, the attractor states store the task-relevant information while the transient dynamics processes the information (Figs 3 and 6). This yields the prediction that a drop in performance resulting from an additional delay between the current stimulus and the execution of the action can be avoided by introducing another stimulus pushing the system into a transient state (Fig. 6).

Furthermore, besides stimuli from the environment, also stimuli from other brain mechanisms, as long-term memory (LTM), are characterized by unreliable timing. We show that in established theoretical neuronal network models of LTM[27,28] the time needed for a cue-triggered recall of stored information varies dependent on the initial conditions of the recall-triggering cue and the neuronal network (Fig. 7). Due to the continuous coupling between WM and LTM, which is fundamental in order to solve complex tasks[29–32], this variance in recall timings has to be reflected in the dynamics of the WM. Thus, we show that, similar to the *N*-back task, only a neuronal network with a combination of attractor and transient dynamics enables a continuous and reliable coupling between WM and LTM which can be used to solve a complex multi-phase task (Fig. 8). This describes, to our knowledge, the first theoretical model of the functional, dynamic interaction between a WM- and a LTM-network.

Results

Reservoir networks are vulnerable to variances in stimuli timings.

Stimuli received by the working memory (WM), coming from the environment as well as from the long-term memory (LTM), are characterized by an unreliable timing of their occurrence. Thus, to function in a reliable manner, the WM has to reduce the influence of these timing variances. Within the last years, neuronal networks with purely transient dynamics –so-called reservoir networks[25,26] –have been proposed as a theoretical model of WM[5]. A reservoir network consists of a generator network, being composed of N_{G} basically randomly connected neurons, which receives temporally varying input stimuli from a set of N_{I} input neurons and projects signals to a downstream output layer with N_{R} readout neurons. Due to the random connectivity within the generator network, the input stimuli are transformed into a variety of complex traces or, in other words, the inputs are processed in different variants by the network. As a consequence, the readout of a desired processing or target signal requires only the optimization of the weight matrix W^{RG} of the synapses transmitting signals from the generator neurons to the readout neurons. In the following, to ensure generality of our results, the readout weight matrix W^{RG} is optimized by

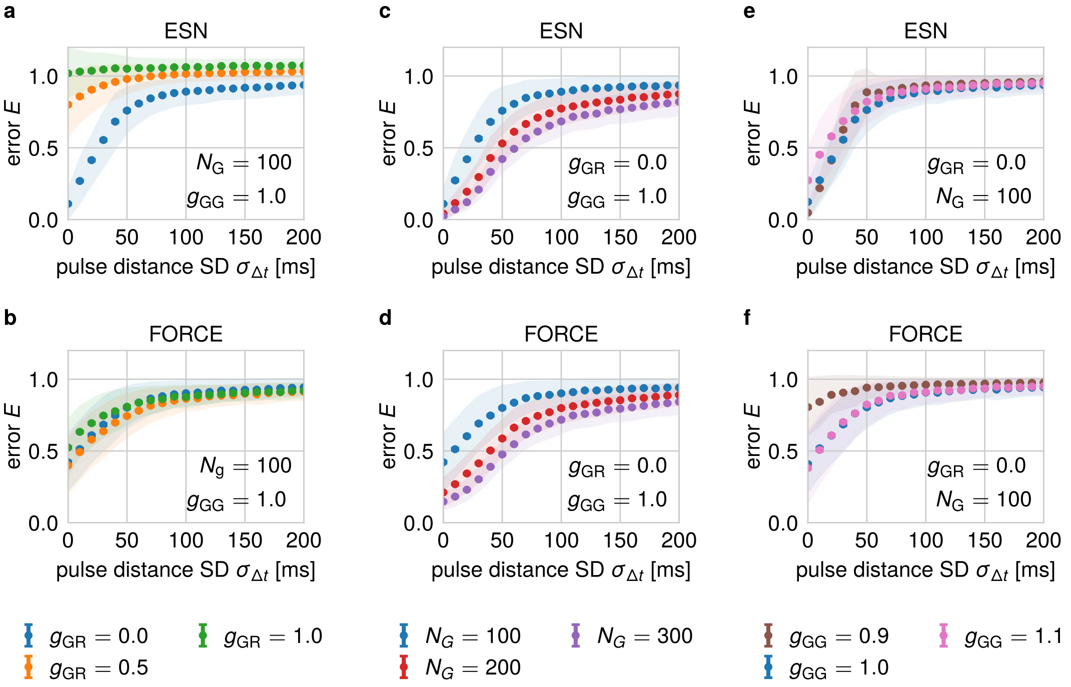

Figure 2. Influence of variances in input timings on the performance of the transient network. The mean normalized readout error E (see Methods) for the benchmark task depicted in Fig. 1 increases with larger standard deviation $\sigma_{\Delta t}$ of the interstimulus intervals of the input stream independent of the used parameters. In (**a**,**c**,**e**), the network is trained using the echo state network approach (ESN). In (**b**,**d**,**f**), the FORCE-learning method is employed. Every data point represents the mean of 1000 network instantiations. The shaded area indicates the standard deviation of the respective error distribution. The error bars show the standard error of the mean. If for one instantiation the error after training is larger than 1.5, we consider the respective training procedure as not converged and exclude it from the mean. (**a**,**b**) The network is trained with three different values of the standard deviation g_{GR} of the feedback-weights from the readout neurons to the generator network. For both training methods, increasing g_{GR} also increases the error E for a given value of $\sigma_{\Delta t}$. The constant parameters are $N_G = 100$ and $g_{gg} = 1.0$. (**c**,**d**) Networks of different sizes, i.e. different values of N_G, are trained to perform the benchmark task. While larger networks perform better for a given value $\sigma_{\Delta t}$, they qualitatively show the same strong sensitivity to variances in input timings. The constant parameters are $g_{GR} = 0$ and $g_{gg} = 1.0$. (**e**,**f**) The influence of different values g_{GG} of the internal weights of the generator network is investigated. Neither increasing nor decreasing of the critical value g_{GG} reduces the error significantly. The constant parameters are $g_{GR} = 0$ and $N_G = 100$.

two alternative strategies (for details see Methods section): On the one hand, we use the offline ESN-approach[25] and, on the other hand, we apply the online FORCE-algorithm[33]. Thus, after optimization, the readout neurons optimally "combine" the signals naturally present in the generator network. Reservoir networks have been shown to posses a high computational capacity as well as a short-term memory capacity[16, 25, 26] which are the two main components of WM.

We investigate the capability of a standard reservoir network to cope with the described timing variances in a setting similar to the N-back task. The neuronal network receives a stream of input stimuli, each of them is a pulse and has either a positive or negative sign (blue line in Fig. 1). At every occurrence of an input stimulus, the network has to produce a non-zero output signal at the readout neuron (green line) of predefined temporal shape (target shape) with a sign equaling the sign of the stimulus received N stimuli before (here, $N = 2$; indicated by arrows). Note that even though here the target shape for the output has the same pulsed shape as the input stimuli, the computational capacity of the network allows that it could be of arbitrary shape (see Supplementary Figure S1 for an example with sine-shaped output signals). In general, to solve this N-back task, the network has to fulfill two sub-tasks: It has to store the sign of the last two input stimuli (storage of information) and, given the next input pulse, it has to produce an output signal of target shape with the sign equaling the pulse presented N stimuli before. The latter depicts the processing of information as the network has to "combine" the stored information (sign) with the transient network dynamics to produce a complex temporal output signal (target shape). Note that the used target shape also implies that the output signal is zero if no input is present. Variances in the timing of occurrence of the input stimuli are introduced by randomly drawing the interstimulus intervals Δt_i from a normal distribution with mean $\mu_{\Delta t}$ and variance $\sigma_{\Delta t}^2$.

The performance of a reservoir network instantiation on the N-back task is evaluated after the training of its readout weight matrix by calculating the root mean square error E determining the difference between target and actual output signal (see Methods). We systematically investigate the influence of the variance in the timing of the

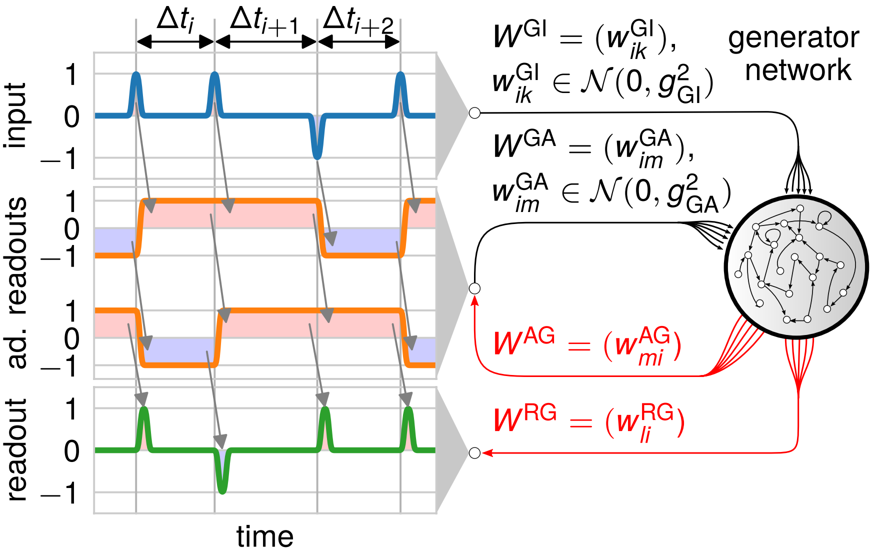

Figure 3. Setup of the benchmark *N*-back task to test the influence of additional, specially-trained readout neurons to cope with variances in the input timings. The input signal as well as the target signal for the readout neuron are the same as before (Fig. 1). Additional neurons, which are treated similar to readout units, are introduced in order to allow for storing task-relevant information. These additional neurons (ad. readouts) have to store the sign of the last and second last received input pulse as indicated by the arrows. The activities from the additional neurons are fed back into the network with weights w_{im}^{GA} drawn from a normal distribution with zero mean and variance g_{GA}^2 basically extending the network. Synaptic weights adapted by the training algorithm are shown in red. The feedback from the readout neurons to the generator network is set to be zero ($g_{GR} = 0$).

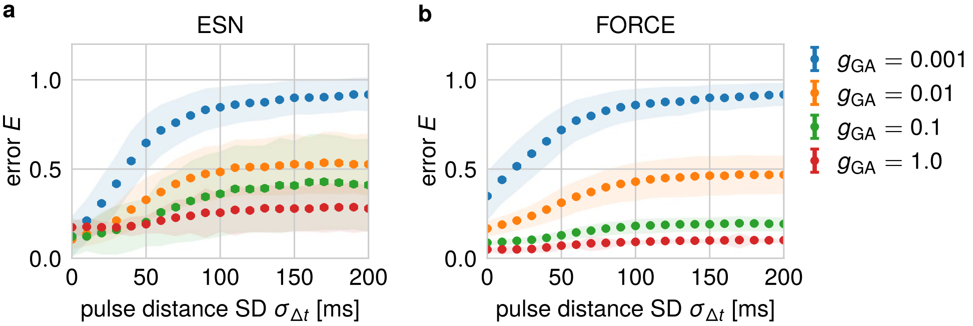

Figure 4. Influence of variances in input timings on the performance of the network with specially-trained neurons. The normalized readout error *E* of a network with specially-trained neurons decreases with larger values of the standard deviation g_{GA} determining the feedback between specially-trained neurons and network. If this standard deviation equals 1, the error stays low and becomes basically independent from the standard deviation $\sigma_{\Delta t}$ of the inter-pulse intervals of the input signal. (**a**) ESN approach; (**b**) FORCE-method.

input stimuli by varying the standard deviation $\sigma_{\Delta t}$ of the interstimulus intervals Δt while keeping the mean $\mu_{\Delta t}$ constant. For each value of the standard deviation, we average the performance over 1000 different (random) network instantiations. Overall, independent of the training method (ESN as well as FORCE) used for the readout weights, the averaged error $\langle E \rangle$ increases significantly with increasing values of $\sigma_{\Delta t}$ until it converges to its theoretical maximum at 1 at about $\sigma_{\Delta t} \approx 100$ ms (Fig. 2). Note that errors larger than 1 are artifacts of the used training method. The increase of the error (or decrease of the performance) with larger variances in the stimuli timings is independent of the parameters of the reservoir network. For instance, we tested the influence of different values of the variance g_{GR}^2 of the feedback weight matrix W^{GR} from the readout neurons to the generator network (Fig. 2a for ESN and b for FORCE). For the present *N*-back task, feedback of this kind does not improve the performance, although several theoretical studies show[33–35] that feedback enhances the performance of reservoir networks in other tasks. In contrast, we find that increasing the number of generator neurons N_G reduces the error for a broad regime of the standard deviation $\sigma_{\Delta t}$ (Fig. 2c and d). Nevertheless, the qualitative relationship is unchanged and the improvement is weak implying a need for large numbers of neurons to solve this rather simple task for medium values of the standard deviation. Another relevant parameter of reservoir networks is the standard deviation g_{GG} of the distribution of the synaptic weights within the generator network determining the spectral radius of the weight matrix[25]. In general, the spectral radius determines whether the network operates in a sub-critical,

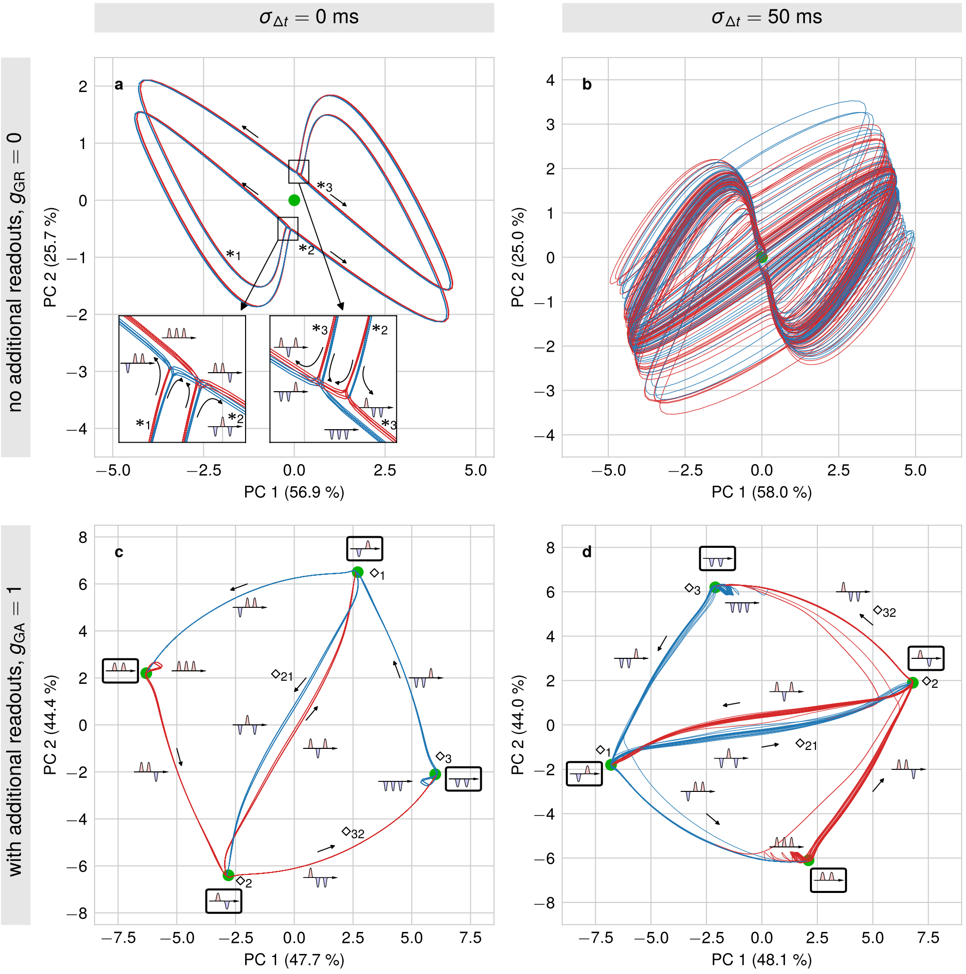

Figure 5. Neural network dynamics during performing the benchmark task projected onto the first two principal components. Trajectory sections which should trigger a positive pulse at the readout units are drawn in red while those which should trigger a negative response are shown in blue. The small arrows indicate the direction in which the system flows along the trajectory. The small pictograms indicate the recent history of the input pulses along the time axis. Green dots indicate attractor states (manually added). (**a**) The network without additional readouts (Fig. 1) stores the history of stimuli on transients. (**b**) By introducing variances in input timings, these transients smear impeding a proper readout. (**c**) The additional readouts (or specially-trained neurons; Fig. 3) "structure" the dynamics of the system by introducing several attractor states each storing the history of the last two stimuli. (**d**) Even in the presence of timing variances the attractor-dominated structure in phase space is preserved enabling a proper readout. Parameters: mean inter-pulse interval $\mu_{\Delta t} = 100$ ms; (**a**) $g_{GR} = 0$, $\sigma_{\Delta t} = 0$ ms; (**b**) $g_{GR} = 0$, $\sigma_{\Delta t} = 50$ ms; (**c**) $g_{GA} = 1$, $\sigma_{\Delta t} = 0$ ms; (**d**) $g_{GA} = 1$, $\sigma_{\Delta t} = 50$ ms. Details see Supplementary Discussion S1.

critical or chaotic regime and also influences the time scale of the reservoir dynamics[33, 36]. Here, we find that both an increase as well as a decrease of g_{GG} of about 10% decrease the performance of the system (Fig. 2e,f). Additionally, it turns out that all findings remain valid also when the performance of the network is evaluated in a less restrictive manner by only distinguishing three discrete states of the readout and target signals (Supplementary Figure S2).

In summary, independent of the used parameter values, we find that if the input stimuli occur in an unreliable manner, a reservoir network with purely transient dynamics has a low performance in solving the *N*-back task. This raises doubts about its applicability as a plausible theoretical model of the dynamics underlying WM.

Specially-trained neurons improve the performance. To obtain a neuronal network which is robust against variances in the timing of the input stimuli, we modify the reservoir network to allow for more stable memory storage. For this, we add (here, two) further neurons to the system and treat them as additional readout neurons by training (ESN as well as FORCE) the weight matrix W^{AG} between generator network and added neurons (similar to the readout matrix W^{RG}). Different to the readout neurons, the target signals of the added neurons are defined such that, after training, the neurons produce a constant positive or negative activity depending on the sign of the last or second last input stimuli, respectively (Fig. 3). The activities of the additional neurons are fed back into the reservoir network via the weight matrix W^{GA} (elements drawn from a normal distribution with zero

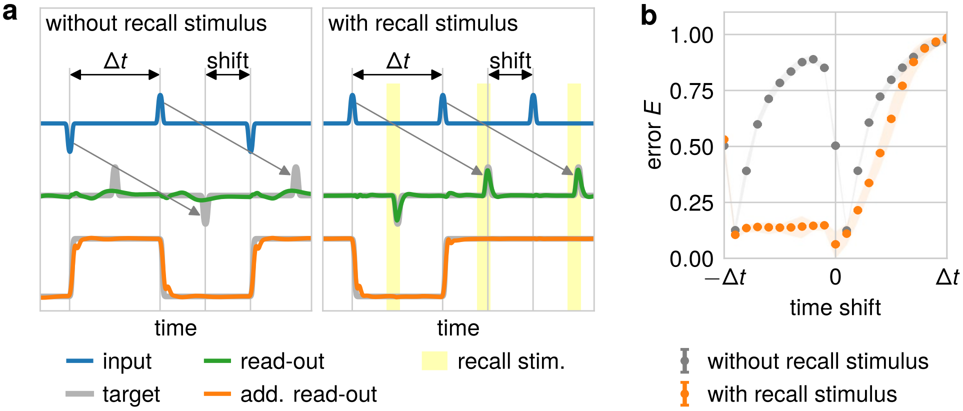

Figure 6. Prediction of the influence of an additional recall stimulus. (**a**) An additional temporal shift is introduced between input and output pulse. In the second setup (lower row) a recall stimulus is applied to the network to trigger the output. This recall stimulus is not relevant for the storage of the task-relevant sign. (**b**) In general the temporal shift increases the error of the system (gray dots; each data point indicates the average over 20 trials) as the system has already reached an attractor state. Introducing a recall stimulus (orange dots) decreases the error for all negative shifts as the system is pushed out of the attractor and the task-relevant information can be read out. This effect diminishes for positive temporal shifts as the system has already forgotten the corresponding information.

mean and variance g_{GA}^2) basically extending the generator network. This procedure ensures that the network memorizes the later-on required information[14, 37]. Note that the feedback from the readout neurons to the generator network is neglected ($g_{\mathrm{GR}} = 0$).

As above, we evaluate the performance of the extended network while solving the N-back task. In general, for a weak feedback from the additional neurons to the generator network (small values of g_{GA}), larger standard deviations $\sigma_{\Delta t}$ of the interstimulus intervals Δt result in larger errors E (Fig. 4a for ESN and b for FORCE). However, increasing the standard deviation g_{GA} of the synaptic weights from the additional neurons to the generator network decreases the influence of the variances in stimuli timings on the performance of the system. For $g_{\mathrm{GA}} = 1.0$, the error is only slightly dependent on the standard deviation $\sigma_{\Delta t}$ of the interstimulus intervals (Fig. 4). The extension of the network by these specially-trained neurons yields a significant improvement compared to the best setup without these neurons (Fig. 2). Please note that this finding also holds for a less restrictive performance evaluation (Supplementary Figure S3). Additionally, the same qualitative finding can also be obtained for significantly larger reservoir networks (Supplementary Figure S4). In the following, we investigate the dynamical principles underlying this increase in performance.

The combination of attractor and transient dynamics increases performance.

Instead of analyzing the complete high-dimensional activity dynamics of the neuronal network, we project the activity vectors onto its two most significant principal components to understand the basic dynamics[38] underlying the performance changes for the N-back task.

For the purely transient reservoir network (without specially-trained neurons; Figs 1 and 2), we investigate the dynamics of the system with $g_{\mathrm{GR}} = 0$, $N_{\mathrm{G}} = 100$, and $g_{\mathrm{GG}} = 1$ as a representative example in more detail (Fig. 5a). The dynamics of the network is dominated by one attractor state at which all neuronal activities equal zero (silent state). However, as the network continuously receives stimuli, it never reaches this state. Instead, dependent on the sign of the input stimulus, the network dynamics runs along specific trajectories (Fig. 5a; red trajectories indicate that the second-last stimulus was positive while blue trajectories indicate a negative sign). The marked trajectory ($*_1 \rightarrow *_2 \rightarrow *_3$) corresponds to a network having recently received a negative and two positive stimuli which now is exposed to a sequence of two negative stimuli (for details see Supplementary Discussion S1). The information about the signs of the received stimuli is stored in the trajectory the network takes (transient dynamics). However, the presence of variances in the timing of the stimuli significantly perturbs this storage mechanism of the network. For $\sigma_{\Delta t} = 50\,\mathrm{ms}$ (Fig. 5b), the trajectories storing positive and negative signs of the second-last stimulus cannot be separated anymore. As a result, the downstream readout neuron fails to extract the task-relevant information.

Extending the reservoir network by the specially-trained neurons changes the dynamics of the system significantly (here, $g_{\mathrm{GA}} = 1$): The network now possesses four distinct attractor states with specific, transient trajectories interlinking them (Fig. 5c). The marked trajectory ($\diamond_1 \rightarrow \diamond_{21} \rightarrow \diamond_2 \rightarrow \diamond_{32} \rightarrow \diamond_3$) corresponds to the same sequence of stimuli as above (for details see Supplementary Discussion S1). Here, the information about the sign of the two last stimuli is stored in the attractor states while the transients, connecting them, are used to process the information and to produce the complex output signal (target shape). Due to the attractor states, which "structure" the dynamics, variance in the timing of stimuli (here, with a standard deviation of $\sigma_{\Delta t} = 50\,\mathrm{ms}$) does not significantly alter the neuronal dynamics (Fig. 5d). The different trajectories and attractor states remain clearly separated. This separation in the presence of timing variance enables the downstream neurons to read out the task-relevant information, which is the dynamical cause of the good performance in the N-back task.

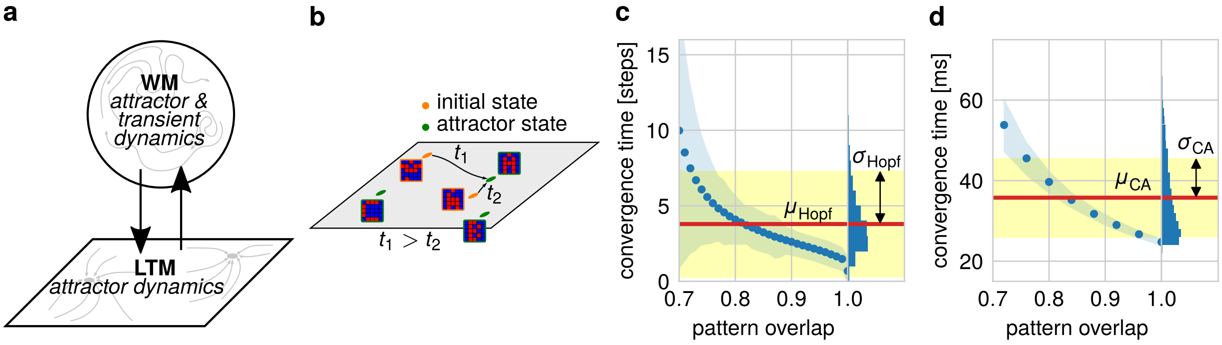

Figure 7. Network models of long-term memory show variances in recall timings. (**a**) As WM and LTM interact continuously, inherent properties of the LTM network influence the function of WM. (**b**) In standard network models of LTM, the recall of a memory representation (e.g., the letter "A") corresponds to the convergence of the neuronal system dynamics to a previously learned attractor (green dots). The time span required until convergence ("convergence time"; here t_1 and t_2) depends on the initial state of the system or the recall stimulus (differently altered "A"s; orange dots). **(c)** 100 random patterns are stored into a standard Hopfield network[43] with $N_{\text{Hopf}} = 1000$ neurons. Patterns with differences in d neurons to one of the stored patterns, corresponding to a pattern overlap of $1 - d/N_{\text{Hopf}}$, are used as initial states for the system. The number of simulation steps to reach the stored pattern represents the convergence time (dots mark average value over 10000 trials with standard deviation indicated by blue shading). The histogram on the right-hand side illustrates the resulting distribution of convergence times for a uniform distribution of initial pattern overlaps (red line shows mean μ_{Hopf} and yellow shading the corresponding standard deviation σ_{Hopf}). **(d)** The recall convergence times analyzed similar to panel (C) for a self-organizing cell-assembly network[28]. A cell assembly is trained to represent a given input pattern. After training, a similar pattern with an overlap of $1 - d/N_{\text{aff}}$, is presented to the network (i.e. d neurons have a different activity compared to the learned pattern). The time span required to activate 90% of the cell assembly represents the convergence time.

Note that, although an attractor state stores information, this information cannot be read out when the output signal has to be time-dependent (non-constant). This includes tasks during which the information has to be read out at specific periods of time and otherwise not. By introducing a time shift between stimulus and information readout (output) even without variances in input timings (constant Δt; Fig. 6), this yields an interesting prediction. A negative shift means that the information has to be read out before the next stimulus is presented. Interestingly, for large negative shifts ($\approx -\Delta t$), the error is small as the readout is required briefly after the last stimulus and the system is still in a transient state. With smaller negative shifts, the error increases as the system reaches the next attractor and the information cannot be read out anymore in a time-resolved way. Note, however, that the pure information about the sign can be easily read out from the reservoir as done by one of the two specially - trained readout units. This information alone, however, cannot be used to generate a time-resolved signal. The situation changes when we introduce another stimulus with task-irrelevant sign which triggers the output: now, the error is small for all negative shifts as the stimulus "kicks" the system out of the attractor back in a transient state such that the stored information can be read out in a time-resolved way. This does not work for positive shifts as, here, the relevant information is already lost (overwritten by sign of next stimulus).

These results demonstrate that the information is reliably stored in the attractor states of the network dynamics while the information processing (temporally specific readout) happens on the transients. This combination of dynamics also enables the reliable interaction of the WM with other brain mechanisms as shown in the following.

Attractor and transient dynamics enable continuous interaction with long-term memory. For solving complex tasks humans and animals incorporate several brain mechanisms which yields, amongst others, to a continuous interaction between working memory and long-term memory (LTM)[29–32] (Fig. 7a). On the one side, information can be transmitted from the WM to the LTM to be stored and to "free" the computational capacities of the WM for processing further information. On the other side, information stored in the LTM can be transmitted back to the WM to be processed. Such interactions imply that the WM has to deal with the inherent properties of the LTM arising from its underlying dynamics. Several experimental and theoretical studies indicate that the neuronal networks implementing LTM are dominated by attractor dynamics[39–42]. Thereby, an attractor state corresponds to a long-term memory representation which is recalled if the network dynamics converges to this state. Note that the convergence time – the time span the system requires to reach the attractor state – is mainly influenced by the recall stimulus and by the initial state of the system (Fig. 7b). As both the recall stimulus and the initial state vary between different recall trials, this variation yields a broad distribution of convergence times (Fig. 7c for the Hopfield model[43] and Fig. 7d for a Hebbian cell assembly model[28]; please see Methods for more details), which is also found in psychophysical experiments[44]. The width of this distribution depends on the parameters of the system, as the network size[45], but it always stays larger than zero implying a variance in the convergence times. In other words, if a complex task requires that information has to be recalled from the LTM and transmitted to the WM, this recalled information reaches the WM with unreliable timings.

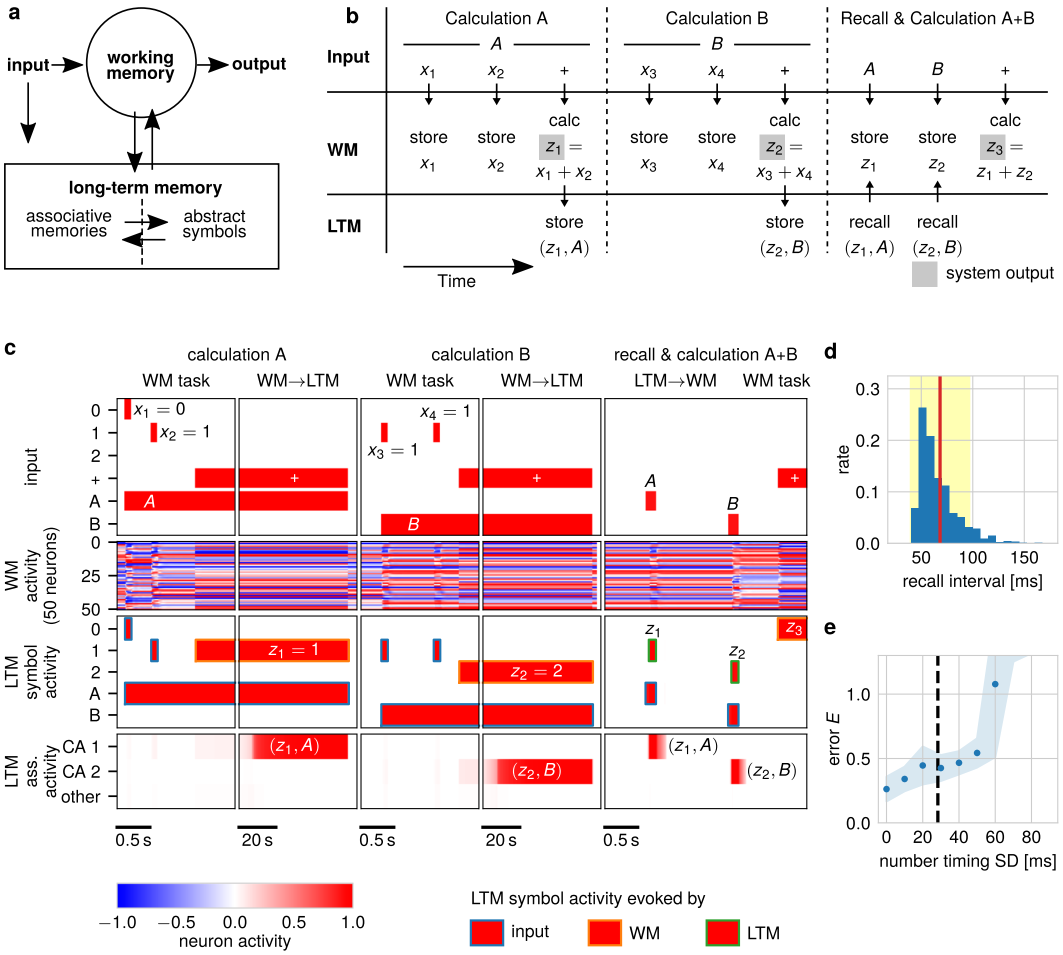

Figure 8. A multi-phase task requires the continuous interaction between WM and LTM. A WM network consisting of attractor states and transient dynamics enables the reliable interaction with an LTM network to solve a complex multi-phase task. Details see main text. (**a**) The external input is projected into WM and LTM. For simplicity, the LTM area is separated into two compartments: the first stores abstract symbols and the second forms input-dependent associations between these symbols. (**b**) Multi-phase task and information flow between input, WM, and LTM. (**c**) External inputs, activities of 10% of the $N_G = 500$ WM-neurons, activities of the neurons within the LTM symbol area and activities within the LTM association area during the three phases. For the activities in the LTM symbol area, we indicate whether this activation is evoked by the external input, the WM, or from the LTM association area. (**d**) Distribution of the time intervals needed for the recall of the information stored in cell assemblies in the LTM-association area for equally distributed trials with 75% to 100% complete context signals. Here, the recall interval is defined as the time span between the onset of the external context stimulus and the point of time at which more than 50% of the neurons in the LTM-symbol area, representing the recalled number, fire at rates higher than 90% of the maximum firing rate. The standard deviation of this distribution is approximately 28 ms. (**e**) Error of a purely transient network with $N_G = 500$ neurons performing the summation task (third phase) with only variances in input timings from the LTM recall alone. According to (**d**), this source of unreliability alone already doubles the readout error (dashed vertical line).

In the following, we will focus on a complex multi-phase task requiring the interaction between WM and LTM (Fig. 8a,b). For simplicity, we assume that the LTM has already formed representations of abstract symbols as numbers and learning context. Now, the LTM has to form associations between these abstract symbols dependent on external inputs and inputs from the WM. Here, we use a self-organizing cell assembly network to form and store these associations (similar to the system used for Fig. 7d[17,28]). The WM receives inputs (external and from the LTM), processes them, and provides the system's output. The dynamics of the WM are governed by the combination of attractor and transient dynamics described above. This combination enables the WM to deal with the unreliability of the LTM-network to provide the task-dependent information.

The task consists of several phases (Fig. 8b,c): In the first phase (context A), the WM-network receives and stores two stimuli (x_1 and x_2) each representing a number, followed by a +-signal instructing the WM to add both numbers (for simplicity we use for all calculations the modulo-three-operator, thus, $z_1 = x_1 + x_2 \bmod 3$). The result z_1 is transferred to the LTM which forms a cell assembly storing the association between context A and result z_1. Thus, the WM processes the information and the LTM stores the result of this processing for later reuse. This "frees" the WM making its processing capacities available for other tasks, as needed in the second phase (context *B*) during which these capacities are used for performing a second calculation. Similar to the first phase, two numbers are presented (x_3 and x_4) and the WM calculates the sum of them. The result is maintained in the LTM by forming a new cell assembly storing the association between context *B* and result z_2. In the third phase (context $A + B$), the information stored in the LTM is transferred back into the WM for further processing. Namely, by externally activating the context signal *A* (*B*), the LTM recalls the stored association and transfers z_1 (z_2) to the WM. Importantly, due to the variance in convergence times of the LTM (Fig. 7d), the WM receives the information z_1 and z_2 with an unreliable timing. However, as it consists of attractor and transient dynamics (Fig. 8c, WM activity), the WM-network can process the information and sum both numbers up. The result $z_3 = z_1 + z_2$ mod3 is the final result of the task. Note that, as in each phase the information relevant later on is stored in the LTM, the performance of the system is independent of the time span between the different phases and the WM can also perform other tasks in between. Furthermore, due to the different time scales and phases involved, both networks – WM and LTM – and their interaction are required to solve this multi-phase task.

The multi-phase task implies several sources of unreliability of input timings perturbing the proper function of the WM. External inputs can be unreliable (similar to Figs 1, 2, 3 and 4) as well as inputs from the LTM (Fig. 7). Even when the external signals are accurate in timing and the LTM is at each recall in the same initial state (here in the silent state; see Fig. 8c, LTM ass. activity), the context cue can induce unreliability. Namely, differences in the context cue triggering the recall of the corresponding association (third phase) compared to the original context signal presented during learning (first and second phase) yield a distribution of LTM recall timings with a significant standard deviation ($\sigma_{\text{recall}} \approx 28$ ms, Fig. 8d). Already this cue-induced variation alone leads to a doubling of the error when using a purely transient network as WM (dashed line in Fig. 8e). All these different sources of unreliability together impede the proper function of purely transient networks to solve this task. Therefore, all our attempts to solve this task with such a purely transient network failed. This indicates that the dynamics underlying working memory should consist of a combination of transient and attractor dynamics.

Discussion

The neuronal network dynamics underlying the proper function of working memory (WM) is still an unresolved question. Experimental findings are diverse with some studies supporting the view that WM operates mainly by transient dynamics[18,19] while others indicate that persistent activities, i.e. attractor states, suffices to explain WM functions[9,10]. Here, we considered the *N*-back task with variances in the timing of input stimuli[22] to draw conclusions on this dynamics. First, we showed that in purely transient systems the information about the *N* past stimuli is stored, as expected, in distinguishable trajectories. However, if the variance in the input timings increases, the trajectories are disturbed resulting in large overlaps between them which impede the readout of the stored information by downstream neurons. In contrast, introducing attractor states in the dynamics "structures" the phase space of the system: It stores the history of the past stimuli by remaining in the corresponding attractor. Only if a new stimulus is presented, independent of the timing, the system's dynamics traverses by "tubes" of transient dynamics to another, history-dependent attractor. This phase of transient dynamics between the attractor states is sufficient to perform complex temporal computations.

The most common type of purely transient network models of WM are reservoir networks[5,21,25,26,46]. The robustness of their performance when confronted with noise in the input or within the network has been extensively studied[34,47–50]. However, the susceptibility of such systems to variances in the timing of the input stimuli (Figs 1 and 2) has – to the best of our knowledge –not been considered and found before. Due to the universality of reservoir networks[25], we expect that the here-presented findings can be generalized to a large class of purely transient systems implying that purely transient dynamics in general are inadequate to describe the dynamics underlying WM. Instead, a combination of transient and attractor dynamics is required (Figs 3 and 4).

Systems consisting of transient dynamics and attractor states have been proposed and investigated before[14,37,38]. In general, these attractors are formed by introducing additional readout neurons which feed back into the generator network. In fact, several studies[33–35] indicate that a feedback from the readout neurons to the generator network enhances the stability and the performance of the network and can even lead to a recurrent neural network with universal computing power[34]. Additionally, the feedback can also be replaced by adapting the synaptic weights within the network in order to introduce attractor states[51]. Note that the attractor states do not have to be stable fixed points of the dynamics. It could be sufficient to introduce slow states[38] or heteroclinic channels[4,52] to make the system robust against variances in input timings. Furthermore, given that neuronal systems have multiple possibilities to store information[53], the attractors could also be realized in the synaptic dynamics[54,55] instead of (persistent) neuronal activities.

The here-derived hypothesis that the combination of transient dynamics and attractor states underlies the dynamics of WM is also supported by several experimental studies: For instance, it was shown that during a monkey WM-task, stimuli trigger the neuronal dynamics of prefrontal cortex neurons to perform fast (transient) transitions through different states until reaching a low-energy attractor state[56]. Similar dynamics are also found on larger brain scales in a fMRI study[57]. Interestingly, in the odor system of the locust, an odor stimulus evokes a succession of states resulting in an odor specific fixed point. However, the separation of different stimuli (a computational task) is optimal during the initial transient dynamics[4,12] supporting our hypothesis.

To solve more complex tasks, the WM has also to consider information stored in the long-term memory (LTM). The resulting interaction between WM and LTM is extensively investigated in psychological and psychophysical experiments[29–32] and its neuronal implementation[2,10,58–60]. However, from the theoretical side, this interaction has not been studied intensively (but see refs 61 and 62). Here, we show that LTM-systems are unreliable in recall timings (Fig. 7) which implies that the WM-network has to be robust against these unreliabilities to functionally interact with the LTM (Fig. 8). This robustness is achieved by the combination of attractor and transient dynamics underlying working memory.

In summary, in the present study we consider the variance in timings of the inputs to draw conclusions about the underlying basic neuronal dynamics of WM. In particular, we show that a reliable WM-system requires a combination of attractor states and transient dynamics. Furthermore, we argue that these different dynamic regimes have also different functional roles – attractors store information while transients process information – enabling a continuous interaction with an adaptive LTM-system and yielding experimental verifiable predictions. This provides a further step in understanding the principles generating the functionally important dynamics of working memory.

Methods

Reservoir Network. The network consist of an input layer, a generator network, and readout neurons. Accordingly, neurons in the actual reservoir are named generator neurons. With N_G generator neurons, N_I input signals and N_R readout neurons, and, if any, N_A additional specially-trained neurons, the dynamics of the generator neuron i is given by a membrane potential u_i^{G} described by

$$\tau \frac{\mathrm{d}u_i^{\mathrm{G}}}{\mathrm{d}t} = -u_i^{\mathrm{G}} + \sum_{j=1}^{N_\mathrm{G}} w_{ij}^{\mathrm{GG}} F_j^{\mathrm{G}} + \sum_{k=1}^{N_\mathrm{I}} w_{ik}^{\mathrm{GI}} I_k + \sum_{l=1}^{N_\mathrm{R}} w_{il}^{\mathrm{GR}} R_l \left(+ \sum_{m=1}^{N_\mathrm{A}} w_{im}^{\mathrm{GA}} A_m \right) \tag{1}$$

with time constant τ, firing rates $F_j^{\mathrm{G}} = \tanh\left(u_j^{\mathrm{G}}\right)$ of generator neurons, input signals I_k, readout signals R_l and, if present, specially-trained additional readouts A_m. The synaptic weights w_{ij}^{GG} within the generator network are drawn from a normal distribution with zero mean and variance $g_{\mathrm{GG}}^2/N_\mathrm{G}$. Similarly, the synaptic weight w_{il}^{GR} from the readout neuron l back to the generator neuron i is drawn from a normal distribution with zero mean and variance $g_{\mathrm{GR}}^2/N_\mathrm{R}$ and the weight w_{im}^{AR} from specially-trained neuron m is drawn from a normal distribution with zero mean and variance $g_{\mathrm{GA}}^2/N_\mathrm{A}$. Every generator neuron i receives signals from exactly one randomly chosen input signal k scaled by a weight w_{ik}^{GI} drawn from a normal distribution with variance g_{GI}^2 and zero mean.

The current activity value R_l of the linear readout neuron $l = 1, \ldots, N_\mathrm{R}$ is given by

$$R_l = \sum_{i=1}^{N_\mathrm{G}} w_{li}^{\mathrm{RG}} F_i^{\mathrm{G}}. \tag{2}$$

The initial values of the weights w_{li}^{RG} are drawn from a normal distribution with zero mean and variance N_G^{-1}. These weights are adapted by different supervised algorithms described below. If there are any specially-trained additional readout units in the network, they follow identical dynamics as the default readout neurons.

If not stated otherwise, the used parameter values are $\tau = 10\,\mathrm{ms}$, $N_G = 100$, $N_I = 1$, $N_R = 1$, $g_{\mathrm{GG}} = 1.0$, $g_{\mathrm{GI}} = 1.0$. All equations are solved by using the Euler method with a time step of $\hat{t} = 1\,\mathrm{ms}$.

Echo State Network Approach. Following the echo state network (ESN) approach[25] to train the weights from the reservoir network to the readouts w_{li}^{RG}, the network is sampled for a given number S of time steps. The activities of the generator neurons are collected in a state matrix M of dimension $N_\mathrm{G} \times S$ with every row containing the activities at a specific time step. The corresponding target signals of the readout neurons are collected in a teacher matrix T of dimension $S \times N_\mathrm{R}$. Optimizing the mean squared error of the readout signals is achieved by calculating the pseudoinverse M^{-1} of M and setting the weight matrix accordingly:

$$W^{\mathrm{RG}} = M^{-1}\, T. \tag{3}$$

Note that during the sampling phase, instead of the actual activities of the readout neurons the values of the target signals modified by Gaussian noise with variance $\sigma_{\mathrm{noise}}^2$ are fed back to the generator network. We use $\sigma_{\mathrm{noise}} = 0.1$.

FORCE Approach. In contrast to the ESN approach, FORCE learning[33] is an online-learning method. As originally proposed, we utilize the recursive least-squares (RLS) algorithm[63] to adapt the readout weights fast enough to keep the actual activities of the readout neurons close to the target values from the very beginning. During learning, in every simulation step, the readout weight vector for readout neuron l at time t is adapted according to

$$w_l^{\mathrm{RG}}(t) = w_l^{\mathrm{RG}}(t - \hat{t}) - e_l(t)(P(t)F^{\mathrm{G}}(t))^T. \tag{4}$$

Here, $\hat{t}$ denotes the step width of the simulation and

$$e_l(t) = (w_l^{\mathrm{RG}}(t - \hat{t}))^T F^{\mathrm{G}}(t) - f_l(t) \tag{5}$$

is the difference between the readout signal and the target function $f_l(t)$. The matrix $P(t)$ has the dimension $N_\mathrm{G} \times N_\mathrm{G}$ and is updated according to

$$P(t) = P(t - \Delta t) - \frac{P(t - \hat{t})F^{\mathrm{G}}(t)F^{\mathrm{G}}(t)^T P(t - \hat{t})}{1 + F^{\mathrm{G}}(t)^T P(t - \hat{t})F^{\mathrm{G}}(t)} \tag{6}$$

with an initial value of $P(0) = \alpha^{-1}\mathbb{1}$ where $\mathbb{1}$ is the identity matrix. We set $\alpha = 100$.

Benchmark Task. For the benchmark task (Fig. 1), input pulses are generated at random time intervals drawn from a normal distribution with mean $\mu_{\Delta t}$ and variance $\sigma_{\Delta t}$. Every pulse is modeled as a convolution of a constant signal with length t_{pulse}, unit magnitude and random sign and a Gaussian window with variance σ^2_{smooth}. To avoid overlaps between pulses, we restrict the time interval between two pulses to a minimum of $2 \cdot t_{\text{pulse}}$. The target readout signal consists of pulses of identical shape whose signs depend on the sign of the respective second-last input pulse. To account for the delays within the generator network, each target pulse is generated with a short delay t_{delay} after the corresponding input pulse.

In the case with specially-trained neurons (or additional readout units) (Fig. 3), we add two further target signals. These signals have a value of either +1 or −1 with the sign depending on the sign of the last or second-last input pulse, respectively. Also here, the pulses start with a short delay t_{delay} with respect to the input pulses and they are smoothed by a convolution with a Gaussian window with variance σ^2_{smooth}.

To evaluate the performance of the network in generating the desired target signal $f_l(t)$ at the read out neuron l, we compare the actual activity $R_l(t)$ with the desired target signal $f_l(t)$ and calculate the root mean square error (RMS). In order to normalize this value, it is divided by the RMS of the target signal:

$$E_l = \frac{\sqrt{\sum_t (R_l(t) - f_l(t))^2}}{\sqrt{\sum_t f_l(t)^2}}, \quad l = 1, \ldots, N_{\text{R}}. \tag{7}$$

For the benchmark task, there is only one regular readout unit ($l = 0$). We therefore omit the index such that $E = E_0$. Every parameter configuration is evaluated in 1000 independent network instantiations.

When adding the recall stimulus in the benchmark task (Fig. 6), it is applied 10 ms before the onset of the target pulse via an additional input neuron ($N_I = 2$) and given by a short positive unit pulse of length t_{pulse} smoothed by a convolution with a Gaussian window with variance σ^2_{smooth}.

The used values are $\mu_{\Delta t} = 200$ ms, $t_{\text{pulse}} = 10$ ms, $t_{\text{delay}} = 10$ ms, and $\sigma_{\text{smooth}} = 2$ ms.

Hopfield Network. We use a standard implementation of the Hopfield network[43]. It consists of $N_{\text{Hopf}} = 1000$ binary neurons. In every time step, every neuron i is in one of two possible states characterized by a firing rate of either $F_i = 1$ or $F_i = -1$. The activity states of all neurons are updated according to

$$F_i(t+1) = \begin{cases} 1, & \text{if} \sum_{j=1}^{N_{\text{Hopf}}} w_{ij}^{\text{Hopf}} F_j(t) \geq 0 \\ -1, & \text{if} \sum_{j=1}^{N_{\text{Hopf}}} w_{ij}^{\text{Hopf}} F_j(t) < 0, \end{cases} \tag{8}$$

where w_{ij}^{Hopf} is the synaptic weight projecting from neuron j to neuron i (w.l.o.g., $w_{ii}^{\text{Hopf}} = 0$). We store $K = 100$ random patterns $\hat{F}_j$ into the network by choosing

$$w_{ij}^{\text{Hopf}} = \frac{1}{K} \sum_{\mu}^{K} \hat{F}_i^{\mu} \hat{F}_j^{\mu}. \tag{9}$$

The network is initialized with a binary vector F^0 whose entries differ in d positions from $\hat{F}^1$ corresponding to an overlap of $1 - d/N_{\text{Hopf}}$. We simulate the development of the network until a stable pattern is reached. Hence, the number of simulated time steps corresponds to the convergence time. In a few cases, the system does not a reach a stable pattern. The respective trials are excluded from the analysis. For every distance d with $0 \leq d \leq N$, we repeated the simulation 10,000 times.

Self-Organized Cell Assemblies. The here-used LTM-model is based on Hebbian cell assembly models[28]. We introduce some modifications to allow for a self-organized allocation of memory representations. The main part of the model is an $n \times n$-grid of excitatory neurons, each described by a membrane potential u_i, $i = 1, \ldots, N_{\text{CA}}$ ($N_{\text{CA}} = n^2$), according to

$$\tau \frac{\mathrm{d}u_i}{\mathrm{d}t} = -u_i + \sum_{j=1}^{N_{\text{CA}}} w_{ij}^{\text{lat}} F_j + w_{\text{exc,inh}} F^{\text{inh}} + \sum_{k=1}^{N_{\text{aff}}} w_{ik}^{\text{aff}} F_k^{\text{aff}} \tag{10}$$

with time constant τ (the same value as for the time constant of the reservoir neurons). F_i represents the firing rate of neuron i given by

$$F_i = \frac{1}{1 + \exp(\beta(\varepsilon - u_i))} \tag{11}$$

with parameters β and ε. Every neuron i on the grid receives lateral inputs with synaptic weights w_{ij}^{lat} from all grid neurons j whose euclidean distance from neuron i (measured in grid units) is smaller than the excitatory interaction radius r_{exc} (otherwise $w_{ij}^{\text{lat}} = 0$). In order to avoid boundary effects, periodic boundary conditions are applied.

In addition to the excitatory neurons, we consider an inhibitory population which receives signals from all excitatory grid neurons and projects signals back to all of them. The mean-field dynamics of the membrane potential u^{inh} and the firing rate F^{inh} of this population is given by

$$\tau_{\mathrm{inh}}\frac{\mathrm{d}u^{\mathrm{inh}}}{\mathrm{d}t} = -u^{\mathrm{inh}} + \sum_{i=1}^{N_{\mathrm{CA}}} w_{\mathrm{inh,exc}}F_i \tag{12}$$

and

$$F^{\mathrm{inh}} = \frac{1}{1 + \exp(\beta_{\mathrm{inh}}(\varepsilon_{\mathrm{inh}} - u^{\mathrm{inh}}))}. \tag{13}$$

Here, $w_{\mathrm{inh,exc}}$ and $w_{\mathrm{exc,inh}}$ designate the average synaptic weight from excitatory neurons to the inhibitory population and vice versa.

A second layer of N_{aff} afferent neurons projects signals onto the grid layer via afferent synapses with weights w_{ik}^{aff}. The firing rates F_k^{aff} of these neurons are externally controlled and represent the input pattern presented to the network. Every grid neuron i receives synapses from exactly n_{aff} randomly chosen input neurons.

Nonzero lateral synapses as well as the nonzero afferent synapses are governed by a synaptic plasticity rule composed of a Hebbian term, correlating the postsynaptic activity F_i and the presynaptic activity F_j, and a synaptic scaling term, driving the postsynaptic activity towards a target value F_T[28, 64]:

$$\tau_w\frac{\mathrm{d}w_{ij}}{\mathrm{d}t} = F_iF_j + (F_T - F_i)w_{ij}^2. \tag{14}$$

To obtain the memory convergence times for this model (Fig. 7d), we consider several phases: In the first phase, the learning phase, we fully activate half of the afferent neurons (representing an input pattern) and simulate the network long enough to obtain a stable cell assembly (memory representation) in the grid layer. All neurons with a firing rate F_i larger than 0.5 are considered to be part of the assembly. In the next phase, we deactivate the input to assure that the resulting network structures do not imply persistent activities. In the recall phase, a similar but slightly different input pattern where $d/2$ previously active afferent neurons are now inactive and $d/2$ originally inactive afferent neurons are now active is presented to the network. This corresponds to a total pattern difference of d and therefore to an overlap of $1 - d/N_{\mathrm{aff}}$. Applying the modified afferent pattern, we measure the time the network requires until 90% of the neurons which have been already active during learning (representing the stored pattern) get active. The respective time interval is considered as the convergence time of the stored memory. For every difference d with $0 < d < N/2$ this procedure is repeated for 1000 times. Trials during which 90% activation is not reached are not considered for further analyses.

The used parameters values are $n = 30$ ($N_{\mathrm{CA}} = 900$), $\tau = 0.01$ s, $w_{\mathrm{exc,inh}} = -20.0$, $\beta = 1$, $\varepsilon = 12$, $r_{\mathrm{exc}} = 3$, $\tau_{\mathrm{inh}} = 0.02$ s, $w_{\mathrm{inh,exc}} = 1.0$, $\beta_{\mathrm{inh}} = 0.1$, $\varepsilon_{\mathrm{inh}} = 100$, $N_{\mathrm{aff}} = 50$, $n_{\mathrm{aff}} = 20$, $\tau_w = 10$ s, and $F_T = 0$.

WM-LTM interaction. For the interaction of WM with LTM, as sketched in Fig. 8, we trained a reservoir network ($N_G = 500$) with $N_I = 4$ input signals ("0", "1", "2", "+") and $N_R = 9$ readouts. The first three of these readouts represent the actual output of the WM ("0", "1", "2"). The second group of three holds additional assisting outputs which encode the last received number input. Accordingly, the third group encodes the second-last received number input. Already during the reservoir training, the actual readouts of the reservoir are connected to the LTM-symbol area. For simplicity, this area is modeled as a collection of single excitatory neurons with dynamics and parameters as described above. Every number is represented by a group of ten neurons, the two context signals by twenty neurons each. The reservoir output signals are fed into the respective symbol neurons with strong synaptic weights $w^{\mathrm{SR}} = 20$. The activities of the symbol neurons, in turn, are directly taken as input signals for the reservoir.

The reservoir network is trained using the FORCE approach as described above. All inputs are smoothed by convolution with a Gaussian kernel with a standard deviation of $\sigma_{\mathrm{smooth}} = 5$ ms. The pulses representing numbers have a length drawn from a uniform distribution with a minimum of $t_{\mathrm{num,min}} = 50$ ms and a maximum of $t_{\mathrm{num,max}} = 100$ ms. The pulses representing the "+"-operation have a length uniformly drawn between $t_{+,\mathrm{min}} = 100$ ms and $t_{+,\mathrm{max}} = 400$ ms. The distances between the individual pulses vary between $\Delta t_{\mathrm{min}} = 100$ ms and $\Delta t_{\mathrm{max}} = 400$ ms. The reservoir network is trained for 100000 time steps and, afterwards, its performance is evaluated for the same number of time steps. This procedure is repeated until the normalized error during evaluation drops below 0.05.

After training the reservoir network, the LTM-symbol area is taken as an input to the LTM-association area which is modeled by the system of self-organized cell assemblies described above. Apart from the higher number of afferent synapses ($n_{\mathrm{aff}} = 30$) per excitatory grid neuron, we use the same parameters as before. However, in order to be able to recall actual symbols from the cell assembly activity, additional feedback synapses from the LTM-association area back to the symbol area are introduced. These are governed by the same synaptic plasticity processes as the lateral and afferent synapses. Every neuron in the symbol area receives synapses from $n_{\mathrm{fb}} = 250$ randomly chosen LTM-association neurons. The feedback synapses are initialized with a weight which equals zero.

References

1. Unsworth, N. On the division of working memory and long-term memory and their relation to intelligence: A latent variable approach. *Acta Psychol. (Amst.)* **134**, 16–28, doi:10.1016/j.actpsy.2009.11.010 (2010).
2. Baddeley, A. D. Working memory: theories, models, and controversies. *Annu. Rev. Psychol.* **63**, 1–29, doi:10.1146/annurev-psych-120710-100422 (2012).
3. Durstewitz, D., Seamans, J. K. & Sejnowski, T. J. Neurocomputational models of working memory. *Nat. Neurosci.* **3**(Suppl), 1184–1191, doi:10.1038/81460 (2000).

4. Rabinovich, M. I., Huerta, R. & Laurent, G. Transient dynamics for neural processing. *Science* **321**, 48–50, doi:10.1126/science.1155564 (2008).
5. Barak, O. & Tsodyks, M. Working models of working memory. *Curr. Opin. Neurobiol.* **25**, 20–24, doi:10.1016/j.conb.2013.10.008 (2014).
6. Fuster, J. M. & Alexander, G. E. Neuron activity related to short-term memory. *Science* **173**, 652–654, doi:10.1126/science.173.3997.652 (1971).
7. Curtis, C. E. & D'Esposito, M. Persistent activity in the prefrontal cortex during working memory. *Trends Cogn. Sci.* **7**, 415–423, doi:10.1016/S1364-6613(03)00197-9 (2003).
8. Deco, G. & Rolls, E. T. Attention and working memory: A dynamical model of neuronal activity in the prefrontal cortex. *Eur. J. Neurosci.* **18**, 2374–2390, doi:10.1046/j.1460-9568.2003.02956.x (2003).
9. Wimmer, K., Nykamp, D. Q., Constantinidis, C. & Compte, A. Bump attractor dynamics in prefrontal cortex explains behavioral precision in spatial working memory. *Nat. Neurosci.* **17**, 431–439, doi:10.1038/nn.3645 (2014).
10. Riley, M. R. & Constantinidis, C. Role of prefrontal persistent activity in working memory. *Front. Syst. Neurosci.* **9**, 181, doi:10.3389/fnsys.2015.00181 (2016).
11. Rainer, G. & Miller, E. K. Timecourse of object-related neural activity in the primate prefrontal cortex during a short-term memory task. *Eur. J. Neurosci.* **15**, 1244–1254, doi:10.1046/j.1460-9568.2002.01958.x (2002).
12. Mazor, O. & Laurent, G. Transient dynamics versus fixed points in odor representations by locust antennal lobe projection neurons. *Neuron* **48**, 661–673, doi:10.1016/j.neuron.2005.09.032 (2005).
13. Jaeger, H. & Eck, D. Can't get you out of my head: A connectionist model of cyclic rehearsal. In Wachsmutch, I. & Knoblich, G. (eds.) *Modeling Communication with Robots and Virtual Humans: Second ZiF Research Group International Workshop on Embodied Communication in Humans and Machines, Bielefeld, Germany, April 5–8, 2006, Revised Selected Papers* 310–335 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2008).
14. Pascanu, R. & Jaeger, H. A neurodynamical model for working memory. *Neural Netw.* **24**, 199–207, doi:10.1016/j.neunet.2010.10.003 (2011).
15. Buonomano, D. V. & Maass, W. State-dependent computations: spatiotemporal processing in cortical networks. *Nat. Rev. Neurosci.* **10**, 113–125, doi:10.1038/nrn2558 (2009).
16. Maass, W. Liquid state machines: Motivation, theory, and applications. In Cooper, S. B. & Sorbi, A. (eds.) *Computability in Context* 275–296 (Imperial College Press, London, 2011).
17. Tetzlaff, C., Dasgupta, S., Kulvicius, T. & Wörgötter, F. The use of hebbian cell assemblies for nonlinear computation. *Sci. Rep.* **5**, 12866, doi:10.1038/srep12866 (2015).
18. Jun, J. K. *et al.* Heterogenous population coding of a short-term memory and decision task. *J. Neurosci.* **30**, 916–929, doi:10.1523/JNEUROSCI.2062-09.2010 (2010).
19. Hussar, C. R. & Pasternak, T. Memory-guided sensory comparisons in the prefrontal cortex: contribution of putative pyramidal cells and interneurons. *J. Neurosci.* **32**, 2747–2761, doi:10.1523/JNEUROSCI.5135-11.2012 (2012).
20. Zhang, X., Yi, H., Bai, W. & Tian, X. Dynamic trajectory of multiple single-unit activity during working memory task in rats. *Front. Comput. Neurosci.* **9**, 117, doi:10.3389/fncom.2015.00117 (2015).
21. Barak, O., Sussillo, D., Romo, R., Tsodyks, M. & Abbott, L. F. From fixed points to chaos: three models of delayed discrimination. *Prog. Neurobiol.* **103**, 214–222, doi:10.1016/j.pneurobio.2013.02.002 (2013).
22. Koppe, G. *et al.* Temporal unpredictability of a stimulus sequence affects brain activation differently depending on cognitive task demands. *Neuroimage* **101**, 236–244, doi:10.1016/j.neuroimage.2014.07.008 (2014).
23. Baddeley, A. D. Working memory: looking back and looking forward. *Nat. Rev. Neurosci.* **4**, 829–839, doi:10.1038/nrn1201 (2003).
24. Owen, A. M., McMillan, K. M., Laird, A. R. & Bullmore, E. N-back working memory paradigm: A meta-analysis of normative functional neuroimaging studies. *Hum. Brain Mapp.* **25**, 46–59, doi:10.1002/hbm.20131 (2005).
25. Jaeger, H. The "echo state" approach to analysing and training recurrent neural networks. Tech. Rep., GMD - German National Research Institute for Computer Science (2001).
26. Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531–2560, doi:10.1162/089976602760407955 (2002).
27. Hopfield, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. USA* **81**, 3088–3092, doi:10.1073/pnas.81.10.3088 (1984).
28. Tetzlaff, C., Kolodziejski, C., Timme, M., Tsodyks, M. & Wörgötter, F. Synaptic scaling enables dynamically distinct short- and long-term memory formation. *PLoS Comput. Biol.* **9**, e1003307, doi:10.1186/1471-2202-14-S1-P415 (2013).
29. Baddeley, A. D., Papagno, C. & Vallar, G. When long-term learning depends on short-term storage. *J Mem. Lang.* **27**, 586–595, doi:10.1016/0749-596X(88)90028-9 (1988).
30. Hulme, C., Maughan, S. & Brown, G. D. A. Memory for familiar and unfamiliar words: Evidence for a long-term memory contribution to short-term memory span. *J Mem. Lang.* **30**, 685–701, doi:10.1016/0749-596X(91)90032-F (1991).
31. Poirier, M., Dhir, P., Saint-Aubin, J., Tehan, G. & Hampton, J. The influence of semantic memory on verbal short-term memory. In *European Perspectives on Cognitive Science* (New Bulgarian University Press, Sofia, 2011).
32. Marton, K. & Eichorn, N. Interaction between working memory and long-term memory. *Z. Psychol* **222**, 90–99, doi:10.1027/2151-2604/a000170 (2014).
33. Sussillo, D. & Abbott, L. F. Generating coherent patterns of activity from chaotic neural networks. *Neuron* **63**, 544–557, doi:10.1016/j.neuron.2009.07.018 (2009).
34. Maass, W., Joshi, P. & Sontag, E. D. Computational aspects of feedback in neural circuits. *PLoS Comput. Biol.* **3**, e165, doi:10.1371/journal.pcbi.0020165 (2007).
35. Gros, C. Cognitive computation with autonomously active neural networks: An emerging field. *Cognit. Comput.* **1**, 77–90, doi:10.1007/s12559-008-9000-9 (2009).
36. Sompolinsky, H. & Crisanti, A. Chaos in random neural networks. *Phys. Rev. Lett.* **61**, 259–262, doi:10.1103/PhysRevLett.61.259 (1988).
37. Enel, P., Procyk, E., Quilodran, R. & Dominey, P. F. Reservoir computing properties of neural dynamics in prefrontal cortex. *PLoS Comput. Biol.* **12**, e1004967, doi:10.1371/journal.pcbi.1004967 (2016).
38. Sussillo, D. & Barak, O. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Comput.* **25**, 626–49, doi:10.1162/NECO_a_00409 (2013).
39. Miyashita, Y. Neuronal correlate of visual associative long-term memory in the primate temporal cortex. *Nature* **335**, 817–820, doi:10.1038/335817a0 (1988).
40. Amit, D. J., Brunel, N. & Tsodyks, M. Correlations of cortical hebbian reverberations: theory versus experiment. *J. Neurosci.* **14**, 6435–6445 (1994).
41. Wood, R., Baxter, P. & Belpaeme, T. A review of long-term memory in natural and synthetic systems. *Adapt. Behav.* **20**, 81–103, doi:10.1177/1059712311421219 (2011).
42. Brunel, N. Is cortical connectivity optimized for storing information? *Nat. Neurosci.* **19**, 749–755, doi:10.1038/nn.4286 (2016).
43. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **79**, 2554–2558, doi:10.1073/pnas.79.8.2554 (1982).

44. Juola, J. F., Fischler, I., Wood, C. T. & Atkinson, R. C. Recognition time for information stored in long-term memory. *Percept. Psychophys.* **10**, 8–14, doi:10.3758/BF03205757 (1971).
45. Kohring, G. A. Convergence time and finite size effects in neural networks. *J. Phys. A Math. Gen.* **23**, 2237–2241, doi:10.1088/0305-4470/23/11/047 (1990).
46. Lukoševičius, M., Jaeger, H. & Schrauwen, B. Reservoir computing trends. *Künstl. Intell.* **26**, 365–371, doi:10.1007/s13218-012-0204-5 (2012).
47. Maass, W. & Sontag, E. D. Analog neural nets with gaussian or other common noise distributions cannot recognize arbitrary regular languages. *Neural Comput.* **11**, 771–782, doi:10.1162/089976699300016656 (1999).
48. Rabinovich, M. I., Varona, P., Selverston, A. I. & Abarbanel, H. D. I. Dynamical principles in neuroscience. *Rev. Mod. Phys.* **78**, 1213–1265, doi:10.1103/RevModPhys.78.1213 (2006).
49. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comp. Sci. Rev.* **3**, 127–149, doi:10.1016/j.cosrev.2009.03.005 (2009).
50. Laje, R. & Buonomano, D. V. Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nat. Neurosci.* **16**, 925–933, doi:10.1038/nn.3405 (2013).
51. Sussillo, D. & Abbott, L. F. Transferring learning from external to internal weights in echo-state networks with sparse connectivity. *PLoS One* **7**, e37372, doi:10.1371/journal.pone.0037372 (2012).
52. Bick, C. & Rabinovich, M. I. Dynamical origin of the effective storage capacity in the brain's working memory. *Phys. Rev. Lett.* **103**, 218101, doi:10.1103/PhysRevLett.103.218101 (2009).
53. Tetzlaff, C., Kolodziejski, C., Markelic, I. & Wörgötter, F. Time scales of memory, learning, and plasticity. *Biol. Cybern.* **106**, 715–726, doi:10.1007/s00422-012-0529-z (2012).
54. Mongillo, G., Barak, O. & Tsodyks, M. Synaptic theory of working memory. *Science* **319**, 1543–1546, doi:10.1126/science.1150769 (2008).
55. Rose, N. S. *et al*. Reactivation of latent working memories with transcranial magnetic stimulation. *Science* **354**, 1136–1139, doi:10.1126/science.aah7011 (2016).
56. Stokes, M. G. *et al*. Dynamic coding for cognitive control in prefrontal cortex. *Neuron* **78**, 364–375, doi:10.1016/j.neuron.2013.01.039 (2013).
57. Courtney, S. N., Ungerleider, L. G., Keil, K. & Haxby, J. V. Transient and sustained activity in a distributed neural system for human working memory. *Nature* **386**, 608–611, doi:10.1038/386608a0 (1997).
58. Donkin, C., Nosofsky, R. M., Gold, J. M. & Shiffrin, R. M. Discrete-slots models of visual working-memory response times. *Psychol. Rev.* **120**, 873–902, doi:10.1037/a0034247.Discrete-Slots (2013).
59. Drover, J. D. Timing over tuning: Overcoming the shortcomings of a line attractor during a working memory task. *PLoS Comput. Biol.* **10**, e1003437, doi:10.1371/journal.pcbi.1003437 (2014).
60. Eriksson, J., Vogel, E. K., Lansner, A., Bergström, F. & Nyberg, L. Neurocognitive architecture of working memory. *Neuron* **88**, 33–46, doi:10.1016/j.neuron.2015.09.020 (2015).
61. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780, doi:10.1162/neco.1997.9.8.1735 (1997).
62. Burgess, N. & Hitch, G. Computational models of working memory: Putting long-term memory into context. *Trends Cogn. Sci.* **9**, 535–541, doi:10.1016/j.tics.2005.09.011 (2005).
63. Haykin, S. S. *Adaptive Filter Theory* 5th edn. (Prentice Hall, Upper Saddle River, NJ, 2002).
64. Tetzlaff, C., Kolodziejski, C., Timme, M. & Wörgötter, F. Synaptic scaling in combination with many generic plasticity mechanisms stabilizes circuit connectivity. *Front. Comput. Neurosci.* **5**, 47, doi:10.3389/fncom.2011.00047 (2011).

Acknowledgements

We thank Prof. Dr. Florentin Wörgötter and Dr. Michael Fauth for fruitful discussions and critics on the manuscript. The research leading to these results has received funding from the Federal Ministry of Education and Research (BMBF) Germany to the Göttingen Bernstein Center for Computational Neuroscience under grant numbers 01GQ1005A [TN] and 01GQ1005B [CT] and from the International Max Planck Research School for Physics of Biological and Complex Systems (IMPRS-PBCS) by stipends of the country of Lower Saxony with funds from the initiative "Niedersächsisches Vorab" and of the University of Göttingen [TN]. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. We acknowledge support by the German Research Foundation and the Open Access Publication Funds of the Göttingen University.

Author Contributions

T.N. and C.T. planned the presented analyses and experiments. Implementation and analysis of the data was done by T.N. T.N. and C.T. wrote the manuscript.

Additional Information

Supplementary information accompanies this paper at doi:10.1038/s41598-017-02471-z

Competing Interests: The authors declare no competing interests.

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