Initial state randomness improves sequence learning in a model hippocampal network

A. P. Shon,* X. B. Wu,† D. W. Sullivan,‡ and W. B Levy§
Department of Neurological Surgery, University of Virginia, P.O. Box 800420, Charlottesville, Virginia 22908-0420
(Received 27 March 2001; published 6 March 2002)

Randomness can be a useful component of computation. Using a computationally minimal, but still biologically based model of the hippocampus, we evaluate the effects of initial state randomization on learning a cognitive problem that requires this brain structure. Greater randomness of initial states leads to more robust performance in simulations of the cognitive task called transverse patterning, a context-dependent discrimination task that we code as a sequence prediction problem. At the conclusion of training, greater initial randomness during training trials also correlates with increased, repetitive firing of select individual neurons, previously named local context neurons. In essence, such repetitively firing neurons recognize subsequences, and previously their presence has been correlated with solving the transverse patterning problem. A more detailed analysis of the simulations across training trials reveals more about initial state randomization. The beneficial effects of initial state randomization derive from enhanced variation, across training trials, of the sequential states of a network. This greater variation is not uniformly present during training; it is largely restricted to the beginning of training and when novel sequences are introduced. Little such variation occurs after extensive or even moderate amounts of training. We explain why variation is high early in training, but not later. This automatic modulation of the initial-state-driven random variation through state space is reminiscent of simulated annealing where modulated randomization encourages a selectively broad search through state space. In contrast to an annealing schedule, the selective occurrence of such a random search here is an emergent property, and the critical randomization occurs during training rather than testing.

DOI: 10.1103/PhysRevE.65.031914 PACS number(s): 87.18.Sn, 87.19.La, 05.40.Ca

I. INTRODUCTION

Random fluctuations are generally an undesirable feature of information processing systems including sequence learning neural networks (e.g., [1]). Investigations of both biological and artificial neural systems have, however, shown that such fluctuations can improve performance [2,3,4,5].

Using our hippocampal model [5], we have previously shown that random fluctuations are important for robust learning and that the model is sensitive to initial conditions. By quantifying this sensitivity in a recurrent, sequence-learning neural network model and by correlating it with learned performance, we have made three, related observations about the role of randomization in learning the cognitive task called transitive inference (TI). First, this model is sensitive to randomization of the network state produced by an external influence even when randomization is limited to time step zero of each trial. Second, the model is most sensitive to this randomization when the probability of neuronal firing on time step zero equals the nominal, preset activity level implicit in the parametrization of these randomly connected networks. Third, a strong positive correlation exists between the model's success in learning TI and the effect that time step zero randomization has on the next network state. For example, the best performance on this cognitive task occurs when randomization at time zero is chosen so as to produce the greatest randomizing effect on a future state of the network.

Here we extend these results in three ways to gain greater insight into the mechanisms by which initial state randomization controls randomization of later states of the network simulations. In contrast to the previous study [5], where the number of neurons firing in the initial vector $\mathbf{Z}(0)$ was varied, here we investigate the effect of randomly varying the firing neurons for a specified fraction of each $\mathbf{Z}(0)$ vector, while the total number of neurons is kept fixed. (For example, suppose there are 1000 neurons and total activity is set to 90 particular neurons out of these 1000; then if the specified randomization is 50%, 45 out of these 90 will be randomly exchanged with the other 910 neurons.) In colloquial terms, the random variation of $\mathbf{Z}(0)$ from trial to trial corresponds to beginning each training trial with a different state of mind.

A second difference from our previous study is the use of a different cognitive task, transverse patterning (TP). Both the TI studied previously and TP studied here require normal hippocampal function [6,7] if they are to be learned. (Thus a good network model of the hippocampus [8,9] should be able to learn both tasks.) Interestingly, TI and TP are complementary cognitive tasks in the sense that TI is a context dependent, linear inferential task and TP is a context dependent, nonlinear inferential task. In particular, success at TP requires learning when each of three symbols is correct as in the following situations: when *A* and *B* are concurrent, *A* is correct; when *B* and *C* are concurrent, then *B* is correct; and when *A* and *C* are concurrent, then *C* is the right answer (as in the playground game rock-paper-scissors).

*Present address: Department of Computer Science and Engineering, University of Washington, Box 352350, Seattle, WA 98195-2350. Email address: aaron@cs.washington.edu
†Email address: xw3f@virginia.edu
‡Email address: dws3t@virginia.edu
§Author to whom correspondence should be addressed. Email address: wbl@virginia.edu

A third difference is the substantial extension of our understanding of the mechanisms through which initial state randomization works. By performing systematic measurements of network states as functions of initial state randomizations, we are able to develop and define the hypothesis of a randomly driven state space search during early phases of learning.

Because this model is not derived from the typical recurrent neural model used by physicists [10] (although it is related to many of these models), an overview of the model seems appropriate. This overview consists of the cognitive, biological, and computational concerns that motivated this network.

Based on hypothesized hippocampal computational functions that explain generalized hippocampal function in behavioral and cognitive contexts [8,9], the model performs sequence learning while at the same time functions as a random recoder. From these two basic characteristics arises an ability to solve problems, whose solutions are only possible by use of contextual information (e.g., Refs. [9,11]). The idea that the hippocampus is a sequence learning/predicting device is controversial. While several research groups seem to agree [12,13,14,15,16], there are other opinions (e.g., Refs. [17,18]).

In terms of specifics, the model uses McCulloch-Pitts neurons modified for divisive inhibition. The connectivity of the model is recurrent, sparse, excitatory, and random except for feedback and feedforward inhibition that are global. Neurons driven by external inputs are incorporated without distinction into the recurrent network. That is, externally activated neurons are also randomly embedded into the network's sparse, recurrent connectivity.

Associative synaptic modification is an autonomous property of each excitatory synapse. The synaptic modification rule is local, asymmetrically time spanning with both potentiation and depression possible [19,13,20,21].

Finally, the overall model performance is not based on any asymptotic theory. Although isolated synapses (i,j) will tend to take on a strength proportional to the conditional mean E [presynaptic input i at $(t-1)$|postsynaptic neuron j fires at t], the entire network cannot be seen as converging to some asymptotic limit. At the end of training, there are transient attractors, but rarely are there stable points in state space after training on the transverse patterning problem. Although somewhat disappointing from a theoretical viewpoint, the model reproduces the learning rates seen in behaving animals (see Ref. [22] for one such comparison), which is really the final arbiter of the model's success or failure.

In this study, we manipulate only one of the two sources of randomness that exist in our minimal model of hippocampal CA3. One source of randomness, which makes each simulation different, is the randomly determined initial connectivity of each simulation. The second source of randomness, and the object of manipulation and study here, is the composition of the set of neurons that fire just before each training and test trial. This set is specified by the vector $\mathbf{Z}(0)$.

Defining randomness in terms of how many neurons are perturbed and how many are kept constant (see Sec. II for details), the results of the computational simulations demonstrate a strong positive correlation between $\mathbf{Z}(0)$ randomness and probability of learning TP (Fig. 1). The results also demonstrate a strong positive correlation between $\mathbf{Z}(0)$ randomness and the average firing periods of specific sets of neurons, called *local context neurons*. These neurons fire only on specific and contiguous time steps of a particular sequence *and thus are subsequence detectors, and are analogous to the hippocampal place cells* [23]. Local context neurons were previously shown [22] to play an important role when simulations of the model solve the transverse patterning problem. This, and additional results, reveal that $\mathbf{Z}(0)$ randomization drives a searchlike process to produce more robust learning.

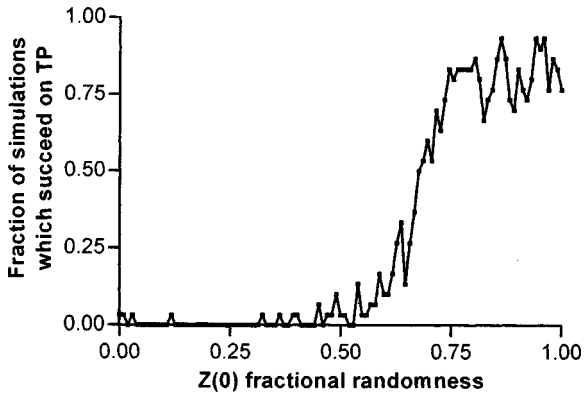

FIG. 1. Randomness of $\mathbf{Z}(0)$ facilitates learning of TP. Performance begins improving once approximately 50% of the $\mathbf{Z}(0)$ neurons are randomized from trial to trial. The fraction of the simulations that successfully learned TP (see Sec. II C) is plotted as a function of the number of randomly chosen neurons in each $\mathbf{Z}(0)$ state of a simulation. See Sec. II E and the Introduction for our definition of randomness and an example of a randomization. Thirty different simulations (each with different random connections between its neurons) were trained and tested at each level of $\mathbf{Z}(0)$ randomness. Inhibition constants were selected to achieve approximately 10% activity. Each $\mathbf{Z}(0)$ vector contained 102 active neurons. All $\mathbf{Z}(0)$ vectors were sets of neurons orthogonal to the set of external inputs. Simulation parameters: $N=1024$, $m_e=32$, $\epsilon=0.05$, $K_I=0.048$, $K_R=0.048$, $K_0=0$, $w=0.4$, and $c=0.1$.

Finally, our studies of neuronal excitation distributions and an accompanying theorem provide a firmer basis for understanding how initial state variations help the model develop robust internal codes to solve problems.

II. METHODS

A. Network architecture

Our computationally minimal model [Fig. 2(a)] of the hippocampus consists of a sparsely connected (10%) recurrent network of McCulloch-Pitts neurons [9]. External inputs represent signals to CA3 from the entorhinal cortex and the dentate gyrus. However, most of the excitation is recurrent.

In this model, a neuron's internal excitation on time step t of a trial is the sum of the weights of its recurrent inputs that were active on time step $(t-1)$. This excitation is divided by a term representing shunting inhibition to obtain a value on

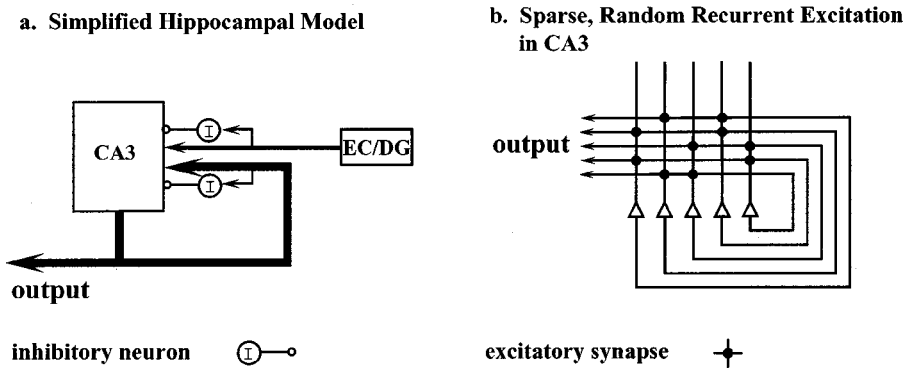

FIG. 2. The basic hippocampal model. (a) The external input to *CA*3 is sparse and excitatory, representing a combination of the inputs from entorhinal cortex (EC) and dentate gyrus (DG). The strongest input to the network is its own recurrent excitation. Both the external and recurrent inputs are accompanied by proportional activation of inhibitory neurons (*I*). The output of the network is the firing vector of the *CA*3 neurons themselves. (b) Schematic depiction of the sparse nature of the recurrent excitatory synapses of the *CA*3 neurons. Key: *CA*3 pyramidal neurons (Δ), recurrent excitatory synapses on these neurons (●).

the interval [0, 1] [24]. Formally, the internal excitation $y_j(t)$ of neuron j on time step t is

$$y_j(t)=\frac{\sum_{i=1}^{N} w_{ij}c_{ij}z_i(t-1)}{\sum_{i=1}^{N} w_{ij}c_{ij}z_i(t-1)+K_R m(t-1)+K_0+K_I\sum_{i=1}^{N} x_i(t)}, \tag{1}$$

where N is the total number of neurons. A synaptic connection from neuron i to neuron j is represented by c_{ij} (1 if a connection is present, 0 if no connection is present). The weight of the synapse from neuron i to neuron j is represented by w_{ij}. The variable $z_i(t-1)$ represents the binary $[z_i(t)\in\{0,1\}]$ firing of recurrent neuron i on time step $(t-1)$. The binary variable $x_i(t)$ represents the firing of an external input to neuron i on time step t. The variable $m(t-1)$ represents the number of neurons in the network that fired on time step $(t-1)$. The denominator of Eq. (1) contains three inhibitory parameters: K_I scales feedforward inhibition of external inputs; K_R scales feedback inhibition as a function of recurrent neuronal firing; and K_0 represents a resting conductance. These three inhibitory parameters are set such that activity levels are maintained around a desired value. We use the method described in Ref. [24] to determine the best values for these inhibitory parameters.

After calculating each neuron's internal excitation, an output firing decision occurs—either a 0 or a 1, representing the absence or presence of an action potential respectively. The firing function z_j is determined on time step t by the equation

$$z_j(t)=\begin{cases}1 & \text{if } y_j(t)\geqslant 0.5 \quad \text{or if } x_j(t)=1,\\ 0 & \text{otherwise}\end{cases}. \tag{2}$$

At the beginning of each simulation, the neuronal connections are specified randomly, such that each neuron receives the same number of recurrent synaptic connections. While the number of connections made by each neuron varies and follows a binomial distribution around the mean value Nc (where N is the number of neurons and c is percent connectivity). Here c was always 10%; thus our model is characterized by sparse connectivity [Fig. 2(b)].

Two types of trials may take place during a simulation: *training* and *test*. During training, a network is presented with a prespecified input sequence and synaptic modification is allowed to occur. A training trial begins at time step zero with synaptic modification inactivated, and the firing state of the network specified by the vector $\mathbf{Z}(0)$. Then follows the presentation of the training input sequence, with synaptic modification allowed throughout the sequence. Between training trials, the network autonomously adjusts its inhibitory parameters (K_R, K_I, and K_0) to keep activity levels relatively stable (for details, see Ref. [25]). A test trial also begins with an initial firing state specified by $\mathbf{Z}(0)$. During a test trial, the network is presented with a part of the training sequence and is expected to produce an output corresponding to one of three possible answers. No synaptic modification takes place within a test trial.

B. The synaptic modification rule

This model learns via locally driven synaptic modification, using a time-dependent associative synaptic modification rule [8,26]. On each time step of each training trial, the weight of every synapse in the network is updated according to the equation

$$w_{ij}(t+1)=w_{ij}(t)+z_j(t)\varepsilon[z_i(t-1)-w_{ij}(t)], \tag{3}$$

where ε is a small positive constant, typically 0.05. Synaptic modification in this model depends completely on the firing of postsynaptic neuron *j*: if the postsynaptic neuron does not fire, the synaptic weight does not change. If the postsynaptic neuron fires on the time step immediately after the presynaptic neuron fires, then the connection w_{ij} is strengthened. Conversely, if the postsynaptic neuron fires when the presynaptic neuron has not fired on the previous time step, then the synaptic weight decreases. After many training trials, synaptic modification produces synaptic weights that are conditional probabilities as $E[Z_i(t-1)|Z_j(t)=1]$ [27].

C. The transverse patterning problem

The TP task [28] requires a choice from among three stimuli, *A,B,C,* which are presented to the network as simultaneous pairs (*AB,BC,* or *CA*). We refer to trials for each of these three stimulus pairs as a *subtask*. Figure 3 illustrates the firing of external neurons used in training to define the two possible versions (i.e., right and wrong) of the *AB* subtask.

In TP experiments, the experimenter trains the subject (human, animal, or a simulated neural network) to choose the correct stimulus in each pair. First, a stimulus pair is presented to the subject, then the subject chooses one stimulus in that pair, and then the subject is told whether its decision was correct or incorrect. This training cycle occurs repeatedly. The stimulus pairings are circular with respect to the correct decision: the stimulus *A* is correct when *AB* co-occurs, the stimulus *B* is correct when *BC* co-occurs, and the stimulus *C* is correct when *CA* co-occurs. As a result of the symmetry of right and wrong, the correct decision can only be made using the contextual knowledge of the stimulus pair itself.

To train a network on TP, we create six training sequences, two for each subtask (e.g., Fig. 3) corresponding to the correct and incorrect decisions. Each training sequence consists of three orthogonal (nonoverlapping) patterns of externally fired neurons. The first pattern (*AB*, *BC*, or *CA*) represents a stimulus pair, the second pattern a ''decision'' (*a*,*b*, or *c*) represents the choice between the two stimuli, and the third pattern (+ or −) represents the outcome (right or wrong) of the decision. Each pattern is presented for three consecutive time steps. We use this presentation method because it enhances performance on the transverse patterning problem [22] and because stimuli are not instantaneous events. Figure 3 illustrates the configuration of external inputs during training for the *AB* subtask. The six training sequences for TP are:

Subtask	Sequence w/correct decision	Sequence w/incorrect decision
AB	$(AB)(AB)(AB)aaa+++$	$(AB)(AB)(AB)bbb---$
BC	$(BC)(BC)(BC)bbb+++$	$(BC)(BC)(BC)ccc---$
CA	$(CA)(CA)(CA)ccc+++$	$(CA)(CA)(CA)aaa---$

Here we employed a progressive training paradigm, as has been used in behavioral experiments ([6,29] see Ref. [30] for a computationally based comparison of different training methods that reflect the animal and human literature). In this paradigm, training occurs in blocks of trials for each subtask. Within a block, training trials are mixed between the two training sequences (+ and −) for the subtask. This mixing between the two training sequences is pseudorandom and is constant across simulations. The progressive training paradigm begins with four blocks of 30 training trials consisting of only the *AB* subtask. Blocks of 20 training trials on the *BC* subtask are then interspersed with additional blocks of five training trials on the *AB* subtask. Blocks of five trials on the *CA* subtask are then introduced, intermixed with blocks of

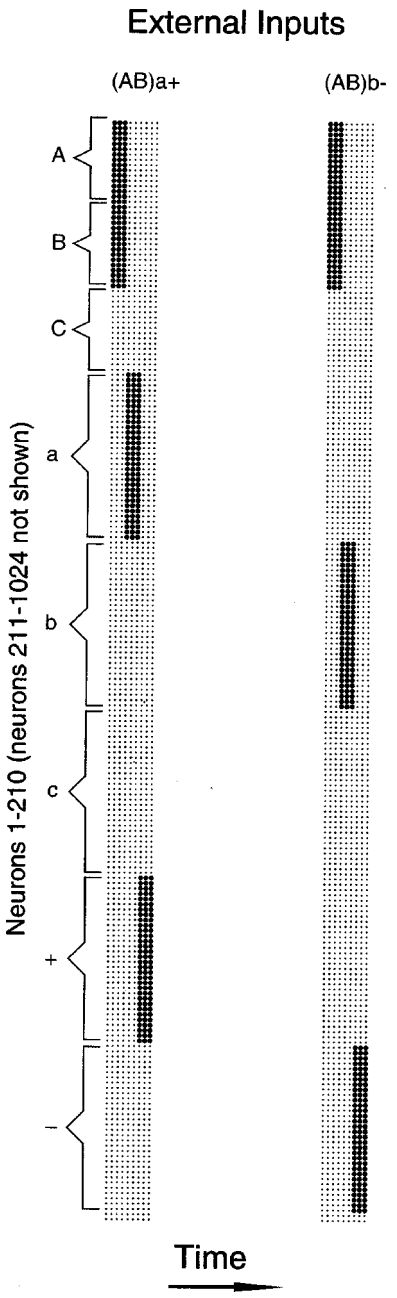

FIG. 3. Firing diagrams for a TP subtask. The left firing diagram, $(AB)a+$, shows an input sequence for an *AB* training trial with a correct decision, a, and its positive outcome, +. The firing diagram to the right, $ABb-$, shows the external input for an incorrect trial where the, *b*, decision is made and its negative result is signaled, −. Note the orthogonal nature of the external inputs: stimulus *A*, neurons 1–16; stimulus *B*, neurons 17–32; decision pattern a, neurons 49–80; decision pattern *b*, neurons 81–112; outcome for the correct response—neurons 145–176; outcome for the incorrect response—neurons 177–208. Because the network is randomly connected, the spatial juxtaposition of two neurons in this schematic is irrelevant to network performance and is used solely for explanatory convenience. Only 210 of the 1024 neurons are shown.

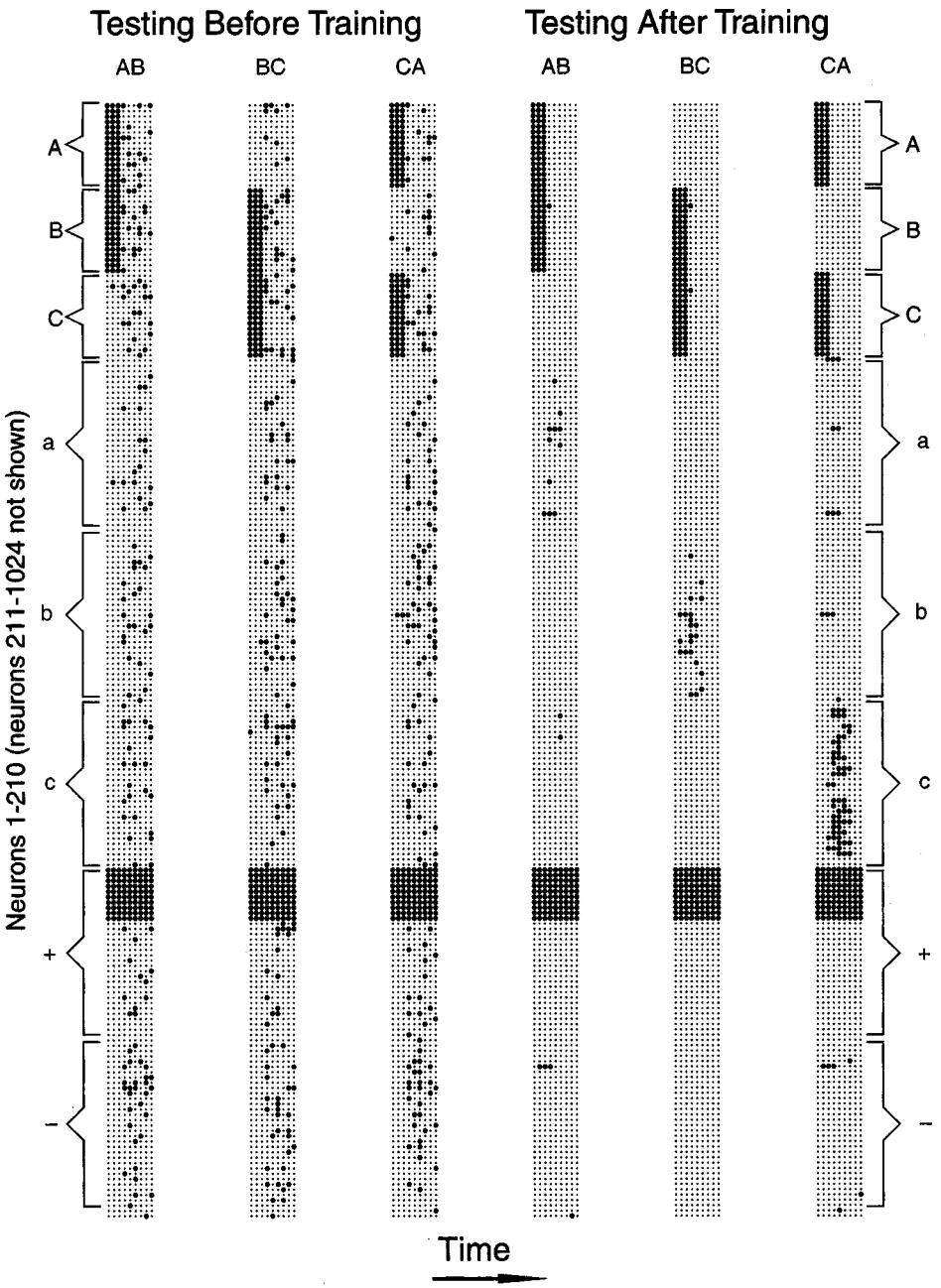

FIG. 4. Training is necessary for strong, reliable correct decision patterns. Illustrated here are firing patterns for test trials given before and after training. The first three firing diagrams are of test trials before training, to stimulus pairs *AB*, *BC*, and *CA*, respectively. Note that the firing patterns of decision neurons (indicated by *a*, *b*, and *c*) are seemingly more random prior to training. However, after training, externally defined decision firing patterns *a*, *b*, or *c* are clearly produced by the network when tested on *AB*, *BC*, or *CA*, respectively.

five training trials on the *AB* and *BC* subtasks. In the last phase of training, trials of all three subtasks are fully intermixed. Each simulation of TP consisted of 300 training trials.

The proportion of training trials consisting of a stimulus pair (e.g., *AB*) and the correct decision pattern with its accompanying positive outcome pattern (i.e., *a*, + in this example) as opposed to the incorrect decision pattern and the negative outcome pattern, varied as training progressed. Specifically, the proportion of correct responses increased over training exactly at the rate reported by Alvarado and Rudy [6].

Following each learning trial, we assessed learning using the method of induced attractors (see Ref. [31]). Each test trial was nine time steps long, plus the starting state $\mathbf{Z}(0)$, with a stimulus pair presented on time steps one, two, and three, and ten neurons from the positive outcome pattern active throughout. We decode the network's decision based on the recurrently induced firing of the external neurons that define the two possible decision patterns for each particular stimulus pair. For example, Fig. 4 illustrates testing before and after training for each of the three subtasks. Before training, the simulations produce highly variable firing patterns of the decision neurons. After training, however, there is a clear, strong, and relatively stable firing of neurons that represent the correct decision, and this is true for each subtask.

To quantify this decision, we determine the average number of neurons in each decision pattern that fired during the last three time steps of the test trial. If more neurons belonging to the correct externally defined decision pattern (e.g., the decision ''*a*'' when *AB* is the stimulus pair) fired than neurons belonging to the incorrect decision pattern, then the network made the correct decision on the subtask tested, and

the test trial is scored as a 1. If more neurons belonging to the incorrect decision pattern (e.g., "*b*" when *AB* is the stimulus pair) fired than in the correct decision pattern, then an incorrect decision has been made, and the trial is scored as a 0. If an equal number of neurons belonging to both decision patterns fired, the trial is scored as 0.5. Typical successful test trials, characterized by higher activity of the neurons representing the correct decision pattern, are shown for all three subtasks in Fig. 4.

Success in test trials (which are interleaved with the final 30 training trials) is used to quantify learned performance. As defined by Alvarado and Rudy [6], a simulation learned TP only if it generated the correct decision at least 85% of the time on all three subtasks during these final 30 test trials (using alternative definitions we have determined that this threshold criterion does not introduce artificial trends into the results).

D. Local context neurons

We previously [1] described the formation of transiently self-exciting assemblies of neurons (see also Ref. [8], Fig. 11 and 12). Such assemblies are characterized by neurons that fire repetitively in response to a specific set of temporally contiguous patterns in a sequence. A repetitively firing neuron is called a *local context neuron* because it locates, in time, a subsequence that provides contextual information. For example, if a neuron fires only on time steps 3, 4, and 5 of a given sequence, it is a local context neuron of length 3 for that sequence. A neuron that fires on multiple, noncontiguous time steps is not a local context neuron, but such neurons are extremely rare at the end of training.

Figure 5, particularly the recurrently activated neurons on the right half of the figure, illustrates the local context neurons formed during a TP simulation. Early in training, neurons 211–420, which are not directly activated by external inputs, fire only in a sparse, scattered manner. In contrast, after training, if one of these neurons fires, it tends to fire for several contiguous time steps. The formation of local context neurons is important for a simulation to succeed at TP [22].

To examine how randomness in $\mathbf{Z}(0)$ affects formation of local context neurons during TP simulations, we determined average local context length for each simulation and averaged those values over all 30 simulations for each degree of fractional randomness tested.

E. Varying randomness in Z(0)

The initial state vector, $\mathbf{Z}(0)$, specifies the binary firing states $Z_i \in \{0,1\}$ of each neuron at time step zero of each training and test sequence. In contrast to Wu and Levy [5], where the positively valued neurons were fully randomized and their Hamming length systematically varied, here $\mathbf{Z}(0)$ is constrained to a fixed length that corresponds to the optimal length implied by the earlier study. Moreover, only a fraction of this length (from 0 to 100%) is randomized in a set of simulations. The number of randomly firing neurons divided by the total number of firing neurons defines the fractional randomness of a $\mathbf{Z}(0)$. That is, for one full simulation of TP training and testing, the binary state vector $\mathbf{Z}(0)$

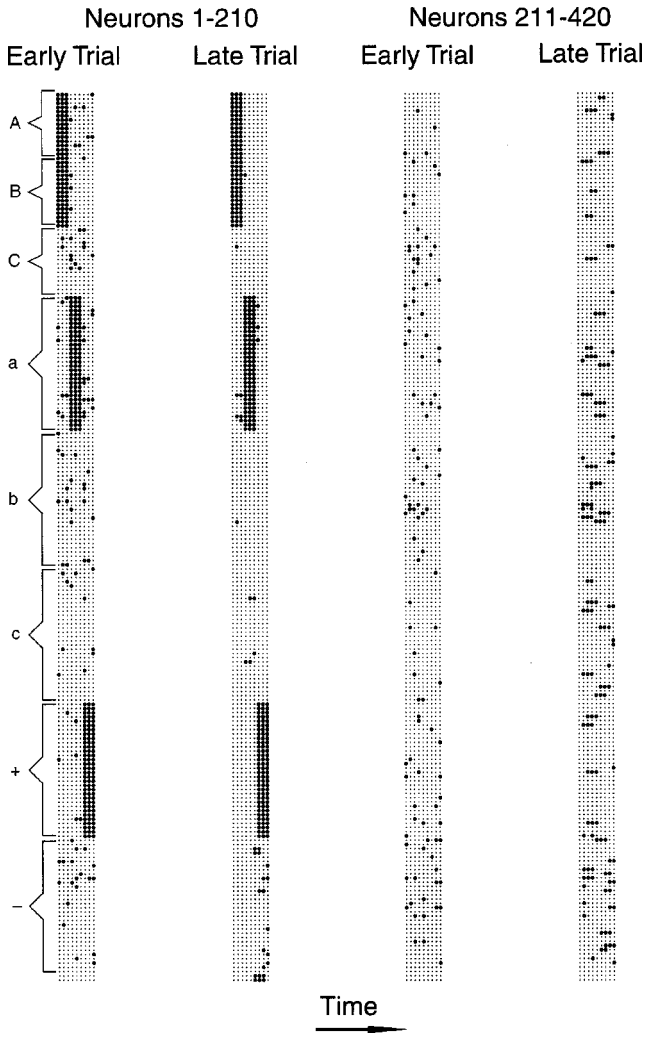

FIG. 5. Context neurons develop with training. Notice the frequent sequential firing of the recurrent neurons (211–420) occurring late in training compared to early in training. This repetitive firing stands in contrast to the sparse, somewhat random firing exhibited in early training. The two leftmost diagrams show firing by mostly externally driven neurons, and here only a few neurons are recurrently driven to fire. The example here is for an $Aba+$ training sequence. The external (i.e., input) codes are: stimulus *A*, neurons 1–16; stimulus *B*, neurons 17–32; decision pattern a, neurons 49–80; outcome for the correct response—neurons 145–176. Because the network is randomly connected, the spatial juxtaposition of two neurons in this schematic is irrelevant to network performance and is used solely for explanatory convenience. Only 420 of the 1024 neurons are shown.

has a fixed, specified fractional randomness, and from trial to trial, each $\mathbf{Z}(0)$ consists of a fixed, nonrandom positively valued subspace of the $\{0,1\}^N$ state space accompanied by neurons that are varied randomly over the remaining state space from trial to trial (except when fractional randomness is zero). To create a $\mathbf{Z}(0)$ with a particular fractional randomness, we construct two preliminary subvectors. The first subvector represents the group of neurons that *must* fire as part of all $\mathbf{Z}(0)$ vectors on all training and test sequences in the simulation. Neurons coding input patterns are excluded

from this first subvector. The second subvector consists of the neurons that *may* fire in all $\mathbf{Z}(0)$ vectors. For each trial, neuronal firing in the second subvector is uniformly random given the required number that must fire.

This method of constructing $\mathbf{Z}(0)$ according to fractional randomness provides an approximate control of trial-to-trial $\mathbf{Z}(0)$ variation. The average normalized Hamming distance between successive $\mathbf{Z}(0)$'s during training is a function of the fractional randomness used in $\mathbf{Z}(0)$ construction. This function and its derivation can be found in Appendix B.

To map the relationship between TP performance and fractional randomness used in training, thirty TP simulations are averaged for each of 100 different levels of fractional randomness ranging from 0 to 1.

F. Measuring the effects of Z(0) randomness on firing patterns

To quantify the model's sensitivity to $\mathbf{Z}(0)$ randomizations, the distance in state space between the temporally matched firing states for a pair of simulations is measured using Hamming distance. To generalize across simulations using different activity levels, we normalize Hamming distance by activity. Thus, where d_H is Hamming distance, d_{nH} is normalized Hamming distance, $Z(t)$ and $Z'(t)$ are firing vectors from a paired simulation, and $m(t)$ is the number of neurons that are active on time step t:

$$d_{nH}[Z(t),Z'(t)]=\frac{d_H[Z(t),Z'(t)]}{2m(t)}.$$

G. Dynamic modulation of inhibition

We ran simulations with and without dynamic modulation of inhibition [25]. When modulated inhibition is used, the values of K_R, K_I, and K_0 are adjusted from trial to trial to maintain, in an approximate sense, a desired activity level. Simulations run without dynamic modulation of inhibition can exhibit significant fluctuations in activity from trial to trial. Even so, all results described below are robust without regard to modulated inhibition.

III. SIMULATION RESULTS

A. Randomness in initial firing patterns assists performance in cognitive tasks

Simulations show a positive role for $\mathbf{Z}(0)$ randomness in learning TP. The amount of trial-to-trial variability of $\mathbf{Z}(0)$ (fractional randomness) is monotonically related to performance, and the best performance is achieved when $\mathbf{Z}(0)$ is fully randomized (fractional randomness$=1$) for each trial (Fig. 1).

Another baseline control is the use of no active neurons in $\mathbf{Z}(0)$ as opposed to the fixed, nonfluctuating neurons of $\mathbf{Z}(0)$ with fractional randomness of zero, as illustrated in Fig. 1. When $\mathbf{Z}(0)$ randomness is zero, only 57.5% (23/40) of the simulations are successful compared to 81.5% (636/780, standard deviation$=7.1\%$) of simulations with full (75–100%) randomization. Thus, regardless of the controls (see also, Refs. [5,24]), initial state randomization helps learning.

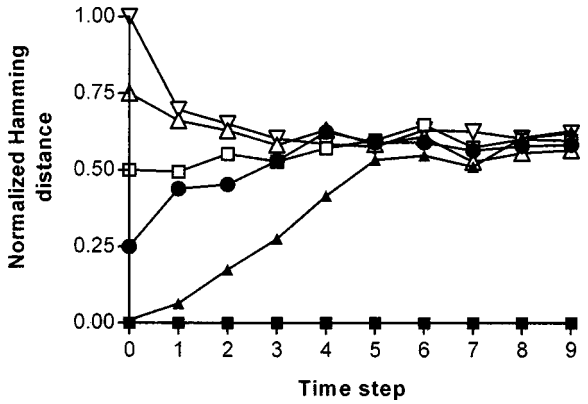

FIG. 6. Before training, external inputs limit recurrent firing to an asymptotic level of randomness although how a simulation reaches this level depends on the amount of $\mathbf{Z}(0)$ randomization. As must be the case, 0% perturbation, where $\mathbf{Z}(0)$ and $\mathbf{Z}'(0)$ are identical, produces no difference in future network state, because network computations are deterministic. However, just perturbing a single neuron at time step zero is enough to produce asymptotic randomness in just a few cycles. For each curve, a single network (with no synaptic modification) was presented twice with the same input sequence; each presentation started with a different initial state [$\mathbf{Z}(0)$ or $\mathbf{Z}'(0)$] but was then followed by the identical external sequence on successive time steps. For each curve, the size of the $\mathbf{Z}(0)$ vs $\mathbf{Z}(0)'$ randomization corresponds to the point plotted at time zero. Each curve plots the normalized Hamming distance between the two firing sequences over nine time steps. Simulations were parametrized as: $N=1024$, $m_e=32$, $\epsilon=0$, $K_I=0.02$, $K_R=0.047\,626\,8$, $K_0=0.777\,933$, $w=0.4$, $c=0.1$. Key: 0% perturbation (■), 1% perturbation (▲), 25% perturbation (●), 50% perturbation (□), 75% perturbation (△), 100% perturbation (∇).

B. Randomness does not disrupt learning

Before training, arbitrary external inputs strongly influence recurrent firing. This is demonstrated in Fig. 6. That is, external inputs influence the path that the firing sequence of the network can follow through the $\{0,1\}^N$ state space. When a single network is twice presented with a simple sequence, the normalized Hamming distance between the two sequences reaches an asymptote at approximately 0.6, regardless of the difference in $\mathbf{Z}(0)$'s. Thus, $\mathbf{Z}(0)$ effects partially dissipate over the course of a single trial. But as a test trial includes fewer time steps of strong external activation, we still need to determine why large $\mathbf{Z}(0)$ randomization does not disrupt testing.

As it turns out, there is less and less sensitivity to variation of $\mathbf{Z}(0)$ as training progresses. In psychological terms, the learned responses to the input sequence become increasingly reliable as training progresses. This reduced sensitivity to initial firing state variation is most simply demonstrated by repeatedly training on one input sequence. Consider a simulation consisting of training on the same nine time step sequence for 125 trials. After each training trial, the simulation is tested twice on the training sequence using two orthogonal $\mathbf{Z}(0)$s. The nine firing states of each pair of test trials were compared, matching time step for time step, and

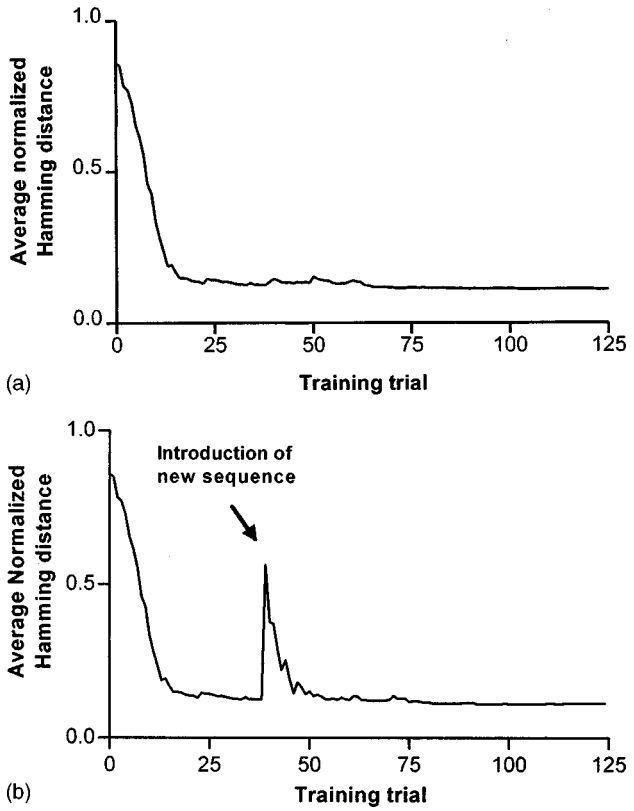

FIG. 7. (a) Sensitivity to $\mathbf{Z}(0)$ randomization declines rapidly as training progresses, but it remains nonzero even after 125 trials. In only 15 training trials, perturbation of firing sequences due to $\mathbf{Z}(0)$ randomization arrives at its approximate minimal value. (b) Introducing a new input sequence during training temporarily increases sensitivity to random perturbation of the initial firing state. The new sequence is fully orthogonal to the first sequence introduced on training trial 39. Both parts use the same general methods. Two test trials accompany each training trial. Normalized Hamming distances quantify each time step of these two tests. Plotted here are the average of the nine Hamming distances for each pair of test trials. The training sequence was nine time steps long, and the set of firing externals shifted by eight neurons on each time step. A different $\mathbf{Z}(0)$ was used for each presentation such that $d_{nH}[\mathbf{Z}(0),\mathbf{Z}'(0)]=1$. Simulation parameters: $N=1024$, $K_0=0.510\,365$, $K_R=0.048\,109$, $K_I=0.02$, $m_e=16$, all other parameters are the same as in Fig. 1.

averaged. The average normalized Hamming distance was 0.86 before training, and dropped to 0.19 by training trial number 15; Fig. 7(a) details the comparison for all training trials. Note the large $\mathbf{Z}(0)$ randomness effect occurring at the beginning of training gives way to a smaller effect as training progresses, and it does so rather quickly. As a result of this growing insensitivity to $\mathbf{Z}(0)$, randomization is no longer disruptive after sufficient training. However, this observation must be further considered in light of the progressive training paradigm, which successively introduces new sequences during training.

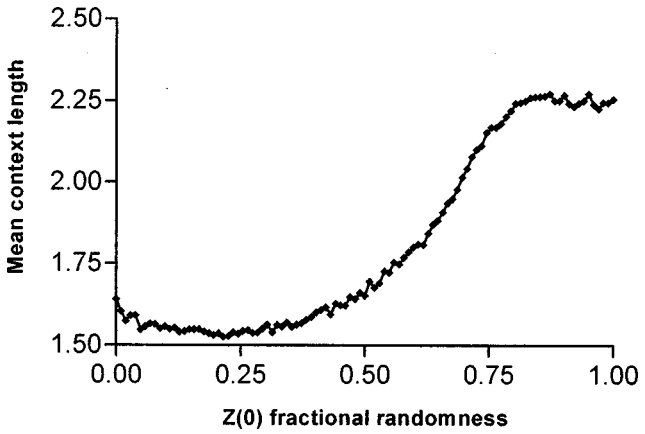

FIG. 8. Randomness of $\mathbf{Z}(0)$ leads to the formation of longer local-context units. Note that the curve shown here is qualitatively similar to the curve in Fig. 1: specifically once fractional randomness of $\mathbf{Z}(0)$ is greater than 50%, mean context length increases dramatically and an asymptote is reached at about 77% randomness. Plotted here is the mean of the context neuron distribution after all 300 training trials, averaged over all 30 simulations used in TP. Because data from each training sequence produced quantitatively similar results, we only show context lengths for the one training sequence, $(AB)(AB)(AB)(a)(a)(a)(+)(+)(+)$. The simulation parameters were the same as for Fig. 1. (See Sec. II E for the definition of local-context neurons.)

C. A novel training sequence increases sensitivity to $Z(0)$ randomization

Although sensitivity to $\mathbf{Z}(0)$ randomization decreases across training on a single sequence, introducing a novel external input sequence during training transiently reverses this trend. Figure 7(b) illustrates the early, quick, decrease in sensitivity to $\mathbf{Z}(0)$ randomization, but more importantly, it illustrates a reversal of this decrease triggered by the introduction of a novel sequence. When the new sequence is first introduced at trial 40, the average normalized Hamming distance between state vectors immediately jumps from 0.13 to 0.56. But just as before, this increased distance drops off quickly ($\sim$15 training trials), reaching roughly the same asymptotically low level of sensitivity as when trained on the initial sequence. Apparently, the novel input sequence pushes the network into a new region of state space, forcing it to use neurons whose synapses have not yet been modified.

D. $Z(0)$ Randomization affects formation of local context neurons

Greater average local context length has been correlated with better performance [1,22]. Here we can study this correlation in a much more direct fashion. By varying $\mathbf{Z}(0)$ as in Fig. 1, we vary average context length as well as performance. In fact, the two measures vary in the same highly nonlinear way as a function of randomization. Comparing Fig. 1 and Fig. 8, we see a striking similarity to the relationship between randomness and performance on the transverse

patterning problem. Both context length and success rate undergo a dramatic jump between fractional randomness 0.5 and 0.75, and level off after fractional randomness 0.75. Thus, there is a strong linear correlation when the two dependent variables are regressed against each other. The linear regression of average context length vs TP performance gives $r^2=0.95$ with a slope of 1.214 and y intercept $=-1.919$.

IV. MODEL DYNAMICS IN RANDOM SEARCH

At the heart of our hippocampal theory is sequence learning as recoding [8]. It is natural to use terms such as recoding or encoding because after training, there is a reliable mapping from a particular set of active external input neurons to a particular set of recurrently activated neurons. Not surprisingly, the codes are highly dependent on the external input sequences presented during training (see Ref. [22]). Understanding how a simulation learns good codes requires an analysis of how the simulation moves through state space during training. Here we begin the development of a quantitative prediction of how $\mathbf{Z}(0)$ affects the state space path of an untrained network.

In Sec. IV A, we discuss effects of $\mathbf{Z}(0)$ randomization on the excitation of neurons relative to threshold. Based upon the ideas outlined in Sec. IV A, Sec. IV B details a quantitative theory of randomization on the first time step as a function of randomization in $\mathbf{Z}(0)$. Both parts use the concepts of excitation distribution and functional threshold.

A. Neuronal excitation before training

Although the excitation received by each individual neuron changes radically across a single trial, the shape of the excitation distribution remains relatively unchanged from time step to time step within a trial. The excitation distribution and the position of the functional threshold offers important insights into the parametric sensitivities of the model [24,32]. On each time step of a simulation, feedback and feedforward inhibition change the amount of recurrent excitation needed to fire a neuron. This fluctuating minimum amount of excitation to fire a neuron is the *functional threshold*. Rewriting Eq. (1), the following formula determines functional threshold when all synaptic weights are equal to w:

$$S=\left\lceil \frac{1}{w}[K_R m(t-1)+K_I m_e+K_0] \right\rceil .$$

Using an untrained simulation presented with a simple twenty-pattern sequence, Fig. 9 locates the functional threshold relative to the excitation distribution where an excitation is the numerator of Eq. (1) for each neuron. Note the large number of neurons whose excitations are very close to the functional threshold. This indicates that many neurons in an untrained network receive either just enough excitation to fire, or receive just under the excitation needed to fire. We claim that these near-threshold neurons are more sensitive to changes in the network firing state on the previous time step than neurons with excitations far from functional threshold.

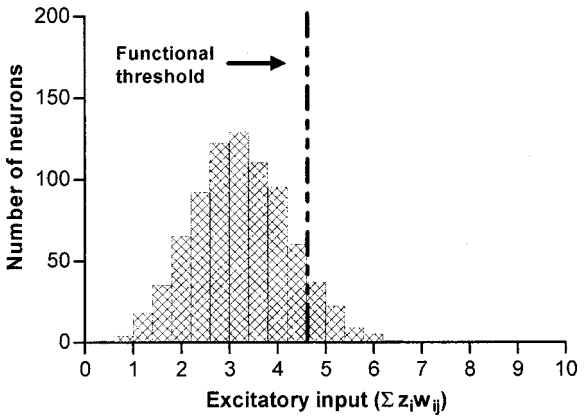

FIG. 9. Before training the excitation distribution is approximately normal with the functional threshold appropriate for the activity level. Neurons to the right of the functional threshold will fire, those to the left will not. Only neurons that will be incorporated into the network's firing at the end of training contribute to the excitation distribution. This distribution was measured from time step 10 of a simple sequence 20 time steps (patterns) long. The shape of the distribution remained unchanged regardless of the time step generating the data. The simulation parameters were the same as those used for Fig. 1.

The idea relevant to $\mathbf{Z}(0)$ is that any small change could push the excitations near the firing threshold to the other side of the threshold. That is, changing the set of neurons that fire on the previous time step leads to large differences in which neurons actually fire on the next time step, although the shape of the excitation distribution remains unchanged. Thus, we hypothesize that those neurons near functional threshold hold the key to understanding $\mathbf{Z}(0)$ effects.

As a simulation is trained on specific sequences, synaptic weights modify in response to those sequences (see, for example, Ref. [33]). As a result, excitation distributions after training differ markedly from excitation distributions before training, particularly near the functional threshold. After training, fewer neurons are in the vicinity of the functional threshold than before training (compare Fig. 10 to Fig. 9). This tendency for neural excitations to move away from the functional threshold makes it more difficult for $\mathbf{Z}(0)$ based randomizations to affect firing after time step one. Thus, the histograms of excitation explain why initial sensitivity drops as training progresses, and they explain why $\mathbf{Z}(0)$ randomness does not harm test performance.

B. Predicting sensitivity to initial conditions before training

In line with the suggestion that $\mathbf{Z}(0)$ randomness perturbs the neurons that have near-threshold internal excitations, we develop a mathematical expression that describes the effect of $\mathbf{Z}(0)$ randomness on neurons as a function of their excitation relative to functional threshold. This theoretical approach confirms the validity of the shuffling around threshold concept as described above, and it makes this hypothesis exact. Specifically, we consider a random process that shuffles histogram occupancies to the left and to the right of the threshold and calculate the general expression that esti-

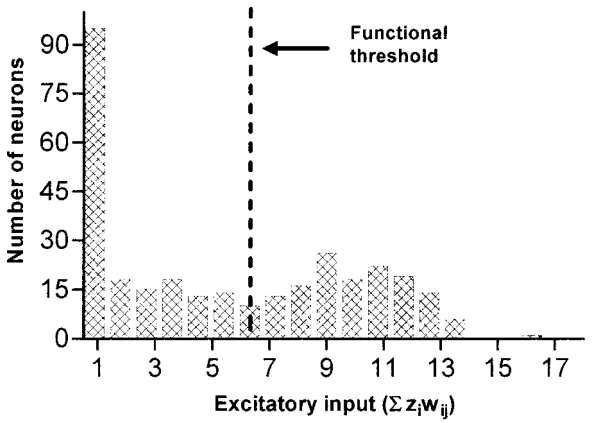

FIG. 10. Training alters the excitation distribution including the number of neurons in the vicinity of functional threshold. After training, fewer neurons are located near the functional threshold than before training (cf. Fig. 9). Combined with the finding that training reduces sensitivity to $\mathbf{Z}(0)$ perturbations, this change in excitation distribution around threshold supports the idea that $\mathbf{Z}(0)$ sensitivity is greatest when many neurons have excitation levels near the functional threshold. The excitation data comes from a training simulation on the $(AB)(AB)(AB)(a)(a)(a)(+)(+)(+)$ sequence (see Methods) after all training is completed. Excitations were recorded from the simulation's response to time step 6 (approximately the middle of the externally driven sequence of inputs). Only neurons that were incorporated into the network's firing at the end of training contribute to the excitation distribution. Simulations were parametrized as in Fig. 1.

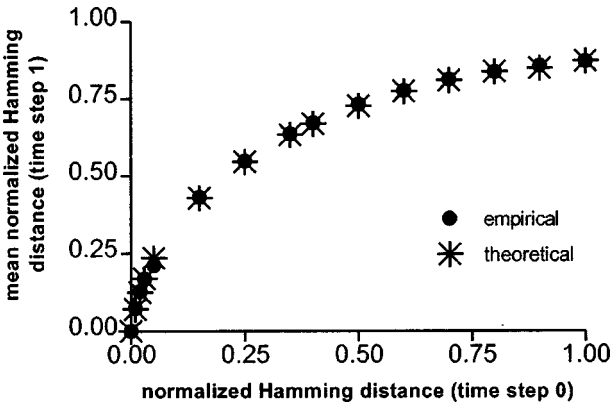

FIG. 11. The theorem based prediction of sensitivity to initial state randomness closely follows empirical observations. Plotted here are pairs of points. One point (*) is the expected sensitivity to initial state randomness produced by the theorem (see Appendix A). The other point (●) of each pair is itself derived from pairs of simulations. This plotted data point is the average of 70 such pairs of simulations. Each set of 70 paired simulations is parametrized with initial state vectors that differ according to the normalized Hamming distance indicated by the *x* axis. As can be deduced by Eq. (B1) in Appendix B, there is an explicit relationship between fractional randomness (defined in Sec. II D) and expected normalized Hamming distance: $r=E(1-a)/(1-a-Ea)$, where r represents fractional randomness, a represents activity, and E represents expected normalized Hamming distance. As described in Sec. II F, sensitivity of initial conditions is quantified in terms of normalized Hamming distance, which measures the distance in state space between temporally matched firing states for a pair of simulations. Each data point plots normalized Hamming distance, measured on time step 1, each pair having a unique randomly generated connectivity. The simulations were parametrized as: $N=8000$, $w=0.4$, $a_0=2.5$, $m_e=20$, $\theta=0.5$, $K_R=0.053\,952\,5$, $K_I=0.01$, $K_0=1.195\,25$.

mates the effect of this shuffling. The calculation predicts $E\{d_{nH}[\mathbf{Z}(1),\mathbf{Z}'(1)\}$ given $d_H[\mathbf{Z}(0),\mathbf{Z}'(0)]$.

The shuffling process is quantified as a two-step process using two functions, L and R. The function L maps an initial state $\mathbf{Z}(0)$ to a "loss" state $\mathbf{Z}^L(0)$ by deactivating Δ active neurons in $\mathbf{Z}(0)$. The function R maps $\mathbf{Z}^L(0)$ to a "reactivated" initial state $\mathbf{Z}'(0)$ by activating Δ inactive neurons in $\mathbf{Z}^L(0)$ with the restriction that none of the Δ newly activated neurons can be one of the neurons deactivated by the L function.

Theory. Specify the set of constants, $\beta=\{N,a_0,c,S,m_e\}$ (see Methods for definition of constants). Then,

$$E\{d_{nH}[\mathbf{Z}(1),\mathbf{Z}'(1)]|\beta,d_H[\mathbf{Z}(0),\mathbf{Z}'(0)]=2\Delta\}$$
$$=\frac{(N-m_e)}{\left((N-m_e)\sum_{h=S}^{Nc}\binom{Na_0}{h}\binom{N-Na_0}{Nc-h}+\binom{N}{Nc}m_e\right)}$$
$$\times\sum_{f=0}^{S-1}\sum_{k=0}^{S-1}\sum_{h=S}^{Nc}\binom{\Delta}{f-k}\binom{N-Na_0-\Delta}{Nc-(h+f-k)}\binom{\Delta}{h-k}$$
$$\times\binom{Na_0-\Delta}{k},$$

where S is the critical number of inputs needed to fire any neuron not fired externally on time step 1.

Appendix A derives this result and shows how a small difference in the vector $\mathbf{Z}(0)$ can quickly lead to divergent network paths through state space. In Fig. 11, the predictions of this theory are compared with empirical observations obtained from simulations. Based on this comparison, it seems we have a quantitatively acceptable theory of the model. Averaged over 70 different randomly connected networks of 8000 neurons running at approximately 1.25% activity, empirical observation of sensitivity to initial conditions tightly follow the theorem's prediction (the theoretical calculation deviates outside the standard error of the mean for only two out of 15 points). Such comparisons are robust across number of neurons and activity level.

V. DISCUSSION

Computational simulations here and elsewhere [5] demonstrate that randomizing the initial state will enhance learning by our hippocampal CA3 model. Simulations with large randomizations of $\mathbf{Z}(0)$ during training are more successful during testing than simulations in which the initial state vector is left constant.

These beneficial effects, which arise from initial state randomness during training, correlate with a greater range of activity states visited by a simulation in the early training

trials. This greater range of the simulation's path through state space across training trials, referred to as ''search,'' is itself correlated with more robust performance which, by definition, must arise from better encodings. Fortunately, as synaptic weights and codewords mature with each trial of training, a simulation grows less and less sensitive to the effects of initial state randomization. As a result, the effect that enhances search does not disrupt testing.

The observed beneficial effects of initial state randomization in our model are analogous, in a certain sense, to the more well-known effects of randomness in simulated annealing [34] (and see also Refs. [35,36] for a possible application corresponding to the biology of the hippocampus). Controlled randomization of movement through state space in simulated annealing improves performance by forcing a system to explore regions of state space outside of a local optimum. Moreover, and similar to observations of Fig. 6, the effect of this randomization in simulated annealing, the distance covered by this exploration through state space lessens over time. However, in simulated annealing, temperature is typically decreased over time by an external control mechanism and a special schedule [37,38]. In contrast, the lessening effect of randomness over time in this model is an emergent property of the model itself, a property which was not explicitly designed into the model.

Based on the observations here, the time of and just after the introduction of a new sequence seems most vital because it is then that the broad search of state space occurs. Introducing a novel training sequence of active external inputs forces the network to use previously unused neurons. During the first few trials after the introduction of a new sequence, the weights of these newly active neurons have not yet stabilized, and relatively many neurons have excitation levels near the functional threshold, as in Fig. 8. It is during this period of larger trial-to-trial variation of neuronal activation that more random variations of the initial state create functionally important fluctuations in the neurons near threshold. The theorem of Sec. IV and Appendix A substantiates how critical the neurons just to either side of threshold are in this process.

This hypothesized expanded search, leading to better sets of coactive neurons, which is to say good codes, is defined further by the observation that randomization of the initial state vector increases the average length of local context neurons. [Average context length has a strong positive linear correlation with TP performance ($r^2=0.95$).] That is, average local context length is a way of quantifying good codes for solving TP. In particular, the existence of such a defined good code with larger average context lengths implies that a simulation has associated temporally distant portions of the training sequence. As a result of these longer-lasting local context neurons, there is a more stable cell-firing bridge to span temporally distant portions of the training sequence. Thus, initial state randomness helps the network create patterns of cell firing that lead to improved sequence prediction.

Our current hypotheses are supported, both qualitatively and quantitatively, through the idea of the sensitivity of the model's firing patterns to initial state randomness. The theorem given in Appendix A shows how differences in the initial state before training quickly affect differences in the model's neuronal firing patterns. Figure 11 provides an empirical demonstration of the theorem's accuracy. By providing a means of quantifying the rate of search through state space as a function of initial conditions, the theorem comprises a critical first step in understanding how the model finds viable firing patterns to recode input sequences.

ACKNOWLEDGMENTS

This work was supported by NIH MH48161 to W.B L., MH57358 to N. Goddard (with Subcontract No. 163194-54321 to W.B L.), by the NDSEG Fellowship Program to A.P.S. and by the Department of Neurosurgery. We also thank Dr. Nancy Desmond for her constructive criticisms of earlier versions of the manuscript.

APPENDIX A: QUANTITATIVE ANALYSIS OF SENSITIVITY TO INITIAL CONDITIONS

Preliminaries: Consider a sparse, randomly connected recurrent neural network described by Eqs. (1) and (2), with fixed parameters N, m_e, w, a_0, K_R, K_I, and K_0. To measure the sensitivity of firing sequences to $\mathbf{Z}(0)$ randomness, two simulations are run with the same connectivity. One simulation is initialized by state vector $\mathbf{Z}(0)$ and the other simulation is initialized by state vector $\mathbf{Z}'(0)$. Each simulation produces a set of firing patterns on the next time step. These are called $\mathbf{Z}(1)$ and $\mathbf{Z}'(1)$, respectively. Here, by definition, the number of active neurons on time step zero is Na_0, or equivalently for these binary neurons,

$$a_0 = P[Z_j(0)=1] = P[Z'_j(0)=1].$$

Define a positive integer Δ, such that,

$$\Delta \equiv \frac{d_H[\mathbf{Z}(0),\mathbf{Z}'(0)]}{2},$$

where d_H is Hamming distance. We will use Δ to represent the number of neurons perturbed from $\mathbf{Z}(0)$ to $\mathbf{Z}'(0)$.

Indicate the probability that a neuron j receives h active inputs as

$$P\left(\sum_{i=1}^{N} c_{ij} z_i = h\right).$$

Further, define S as the minimum number of active inputs needed to fire a neuron in $\mathbf{Z}(1)$ or $\mathbf{Z}'(1)$.

Theorem. Specify the set of constants, $\beta = \{N, a_0, S, c, m_e\}$. Then

$$E\{d_{nH}[\mathbf{Z}(1),\mathbf{Z}'(1)]|\beta,\Delta\}$$
$$= \frac{(N-m_e)}{\left[(N-m_e)\sum_{h=S}^{Nc}\binom{Na_0}{h}\binom{N-Na_0}{Nc-h}+\binom{N}{Nc}m_e\right]} \times \sum_{f=0}^{S-1}\sum_{k=0}^{S-1}\sum_{h=S}^{Nc}\binom{\Delta}{f-k}\binom{N-Na_0-\Delta}{Nc-(h+f-k)}\binom{\Delta}{h-k} \times \binom{Na_0-\Delta}{k}.$$

Note that this theorem is derived from the product of three hypergeometric distributions. To prove the theorem, we offer four supporting lemmas.

Lemma 1. Let S be the critical number of inputs needed to fire a neuron via recurrent inputs on time step 1. Then S is given by

$$S=\left\lceil\frac{K_0+K_R m(t-1)+K_I m_e}{w}\right\rceil.$$

Proof. See Ref. [24].

Lemma 2. Recall the definition of the set β. The probability of a neuron j receiving h active inputs is given by the hypergeometric distribution

$$P\left(\sum_{i=1}^{N} c_{ij}z_j=h\right)=\frac{\binom{Na_0}{h}\binom{N-Na_0}{Nc-h}}{\binom{N}{Nc}}.$$

Proof. A sample of size Nc is to be randomly chosen from N neurons of which Na_0 are active neurons and N-Na_0 are inactive neurons. Let h be the number of neurons in the sample picked that are active. The probability of choosing h active neurons out of a sample of Nc neurons is the hypergeometric given above. Q.E.D.

Perturbation of $Z(0)$: To facilitate discussion of Lemmas 3 and 4, we define two functions, one called L and another called R. These functions allow us to perform a stepwise analysis of the process through which we randomly perturb neurons to create $\mathbf{Z}'(0)$ from $\mathbf{Z}(0)$.

The function L maps an initial state vector $\mathbf{Z}(0)$ to a ''loss'' state vector $\mathbf{Z}^L(0)$ by randomly deactivating Δ active neurons in $\mathbf{Z}(0)$ (i.e., resetting their binary firing states from 1 to 0). We denote $L[\mathbf{Z}(0)]$ as $\mathbf{Z}^L(0)$. $\mathbf{Z}^L(0)$ is a state vector identical to $\mathbf{Z}(0)$, except that Δ active neurons in $\mathbf{Z}(0)$ have been deactivated. Thus,

$$\sum_{j=1}^{N} Z_j^L(0)=\sum_{j=1}^{N} Z_j(0)-\Delta.$$

The function R maps $\mathbf{Z}^L(0)$ to a ''reactivated'' initial state vector $\mathbf{Z}'(0)$ by randomly activating Δ inactive neurons in $\mathbf{Z}^L(0)$, with the restriction that none of the Δ newly activated neurons can be one of the neurons previously deactivated by the L function. Thus, $R[\mathbf{Z}^L(0)]=\mathbf{Z}'(0)$. Since the total activities of $\mathbf{Z}(0)$ and of $\mathbf{Z}'(0)$ are equal, we can write

$$\sum_{j=1}^{N} Z_j'(0)=\sum_{j=1}^{N} Z_j^L(0)+\Delta=\sum_{j=1}^{N} Z_j(0)=Na_0.$$

Lemma 3. Recall the definition of the set β. The probability that a neuron j receives k active inputs when Δ active neurons are removed from $Z(0)$ given that neuron j originally received h active inputs from $Z(0)$ is given by

$$P\left(\sum_{i=1}^{N} c_{ij}Z_i^L=k|\sum_{i=1}^{N} c_{ij}Z_i=h\right)=\frac{\binom{\Delta}{h-k}\binom{Na_0-\Delta}{k}}{\binom{Na_0}{h}}.$$

Proof. The number of ways a neuron j can receive h active inputs is

$$\binom{Na_0}{h}.$$

Of these, the k inputs left active in $\mathbf{Z}^L(0)$ can be chosen, without replacement, from $Na_0-\Delta$ neurons [since $\mathbf{Z}^L(0)$ contains Δ fewer active neurons than does $\mathbf{Z}(0)$] giving

$$\binom{Na_0-\Delta}{k},$$

different possibilities.

But the loss of $h-k$ active inputs can only come from the Δ active neurons removed from $\mathbf{Z}(0)$ to form $\mathbf{Z}^L(0)$ giving

$$\binom{\Delta}{h-k},$$

possibilities.

These last two counts are independent, and thus are multiplied together to get total number of ways Z^L can occur. Q.E.D.

Lemma 4. Recall the definition of the set β. The probability that a neuron j receives f active inputs from $Z'(0)$ given that neuron j received h active inputs from $Z(0)$ and k active inputs from $Z^L(0)$ is given by

$$P\left(\sum_{i=1}^{N} c_{ij}Z_i'=f|\sum_{i=1}^{N} c_{ij}Z_i^L=k,\sum_{i=1}^{N} c_{ij}Z_i=h\right)$$
$$=P\left(\sum_{i=1}^{N} c_{ij}Z_i'-\sum_{i=1}^{N} c_{ij}Z_i=f-k|\right.$$
$$\left.\sum_{i=1}^{N} c_{ij}Z_i^L=k,\sum_{i=1}^{N} c_{ij}Z_i=h\right)$$
$$=\frac{\binom{\Delta}{f-k}\binom{N-Na_0-\Delta}{Nc-(h+f-k)}}{\binom{N-Na_0}{Nc-h}}.$$

Proof. The proof is analogous to that for Lemma 3 except here we must take into account that Na_0 neurons in Z^L cannot be perturbed by the function R, and that h of the Nc inputs cannot be part of the inputs selected by this function. Q.E.D.

Proof of the theorem. For any given $\{\mathbf{Z}(1),\mathbf{Z}'(1)\}$ pair, let us enumerate across all neurons, the effect of the perturbation Δ,

$$n_1=|\{j:Z_j(1)=1,Z'_j(1)=0\}|,$$

$$n_2=|\{j:Z_j(1)=0,Z'_j(1)=1\}|,$$

$$n_3=|\{j:Z_j(1)=1,Z'_j(1)=1\}|,$$

$$n_4=|\{j:Z_j(1)=0,Z'_j(1)=0\}|.$$

These are all the possibilities partitioned, thus $n_1+n_2+n_3+n_4=N$. Because the number of active neurons in $\mathbf{Z}(1)$ and $\mathbf{Z}'(1)$ are by assumption equal, $n_1+n_3=n_2+n_3$, and thus, $n_1=n_2$. But by the standard Hamming distance definition, $d_H[Z(1),Z'(1)]=n_1+n_2$, which gives us a normalized value of

$$d_{nH}[Z(1),Z'(1)]=(n_1+n_2)/2(n_1+n_3)$$
$$=2n_1/2(n_1+n_3)=n_1/(n_1+n_3),$$

which is just the empirical conditional probability $P[Z'_j(1)=0|Z_j(1)=1]$. That is,

$$d_{nH}[Z(1),Z'(1)]=P[Z'_j(1)=0|Z_j(1)=1]. \quad \text{(A1)}$$

The problem is thus reduced to determining the conditional probability in Eq. (A1). Denote the set of all neurons that are fired externally as M, where M has cardinality m_e,

$$P[Z'_j(1)=0|Z_j(1)=1]=P[j\in M,Z'_j(1)=0|Z_j(1)=1]$$
$$+P[j\notin M,Z'_j(1)=0|Z_j(1)=1].$$

Externally fired neurons cannot be turned off, thus

$$=P[j\notin M,Z'_j(1)=0|Z_j(1)=1]=P[Z'_j(1)=0|Z_j(1)=1,j\notin M]P[j\notin M|Z_j(1)=1]$$
$$=\{P[Z^L_j(1)=0,Z'_j(1)=0|Z_j(1)=1,j\notin M]+P[Z^L_j(1)=1,Z'_j(1)=0|Z_j(1)=1,j\notin M]\}.$$

$$P[j\notin M|Z_j(1)=1]=\{P[Z'_j(1)=0,Z^L_j(1)=0|Z_j(1)=1,j\notin M]+0\}P[j\notin M|Z_j(1)=1]$$
$$=P\left(\sum_{i=1}^N c_{ij}Z'_j(1)<S,\sum_{i=1}^N c_{ij}Z^L_j(1)<S|\sum_{i=1}^N c_{ij}Z_j(1)\geqslant S\right)P[j\notin M|Z_j(1)=1]$$
$$=P[j\notin M|Z_j(1)=1]\sum_{f=0}^{S-1}\sum_{k=0}^{S-1}P\left(\sum_{i=1}^N c_{ij}Z'_j(1)=f,\sum_{i=1}^N c_{ij}Z^L_j(1)=k|\sum_{i=1}^N c_{ij}Z_j(1)\geqslant S\right)$$
$$=P[j\notin M|Z_j(1)=1]\sum_{f=0}^{S-1}\sum_{k=0}^{S-1}\frac{P\left(\sum_{i=1}^N c_{ij}Z'_j(1)=f,\sum_{i=1}^N c_{ij}Z^L_j(1)=k,\sum_{i=1}^N c_{ij}Z_j(1)\geqslant S\right)}{P\left(\sum_{i=1}^N c_{ij}Z_j(1)\geqslant S\right)}$$
$$=\frac{P[j\notin M|Z_j(1)=1]}{P[Z_j(1)=1|j\notin M]}\sum_{f=0}^{S-1}\sum_{k=0}^{S-1}\sum_{h=S}^{Nc}P\left(\sum_{i=1}^N c_{ij}Z'_j(1)=f,\sum_{i=1}^N c_{ij}Z^L_j(1)=k,\sum_{i=1}^N c_{ij}Z_j(1)=h\right)$$
$$=\frac{P(j\notin M)}{P[Z_j(1)=1]}\sum_{f=0}^{S-1}\sum_{k=0}^{S-1}\sum_{h=S}^{Nc}P\left(\sum_{i=1}^N c_{ij}Z'_j(1)=f,\sum_{i=1}^N c_{ij}Z^L_j(1)=k,\sum_{i=1}^N c_{ij}Z_j(1)=h\right). \quad \text{(A2)}$$

By the definition of conditional probability, one has

$$P\left(\sum_{i=1}^N c_{ij}Z'_j(1)=f,\sum_{i=1}^N c_{ij}Z^L_j(1)=k,\sum_{i=1}^N c_{ij}Z_j(1)=h\right)$$
$$=P\left(\sum_{i=1}^N c_{ij}Z_j(1)=h\right)P\left(\sum_{i=1}^N c_{ij}Z^L_j(1)=k|\sum_{i=1}^N c_{ij}Z_j(1)=h\right)P\left(\sum_{i=1}^N c_{ij}Z'_j(1)=f|\sum_{i=1}^N c_{ij}Z^L_j(1)=k,\sum_{i=1}^N c_{ij}Z_j(1)=h\right).$$

The three probabilities in the above equation are given by Lemmas 2, 3, and 4, respectively. So, substituting into Eq. (A2), replacing these probabilities with their respective hypergeometrics, simplifying, and taking the expected value over multiple simulations, we have,

$$E\{d_{nH}[Z(1),Z'(1)]\} = \frac{(N-m_e)}{\left((N-m_e)\sum_{h=S}^{Nc}\binom{Na_0}{h}\binom{N-Na_0}{Nc-h}+m_e\binom{N}{Nc}\right)} \times\sum_{f=0}^{S-1}\sum_{k=0}^{S-1}\sum_{h=S}^{Nc}\binom{\Delta}{f-k}\binom{N-Na_0-\Delta}{Nc-(h+f-k)}\binom{\Delta}{h-k} \times\binom{Na_0-\Delta}{k},$$

by construction of those probabilities and the definition of S given by Lemma 1. Q.E.D.

APPENDIX B: RELATING NORMALIZED HAMMING DISTANCE TO FRACTIONAL RANDOMNESS

In order to vary randomness in a systematic and computationally efficient way, the TP simulations used fractional randomness (see Methods), defined as the independent variable r. Despite this, the theory described in Appendix A quantifies $\mathbf{Z}(0)$ randomness in terms of normalized Hamming distance. Fortunately, there is a strong relationship between fractional randomness and the Hamming distance between $\mathbf{Z}(0)$'s used in different trials.

The relationship between fractional randomness, r, and normalized Hamming distance comes from the definitions of fractional randomness and normalized Hamming distance themselves (see Sec. II E and II F). Of particular importance is the procedure that uses fractional randomness to randomly vary $\mathbf{Z}(0)$ from trial to trial in a controlled manner. We will find that the method of fractional randomness provides a good approximate control over trial-to-trial $\mathbf{Z}(0)$ normalized Hamming distance.

Recall that to create $\mathbf{Z}(0)$ with fractional randomness r, one set of active neurons is fixed from trial to trial, and the rest of the active neurons are chosen randomly from the remaining neurons in $\mathbf{Z}(0)$. Define g as a random variable that represents the number of active neurons that could have been randomly perturbed, but were not, and define q as the number of active neurons held fixed (that is, could not be perturbed). Thus, the total number of active neurons (represented by Na_0) is the sum of those that were perturbed (half the Hamming distance), plus those that could have been perturbed but were not (g), plus the ones held fixed (q),

$$Na_0=\frac{d_H[Z(0),Z'(0)]}{2}+g+q.$$

Rearranging this equation in terms of normalized Hamming distance,

$$d_{nH}[Z(0),Z'(0)]=\frac{Na_0-q-g}{Na_0}.$$

We are interested in the average normalized Hamming distance, as a function of g. To find this, note the probability that g neurons are randomly selected to fire in both $\mathbf{Z}(0)$ and $\mathbf{Z}'(0)$,

$$P(g=x)=\frac{\binom{Na_0-q}{x}\binom{N-Na_0}{Na_0-q-x}}{\binom{N-q}{Na_0-q}}.$$

This is a hypergeometric distribution, with expectation

$$E[g]=\frac{(Na_0-q)^2}{(N-q)}.$$

Thus, the expected normalized Hamming distance is

$$E\{d_{nH}[Z(0),Z'(0)]\}=\frac{Na_0-q-\dfrac{(Na_0-q)^2}{(N-q)}}{Na_0}.$$

Recalling the definition of fractional randomness,

$$r=\frac{Na_0-q}{Na_0},$$

which implies

$$q=Na_0-Na_0r.$$

Now, substituting for q produces an expression that determines expected normalized Hamming distance in terms of activity and fractional randomness

$$E\{d_{nH}[Z(0),Z'(0)\} = \frac{Na_0r-\dfrac{(Na_0r)^2}{(N+Na_0r-Na_0)}}{Na_0} = r-\frac{a_0r^2}{(1-a_0+a_0r)}=\frac{r(1-a_0)}{(1-a_0+a_0r)}. \quad \text{(B1)}$$

This difference between average normalized Hamming distance and fractional randomness increases as fractional randomness or activity increases, but not by much. In particular, when fractional randomness is zero, expected normalized Hamming distance is zero, whereas, at fractional randomness of one, expected normalized Hamming distance is $1-a_0$. For example, when fractional randomness$=1$ and $a_0=0.1$ (as in our TP simulations), the average normalized Hamming distance between any two $\mathbf{Z}(0)$'s used in training is 0.9.

[1] X. B. Wu, R. A. Baxter, and W. B. Levy, Biol. Cybern. **74**, 159 (1996).
[2] P. C. Gailey, A. Neiman, J. J. Collins, and F. Moss, Phys. Rev. Lett. **79**, 4701 (1997).
[3] H. Liljenström and X. B. Wu, Int. J. Neural Syst. **6**, 19 (1995).
[4] J. Buhmann and K. Schulten, Biol. Cybern. **56**, 313 (1987).
[5] X. B. Wu and W. B. Levy, Neurocomputing **26-27**, 601 (1999).
[6] M. C. Alvarado and J. W. Rudy, J. Exp. Psychol. **18**, 145 (1992).
[7] J. A. Dusek and H. Eichenbaum, Proc. Natl. Acad. Sci. U.S.A. **94**, 7109 (1997).
[8] W. B. Levy, in *Computational Models of Learning in Simple Neural Systems*, edited by R. D. Hawkins and G. H. Bower (Academic Press, New York, 1989), p. 243.
[9] W. B. Levy, Hippocampus **6**, 579 (1996).
[10] J. J. Hopfield, Proc. Natl. Acad. Sci. U.S.A. **79**, 2554 (1982).
[11] W. B. Levy, in *Computational Neuroscience: Trends in Research*, edited by J. M. Bower (Plenum Press, New York, 1997) pp. 379–383.
[12] M. E. Hasselmo, Prog. Brain Res. **121**, 3 (1999).
[13] G. V. Wallenstein, H. Eichenbaum, and M. Hasselmo, TINS **21**, 317 (1998).
[14] K. I. Blum and L. F. Abbott, Neural Comput. **8**, 85 (1996).
[15] C. V. Buhusi and N. A. Schmajuk, Hippocampus **6**, 621 (1996).
[16] M. R. Mehta, M. C. Quirk, and M. A. Wilson, Neuron **25**, 707 (2000).
[17] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, Psychol. Rev. **102**, 419 (1995).
[18] E. T. Rolls and A. Treves, *Neural Networks and Brain Function* (Oxford University Press, Oxford, 1998).
[19] W. B. Levy and O. Steward, Brain Res. **175**, 233 (1979).
[20] W. B. Levy, in *Proceedings of the Fourth Annual Conference of Cognitive Science Society* (L. Erlbaum, Hillsdale, NJ, 1982), p. 135.
[21] A. A. Minai and W. B. Levy, in *Proceedings of the INNS World Congress on Neural Networks* (L. Erlbaum, Hillsdale, NJ, 1993), pp. II–505.
[22] X. B. Wu, J. Tyrcha, and W. B. Levy, Biol. Cybern. **79**, 203 (1998).
[23] J. O'Keefe and L. Nadel, *The Hippocampus as a Cognitive Map* (Clarendon Press, Oxford, 1978).
[24] A. C. Smith, X. B. Wu, and W. B. Levy, Network Comput. Neural Syst. **11**, 63 (2000).
[25] S. Polyn and W. B. Levy, Neurocomputing **38-40**, 823 (2001).
[26] W. B. Levy and O. Steward, Neuroscience **8**, 791 (1983).
[27] W. B. Levy, C. M. Colbert, and N. L. Desmond, in *Neuroscience and Connectionist Models*. edited by M. A. Gluck and D. E. Rumelhart (Lawrence Erlbaum, Hillsdale, NJ, 1990), p. 187.
[28] K. W. Spence, Psychol. Rev. **59**, 89 (1952).
[29] J. W. Rudy, J. P. Keith, and K. Georgen, Dev. Psychobiol. **26**, 171 (1993).
[30] A. P. Shon, X. B. Wu, and W. B. Levy, Neurocomputing **32-33**, 995 (2000).
[31] W. B. Levy, X. B. Wu, and R. A. Baxter, Int. J. Neural Syst. **6**, 71 (1995).
[32] A. A. Minai and W. B. Levy, Biol. Cybern. **70**, 177 (1993).
[33] A. Amarasingham and W. B. Levy, Neural Comput. **10**, 25 (1997).
[34] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, Science **220**, 671 (1983).
[35] G. V. Wallenstein and M. E. Hasselmo, J. Neurophysiol. **78**, 393 (1997).
[36] V. A. Sohal and M. E. Hasselmo, Neural Comput. **10**, 869 (1998).
[37] M. D. Huang, F. Romeo, and A. Sangiovanni-Vincentelli, in *Proceedings of the IEEE International Conference on Computer Aided Design—Digest of Technical Papers, Santa Clara, CA* (IEEE, Princeton, NJ, 1986), p. 381.
[38] S. Geman and D. Geman, IEEE Trans. Pattern Anal. Mach. Intell. **6**, 721 (1984).