SUPPLEMENTAL INFORMATION

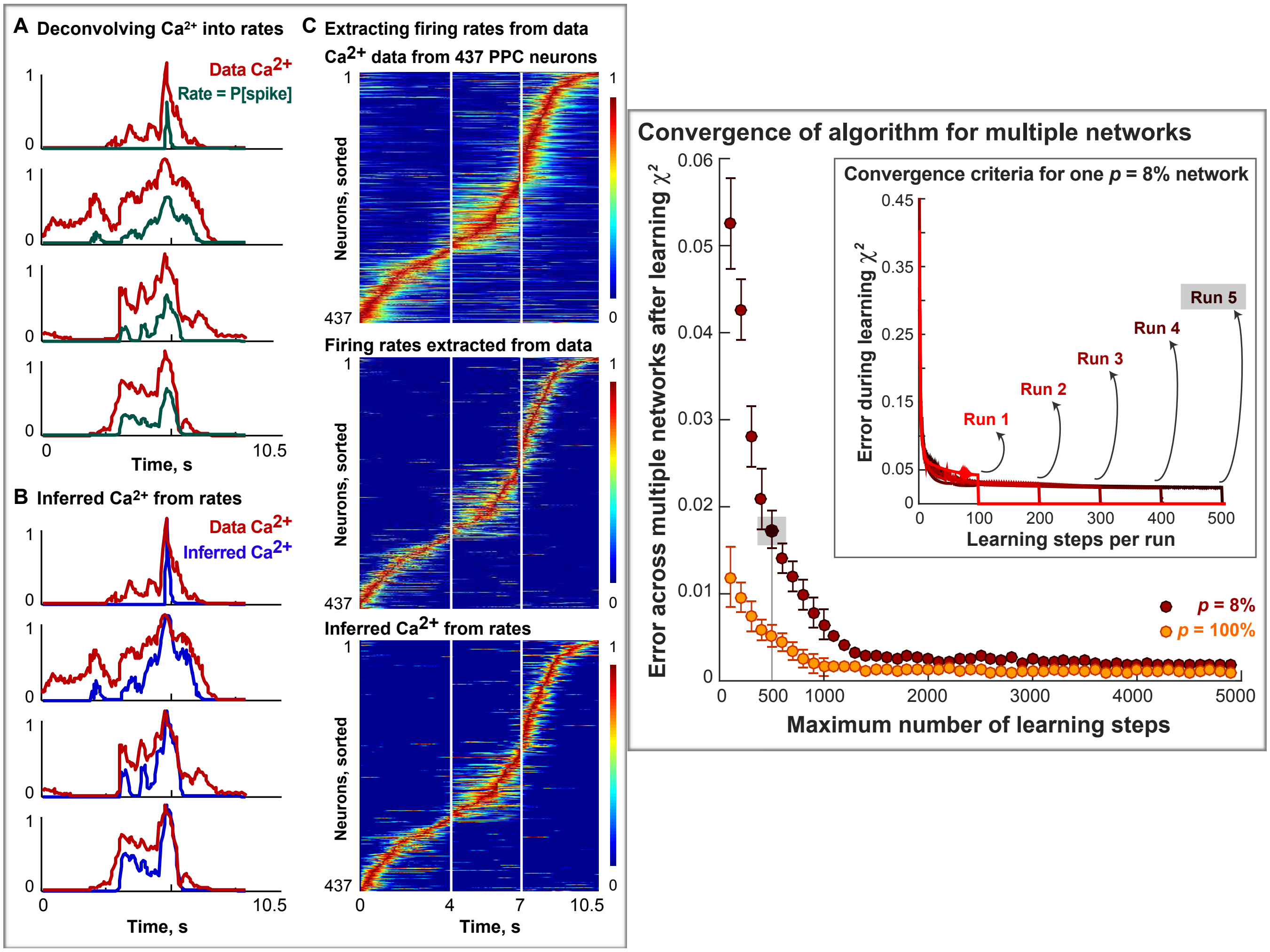

Figure S1 (related to Figure 1): Left Panel, Extracting Target Functions from Data by Deconvolution and Right Panel, Convergence of the PINning Algorithm

A. The use of deconvolution algorithms for the extraction of firing rates is illustrated here. A few example PPC neurons showing the Ca^{2+} fluorescence signals in red (replotted from [Harvey, Coen & Tank, 2012]) and their extracted firing rates (normalized to a maximum of 1) in green. These rates were obtained using both methods described in Experimental Procedures 3. The implementation of the Bayesian algorithm (method #2 from Experimental Procedures 3) yields spike times along with the statistical confidence of a spike arriving at that particular time point (P[spike]). However, since the frame rate of the imaging experiments in [Harvey, Coen & Tank, 2012] was relatively slow (64ms per frame), we used these “probabilistic” spike trains as a normalized firing rate estimate. To verify, we first deconvolved single trial Ca^{2+} data, and then performed an average over all the single trial P[spike] estimates to obtain the smooth firing rates for each of the 437 sequential neurons that we show in the middle panel of (C).

B. To see how accurate the firing rate estimates extracted from the data were, we re-convolved the extracted firing rates of the example units in (A) through a difference of exponentials with a rise time of 52ms and decay time of 384ms. The PPC Ca^{2+} data is plotted in red and the inferred Ca^{2+} is plotted in blue here.

C. Top panel shows the original Ca^{2+} fluorescence signals from the PPC (similar to Figure 2A of main text, adapted from Figure 2c in [Harvey, Coen & Tank, 2012]). Middle panel shows the firing rates (from 0 to a maximum of 6Hz, normalized by the peak to 1) extracted from these data by using deconvolution methods. Bottom panel shows the inferred Ca^{2+} signals obtained by convolving these extracted firing rates by a Ca^{2+} impulse response function as explained for (B) (see also Experimental Procedures 3).

D. Right Panel shows the convergence of the PINning algorithm. The convergence of our PINning algorithm (Experimental Procedures 4) is shown here for multiple sequential networks generating sequential outputs (similar to Supplemental Figure 4). We plot the χ^2 error between the target functions and the outputs of several PINned networks as a function of the number of learning steps for two values of $p - p = 8\%$ plastic synapses (mean values in the red circles, means computed over 5 instantiations each) and $p = 100\%$ plastic synapses (mean values in the yellow circles, means computed over 5 instantiations each) for comparison purposes. When the χ^2 error drops below 0.02, which for both sparsely and fully PINned networks occurs before the 500th learning step, we terminate the learning and simulate the network with the PINned connectivity matrix (denoted as $\boldsymbol{J_{PINned,\,p\%}}$ in general) for an additional 50 steps before the program graphs the network outputs (firing rates, inferred calcium to compare with data, statistics of $\boldsymbol{J_{PINned,\,p\%,}}$ etc.). The point highlighted in the gray square corresponds to a PINned network with $p = 8\%$ that ran for 500 learning steps, at the end of which, the χ^2 error was 0.018. Additionally, this network had a *pVar* (Experimental Procedures 7) of 92%. **Inset** shows the speed of convergence for runs of different lengths for the $p = 8\%$ PINned network. Run #5 corresponds to the example network highlighted in the gray square.

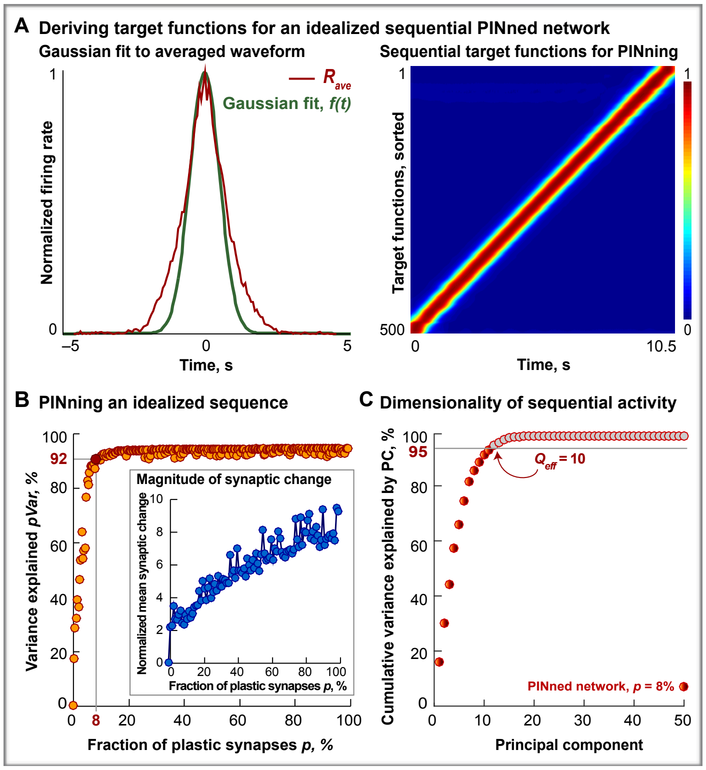

Figure S2 (related to Figures 3 and 6): Idealized Sequence-Generating PINned Network

A. Left panel shows a Gaussian with mean = 0 and variance = 0.3, denoted by *f(t)* (green trace), that best fits the neuron-averaged waveform (red trace, R_{ave}, identical to Figure 1D). This waveform *f(t)* is used to generate the target functions (right panel) for a network of 500 rate-based model neurons PINned to produce an idealized sequence of population activity (*bVar* = 100%).

B. Effect of increasing the PINning fraction, *p*, in a network producing the single idealized sequence is shown here. *pVar* (Experimental Procedures 7) plotted as a function of *p,* increases from 0 for a random unmodified network (*p* = 0) network and plateaus at *pVar* = ~92% for and above *p* = 8%. The sequence-facilitating properties of the connectivity matrix in the *p* = 8% network highlighted in red are analyzed in Figure 4 of the main text. **Inset** shows the magnitude of synaptic change required to generate an idealized sequence as a function of *p*, computed as in Figure 2E (Experimental Procedures 8). The overall magnitude grows from a factor of ~3 for sparsely PINned networks (*p* = 8%) to between 8 and 9 for fully PINned networks (*p* = 100%) producing a single idealized sequence that match the targets in (A). As explained in the main text, although the individual synapses change more in sparsely PINned (small *p*) networks, the total amount of change across the synaptic connectivity matrix is smaller.

C. Dimensionality of sequential activity is computed (as in Figure 2F, Experimental Procedures 5) for the 500-neuron PINned network generating an idealized sequence (orange circles) with *p* = 8% and *pVar* = 92%. Of the 500 possible PCs that can capture the total variability in the activity of this 500-neuron network, 10 PCs account for over 95% of the variance when *p* = 8%, i.e., Q_{eff} = 10.

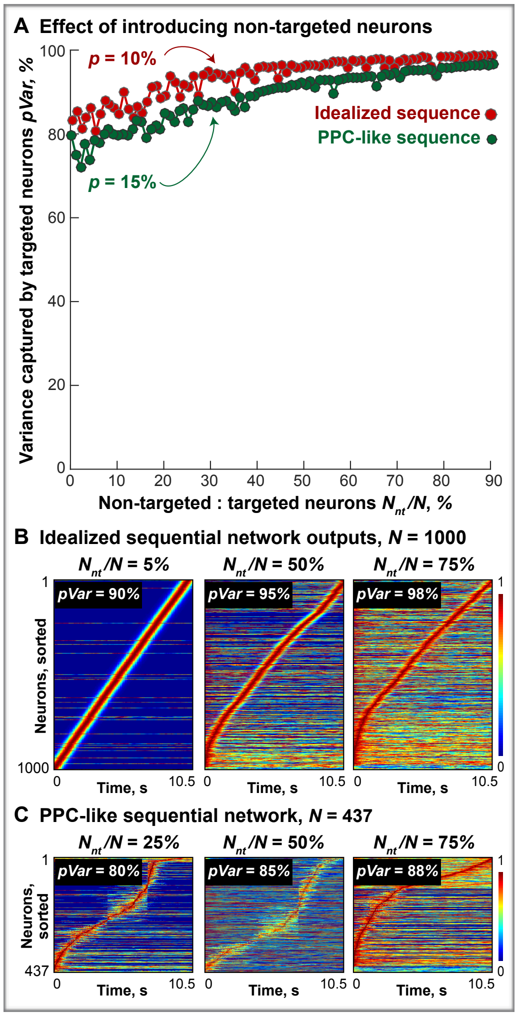

Figure S3 (related to Figures 1, 2 and 5): Simulating Unobserved Neurons By Including Non-Targeted Neurons In PINned Networks

A. Variance of the target functions captured by the outputs of the network neurons that have been PINned, *pVar* (evaluated as described in Experimental Procedures 7) is plotted here as a function of the ratio of non-targeted to targeted neurons in the network, denoted by N_{nt}/N. Overall, *pVar* does not decrease appreciably when the relative number of untrained neurons introduced into the PINned network is increased. Interestingly however, *pVar* improves slightly for sparsely PINned (p = 10% networks in the red circles for an idealized sequence-generating network and p = 15% networks in green for a PPC-like sequence) when untrained neurons are introduced. This improvement gets smaller as p increases (not shown). The inclusion of non-targeted neurons in the networks constructed by PINning simulates the effect of unobserved but active neurons that may exist in the experimental data and might influence neural activity. It should be noted, however, that these additional neurons, however, might add irregularity to the sorted outputs of the full network (including both targeted and non-targeted neurons) and reduce the stereotypy of the overall outputs (indicated by a decrease in the *bVar* computed over the full network, not shown). This effect is independent of p.

B. Example network outputs are shown here for 3 values of N_{nt}/N. for a network generating an idealized sequence similar to Supplemental Figure 4. *pVar* = 90% for N_{nt}/N = 5%, 95% for N_{nt}/N = 50% and 98% for N_{nt}/N = 75%. The sequences become noisier overall as more randomly fluctuating untrained neurons are introduced, but the percent variance of the targets captured by the PINned neurons remains largely unaffected, even showing a slight improvement.

C. Same as panel (B), except for a 437-neuron network constructed by PINning to generate a PPC-like sequence similar to Figure 2C with different fractions of neurons left untrained. *pVar* = 80% for N_{nt}/N = 25%, 85% for N_{nt}/N = 50% and 88% for N_{nt}/N = 75%, confirming the same general trend as above.

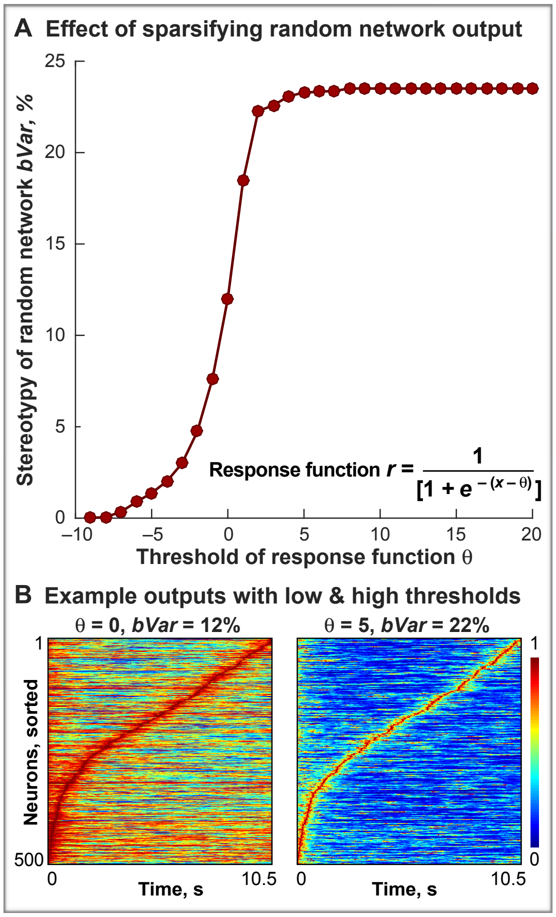

Figure S4 (related to Figure 1): Changing The Threshold Of The Response Function And Sparsifying The Random Network Output

A. Effect of sparsifying the outputs of random networks of model neurons constructed with different thresholds, denoted by θ, is shown here. Stereotypy of the sequences made by sorting the outputs of these random networks, *bVar*, increases from 0 to 12% for $\theta = 0$ networks used as initial configurations throughout the paper, and saturates at 22% for networks with threshold values of $\theta > 2$.

B. Example outputs from two random networks, one with $\theta = 0$ (left, with *bVar* = 12%, this is identical to Figure 1B) and one with $\theta = 5$ (right, with *bVar* = 22%) are shown here. Firing rates are normalized by the t_{COM} and sorted to yield the sequences shown here.

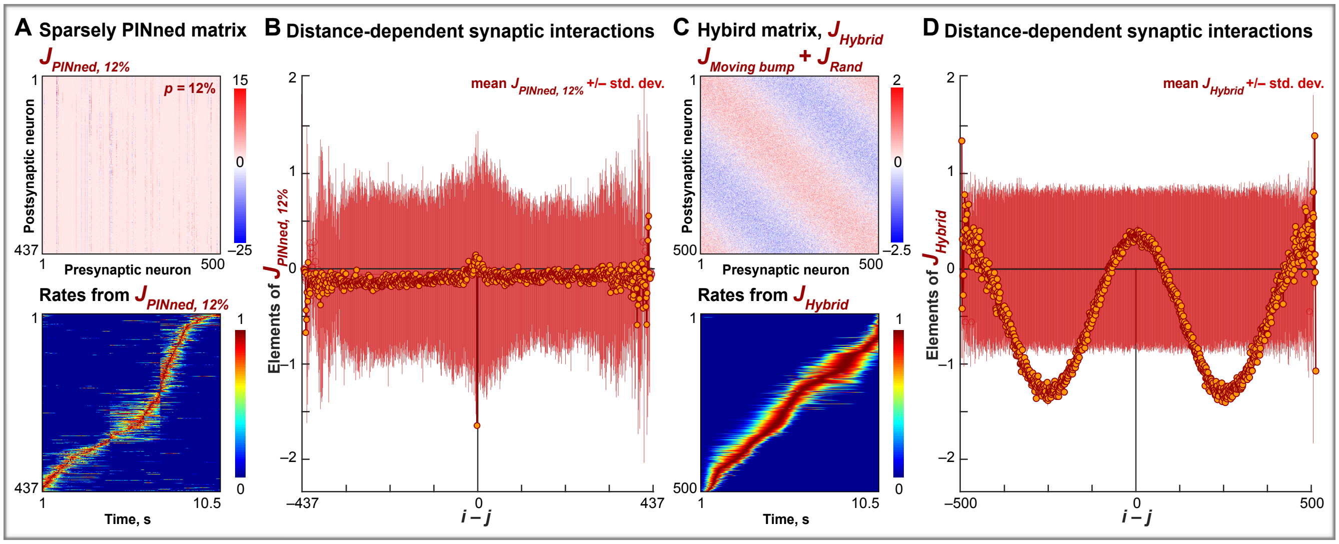

Figure S5 (related to Figure 3): Comparison of sparsely PINned matrix, $J_{PINned,\ 12\%}$ and an Additive Hybrid Connectivity Matrix of the Form, $J_{Hybrid} = J_{Moving\ bump} + J_{Rand}$

A. Synaptic connectivity matrix of a 437-neuron network with $p = 12\%$ that produces a PPC-like sequence with $pVar = 85\%$, denoted by $J_{PINned,\ 12\%}$, and the firing rates obtained (identical to Figure 2C, also highlighted by the red circle in Figure 2D) are shown here.

B. Influence of neurons away from the sequentially active neurons is estimated by computing the mean (circles) and the standard deviation (lines) of the elements of J_{Rand} (in blue) and $J_{PINned,\ 8\%}$ (in orange) in successive off-diagonal "stripes" away from the principal diagonal (as described in Experimental Procedures 11 and analogous to Figure 3 in the main text). These quantities are plotted as a function of the "inter-neuron distance", $i - j$. In units of $i - j$, 0 corresponds to the principal diagonal or self-interactions, and the positive and the negative terms are the successive interaction magnitudes of neurons a distance $i - j$ away from the primary sequential neurons.

C. Same as panel (A), except for the synaptic connectivity matrix from an additive hybrid of the form, $J_{Hybrid} = J_{Moving\ bump} + J_{Rand}$ is shown here, where, $J_{Rand.}$ is a random matrix similar to the one shown in the lower panel of Figure 3A and $J_{Moving\ bump}$ is the connectivity for a moving bump model [Yishai, Bar-Or & Sompolinsky, 1995]. The hybrid matrix contains a structured and a random part, and is constructed by the addition of a moving bump connectivity matrix [Yishai, Bar-Or & Sompolinsky. 1995], $J_{\text{Moving bump}} = -J_0 + J_2[\cos(\phi_i - \phi_j)] + 0.06 \times [\sin(\phi_i - \phi_j)]$, $\phi = 0, \ldots, \pi$) and a random matrix, J_{Rand}, similar to the one used to initialize PINning (lower panel of Figure 3A). Mathematically, the hybrid is of the form,

$$J_{\text{Hybrid}} = \frac{1}{N}[A_B J_{\text{Moving bump}}] + \frac{1}{\sqrt{N}}[A_R J_{\text{Rand}}]$$

, where A_B is the relative amplitude of the structured or moving bump part, scaled by network size, N, and A_R is the relative amplitude of the random part of the N-neuron hybrid network, scaled by $\sqrt{N}$. The lower panel shows the firing rates from the additive hybrid network whose connectivity is given by J_{Hybrid}. $pVar$ for the output of this hybrid network is only 1%, however, its stereotypy, $bVar = 92\%$.

D. Same as (B), except for $J_{Hybrid.}$ Band-averages (orange circles) are bigger and more asymmetric compared to those for $J_{PINned,\ 12\%}$. Notably, these band-averages are positive for $i - j = 0$ and in the neighborhood of 0. Fluctuations around the band-averages (red lines) for J_{Hybrid} are less structured than those for $J_{PINned,\ 12\%}$ and result from J_{Rand}.

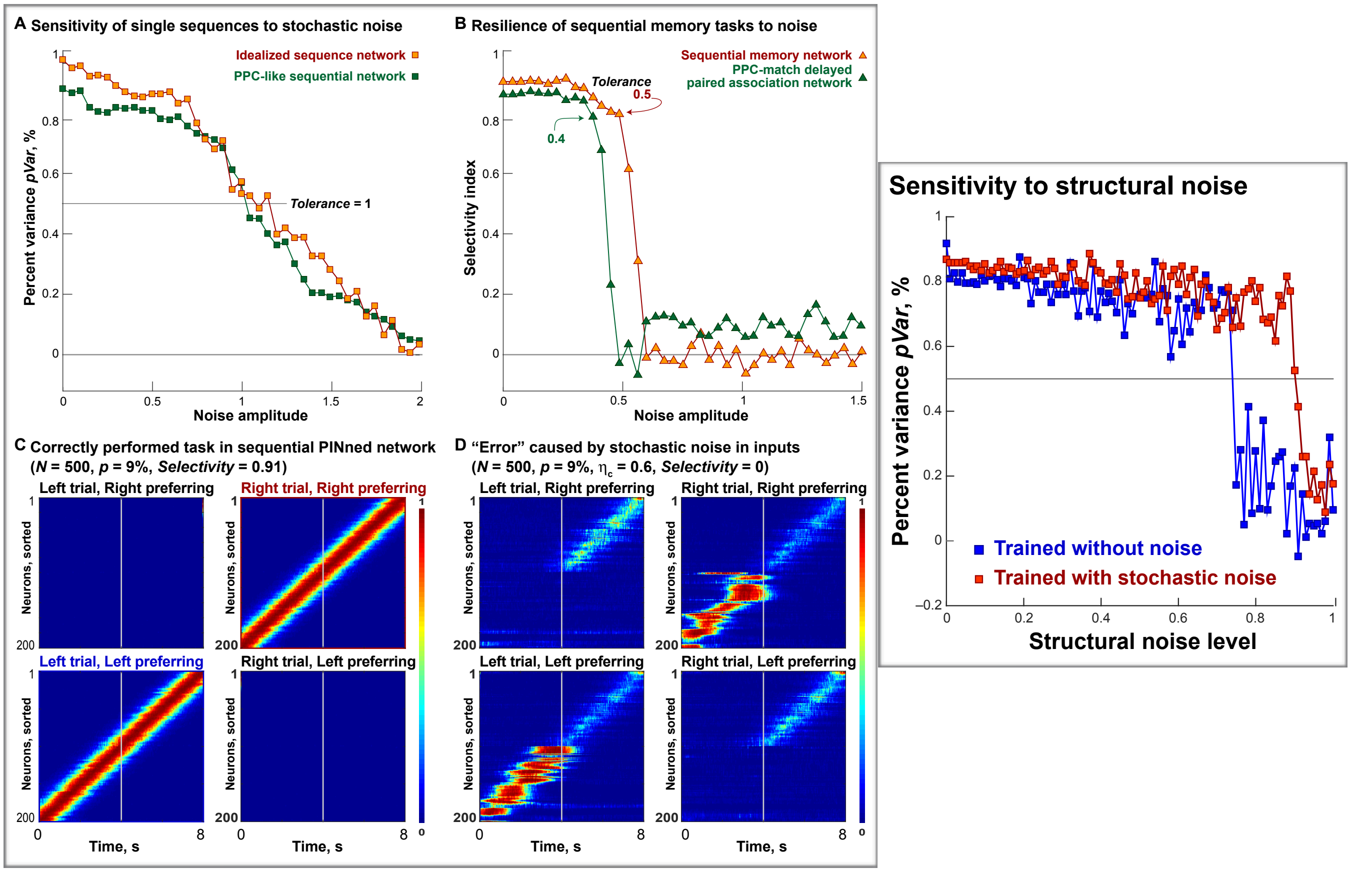

Figure S6 (related to Figures 3 and 6): Left Panel, Robustness of PINned Networks to Stochastic Noise in Inputs and Right Panel, Sensitivity to Structural Noise

A. Stochastic noise, η is injected as an additional current to test whether, and how much, the neural sequences learned through PINning are stable against perturbations (as described in Experimental Procedures 2), and the results are shown here. Percent variance of the target functions explained by the network outputs, *pVar*, drops as amplitude of the injected noise is increased. The maximum tolerance is indicated by the gray line and for single sequences, we define it as the amplitude of noise at which *pVar* drops below 50% and denote it by η_c. For the idealized sequence and for the PPC-like sequence, $\eta_c = 1$.

B. Noise tolerance of memory networks that implement working memory through an idealized sequence (orange triangles) and through PPC-like sequences (green triangles, identical to the network shown in Figure 5) is plotted here. Selectivity of both networks drops at different values of η, indicating the maximum resilience or noise-tolerance of each, denoted by η_c – the sequential memory network has a maximum tolerance of $\eta_c = 0.5$ while the PPC-match memory network has a maximum noise tolerance of $\eta_c = 0.4$.

C. Correctly performed delayed paired association task with delay period memory of the cue identity implemented through two idealized sequences (each similar to Supplemental Figure 4A) in a network of 500 neurons with $p = 9\%$ plastic synapses. For the outputs shown here, the turn period is omitted for clarity, and the delay period starts at the 5s time-point in the 10.5s-long task. The performance of this network, quantified by the selectivity index (Experimental Procedures 9) is 0.91.

D. The same network from panel (D) fails to perform the task (selectivity = 0) because the high levels of stochastic noise present in the inputs ($\eta = 0.6$, here) quenches delay period memory.

E. Right Panel shows sensitivity of PINned networks to structural noise in the synaptic connectivity matrix. Structural noise (described by $\text{level} \times \frac{\mathcal{N}(0,1)}{\sqrt{N}}$, where N = 500 here) is added to the connections in the sparsely PINned connectivity matrix, $\boldsymbol{J_{PINned,\ 8\%}}$ to test whether, and by how much, the synaptic connections are finely tuned. *pVar*, computed as in Experimental Procedures 9, is plotted as a function of structural noise amplitude, for sequential networks obtained by PINning in the presence of stochastic noise in the inputs (red squares) and for networks trained without any noise (blue squares). Gray line is at *pVar* = 50%, the noise-free-PINned network has a tolerance (noise amplitude at which *pVar* drops below 50%) of 0.6, and the network trained with noise, 0.8. Training in the presence of stochastic noise therefore leads to slightly more robust networks, although the drop in *pVar* with the addition of structural noise does not fully recover with training noise.

Supplemental Experimental Procedures S7: Details of Experimental Procedures in the Main Text

7.1. Network elements (see also Experimental Procedures 1 in main text)

We consider a network of *N* fully interconnected neurons described by a standard firing rate model. Each model neuron is characterized by an activation variable, x_i for *i = 1, 2, ... N*, where, *N* = 437 for the PPC-like sequence in Figures 1D and 2A, *N* = 500 for the single idealized sequence in Supplemental Figure 4 and the multi-sequential memory task in Figure 6, and *N* = 569 for the 2AFC task in Figure 5 (we generally build networks of the same size as the experimental dataset we are trying to model, however the results obtained remain applicable to larger networks, see for example, Supplemental Figure 7), and a nonlinear response function, $\phi(x) = \dfrac{1}{1+e^{[-(x-\theta)]}}$. This function ensures that the firing rates, $r_i = \phi(x_i)$, go from a minimum of 0 to a maximum at 1. Adjusting θ allows us to set the firing rate at rest, $x = 0$, to some convenient and biologically realistic background firing rate, while retaining a maximum gradient at $x = \theta$. We use $\theta = 0$ (but see also Supplemental Figure 8).

We introduce a recurrent synaptic weight matrix ***J*** with element J_{ij} representing the strength of the connection from presynaptic neuron *j* to postsynaptic neuron *i* (schematic in Figure 1A) The individual synaptic weights are initially chosen independently and randomly from a Gaussian distribution with mean and variance given by $\langle\, J_{ij} \rangle_J = 0$ and $\langle J_{ij} \rangle_J^2 = g^2 / N$, and are either held fixed or modifiable, depending on the fraction of plastic synapses *p* that can change by applying a learning algorithm (Experimental Procedures 4).

The activation variable for each network neuron x_i is determined by,

$$\tau \frac{dx_i}{dt} = -\mathrm{x}_i + \sum_j^N J_{ij}\phi(x_j) + h_i\,.$$

In the above equation, τ = 10ms is the time constant of each unit in the network and the control parameter *g* determines whether (*g* > 1) or not (*g* < 1) the network produces spontaneous activity with non-trivial dynamics [Sompolinsky, Crisanti & Sommers, 1988; Rajan, Abbott & Sompolinsky, 2010; Rajan, Abbott & Sompolinsky, 2011]. We use *g* values between 1.2 and 1.5 for the networks in this paper, so that the randomly initialized network generates chaotic spontaneous activity prior to activity-dependent modification (gray trace in the schematic in Figure 2B), but produces a variety of regular non-chaotic outputs that match the imposed target functions afterward (red trace in the schematic in Figure 2B, see also results in Figures 2, 5 and 6, see also Experimental Procedures 4 later). The network equations are integrated using Euler method with an integration time step, *dt* = 1ms. h_i is the external input to the unit *i*.

7.2. Design of External Inputs (see also Experimental Procedures 2 in main text)

During the course of the real two-alternative forced-choice (2AFC) experiment [Harvey, Coen & Tank, 2012], as the mouse runs through the virtual environment, the different patterns projected onto the walls of the maze (colored dots, stripes, pillars, hatches, etc.) translate into time-dependent visual inputs arriving at the PPC. Therefore, to represent sensory (visual and proprioceptive) stimuli innervating the PPC neurons, the external inputs to the neurons in the network, denoted by $h(t)$, are made from filtered and spatially delocalized white noise that is frozen (repeated from trial to trial), using the equation, $\tau_{WN} \dfrac{dh}{dt} = -h(t) + h_0\eta(t)$, where η is a random variable drawn from a Gaussian distribution with 0 mean and unit variance, and the parameters h_0 and τ_{WN} control the scale of these inputs and their correlation time, respectively. We use $h_0 = 1$ and $\tau_{WN} = 1$ s. There are as many different inputs as there are model neurons in the network, with individual model neurons receiving the same input on every simulated trial. A few example inputs are shown in the right panel of Figure 1A.

In addition to the frozen noise *h*, which acts as external inputs to these networks, described above, we also test the resilience of the memory networks we built (Figure 6E) to injected stochastic noise. This stochastic injected noise varies randomly (i.e., is a Gaussian random variable between 0 and 1, drawn from a zero mean and unit variance distribution) and independently at every time step. The diffusion constant of the white noise is given by $A_\eta^2 / 2\tau$, where the amplitude is A_η^2 and τ is the time constant of the network units (we use 10ms, as detailed in Experimental Procedures 1). We define "Resilience" or "Noise Tolerance" as the critical amplitude of this stochastic noise, denoted by η_c, at which the delay period memory fails and the Selectivity Index of the memory network drops to 0 (Figure 6E, see also Supplemental Figure 10).

7.3. Extracting Target Functions From Calcium Imaging Data (see also Experimental Procedures 3 in main text)

To derive the target functions for our activity-dependent synaptic modification scheme termed Partial In-Network Training or PINning, we convert the calcium fluorescence traces from PPC recordings [Harvey, Coen & Tank, 2012] into firing rates using two complementary methods. We find that for this dataset, the firing rates estimated by the two methods agree quite well (Supplemental Figure 1).

The first method is based on the assumption that the calcium impulse response function, which is a difference of exponentials ($\mathrm{K} \propto \mathrm{e}^{-t/384} - e^{-t/52}$, with a rise time of 52ms and a decay time of 384ms [Tian et al, 2009; Harvey, Coen & Tank, 2012]), is approximated by an alpha function of the form $\mathrm{K} \propto \mathrm{te}^{-t/\tau_{\mathrm{Ca}}}$, where there is only a single (approximate) time constant for the filter, τ_{Ca} = 200ms. According to this assumption, the scaled firing rate *s* and calcium concentration, *[Ca^{2+}]* are related by,

$$\tau_{Ca}\frac{dCa_i}{dt} = -Ca_i(t) + x_i(t) \text{ and } \tau\frac{dx_i}{dt} = -x_i(t) + s_i(t),$$

where, x*(t)* is an auxiliary variable.The inverse of the above model is obtained by taking a derivative of the calcium data, writing,

$$x_i(t) = Ca_i(t) + \tau_{Ca}\frac{dCa_i}{dt} \text{ and } s_i(t) = x_i(t) + \tau\frac{dx_i}{dt}.$$

Once we have *s(t),* we rectify it and choose a smoothing time constant τ_R for the firing rate we need to compute. Finally, integrating the equation, $\tau_R \frac{dR_i}{dt} = -R_i(t) + s_i(t)$, and normalizing by the maximum gives us an estimate for the firing rates extracted from the calcium data, denoted by *R* (Supplemental Figure 1).

The second method is a fast Bayesian deconvolution algorithm [Pnevmatikakis et al, 2014; Vogelstein et al, 2010, available online at https://github.com/epnev/continuous_time_ca_sampler] that infers spike trains from calcium fluorescence data. The inputs to this algorithm are the rise time (52ms) and the decay time (384ms) of the calcium impulse response function [Tian et al, 2009; Harvey, Coen & Tank, 2012] and a noise parameter [Pnevmatikakis et al, 2014; Vogelstein et al, 2010]. Typically, if the frame rate for acquiring the calcium images is low enough (the data in [Harvey, Coen & Tank, 2012] are imaged at 64ms per frame), the outputs from this algorithm can be interpreted as a normalized firing rate. To verify the accuracy of the firing rate outputs obtained from trial-averaged calcium data (for example, from Figure 2c in [Harvey, Coen & Tank, 2012]), we smoothed the spike trains we got from the above method for each trial separately through a Gaussian of the form $\sum_i e^{\frac{-(t-t_i)}{2\tau_R^2}}$, normalized by $\sqrt{2\pi} \times \tau_R$, and then averaged over single trials to get trial-averaged firing rates (this smoothing and renormalization procedure has also been recommended for faster imaging times [Pnevmatikakis et al, 2014; Vogelstein et al, 2010]).

Once the values of τ_R and τ_{Ca} are determined that make the results obtained by both deconvolution methods consistent (we used τ_R = 100ms and τ_{Ca} = 384ms), we used the firing rates extracted as target functions for PINning through the transform, $f_i(t) = \ln\left[\frac{R_i(t)}{1-R_i(t)}\right]$. The above expression is obtained by solving the activation function relating input current to firing rate of model neurons, $R_i(t) = \frac{1}{1+e^{-f_i(t)}}$, since the goal of PINning is to match the input to neuron *i*, say, denoted by $z_i(t)$ to its target function, denoted by $f_i(t)$. Finally, to verify our estimates, we re-convolved (Supplemental Figure 1) the output firing rates from the network neurons with a difference of exponentials using a rise time of 52ms and a decay time of 384ms.

7.4. Synaptic Modification Rule For PINning (see also Experimental Procedures 4 in main text)

During PINning, the inputs of individual network neurons are compared directly with the target functions to compute a set of error functions, i.e., $e_i(t) = z_i(t) - f_i(t)$, for *i = 1, 2, ... N.* Individual neuron inputs are expressed as $z_i(t) = \sum_j J_{ij} r_j(t)$, where $r_j(t)$ is the firing rate of the *j*th or the presynaptic neuron.

During learning, the subset of plastic internal weights in the connectivity matrix ***J*** of the random recurrent network, denoted by the fraction *p*, undergo modification at a rate proportional to the error term, the presynaptic firing rate of each neuron, r_j and a *pN x pN* matrix, ***P*** (with elements P_{ij}) that keeps track of the rate fluctuations across the network at every time step. Here, *p* is the fraction of neurons whose outgoing synaptic weights are plastic; since this is a fully connected network, this is also the fraction of plastic synapses in the network. Mathematically, $P_{ij} = < r_i r_j >^{-1}$, the inverse cross-correlation matrix of the firing rates of the network neurons (P_{ij} is computed for all *i* but is restricted to *j = 1, 2, 3,…,pN*). The basic algorithm is schematized in Figure 2B. At time *t*, for *i = 1, 2, ... N* neurons, the learning rule is simply that the elements of the matrix ***J*** are moved from their values at a time step Δt earlier through $J_{ij}(t) = J_{ij}(t-1) + \Delta J_{ij}(t)$. Here, the synaptic update term, according to the RLS/FORCE procedure [Haykins, 2002; Sussillo & Abbott, 2009] (since other methods for training recurrent networks, such as backpropagation would be too laborious for our purposes) follows, $\Delta J_{ij}(t) = c[z_i(t) - f_i(t)] \sum_k P_{jk}(t) r_k(t),$ where the above update term is restricted to the *p*% of plastic synapses in the network, which are indexed by *j* and *k* in the above expression. While c can be thought of as an effective learning rate, it is given by the formula, $c = \frac{1}{1 + r'(t)\mathbf{P}(t)r(t)}$. The only free parameter in the learning rule is ***P(0)*** (but the value to which it is set is not critical [Sussillo & Abbott, 2009]). When there are multiple sequences (such as in Figures 5 and 6), we choose *pN* synapses that are plastic and we use those same synapses for all the sequences.

The matrix ***P*** is generally not explicitly calculated but rather updated according to the rule, $\mathbf{P}(t) = \mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)r(t)r'(t)\mathbf{P}(t-1)}{1 + r'(t)\mathbf{P}(t-1)r(t)}$ in matrix notation, which includes a regularizer [Haykins, 2002]. In our scheme, all indices in the above expression are restricted to the neurons with plastic synapses in the network. The algorithm requires the matrix ***P*** to be initialized to the identity matrix times a factor that controls the overall learning rate, i.e., $P(0) = \alpha \times \mathrm{I}$, and in practice, values from 1 to 10 times the overall amplitude of the external inputs (denoted by h_0 in Experimental Procedures 2) driving the network are effective (other values are explored in [Sussillo & Abbott, 2009]).

For numerically simulating the PINned networks whose sequential outputs are shown in Figures 2C, 5 and 6, the integration time step used is *dt* = 1ms (as described in Experimental Procedures 1 and 2, we use Euler method for integration). The learning occurs at every time step for the *p*% of pre-synaptic neurons with plastic outgoing synaptic weights. Starting from a random initial state (Experimental Procedures 1), we first run the program for 500 learning steps, which include both the network dynamics and the PINning algorithm, and then an additional 50 steps with only the network dynamics after the learning has been terminated (convergence metrics below, see also Supplemental Figure 2). A "step" is defined as one run of the program for the duration of the relevant trial, denoted by *T*. Each step is equivalent to *T* = 10500 time points (10.5s) in Figures 2C and 5E, and 8000 time points (8s) in Figure 6B.

On a standard laptop computer, the first 500 such steps for a 500-unit rate-based network producing a 10s-long sequence (such as in Supplemental Figure 4) take approximately 8 minutes to complete in realtime and the following 50, about 30 seconds in realtime (about 1s/step for the 10s-long single sequence example, scaling linearly with network size *N*, total duration or length of the trial *T*, and number of sequences produced.)

The convergence of the PINning algorithm was assayed as follows: (a) By directly comparing the outputs with the data (as in the case of Figures 2A–C and 5D–E) or the set of target functions used (Supplemental Figure 4) and (b) By calculating and following the χ^2-squared error between the network rates and the targets, both during PINning (inset of Supplemental Figure 2) and at the end of the simulation (Supplemental Figure 2). The performance of the PINning algorithm was assayed by computing the percent variance of the data or the targets captured by the sequential outputs of different networks (*pVar*, see Experimental Procedures 7, see also Figure 1H) or by computing the Selectivity Index of the memory network (Selectivity, see Experimental Procedures 9, see also Figure 5F).

7.5. Dimensionality Of Network Activity (*Qeff*) (see also Experimental Procedures 5 in main text)

We use state space analysis based on PCA (see for example, [Rajan, Abbott & Sompolinsky, 2011; Sussillo, 2014] and references therein) to describe the instantaneous network state by diagonalizing the equal-time cross-correlation matrix of network firing rates given by, $Q_{ij} = \langle (r_i(t) - < r_i >)(r_j(t) - < r_j >) \rangle$, where <> denotes a time average. The eigenvalues of this matrix expressed as a fraction of their sum indicate the distribution of variances across different orthogonal directions in the activity trajectory. We define the effective dimensionality of the activity, Q_{eff}, as the number of principal components that capture 95% of the variance in the dynamics (Figure 2F).

7.6. Stereotypy Of Sequence (*bVar*), % (see also Experimental Procedures 6 in main text)

bVar quantifies the variance of the data or the network output that is explained by the translation along the sequence of an activity profile with an invariant shape. For example, for Figure 1C–G, we extracted an aggregate waveform, denoted by R_{ave} (red trace in Figures 1D and Supplemental Figure 4A), by averaging the t_{COM}-realigned firing rates extracted from trial-averaged PPC data collected during a 2AFC task [Harvey, Coen & Tank, 2012]. Undoing the t_{COM} shift, we can write this function for the aggregate or "typical" bump-like waveform as $R_{\text{ave}}(t-t_i)$. The amount of variability in the data that is explained by the moving bump, $R_{\text{ave}}(t-t_i)$ is given by a measure we call *bVar*.

$bVar = \left[1 - \frac{\langle R_i(t) - R_{\text{ave}}(t-t_i) \rangle^2}{\langle R_i(t) - \bar{R}(t) \rangle^2} \right]$, where $<> = \sum_i^N \sum_t^T$. In the above expressions, *R*'s denote the firing rates extracted from calcium data (Experimental Procedures 3); $R_i(t)$ is the firing rate of the i^{th} PPC neuron at time *t* and $\bar{R}(t)$ is the average over neurons. The total duration,*T* is 10.5s in Figures 1, 2 and 5, and 8s in Figure 6.

In some ways, *bVar* is similar to the ridge-to-background ratio computed during the analysis of experimental data for measuring the level of background activity (see for example, Supplemental Figure 14 in [Harvey, Coen & Tank, 2012]); however, *bVar* additionally quantifies the stereotypy of the shape of the transient produced by individual neurons.

7.7. Percent Variance Of Data Explained By Model (*pVar*), % (see also Experimental Procedures 7 in main text)

We quantify the match between the experimental data or the set of target functions, and the outputs of the model by the amount of variance of the data that is captured by the model, $pVar = \left[1 - \frac{\langle D_i(t) - r_i(t) \rangle^2}{\langle D_i(t) - \bar{D}(t) \rangle^2} \right]$, which is one minus the ratio of the Frobenius norm of the difference between the data and the outputs of the network, and the variance of the data. The data referred to here, denoted by *D*, is trial-averaged data, such as from Figures 1D, 2A and 5D.

7.8. Magnitude Of Synaptic Change (see also Experimental Procedures 8 in main text)

In Figure 2E and in Figure 5G, we compute the magnitude of the synaptic change required to implement a single PPC-like sequence, an idealized sequence and three memory tasks, respectively. In combination with the fraction of plastic synapses in the PINned network, *p*, this metric characterizes the amount of structure that needs to be imposed in an initially random network to produce the desired temporally structured dynamics. This is calculated as, normalized mean synaptic change = $\frac{\sum_{ij} |J_{ij}^{\text{PINned, p\%}} - J_{ij}^{\text{Rand}}|}{\sum_{ij} |J_{ij}^{\text{Rand}}|}$, where, following the same general notation as in the main text, $\boldsymbol{J_{PINned, p\%}}$ denotes the connectivity matrix of the PINned network constructed with p% plastic synapses and $\boldsymbol{J_{Rand}}$ denotes the initial random connectivity matrix ($p = 0$).

7.9. Selectivity Index For Memory Task (see also Experimental Procedures 9 in main text)

In Figures 5F, a Selectivity Index is computed (similar to Figure 4 in [Harvey, Coen & Tank, 2012]) to assess the performance of different PINned networks at maintaining cue-specific memories during the delay period of delayed paired association tasks. This metric is based on the ratio of the difference and the sum of the mean activities of preferred neurons at the end of the delay period during preferred trials, and the mean activities of preferred neurons during opposite trials. We compute Selectivity Index as, $\frac{1}{2}\left[\frac{<r>_{r.t}^{r.n} - <r>_{l.t}^{r.n}}{<r>_{r.t}^{r.n} + <r>_{l.t}^{r.n}} + \frac{<r>_{l.t}^{l.n} - <r>_{r.t}^{l.n}}{<r>_{l.t}^{l.n} + <r>_{l.t}^{l.n}}\right]$, where the notation is as follows:

$<r>_{\text{r.t}}^{\text{r.n}} = \frac{1}{N_{\text{right pref neurons}}} \sum_{i}^{\text{right pref neurons}} r_i$ and $<r>_{\text{r.t}}^{l\text{.n}} = \frac{1}{N_{\text{left pref neurons}}} \sum_{i}^{\text{left pref neurons}} r_i$ are the average firing rates of right-preferring and left-preferring model neurons on right trials;

$<r>_{l\text{.t}}^{\text{r.n}} = \frac{1}{N_{\text{right pref neurons}}} \sum_{i}^{\text{right pref neurons}} r_i$ and $<r>_{l\text{.t}}^{l\text{.n}} = \frac{1}{N_{\text{left pref neurons}}} \sum_{i}^{\text{left pref neurons}} r_i$ are the average firing rates of right-preferring and left-preferring model neurons on left trials of the simulated task. The end of the delay period is at approximately 10s for the network in Figure 5 (after [Harvey, Coen & Tank, 2012]) and at ~7s time point for the network in Figure 6.

7.10. Temporal Sparseness Of Sequences (*N_{Active}/N*) (see also Experimental Procedures 10 in main text)

The temporal sparseness of a sequence is defined as the fraction of neurons active at any instant during the sequence, the fraction, N_{Active}/N. To compute this, first, the normalized firing rate from each model neuron in the network or from data [Harvey, Coen & Tank, 2012], denoted by $R_i(t)$, is realigned by the center-of-mass, $i_{COM}(t)$, given by, $i_{\text{COM}}(t) = \sum_i i * R(i,t) / \sum_i R(i,t)$, where i is the neuron number and t is time. The realigned rates are then averaged over time to obtain $<R>_{\text{time}}$, i.e., $\langle R(i) \rangle_{\text{time}} = \langle R(i - i_{\text{COM}}(t), t) \rangle_{\text{time}}$ after using a circular shift rule to undo the i_{COM}-shift. The standard deviation of the best Gaussian that fits this curve $<R>_{\text{time}}$ is the number, N_{Active}, and the ratio of N_{Active} to the network size, N, yields the temporal sparseness of the sequence, N_{Active}/N. For the data in Figure 2A, for example, N_{Active}/N = 3% (i.e., 16 neurons out of a total of 437). Practically speaking, decreasing this fraction makes the sequence narrower and at a critical value of sparseness (N_{Active}/N = 1.6% in Figure 6), there is not enough current in the network to propagate the sequence. However, up to a point, making the sequences sparser increases the capacity of a network for carrying multiple non-interfering sequences (i.e., sequences with delay period memory of cue identity), without demanding an increase in either the fraction of plastic synapses, p, or in the overall magnitude of synaptic change.

7.11. Analyzing the Structure of PINned Synaptic Connectivity Matrices (see also Experimental Procedures 11 in main text)

This is pertinent to Section 3 (Figures 3 and 4), in which we quantify how the synaptic strength varies with the "distance" between pairs of network neurons in connectivity space, $i - j$, in PINned sequential networks. We first compute the means and the standard deviations of the principal diagonals, i.e., $\sum_{i=j}^{N} \frac{J_{ij}}{N} \rightarrow \sum_{i=1}^{N} \frac{J_{ii}}{N}$, in the 3 connectivity matrices under consideration here – $\boldsymbol{J_{PINned, 8\%}}$, $\boldsymbol{J_{Rand}}$, and just for comparison purposes, $\boldsymbol{J_{PINned, 100\%}}$. Then, we compute the means and the standard deviations of successive off-diagonal “stripes” moving away from the principal diagonal, i.e., $\sum_{i-j=a}\frac{J_{ij}}{N} \rightarrow \sum_{i=a+1}^{N}\frac{J_{i(i-a)}}{N}$, for the same three matrices. These are plotted in Figures 3B for the sparsely PINned matrix, $\boldsymbol{J_{PINned,\ 8\%}}$, relative to the randomly initialized matrix, $\boldsymbol{J_{Rand}}$, and in Figure 3E, for the fully PINned matrix constructed for comparison purposes, $\boldsymbol{J_{PINned,\ 100\%}}$. The same analysis is also used to compare different partially structured matrices in Supplemental Figure 9.

7.12: Cross-Validation Analysis (see also Experimental Procedures 12 in main text)

We first divided the data from 436 neurons in Figure 1D into two separate “synthetic” sequences by assigning the even-numbered cells to one (let’s call this Sequence A, containing 218 neurons) and the odd-numbered cells to another sequence (say, Sequence B, also with 218 neurons). Then we constructed a PINning-based network with 218 model neurons, exactly like we described in the main text, using data from Sequence A as target functions for PINning. Next, we computed the percent variance of the data in Sequence A and the data in Sequence B, denoted by $pVar_{Test\ A,\ Train\ A}$ and $pVar_{Test\ B,\ Train\ A}$, respectively, that are captured by the outputs of the PINned network (Experimental Procedures 7). Following a similar procedure, we also computed $pVar_{Test\ A,\ Seq\ B}$ and $pVar_{Test\ B,\ Seq\ B}$, after PINning a second network against target functions derived from Sequence B.

We obtained the following estimates for all fractions of plastic synapses, p :

$pVar_{Test\ A,\ Train\ A} = 91 \pm 2\%$ $\qquad$ $pVar_{Test\ B,\ Train\ A} = 45 \pm 2\%$

$pVar_{Test\ A,\ Train\ B} = 46 \pm 2\%$ $\qquad$ $pVar_{Test\ B,\ Train\ B} = 90 \pm 2\%$

For comparison purposes, a random network such as the one in Figure 1B only captures a tiny amount of the variability of data (in this notation, $pVar_{Test\ Data,\ Random\ Network} = 0.2\%$, see also, right panel of Figure 1H). Additionally, the data from one set only accounts for 49% of the variance of the data of the other set, i.e., $pVar = 49\%$. Thus the model does almost as well as it possibly could.