On the information in spike timing: neural codes derived from polychronous groups

Zhinus Marzi, João Hespanha and Upamanyu Madhow

Department of Electrical and Computer Engineering

University of California, Santa Barbara

Email: {zh_marzi, hespanha, madhow}@ece.ucsb.edu

Abstract—There is growing evidence regarding the importance of spike timing in neural information processing, with even a small number of spikes carrying information, but computational models lag significantly behind those for rate coding. Experimental evidence on neuronal behavior is consistent with the dynamical and state dependent behavior provided by recurrent connections. This motivates the minimalistic abstraction investigated in this paper, aimed at providing insight into information encoding in spike timing via recurrent connections. We employ information-theoretic techniques for a simple reservoir model which encodes input spatiotemporal patterns into a sparse neural code, translating the polychronous groups introduced by Izhikevich into codewords on which we can perform standard vector operations. We show that the distance properties of the code are similar to those for (optimal) random codes. In particular, the code meets benchmarks associated with both linear classification and capacity, with the latter scaling exponentially with reservoir size.

I. INTRODUCTION

Classical models in computational neuroscience are based on rate coding. The existing state of the art in machine learning (ranging from perceptrons to CNNs and RNNs) also implicitly models rate coding, with the real value at the input or output of a neuron serving as an abstraction of a firing rate. While there is significant experimental evidence and considerable speculation on the importance of spike timing, computational models are far less established. In this paper, we propose and investigate an idealized model in order to derive insight into the neural codes that can be constructed from spike timing information.

Sensory information (e.g., in visual and auditory stimuli) are inherently embedded into spatiotemporal patterns. Experimental evidence in [1], [2] suggests that neurons show dynamical behavior even when the stimulus is fixed: [1] shows that the cat primary visual cortex acts like a fading memory whose current activity contain as much information from previous frame's content as the current frame, while [2] shows that the network responds differently to odor B when it is proceeded by odor A. This is consistent with the dynamical and state dependent behavior provided by recurrent connections. Thus, recurrent connections, which are the key aspect of the model considered here, are strongly neuro-plausible candidates for cortical modeling, since they bring about temporal context by providing state-dependent dynamics.

Our starting point is a key observation by Izhikevich more than a decade ago [3]. He observed that experiments show a great deal of variability in axonal delays, which results in an interesting phenomenon that he terms *polychronization*. Figure 1 illustrates the effect of axonal delays: Neurons A and B provide inputs to Neuron C through axons with delays 3 ms and 5 ms, respectively. Suppose that Neuron C acts as a coincidence detector, firing when spikes from Neurons A and B arrive at the same time (within the same 1 ms bin, in our discrete time model). Given the difference in axonal delays, this happens when the spike from Neuron A is launched 2 ms after the spike from Neuron B. When we have a network of neurons connected via variable delay axons, interesting patterns of firings in space (i.e., across neurons) and time, which Izhikevich terms *polychronous groups,* result from individual spikes initiated at a small number of neurons.
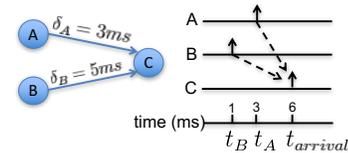

Fig. 1: Polychronization

In this paper, we ask whether polychronization can provide the basis for a concrete computational framework. In the tradition of information theory, we consider the simplest possible model that captures the basic features of the problem. We consider a set of K input neurons connected to a reservoir of N neurons, where $N \gg K$, as depicted in Figure 2(a) (the model is discussed in more detail in Section II). The axonal delays from input neurons to reservoir neurons, and those between reservoir neurons, are randomly chosen from one of T consecutive values. Each reservoir neuron is a coincidence detector, firing when at least m spikes from pre-synaptic neurons fall in the same time bin. For each input spike pattern, there is a corresponding space-time pattern in the reservoir, which we may interpret as an encoding of the input pattern. Figure 2(b) shows an example reservoir response to a particular input pattern.

Thus, we have encoded an input space-time pattern into an output space-time pattern in a higher-dimensional space. The dimension increase from input to output spaces should separate out the patterns from each other more than at the input, and hence provides a "channel code." The information embedded in the reservoir's space-time pattern can be read
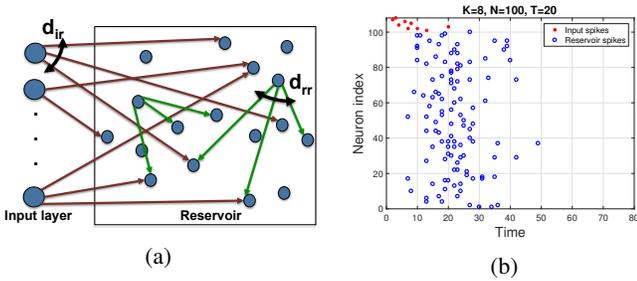
Fig. 2: (a) System model, (b) Spatiotemporal code

out with various degrees of sophistication. In this paper, we show that even the *simplest* approach in which we integrate the number of spikes for each neuron over a relatively large time horizon (effectively converting the rich dynamics within the reservoir into a rate code) provides a powerful neural code that scales "well" with reservoir size N. This readout mechanism maps the space-time input patterns to spatial output patterns by integrating the outputs of the reservoir neurons over predefined intervals. The resulting spatial codeword is a real-valued vector of dimension N. We discuss the properties of these spatial codes in this paper. We are interested in how these codes scale with reservoir size N, and our main results are summarized as follows:

• Our analysis reveals that to achieve good properties in terms of linear separability and memorization capacity, the output degree of the input neurons d_{ir}, should scale linearly with $N\sqrt{T}/K$, while the output degree of the reservoir neurons d_{rr}, can remain constant as we scale the reservoir size.

• We have verified the above scaling laws in terms of the ability of the neural network to encode spatio-temporal patterns to facilitate classification using a linear classifier. Specifically, we have shown that the network placement of codewords enable it to achieve fundamental benchmarks of linear separability established by Cover [4].

• We have also verified that, under the above scaling law, the weight and distance properties of the spatial codes scale with N in a manner similar to that of random codes. This enable us to argue that the number of patterns that can be reliably distinguished scales exponentially with N.

We observe that, for our simple readout model, small perturbations in the input pattern (e.g., the change in the timing of a single spike in one of the neurons) lead to large changes in the output. Thus, the system is well-matched to one-shot learning and memorization (i.e., training and testing using the same patterns).

II. SYSTEM MODEL AND PROPERTIES

We now describe the system model depicted in Figure 2(a) in more detail, and discuss some basic properties and parameter choices. As a minimal idealization of spike timing information, we consider K input neurons, each emitting a single spike at a time chosen from $\{1, ..., T\}$. Thus, there are T^K possible input patterns, corresponding to $K \log_2 T$ bits of information. The input can therefore be represented as a $K \times T$ binary array $\mathbf{U}$

with the (k, t)th entry $u_{kt} = 1$ if input neuron k spikes at time t and $u_{kt} = 0$ otherwise, where $k = 1, ..., K$, $t = 1, ..., T$.

We are interested in encoding each input spatiotemporal pattern into the responses of N reservoir neurons. Each input neuron is connected to d_{ir} reservoir neurons, chosen at random. The reservoir neurons are internally connected to each other, with each reservoir neuron providing input to d_{rr} other reservoir neurons, chosen at random. Thus, the average in-degree of a reservoir neuron is $\frac{K}{N} d_{ir} + d_{rr}$. For simplicity, all synapses are taken to be excitatory, but the role of inhibitory synapses in shaping neural codes in our setting is certainly of great interest for future work.

A reservoir neuron fires if and only if at least m incoming spikes arrive in a given time slot. We set $m = 2$ for all of our numerical results.

We also make a drastic abstraction of spike timing dependent plasticity (STDP) in our model. Specifically, we assume that, in the learning phase, all synapses are of equal strength, so that m incoming spikes that line up in time suffice to make a neuron fire; this enables "instant learning" of input patterns. After the learning phase is over, we may set "unused" incoming synapses (i.e., those for which the output neuron has never fired in the slot directly after the arrival of a spike on the synapse) to zero. Incorporating more sophisticated STDP mechanisms compatible with continual learning and forgetting into our model are a subject of future work.

A. Neural coding

While the input patterns span times 1 through T, the recurrent connections in the reservoir imply that the response to an input may last for more than T time units. In principle, due to reverberant effects, the duration of spikes from reservoir neurons could be indefinitely long, but in practice, we can capture most of the relevant information within a finite horizon that we denote as T_{hz}. In our simulations, we employ $T_{\mathrm{hz}} = 4T$.

The reservoir "space-time" response is a $N \times T_{\mathrm{hz}}$ array $\mathbf{S}$ with (j, t)th entry $s_{jt} = 1$ if reservoir neuron j spikes at time t, and $s_{jt} = 0$ otherwise, where $j = 1, ..., N$, $t = 1, ..., T_{\mathrm{hz}}$. In order to translate the reservoir spatiotemporal information to a purely spatial domain, we count the firings of the reservoir neurons for the duration of the horizon T_{hz}, which results in an N-dimensional codeword vector $\mathbf{X}$ with its jth entry given by

$$x_j = \sum_{t=1}^{T_{\mathrm{hz}}} s_{jt} , \quad j = 1, ..., N$$

From a computational viewpoint, the mapping from input pattern $\mathbf{U}$ to $\mathbf{X}$ is a "hash" from temporal coding to rate coding, resulting in a neural code in a familiar vector space which we can study with conventional techniques. Since this opens the path for efficient classical computation and learning, most of our discussion in this paper is devoted to spatial coding. From the standpoint of neuro-plausibility, there is intriguing experimental evidence regarding the tuning of synaptic integration within cortex [5].

B. Network Parameters

In order for the neural code to properly utilize the available dimensions as we scale N, and to have the desirable characteristics discussed in Section III, we must ensure that a large fraction of the reservoir neurons are stimulated by each input pattern. Firings in reservoir neurons occur due to direct stimulation from the input, as well as due to the recurrent internal connections, but an excessive number of firings due to internal connections lead, we have found, to less discrimination across the neural codewords. We therefore do not scale the internal degrees in the reservoir with N, and scale the degrees of the input nodes to ensure sufficient stimulation.

The firing probability of a reservoir neuron in direct response to the *input* spikes is equivalent to the *coincidence probability* of spike arrival times denoted as $t_{\text{arrival}} = t_i + \delta_{\mathbf{i}}, \quad i \in \{1, 2, ..., I\}$ (Figure 1), where I is the number of input synapses. In our model, input spikes (t_i) and axonal delays (δ_i) are independent random variables with distribution $\mathcal{U}[1, T]$. Hence, t_{arrival} has a symmetric triangular distribution over the range $[2, 2T]$. The coincidence probability for uniform random variables is described by the well known *birthday (collision) problem* [6]: the probability of having at least one coincidence among n realizations of a random varible distributed as $\mathcal{U}[1, d]$. This probability can be approximated by [7]

$$p(n; d) = 1 - e^{\frac{-n(n-1)}{2d}} \tag{1}$$

The distribution of interest to us is triangular rather than uniform, but the coincidence probability for any finite range distribution is bounded below by that of a uniform distribution over the same range [8]. Thus, the coincidence probability of random variable $t_{\text{arrival}} \sim \Delta[2, 2T]$ for spikes coming through I input synapses is lower bounded by

$$p(I; 2T - 1) = 1 - e^{\frac{-I(I-1)}{2(2T-1)}} \tag{2}$$

Using this expression leads to the rule of thumb that the reservoir in-degree should scale as $\Omega(\sqrt{T})$ [1], in order to maintain a constant firing probability. Since synapses coming from input layer are uniformly connected to reservoir neurons, I is the sum of K Bernoulli random variables, with mean $E(I) = \frac{K}{N} d_{ir}$. Hence, d_{ir} should scale as $\Omega(\frac{N}{K}\sqrt{T})$ in order to let the input spikes propagate through the reservoir.

We consider two regimes for scaling of K and N. In the first regime (Section III-A), we fix K and let N get large, so that d_{ir} scales as $\Omega(N)$. In the second regime (Section III-B), we increase K and N with $K.N$ fixed, so that d_{ir} scales as $\Omega(1)$.

C. Chaotic Mapping

Figure 3 indicates that the mapping from input patterns to both the space-time and spatial codes is chaotic, in that the output Hamming distances and Euclidean distances can be large even when the input Hamming distance is small (while the distances for input hamming distance 2 are smaller than the

---

[1]We say that $f = \Omega(g)$ if there exist constant $C > 0$ such that $C < |f/g|$

---

rest, but there is a substantial fraction of large distances even there). This sensitivity may be a drawback for generalization, but it is a positive feature for one-shot learning or pattern memorization. Of course, if we think of our discrete time model as approximating a continuous-time system with a given time resolution, the response will not change abruptly under timing perturbations that are smaller than the resolution.
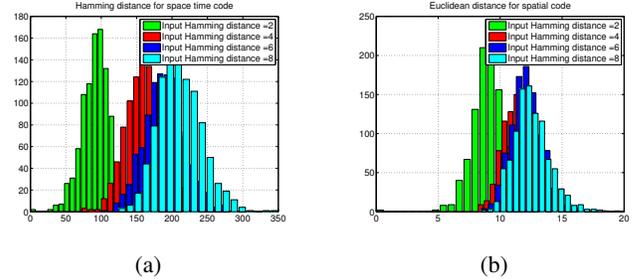

Fig. 3: Histogram of (a) Hamming distance of space-time codes (b) Euclidean distance of spatial codes, for different input hamming distances.

III. Code characteristics

A. Linear Separability

From a computational perspective, one can view the reservoir neurons as mapping the spatio-temporal stimuli applied to the input neurons into spatial output codewords. In the context of our model, this means that the neural network is a (nonlinear) map ϕ from an input space of K-dimensional vectors (each entry representing a firing time for the corresponding input neuron) into an output space of N-dimensional vectors (each entry representing the number of firing for the corresponding reservoir neuron).

The problem of constructing optimal maps from a finite set of input patterns to an N-dimensional output "feature" space is addressed in a classic paper by Cover [4]. Cover considers a classification problem where a set of P patterns represented by features in $\mathbb{R}^d$ are given binary labels and shows that the fraction ρ of the total set of 2^P possible labels can be classified with a linear classifier is given by

$$\rho \leqslant \rho_{\max}(P, d) := \frac{\sum_{k=0}^{d} \binom{P-1}{k}}{2^{P-1}}, \tag{3}$$

with equality achieved when the P patterns are in "general positions in $\mathbb{R}^d$". By this, we mean that every subset of d patterns correspond to linearly independent vectors in $\mathbb{R}^d$. Note that Cover's result in [4] only considers equality in (3), but his proof can easily be adapted to show that points not in general positions do not increase the fraction of linearly classifiable labels.

An important feature of the function $\rho_{\max}(P, d)$ in (3) is that it exhibits a steep change from values close to 1 for $P < 1.5d$ to values close to 0 for $P > 2.5d$, as can be seen in Figure 4, where $\rho_{\max}(P, d)$ is ploted as a function of P/d (for $d = N$).

Since $\rho_{\max}(P, d)$ is a monotonically increasing function of d, the preceding result shows that one can perform a richer

class of classification tasks by increasing the dimension d of the feature space. In fact, one can classify essentially any labeling of P patterns represented by features in $\mathbb{R}^d$, provided that $d > P/1.5$. However, this is only true if we can map our original P patterns into the high dimensional space $\mathbb{R}^d$, *while keeping the high-dimensional features in general positions.* This excludes, for example, any linear transformation, which would necessarily introduce non-general positions when the original patterns lie in $\mathbb{R}^K$ with $K \geqslant d$.

Figure 4 shows that our neural network provides an essentially optimal nonlinear mapping of the input spatiotemporal patterns to spatial output patterns in $\mathbb{R}^N$, in that it maps the input patterns to general positions in $\mathbb{R}^N$. This enables linear classification in the output space $\mathbb{R}^N$ for the largest possible fraction $\rho_{\max}(P, N)$ of possible labels. In order to obtain this figure, $P = 100$ input patterns are randomly generated and labeled into two classes. These patterns are propagated through the spiking network, and we then determine whether or not the patterns can be accurately classified by a SVM classifier. Figures 4a and 4b compare the fraction of patterns/labels that can be linearly classified with the upper bound provided by $\rho_{\max}(P, N)$ (black lines) for several values of the parameters T and K, respectively. For comparison, we also include the fraction of patterns/labels that can be linearly classified for a set of random feature vectors in $\mathbb{R}^N$ with iid Gaussian entries (green lines), which are in general positions in $\mathbb{R}^N$.

Figure 4c contains a plot similar to that in Figure 4a, but instead of presenting curves for several values of T, we fix T and consider several values of the input-to-reservoir degree d_{ir}. This second figure shows that there is a threshold for d_{ir} below which the mapping is not effective. Specifically, we see that when each input neuron is connected to at least 50% of the reservoir neurons, the neural network essentially achieves optimal separability and can linearly classify essentially any labeling of P patterns, provided that $N > P/1.5$. When each input neuron is connected to only 30% of the reservoir neurons, we see some degradation on performance, but the network can still classify most labelings, provided that $N > P/1.25$. However, when the connectivity is down to 20%, the fraction of patterns that can be linearly classified becomes exceedingly small.

Thus, while our neural network produces a good code with sufficient connectivity, there is a threshold below which it is no longer effective. This is well predicted with the approximation for reservoir neuron firing probability (due to the input alone) given by (2): the approximations are 0.01, 0.04 and 0.14 for 20%, 30% and 50% connectivity, respectively. The corresponding numbers via simulations are 0.03, 0.08 and 0.2, respectively, which yield similar qualitative conclusions. A detailed analysis of the specific threshold required to trigger "enough" activity within the reservoir (which should depend on the reservoir's internal connectivity) is an interesting topic for future work.

The discussion above shows that, under an appropriate input-to-reservoir connectivity, the neural network is almost optimal in terms of maximizing the probability of linear classification.

Perhaps not surprisingly, we can show that it also performs well in terms of achieving a small classification error rate for the patterns that are not linearly separable in the output space $\mathbb{R}^N$. To demonstrate this, we present in Figure 5 the classification error rates obtained for the same experiments used to construct Figure 4. We can see that in all the cases for which we obtained optimality in terms of the probability of linear classification, the error rates match those that would be obtained by a set of random feature vectors in $\mathbb{R}^N$ with iid Gaussian entries.

B. Capacity

The linear separability results of the preceding section show that the proposed system produces neural codewords in "general position," similar to those produced by random codes. Of course, the number of linearly separable codewords scales linearly with N. We now remove the constraint of linear separability, and ask whether the number of patterns we can reliably distinguish scales exponentially with N, formulating it as a communication problem as follows:

• There are T^K possible input patterns, or messages, corresponding to $K \log_2 T$ bits of information.
• Each pattern is encoded into a vector of dimension N using our spatial code, hence our *code rate* is $\alpha = \frac{K \log_2 T}{N}$ bits/dimension.
• If we increase code dimension N while keeping code rate α fixed, we have $2^{\alpha N}$ codewords, corresponding to exponential scaling of the memory. Can we do this while keeping the codewords "well separated"?

In information-theoretic terms, "well separated" means that, when a codeword is perturbed by noise or other impairments (determined by a channel model), and we use maximum likelihood decoding, then the probability of error (i.e., decoding into the wrong codeword) tends to zero as N gets large. For any given input pattern, and corresponding codeword $\mathbf{X}$, the probability of maximum likelihood decoding to a different codeword $\mathbf{Y}$ can be bounded by $e^{-\beta \|\mathbf{X} - \mathbf{Y}\|^2}$ for standard noise models such as Gaussian and Poisson. As we show shortly, the pairwise distance squared between codewords grows linearly with N, concentrating around its mean (Figure 6(a)). Thus, a union bound on the error probability, conditioned on the correct codeword being $\mathbf{X}$, is given by

$$P_e \leqslant \sum_{\mathbf{Y} \neq \mathbf{X}} e^{-\beta \|\mathbf{X} - \mathbf{Y}\|^2} \sim 2^{\alpha N} e^{-\beta \gamma N} \qquad (4)$$

assuming that $\|\mathbf{X} - \mathbf{Y}\|^2$ concentrates around γN. Thus, the error probability decays to zero if $\alpha \ln 2 < \beta \gamma$.

Figure 6(a) gives simulation results showing both the mean and variance of squared distances scale linearly with N. We can now apply Chebyshev's inequality to infer that the squared distances indeed concentrate around their mean, which implies that, for an appropriate choice of parameters, we can make error probability tend to zero.

Beyond the basic concentration result, it is also worth checking how much our neural codes "look like" random codes with i.i.d. components. Suppose that we have two N-dimensional
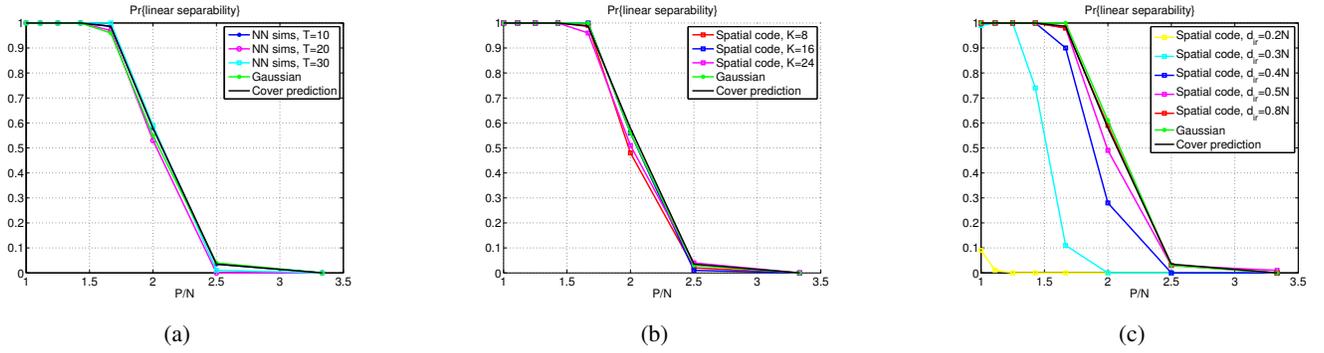
(a)   (b)   (c)

Fig. 4: Spatial coding attains optimum separability over a wide range of network parameters. Default network settings are $T = 20$, $K = 8$, $d_{ir} = 0.8N$ and $d_{rr} = 4$.
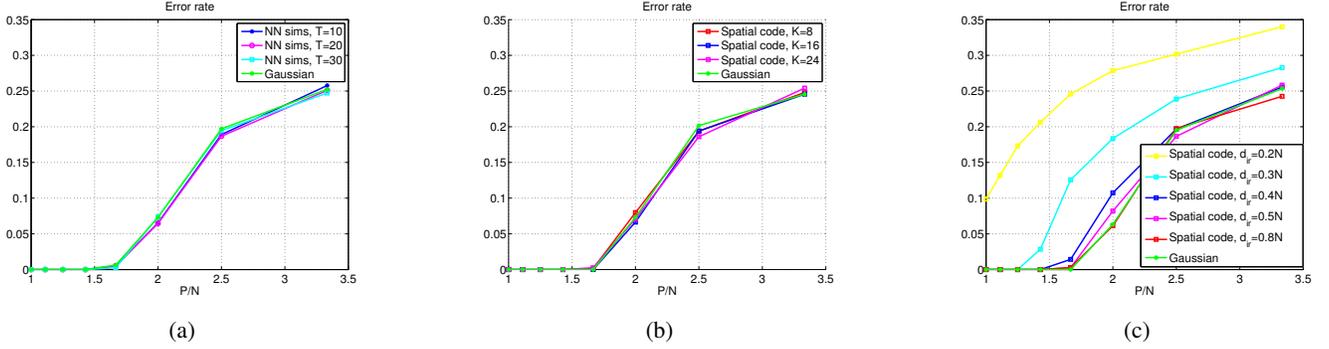

(a)   (b)   (c)

Fig. 5: Spatial coding's error rate matches with random coding. Default network settings are $T = 20$, $K = 8$, $d_{ir} = 0.8N$ and $d_{rr} = 4$.

codewords $\mathbf{X} = (X_1, ..., X_N)$ and $\mathbf{Y} = (Y_1, ..., Y_N)$ from a random code. If the components are i.i.d. and independent across codewords, the mean and variance of the distance squared will clearly scale linearly with N. In addition, however, $(\|\mathbf{X} - \mathbf{Y}\|^2 - E[\|\mathbf{X} - \mathbf{Y}\|^2])/\sqrt{N \mathrm{std}(X_1 - Y_1)^2}$ tends to a standard Gaussian by the central limit theorem. Our neural code also appears to exhibit this property: Figure 6(b) shows that the histograms of distance-squared exhibit small Gaussian-like deviations around the mean.

We note from figure 6(a) that the normalized distance of the codewords, which can be interpreted as robustness of the code against noise, is not a monotonic function of code rate. To explain this behavior, first note that in our simulations, $E(I) = K d_{ir}/N \propto \alpha$, which implies that the firing probability of the reservoir neurons is increasing with code rate α. For very small α ($\alpha = 0.5$), not enough neurons fire and for very large α ($\alpha = 8$), most of the neurons fire most of the time. Both of these scenarios lead to less discrimination among codewords. Thus, there exists a sweet spot for code rate (e.g., $\alpha = 4$) that provides the most discrimination by adjusting input-to-reservoir connectivity.

IV. RELATED WORK

Other prior models trying to utilize spike timing include the tempotron [9], which attempts binary classification of different spike timing patterns using a perceptron-like structure, the
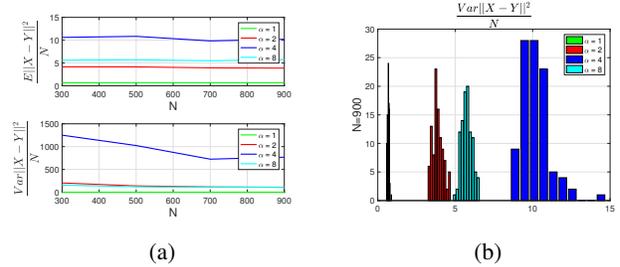

(a)   (b)

Fig. 6: (a) Mean and variance of distance squared scale linearly with N, (b) Distance squared concentrates around its mean, and exhibits Gaussian-like deviations from it (consistent with the central limit theorem). Default network settings are $d_{ir} = \sqrt{T} \log T$, $T = 20$ and $d_{rr} = 4$.

chronotron [10], which attempts to train towards a desired output spike timing pattern. Possible mechanisms for realizing such machines are discussed in [11]. There is a rich history and continued research in experimental neuroscience showing the importance of spike timing; see [12], [13], [14], as well as references in [3], [9], [10]. The chaotic nature of the mappings induced by spike timing has been pointed out in [15], where these observations are interpreted to cast doubt on whether spike timing provides robust enough signals. Our model, which is an abstraction of well-accepted spiking neural network

models [16], embraces the chaotic mappings resulting from spike timing, and shows that these can produce good "channel codes" which could provide the basis for a robust memory. Note that Izhikevich also makes an interesting case for the role of polychronous groups in working memory [17]. Our abstraction is rich enough to produce polychronous groups as in [3], without requiring the detailed models for continuous-time neural dynamics used there.

Prior attempts at abstraction include Howard *et al* [18], who consider a network of coincidence detectors similar to ours, but detailed insight into the neural codes generated by such networks is not provided. In terms of computational models based on polychronous groups, prior work has focused on having at least some synapses with adaptive delays. Paugham-Moisy *et al* [19] consider supervised learning with a reservoir computing model similar to ours, but with a crucial difference: they allow adjustment of the delays from the reservoir neurons to the readout. This can, in principle, compensate for the chaotic nature of the map from the input to the reservoir, but adapting delays has drawbacks from the point of view of both computational neuroscience (questionable neuroplausibility) and neuromorphic design (difficulty of implementation). Izhikevich and Hoppenstead [20] propose a "polychronous wavefront computation" model in which transponders fire under a suitable coincidence between wavefronts initiated by spikes at other transponders, but in order to implement specific function, the location of the transponders (or, in effect, the delays of synapses connecting them) must be programmed. To the best of our knowledge, ours is the first work to conduct a detailed examination of the neural codes obtained from polychronous groups.

Our model falls within the general framework of reservoir computing, since it has a reservoir with fixed properties, followed by a readout mechanism that has design flexibility. However, it differs fundamentally from existing reservoir computing models such as liquid state machines and echo state networks [21], [22]. These are, in essence, fixed nonlinear filters that map the input state to an internal state that is read out after an adaptive linear transformation (adapted to track a target output). However, the nonlinear mappings in such systems are smooth, unlike the chaotic maps associated with coincidence detection and variable delays in our system.

V. Conclusions

The new reservoir computing model proposed here enables translation of information from spike timing into "good", "random like" neural codes in standard vector space. It is worth exploring architectural variants such as layered reservoirs, or reservoirs with geometric connectivity constraints. The specific readout mechanism we propose is simple, but there could be a number of other ways of going from the binary space-time code defined by the firing patterns of the reservoir neurons to a vector spatial code. The role of inhibitory connections in shaping the code is also an interesting topic. While we provide a basic analysis of the neural code, it is of interest to explore its robustness under noise models specifically related to

timing: for example, random Poisson spikes at input or within the reservoir can produce reverberations that are nonlinearly coupled to the input pattern.

It is interesting to note that spike timing plays a critical role in a number of neuromorphic hardware designs, in the form of the address event representation (AER): analog operations and thresholding encode information in spikes, and the spatiotemporal locations of the spikes are fed to processing and control units via a digital bus. First proposed in the early 1990s [23], [24], AER has been used to build "silicon retinas" by a number of research groups [25], [26], [27]. Our work raises the question of whether it might be possible to replace explicit digital encoding of the space-time location of spikes, and subsequent digital processing, by more power-efficient analog embeddings and processing.

References

[1] D. Nikolić, S. Haeusler, W. Singer, and W. Maass, "Temporal dynamics of information content carried by neurons in the primary visual cortex," in *Advances in neural information processing systems*, 2007, pp. 1041–1048.

[2] B. M. Broome, V. Jayaraman, and G. Laurent, "Encoding and decoding of overlapping odor sequences," *Neuron*, vol. 51, no. 4, pp. 467–482, 2006.

[3] E. M. Izhikevich, "Polychronization: computation with spikes," *Neural Computation*, vol. 18, no. 2, pp. 245–282, 2006.

[4] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE transactions on electronic computers*, no. 3, pp. 326–334, 1965.

[5] D. L. F. Garden *et al.*, "Tuning of synaptic integration in the medial entorhinal cortex to the organization of grid cell firing fields," *Neuron*, vol. 60, pp. 875–889, 2008.

[6] M. Gardner, "Mathematical games," *Scientific American*, vol. 222, no. 6, pp. 132–140, 1970.

[7] M. Arnold and W. Glaß, "Simple approximation formulas for the birthday problem," *American Mathematical Monthly*, vol. 120, no. 7, pp. 645–648, 2013.

[8] M. L. Clevenson and W. Watkins, "Majorization and the birthday inequality," *Mathematics Magazine*, vol. 64, no. 3, pp. 183–188, 1991.

[9] R. Gutig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing-based decisions," *Nature Neuroscience*, vol. 9, no. 3, March 2006.

[10] R. V. Florian, "The chronotron: A neuron that learns to fire temporally precise spike patterns," *PLoS ONE*, vol. 7, no. 8, 2012.

[11] C. Albers, M. Westkott, and K. Pawelzik, "Perfect associative learning with spike-timing dependent plasticity," in *Advances in Neural Information Processing Systems*. MIT Press, 2013.

[12] R. VanRullen, R. Guyonneau, and S. J. Thorpe, "Spike times make sense," *Trends in Neurosciences*, vol. 28, no. 1, January 2005.

[13] J. D. Rolston, D. A. Wagenaar, and S. M. Potter, "Precisely timed spatiotemporal patterns of neural activity in dissociated cortical cultures," *Neuroscience*, vol. 148, pp. 294–303, 2007.

[14] I. Birznieks and R. M. Vickery, "Spike timing matters in novel neuronal code involved in vibrotactile frequency perception," *Current Biology*, 2017.

[15] A. Banerjee and A. Pouget, "Dynamical constraints on computing with spike timing in the cortex," in *Advances in Neural Information Processing Systems*. MIT Press, 2003.

[16] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[17] B. Szatmary and E. M. Izhikevich, "Spike-timing theory of working memory," *PLoS Computational Biology*, vol. 6, no. 8, 2010.

[18] M. Howard *et al.*, "Further explorations of a minimal polychronous memory," in *ICAI 2010*, 2010.

[19] H. Paugam-Moisy, R. Martinez, and S. Bengio, "Delay learning and polychronization for reservoir computing," *Neurocomputing*, vol. 71, pp. 1143–1158, 2008.

[20] E. M. Izhikevich and F. C. Hoppenstead, "Polychronous wavefront computations," *International Journal of Bifurcation and Chaos*, vol. 19, no. 5, pp. 1733–1739, 2009.

[21] W. Maass, T. Natschlager, and H. Markram, "Real-time computing without stable states: a new framework for neural computation based on perturbations," *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[22] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks," German National Research Center for Information Technology, Tech. Rep. GMD report 148, 2001.

[23] M. Sivilotti, "Wiring considerations in analog vlsi systems, with application to field-programmable networks," Ph.D. dissertation, California Institute of Technology, Pasadena, CA, 1991.

[24] M. Mahowald, "Vlsi analogs of neuronal visual processing: A synthesis of form and function," Ph.D. dissertation, California Institute of Technology, Pasadena, CA, 1992.

[25] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 5, pp. 416–434, May 2000.

[26] R. Serrano-Gotarredona *et al.*, "Aer building blocks for multi-layer multi-chip neuromorphic vision systems," in *Advances in Neural Information Processing Systems*. MIT Press, 2005.

[27] J. A. L. nero Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco, "A five-decade dynamic-range ambient-light-independent calibrated signed-spatial-contrast aer retina with 0.1-ms latency and optional time-to-first-spike mode," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 10, pp. 2632–2643, Oct 2010.