Robust computation with rhythmic spike patterns

E. Paxon Frady[a,1] and Friedrich T. Sommer[a,1]

[a]Redwood Center for Theoretical Neuroscience, University of California, Berkeley, CA 94720

Edited by Gyorgy Buzsáki, NYU Neuroscience Institute, New York, NY, and approved July 18, 2019 (received for review February 13, 2019)

Information coding by precise timing of spikes can be faster and more energy efficient than traditional rate coding. However, spike-timing codes are often brittle, which has limited their use in theoretical neuroscience and computing applications. Here, we propose a type of attractor neural network in complex state space and show how it can be leveraged to construct spiking neural networks with robust computational properties through a phase-to-timing mapping. Building on Hebbian neural associative memories, like Hopfield networks, we first propose threshold phasor associative memory (TPAM) networks. Complex phasor patterns whose components can assume continuous-valued phase angles and binary magnitudes can be stored and retrieved as stable fixed points in the network dynamics. TPAM achieves high memory capacity when storing sparse phasor patterns, and we derive the energy function that governs its fixed-point attractor dynamics. Second, we construct 2 spiking neural networks to approximate the complex algebraic computations in TPAM, a reductionist model with resonate-and-fire neurons and a biologically plausible network of integrate-and-fire neurons with synaptic delays and recurrently connected inhibitory interneurons. The fixed points of TPAM correspond to stable periodic states of precisely timed spiking activity that are robust to perturbation. The link established between rhythmic firing patterns and complex attractor dynamics has implications for the interpretation of spike patterns seen in neuroscience and can serve as a framework for computation in emerging neuromorphic devices.

spiking neural network | associative memory | oscillations | phasor networks | phase-to-timing

The predominant view held in neuroscience today is that the activity of neurons in the brain and the function of neural circuits can be understood in terms of the computations that underlie perception, motion control, decision making, and cognitive reasoning. However, recent developments in experimental neuroscience have exposed critical holes in our theoretical understanding of how neural dynamics relates to computation. For example, in the prominent existing theories for neural computation, such as energy-based attractor networks (1, 2), population coding (3), and the neural-engineering framework (4), information is represented by analog variables that are typically related to the firing rates of spiking neurons in the real brain. Properties of rate-coding (5–7) are hard to reconcile with experimental observations, such as precise sequences of action potentials (8–11) and computation during perception involving only few spikes per neuron (12), as well as spike-timing codes using the phase of neural oscillations, as in refs. 13 and 14. Thus, theories that go beyond rate-coding and that can elucidate the functions of spike timing and rhythmic activity patterns are required.

Here, we develop such a theory based on complex-valued neural networks and a "phase-to-timing" mapping that translates a complex neural state to the timing of a spike. We first introduce a model of fixed-point attractor networks in complex state space, called "threshold phasor associative memory" (TPAM) networks. The model dynamics is governed by a Lyapunov function, akin to Hopfield networks (15); it has high memory capacity and, unlike Hopfield networks, can store continuous-valued data. Second, we propose spiking implementations of TPAM, one as direct as possible, and one more biologically plausible. Our framework can be used to design circuits of spiking neurons to compute robustly with spike times, potentially trailblazing a path toward fully leveraging recent high-performance neuromorphic computing hardware (16). Concurrently, the framework can help neuroscientists understand the computational purpose of experimental observations, such as sequential and rhythmic firing dynamics, balance between excitation and inhibition, and synaptic delays.

Background

Fixed-point attractor models and, more generally, energy-based models (17) have played an eminent role in the fields of neural networks and theoretical neuroscience. The appeal of these models comes from the fact that their dynamics can be described by the descent of an energy or Lyapunov function, often conceptualized as an "energy landscape." While energy descent does not describe all of neural computation, it enables important types of computations, such as optimization (18) and denoising/error correction, an essential ingredient for making neural computation robust and reliable (15).

The Landscape of Fixed-Point Attractor Neural Networks. Here, we focus on a prominent class of fixed-point attractor networks with Hebbian one-shot learning to store a set of neural activity patterns. To retrieve a pattern, the network is first initialized with a cue—typically, a noisy or partial version of a stored pattern. For retrieval, an iterative dynamics successively reduces the initial noise and converges to a clean version of the stored memory. The models can be distinguished along the following dimensions: real-valued versus a complex-valued neural state space; neurons

Significance

This work makes 2 contributions. First, we present a neural network model of associative memory that stores and retrieves sparse patterns of complex variables. This network can store analog information as fixed-point attractors in the complex domain; it is governed by an energy function and has increased memory capacity compared to early models. Second, we translate complex attractor networks into spiking networks, where the timing of the spike indicates the phase of a complex number. We show that complex fixed points correspond to stable periodic spike patterns. It is demonstrated that such networks can be constructed with resonate-and-fire or integrate-and-fire neurons with biologically plausible mechanisms and be used for robust computations, such as image retrieval.

Author contributions: E.P.F. and F.T.S. designed research; E.P.F. and F.T.S. performed research; E.P.F. contributed new reagents/analytic tools; and E.P.F. and F.T.S. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence may be addressed. Email: epaxon@berkeley.edu or fsommer@berkeley.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1902653116/-/DCSupplemental.

Published online August 20, 2019.

with discretizing versus continuous transfer functions; and the neural firing patterns in the network being sparse versus dense (Fig. 1).

The traditional models of attractor neural networks, which have a symmetric synaptic matrix that guarantees Lyapunov dynamics and fixed points (25), were inspired by the Ising model of ferromagnetism (26). The neural transfer function is step-shaped (describing the spiking versus silent state of a neuron), and the models can store dense activity patterns with even ratio between active and silent neurons (15, 19). The inclusion of "thermal" noise yielded networks with a sigmoid-shaped neural-transfer function (1, 23, 27), which produces continuous real-valued state vectors that could be conceptually related to firing rates of spiking neurons in the brain. Further, following the insight of Willshaw et al. (28), attractor networks for storing sparse activity patterns were proposed, which better matched biology and exhibited greatly enhanced memory capacity (21, 29–31).

One drawback of these traditional memory models is that all attractor states are (close to) binary patterns, where each neuron is either almost silent or fully active near saturation. The restriction to binary memories can be overcome by introducing model neurons that can saturate at multiple (more than 2) activation levels (22, 32–34). This class of models was inspired by the Potts glass model in solid-state physics. Another model with multilevel neurons is the so-called "complex Hopfield network" (20, 35–42). Here, the model neurons are discretized phasors, and, as a result, the states of the model are complex vectors whose components have unity norm, and phase angles chosen from a finite, equidistant set. For a discretization number of 2, complex Hopfield networks degenerate to the real-valued bipolar Hopfield network (15).

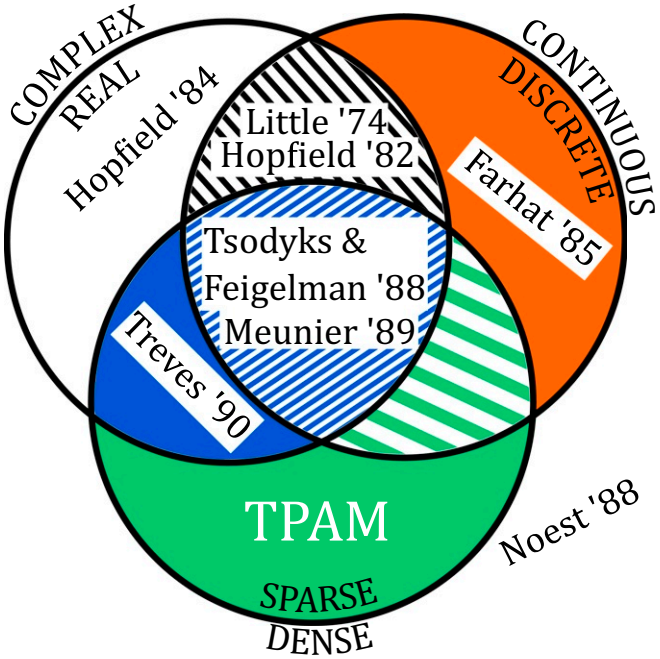

Fig. 1. Venn diagram delineating different types of fixed-point attractor networks. The boundaries represent the following distinctions: *1*) complex-valued versus real-valued neural states; *2*) continuous-valued versus discrete-valued neural states; and *3*) dense versus sparse neural activity patterns. Regions are labeled by key publications describing these models: Little '74, ref. 19; Hopfield '82, ref. 15; Hopfield '84, ref. 1; Farhat '85, ref. 20; Tsodyks & Feigelman '88, ref. 21; Meunier '89, ref. 22; Treves '90, ref. 23; and Noest '88, ref. 24. Our research focuses on the uncharted areas, sparse complex-valued models (TPAM).

A unique strength of a complex state space was highlighted by "phasor networks" (24, 43, 44). In phasor networks, the neural transfer function is a phasor projection—i.e., each neural state carries the continuous phase value of the postsynaptic sum, but has normalized magnitude. Interestingly, even without phase discretization, phasor networks can store arbitrary continuous-valued phasor patterns. Patterns with arbitrary relative phase angles can be stable because of the ring topology of normalized complex numbers.

In existing phasor networks and complex Hopfield networks, all neurons represent phases at every time step. To provide models of greater computational versatility, here, we explore relatively uncharted territory in Fig. 1: attractor networks with complex-valued sparse activity states. These models can also store phasor patterns, in which a fraction of components have zero amplitude that correspond to silent or inactive neurons. Specifically, we introduce and investigate an attractor network model called the TPAM network. As will be shown, pattern sparsity in TPAM enables high memory capacity—as in real-valued models (21)—and also corresponds to spike-timing patterns that are neurobiologically plausible.

Hebbian Sequence Associative Memories. A first foray into temporal neural coding was the development of networks of threshold neurons with Hebbian-type heteroassociative learning that synaptically stores the first-order Markovian transitions of sequential neural activity patterns (45–51). When initialized at or near the first pattern of a stored sequence, the parallel-update and discrete time dynamics of the network will produce the entire sequence, with a pattern transition occurring each time step. These networks have nonsymmetric synaptic matrices, and, therefore, the dynamics are not governed by a Lyapunov function (25). However, for networks that store cyclic pattern sequences of equal lengths, Herz et al. (52, 53) have shown that the network dynamics are governed by an energy function, defined in an extended state space in which each entire sequence is represented by a point.

We will show that sequence-associative networks are related to phasor memory networks, whose fixed points are phase patterns with equidistantly binned phase values (each phase bin representing 1 position in the sequence), yet with a different learning rule and algebra from TPAM networks.

Results

TPAM. We propose a memory model, the TPAM, which can store sparse patterns of complex phasors as fixed-point attractors. The network dynamics is governed by an energy function, which we derive. Further, simulation experiments show that TPAM has high memory capacity and provides an efficient error-correction stage in a neural network for storing images.

Learning and Neural Dynamics. Phasor neural networks (24, 43) were designed to store dense phase-angle patterns in which every component represents a phase angle. Similar to auto-associative outer-product Hebbian learning (54), like in the standard Hopfield network (15), phasor networks employ a complex-conjugate outer-product learning rule:

$$\mathbf{W} = \mathbf{S}\mathbf{S}^{*\top}\ ,\ W_{ij} = \sum_{m=1}^{M} S^r_{im} S^r_{jm}\, e^{\mathrm{i}(S^\phi_{im} - S^\phi_{jm})}, \qquad [1]$$

where $\mathbf{S} \in \mathbb{C}^{N\times M}$ is a matrix of M phasor patterns of dimension N, and $\mathbf{S}^*$ denotes the complex conjugate. A component of one of the stored patterns is given by $S_{im} = S^r_{im}\, e^{\mathrm{i}S^\phi_{im}}$. The entries along the diagonal of $\mathbf{W}$ are set to 0.

In each time step, the complex neural states of the phasor network are multiplied by the complex weight matrix to give the postsynaptic dendritic sum:

$$u_i(t) = \sum_j W_{ij} z_j(t). \quad [2]$$

The neural update is parallel, and the transfer function is a phasor projection, with the complex value of each neuron set to unit magnitude and preserving the phase angle of $u(t)$: $z_i(t+1) = u_i(t)/|u_i(t)|$.

In contrast to the described phasor memory network, the TPAM network is designed for storing patterns in which only a sparse fraction of components $p_{hot} = K/N$ have unit magnitude, and the rest have zero amplitude—i.e., are inactive—with K the number of neurons active in a single pattern and N the total number of neurons. TPAM uses the same learning rule [**1**] and postsynaptic summation [**2**] as the original phasor network, but differs in the neural-transfer function. The neural-transfer function includes a threshold operation on the amplitude of the synaptic sum [**2**]:

$$z_i(t+1) = g(u_i(t), \Theta(t)) := \frac{u_i(t)}{|u_i(t)|} H(|u_i(t)| - \Theta(t)), \quad [3]$$

with $H(x)$ the Heaviside function. If the threshold $\Theta(t)$ is met, the output preserves the phase of the sum vector and normalizes the amplitude. Otherwise, the output is zero.

To maintain a given level of network activation, the threshold setting needs to be controlled as a function of the global network activity (55). Here, we set the threshold proportional to the overall activity:

$$\Theta(t) = \theta \sum_i |z_i(t)| = \theta |\mathbf{z}(t)|, \quad [4]$$

with θ a scalar between 0 and 1, typically slightly less than 1.

The memory recall in TPAM with $N = 400$ neurons is demonstrated in Fig. 2. The network has stored $M = 100$ sparse random phasor patterns with $p_{hot} = 10\%$ and phase values drawn independently from a uniform distribution. The iterative recall is initialized by a partial memory pattern—with some nonzero components set to zero (Fig. 2, *Upper*) and with a superposition of several stored patterns (Fig. 2, *Lower*). In both cases, the network dynamics relaxes to one of the stored memories (approximately).

Energy Function of TPAM Networks. For traditional phasor memory networks (without threshold), Noest (24) showed that the corresponding Lyapunov function is

$$E(\mathbf{z}) = -\frac{1}{2} \sum_{ij} W_{ij} z_i z_j^*. \quad [5]$$

Note that, because [**1**] results in a Hermitian matrix $\mathbf{W}$, [**5**] is a real-valued function. Further note that the dynamics in phasor networks is a generalization of phase-coupled systems well studied in physics, such as the Kuramoto model (56) and the XY model (57), and for describing coherent activity in neural networks (58–60). Those models are governed by a Lyapunov function of the form [**5**], but in which $\mathbf{W}$ is real-valued and symmetric (61).

To see how the inclusion of the threshold operation in the TPAM update [**3**] changes the Lyapunov function, we follow the treatment in ref. 1 by extending [**5**] to describe the dynamics of phasor networks with arbitrary invertible transfer function $f(z)$:

$$E(\mathbf{z}) = -\frac{1}{2} \sum_{ij} W_{ij} z_i z_j^* + \sum_i \int_0^{|z_i|} f^{-1}(v) dv. \quad [6]$$

The neural-transfer function of TPAM, $g(z;\Theta)$ in [**3**], is not invertible. But it can be approximated by a smooth, invertible function by replacing the Heaviside function in [**3**] with an invertible function $f(z)$—for example, the logistic function. In the limit of making the approximation tight—i.e., $f(z) \approx g(z;\Theta)$—the corresponding update is given by [**3**]. For a constant global threshold $\Theta = \Theta(t)$, the Lyapunov function [**6**] of TPAM is:

$$E(\mathbf{z}) = -\frac{1}{2} \sum_{ij} W_{ij} z_i z_j^* + \Theta \|\mathbf{z}\|_1, \quad [7]$$

with $\|\mathbf{z}\|_1$ the L_1 norm of the vector, the sum of its components' amplitudes. According to Eq. **7**, a positive constant global threshold [**3**] has the effect of adding a L_1 constraint term, which encourages a lower activity in the network.

For the dynamic threshold control [**4**], the Lyapunov function for TPAM becomes

$$E(\mathbf{z}) = \sum_{ij} \left(-\frac{1}{2} W_{ij} + \theta \mathbf{I} \right) z_i z_j^*, \quad [8]$$

with $\mathbf{I}$ the identity matrix. According to Eq. **8**, a positive coefficient θ in the dynamic threshold control, [**3**] and [**4**], adds a repulsive self-interaction between active phasors, thereby reducing the activity in the network.

The derived Lyapunov functions help to clarify the difference between constant and linear threshold control. Consider the case of low memory load. With constant threshold, not only are the individual stored patterns stable fixed points, but also their superpositions will be stable. In contrast, dynamic threshold control introduces competition between active stored memory patterns. The coefficient θ can be tuned so that only individual patterns are stable (as done here). When lowered, superpositions of 2 (or more) patterns can become stable, but competition still only allows a limited number of active superpositions. This may be useful behavior for applications outside the scope of this paper.

Information Capacity of TPAM Networks. To understand the function of TPAM, the impact of its different features on memory performance was studied through simulation experiments. After storing M random patterns, we initialized the network to one of the stored patterns with a small amount of noise. The network ran until convergence or for a maximum of 500 iterations. To assess the quality of memory recall, we then compared the network state with the errorless stored pattern.

Fig. 3*A* displays on the y axes "cosine similarity" (i.e., correlation) between the output of the memory and the desired target

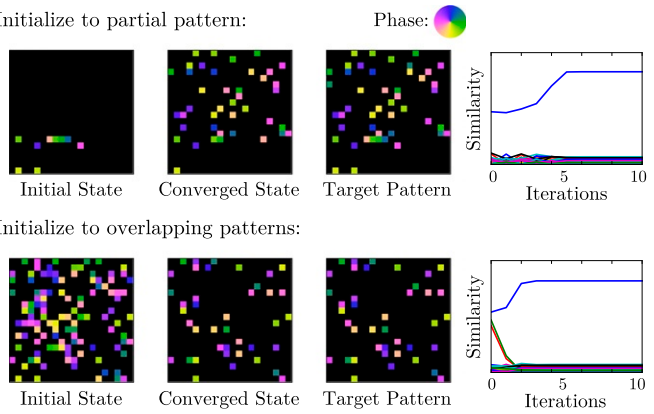

Fig. 2. Memory recall in a TPAM network. Results of 2 retrieval experiments, 1 initialized by a partial memory pattern (*Upper*) and 1 by a superposition of 3 memory patterns (*Lower*), are shown. Both recalls were successful, as indicated by the similarity between "converged" and "target" patterns (phase values are color coded; black corresponds to zero amplitude). Images on the right show that it takes only a few iteration steps until only the overlap with the target memory is high (blue lines).

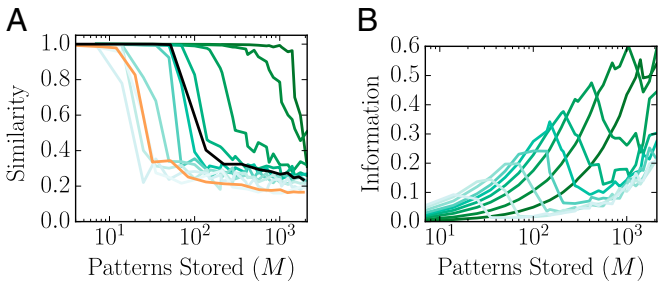

Fig. 3. Capacity of TPAM networks. (*A*) Recall performance of TPAM as a function of stored patterns (green; darker green indicates higher sparsity, over range $p_{hot} = [0.02, 0.05, 0.10, 0.15, 0.2, 0.25, 0.35, 0.5, 0.75, 1.0]$), in comparison with traditional associative memory models: bipolar Hopfield networks (black) (15) and continuous phasor networks (orange) (24). Similarity is the Pearson correlation between converged network state and target state. (*B*) The memory capacity of the TPAM network in bits per synapse (sparsity encoded by shading, as in *A*).

pattern. This normalized metric accounts for both disparity in the phase offset and mismatch in the supports, but does not directly reflect the mutual information between patterns, which also depends on the sparsity level. Fig. 3*A* compares TPAM with different levels of sparsity (green) to the traditional binary Hopfield network (15) (black line) and to the continuous phasor network (24) (orange line). As in the case of the ternary models (22) (*SI Appendix*), the number of patterns that can be recalled with high precision increases significantly with sparsity.

To assess how the memory efficiency of TPAM depends on pattern sparsity, we empirically measured the information in the random phase patterns that can be recalled from the memory (*SI Appendix*). Dividing the recalled information by the number of synapses yielded the "memory capacity" of a network in bits per synapse (30, 31). Measuring the memory capacity in bits, rather than by the number of stored patterns (15), has the advantage that performances can be compared for memory patterns with different sparsity levels.

The measurements of memory capacity in bits/synapse showed that sparsity greatly increased memory capacity (Fig. 3*B*) over dense associative memory networks. Interestingly, this result parallels the increase of memory capacity in binary Hopfield networks with pattern sparsity (21, 29, 30). This holds up to a limit, however, as the networks with the highest sparsity levels had a slightly decreased maximum memory capacity in the simulation experiments. The slow secondary increase of the information capacity is due to a residual of small amounts of unusable information across many patterns in a very-low-fidelity regime (*SI Appendix*).

Indexing and Retrieving Data with TPAM Networks. One option to storing real-world data in TPAM networks is to encode the data in phasor patterns that can be stored in a recurrent TPAM network, as described in the last section. A problem with this approach is that data correlations cause interference in the stored patterns, which is a known issue in traditional associative memories with the outer-product learning rule that reduces the information capacity quite drastically (2).

Here, we explored the ability of TPAM to perform error correction of random indexing patterns within a memory network inspired by the sparse distributed memory (SDM) model (62). The original SDM model consists of 2 feedforward layers of neurons: an indexing stage with random synaptic weights, mapping data points to sparse binary index vectors; and a heteroassociative memory, mapping index vectors back to data points. Our memory architecture deviated from the original SDM model in 3 regards. First, it used complex-valued index patterns. Second, synaptic weights in the indexing stage were learned from the data. Third, it consisted of an additional third stage, an error-correction stage using a recurrent TPAM, that sits between the indexing stage and heteroassociative memory (similar to ref. 40; Fig. 4*A*).

In the following, we denote the data matrix as $\mathbf{P} \in \mathbb{R}^{D \times M}$, where D is the dimensionality of the data and M is the number of data points. As in previous sections, the matrix of index vectors (randomly chosen sparse phasor patterns) is $\mathbf{S} \in \mathbb{C}^{N \times M}$, where N is the number of neurons in the TPAM network.

The indexing stage maps incoming data vectors into index patterns. It is a feedforward network of neurons with the synaptic matrix $\mathbf{W}^I \in \mathbb{C}^{N \times D}$. A simple Hebbian learning rule for heteroassociation is $\tilde{\mathbf{W}}^I = \mathbf{S}\mathbf{P}^\top$. However, to reduce the level of indexing errors due to inherent data correlations, we used learning that involves the pseudoinverse (PINV) of the data, $\mathbf{P}^+ = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^\top$, resulting from the singular value decomposition $\mathbf{P} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Specifically, the synapses in the indexing stage are formed according to:

$$\mathbf{W}^I = \mathbf{S}\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^\top = \mathbf{S}\mathbf{P}^+. \quad [9]$$

This linear transform performs "pattern separation" by amplifying the differences between correlated data vectors. It thereby produces decorrelated projections from the data space to the index space.

The output stage is a heteroassociative memory of the data. This is a feedforward network using simple Hebbian learning for mapping an index pattern back into a data vector. To produce

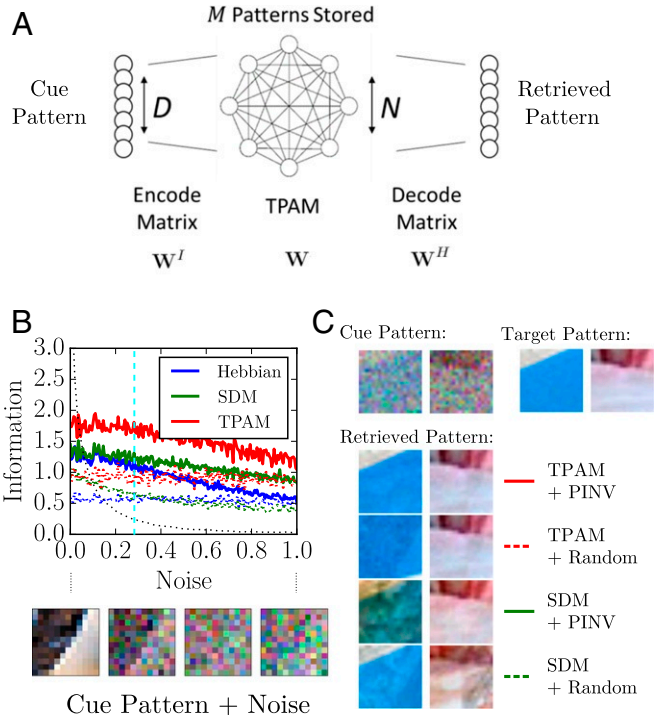

Fig. 4. Storage of natural images in a 3-layer architecture with complex neurons. (*A*) Layered network architecture, consisting of 3 stages: indexing stage with pattern separation, TPAM for error correction, and heteroassociative memory. (*B*) Comparision of the performances of the full architecture versus other heteroassociative memory models. The pattern-separation stage for indexing (solid lines) and the TPAM network for error correction significantly increase the retrieved information (bits per pixel). The information in the noisy cue pattern is plotted for comparision (dashed black line). (*C*) Examples of retrieved images for different models. Cue pattern noise is 0.3 (cyan line in *B*).

real-valued output patterns, the output neural-transfer function projects each component to the real part.

In SDM and heteroassociative memories in general, if the indexing or cue patterns are noisy, the quality of the returned data suffers significantly. To improve the retrieval quality in these cases, we stored the indexing patterns **S** in TPAM according to [**1**]. The TPAM performed error correction on the patterns produced by the indexing stage, and corrected index patterns were decoded by the heteroassociative memory stage.

Empirical comparisons of image storage and retrieval using a simple Hebbian heteroassociative memory, an SDM, and the full network with pattern separation and TPAM for error correction (Fig. 4*B*; see *SI Appendix* for implementation details) were performed with simulation experiments. The simple Hebbian model and the SDM were also extended by incorporating pattern separation in the indexing stage (solid lines in Fig. 4*B* include pattern separation). We stored $M = 20$ image patches of $D = 12 \times 12 \times 3$ pixels into the networks and measured the Pearson correlation ρ ("Similarity") of the retrieved pattern with the true pattern, given a noisy input cue. We computed the total information per pixel as $I^H = -\frac{1}{2}\log_2(1-\rho^2)$ (*SI Appendix*). The full network returned a larger amount of information about the stored data than the simpler models. Errors in retrieved TPAM patterns (Fig. 4*C*) were due to spurious local minima, which are usually superpositions of stored memories. Similarly, errors in SDM were spurious activations of incorrect patterns, leading to readout errors also as superpositions of stored memories. Including the PINV for indexing improves the likelihood of avoiding such superposition errors.

Relating TPAM Networks to Spiking Neural Networks. Here, we exploit a natural link between a complex state and a spike raster through a phase-to-timing mapping. We describe 2 models with spiking neurons that perform the key computations in TPAM: complex synaptic multiplication, summation of complex postsynaptic signals [**2**], and the neural-transfer function [**3**].

Phase-to-Timing Mapping. Each component of a complex state vector in TPAM, $z_i(t) =: z_i^r(t)e^{\mathrm{i}z_i^\phi(t)}$, can be uniquely mapped to a spike pattern in a population of neurons through a phase-to-timing mapping. Specifically, in the context of a cycle with period T, the timing of a spike $s_i^{(t)}$ on a continuous time axis s represents the phase angle $z_i^\phi(t)$, nominally with $s_i^{(t)} = T(z_i^\phi(t)/2\pi + t)$. Phasor values with magnitude zero are represented by silence (Fig. 5). Thus, a fixed-point state in TPAM corresponds to a limit cycle of precisely timed spiking activity, where neurons fire with a period of T or are silent.

With this mapping, we can construct spiking neural networks evolving in continuous time that perform the operations of TPAM. The complex multiplication between presynaptic input and synaptic weight requires the addition of phases. The phase-to-timing mapping also applies to the complex-valued synapses, $W_{ij} =: W_{ij}^r e^{\mathrm{i}W_{ij}^\phi}$, in which synaptic phase-shift translates to synaptic delay by $\zeta_{ij} = T(W_{ij}^\phi/2\pi)$. The synaptic delay adds with the spike timing, which computes the complex product.

In TPAM with time-discrete parallel update scheme, a state $z(t)$ depends exclusively on the previous state $z(t-1)$. The spiking network, however, evolves in continuous time, and there is no special demarcation between time steps. To match the time-discrete update in the spiking network, one could hypothetically construct appropriate synaptic delays by adding or subtracting multiples of T based on the demarcation of the zero phase (Fig. 5). This can be done for 1 pattern, but if a neuron is participating in more than 1 pattern, then it is not possible to guarantee that all fixed points perfectly mirror the discrete update. Neurons will potentially have influence within the same time step or 2 time steps later. Importantly, however, at the fixed points where every spike train is T-periodic, altering the delays by T does not change the postsynaptic coincidence structure. Therefore, the continuous time dynamics is similar to the discrete time dynamics near fixed points, but will, in general, not be exactly the same.

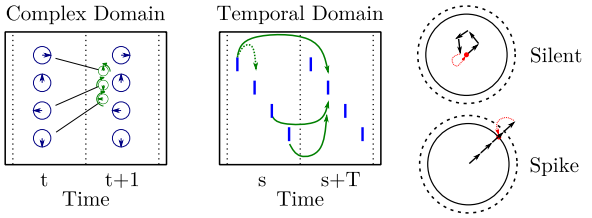

Fig. 5. Mapping states and synapses of TPAM to spiking networks. The complex TPAM state vector can be mapped to a pattern of precisely timed spikes by using the phase-to-timing mapping. Similarly, the complex synapses can be mapped to synaptic delays. At any stable fixed point (as shown), a cycle time T can be subtracted or added to individual synaptic delays, so that the discrete time dynamics is obeyed. However, this cannot be guaranteed simultaneously for multiple patterns. Deterministic spiking above a threshold implements the TPAM transfer function [**3**].

Complex Algebra with Resonate-and-Fire Neurons. The phase-to-timing mapping illustrates how complex phases can be translated to delays and how this can be used to compute complex multiply. To fully implement a spiking version of TPAM, one needs to also perform the complex dendritic sum, which requires more than just coincidence detection (63). In the temporal domain, addition of complex inputs can be translated into the addition of sine waves matched to the cycle period. A simple mechanism that can achieve this uses *resonate-and-fire* neurons that have subthreshold dynamics of damped oscillators (64):

$$\dot{Z}_i(s) = (\lambda + \mathrm{i}\omega)Z_i(s) + \sum_j \sum_k R_{ij}\delta(s - (s_j^{(k)} + \eta_{ij})), \quad [10]$$

with the Cartesian decomposition of the complex state $Z_i(s) = V_i(s) + \mathrm{i}\,U_i(s)$, and where λ is an attenuation parameter and the angular frequency is $\omega = 2\pi/T$. This neuron model will bandpass-filter the incoming spike train and enable it to (approximately) compute the complex summation.

Our model differs in 3 regards from the network studied in ref. 64: 1) A spike is elicited at time $s_j^{(k)}$ if the internal variables cross 2 thresholds, on the real and imaginary axes: $V_i > V_\theta$ and $U_i > 0$. 2) A refractory period after each spike prevents more than 1 spike per cycle. 3) The weights **R** can be complex, influencing both real and imaginary parts of the neuronal state, and have delays η_{ij}. The weights formed by the TPAM learning rule W_{ij} [**1**] can be set through any combination such that $W_{ij} = R_{ij}\,e^{\mathrm{i}2\pi\eta_{ij}/T}$, in particular, with instantaneous synapses $\mathbf{R} = \mathbf{W}$, $\eta_{ij} = 0$ or with real-valued synapses: $\mathbf{R} = \mathbf{W}^r$, $\eta_{ij} = \zeta_{ij}$.

Each presynaptic spike elicits a jump in the internal state (determined by magnitude and phase of the synapse), kick-starting the damped oscillation (Fig. 6*A*). If subsequent jumps are coherent, they add up to increase the oscillation magnitude. If they are decoherent, they generate a random walk without systematically increasing the oscillation magnitude. Fig. 6*B* demonstrates spiking TPAM (with fixed threshold [**4**]).

Biophysical Model. Some biological neurons behave as individual oscillators, as in the model [**10**]. However, oscillatory membrane voltages and periodic spike trains can also be created through network effects, such as the interplay between excitatory and inhibitory populations (65, 66). We next constructed a biophysical model with integrate-and-fire neurons (67, 68) (Fig. 7*A*) and real-valued synaptic connections which obey Dale's

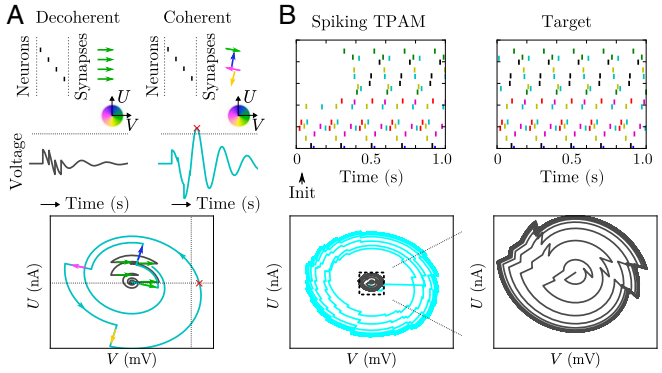

Fig. 6. Complex algebra with resonate-and-fire neurons. (*A*) The spike-timing pattern and the synaptic phase can combine coherently or decoherently. (*B*) A spiking TPAM network with resonate-and-fire neurons, the state initialized with a partial stored pattern. The network converges close to the stored pattern. Internal states of an active (cyan) and inactive (gray) neuron are shown.

principle (69). This model employs network mechanisms to compute the complex dot product. The derivation of this model starts by considering the solution of [**10**] in terms of a synaptic convolution:

$$Z_i(s) = \sum_j \sum_k R_{ij} \left(\delta(s - (s_j^{(k)} + \eta_{ij})) * H(s) e^{(\lambda + \mathrm{i}\omega)s} \right). \quad [11]$$

Rather than complex synapses and damped oscillators forming the synaptic convolution kernels, one can use the trade-off between phase shift and delay to make all synaptic connections real-valued and positive:

$$V_i(s) = \sum_j \sum_k W_{ij}^r \left(\delta(s - (s_k^{(j)} + \zeta_{ij})) * H(s) e^{\lambda s} \cos(\omega s) \right). \quad [12]$$

Eq. **12** shows that the oscillation dynamics of each neuron [**10**] can alternatively be implemented by oscillatory synaptic transmission—e.g., each presynaptic spike is convolved with a damped cosine function. Such balanced (zero mean) convolution kernels can be produced by polysynaptic propagation—for example, if excitatory postsynaptic potentials are followed in a stereotyped fashion by inhibition as has been observed (70).

A complex synapse in TPAM can then be implemented by 3 synapses between spiking neurons that follow Dale's principle: direct excitation (E-E) and indirect inhibition (E-I plus I-E). The collective action of each excitatory spike is to recombine in the postsynaptic excitatory neuron and approximately create a balanced oscillation in the input current.

The inhibitory pathway serves 2 purposes: to act as the negative current in the oscillation and to dynamically adjust the threshold [**4**]. These operations can be achieved with the inhibition acting linearly with nonspecific synaptic delays. A single inhibitory neuron can be used for a population of excitatory neurons, and the recurrent inhibition simply adjusts the offset of the oscillation caused by the recurrent excitation.

When a neuron's membrane potential reaches threshold V_θ, it fires a spike, and the membrane potential is set to the reset potential V_r. After the synaptic delay, the spike causes postsynaptic channels to open. This is modeled as a jump in a synaptic state variable g_{ij} that injects current proportional to the synaptic magnitude $I_i = \sum_j W_{ij}^r g_{ij}$, which then decays exponentially as the channels close (*SI Appendix*).

The time constants of the neural membrane and synaptic variables are tuned based on the cycle time T to create the oscillation. The membrane charging time $\tau_{RC} = 0.5T$ acts as a low-pass filter, helping to smooth out the input impulses into approximately a sine wave. This charging time can also cause extra delay, $\Delta_{RC} = \frac{1}{\omega} \arctan\left(\frac{1}{\omega\tau_{RC}}\right)$, but this charging delay can be accounted for by adjusting delays of synapses. The inhibitory synapses are slow, with a time constant $\tau_s^I = 2T$, while the excitatory synapses are fast, with a time constant $\tau_s^E = 0.5T$. With these settings, the total postsynaptic current elicited by a spike forms (approximately) a sine-wave oscillation with cycle period of T. Depending on presynaptic spike times and synaptic delays, the currents can sum either decoherently or coherently (Fig. 7 *B* and *C*), estimating the complex dot product. Altogether, the TPAM can be implemented with these mechanisms (Fig. 7 *D* and *F*).

The recurrent inhibition can also implement the global normalization needed for the dynamic threshold strategy [**3**], which creates winner-take-all dynamics and keeps the activity sparse. The inhibitory population integrates the activity from the excitatory neurons and routes inhibition back into the population in proportion to the overall activity. The magnitude of this feedback inhibition can be tuned so that only individual patterns are stable fixed points. The gain of this inhibitory feedback is determined by multiple parameters. Each of these gain factors can be computed analytically (with a linear approximation being useful to understand the gain of the spiking neural-transfer function), which is used to tune the parameters (*SI Appendix*).

The deterministic dynamics of the integrate-and-fire neuron will cause the neuron to spike whenever the current drives the voltage above threshold. If the magnitude of the input current oscillation is large enough, then the neuron will fire at a consistent phase (near the zero crossing of the oscillation). However, in this model, the gain of the input oscillation can affect the precise spike timing, but this is mitigated with high gain (*SI Appendix*). For the excitatory neurons, a refractory period slightly over half the cycle time (i.e., $\tau_{ref} = 0.6T$) acts as the Heaviside function on the magnitude. This implements the phasor projection of TPAM by limiting the neuron to 1 spike per cycle, while preserving the phase through the spike timing [**3**]. The parameter value V_θ sets the threshold θ of whether the neuron will fire or not [**4**].

Here, we selectively incremented delays by T to confine the synaptic delays to a range between $0.5T$ and $1.5T$. This choice maximizes interactions between subsequent time steps, but interactions within and across 2 time steps persist. At the fixed points, neither delays nor the attenuation λ affected the dendritic inputs.

Simulation Experiments with the Biophysical Model. For storing a small collection of RGB images, a network architecture with spiking neurons was implemented, as depicted in Fig. 4*A*. The indexing stage transformed the real-valued input image into a complex vector from the encoding matrix (*SI Appendix*). The complex vector was mapped into a timed sequence of input spikes (Fig. 7*D*, Init), which is the initial input to a spiking TPAM network. The spiking TPAM network was decoded with a Hebbian heteroassociative memory. This readout stage used the same integrate-and-fire neurons and synaptic mechanisms to implement the complex dot product for the readout matrix. However, the readout neurons responded proportionally to the input magnitude rather than use a hard thresholding function (i.e., no refractory period; *SI Appendix*).

The network was cued with several overlapping patterns and noise. After initialization through the indexing stage, the spiking activity in the TPAM network quickly settled into an oscillatory pattern (Fig. 7*D*), which corresponded to the index of one of the stored patterns. The global oscillation of the network was generated intrinsically—it did not require any external driving mechanism or internal oscillatory dynamics. The output of the heteroassociative memory stage during the convergence of the TPAM dynamics (Fig. 7*E*) showed how the network rapidly

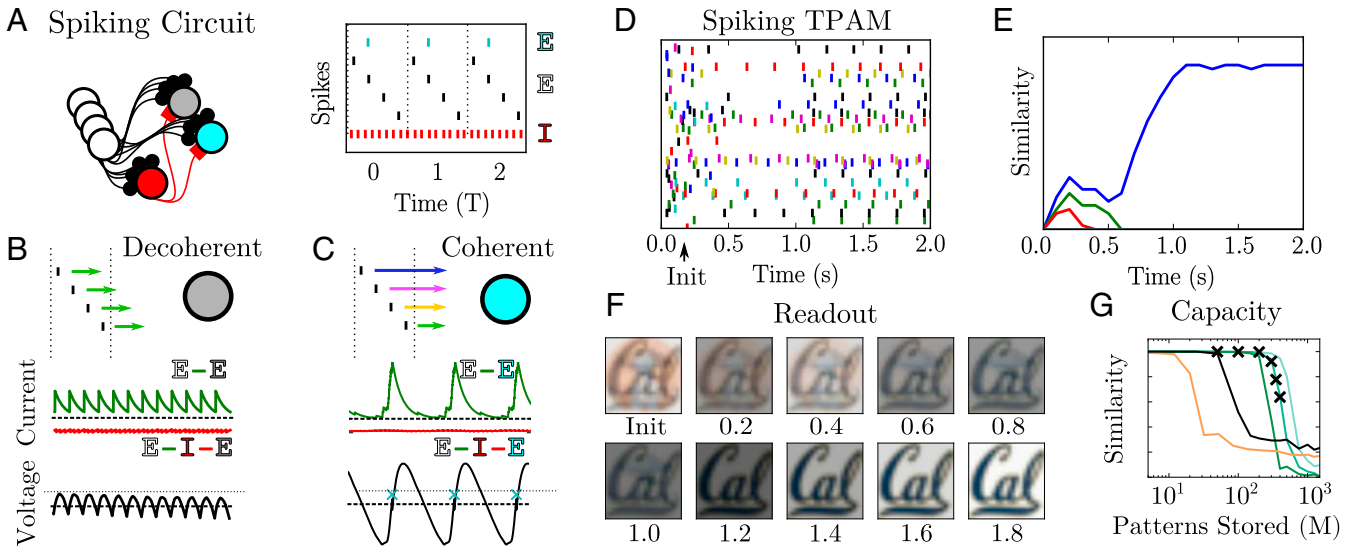

Fig. 7. Biologically plausible TPAM with integrate-and-fire neurons. (*A*) Direct excitation and indirect inhibition produces a postsynaptic current oscillation. Inhibition is nonspecific and acts to offset the oscillation. (*B* and *C*) The effect of a presynaptic spike raster (row 1) depends on the delay relationships of each synapse. The resulting postsynaptic currents (row 2) can be rather small (decoherent; *B*), or quite large (coherent; *C*) (green, excitation; red, inhibition). If the current is large and coherent, the neuron will reach threshold and fire a spike. (*D*) The spiking TPAM was incorporated into a data-indexing network for images. The network is cued with a spiking pattern that encodes an overlap of 3 stored images (Init). After a short time, the network converges to a stored attractor state, a sparse spiking pattern that persistently repeats. (*E*) Evolution of similarities between network state and index patterns of different images. (*F*) Retrieved image as a function of time. (*G*) Retrieval performance of the spiking implementation of TPAM, measured by average similarity between retrieved and target state as function of stored patterns (black x's). The performance of the spiking model matches the performance of a similar TPAM in the complex domain (green lines). The spiking network can store more patterns than the traditional Hopfield network (black) (15) or the dense phasor network (orange) (24).

settled to one of the stored patterns superposed in the input, outcompeting the other 2.

The capacity of the spiking network was examined in simulation experiments (Fig. 7*G*). The spiking network tested was robust even without careful parameter optimization. Retrieval performance based on the number of stored patterns (Fig. 7*G*, black x's) easily exceeded the performance of a traditional bipolar Hopfield network (Fig. 7*G*, black line). The performance curve of the spiking model was predicted by the performance curve of the complex TPAM with similar settings. However, the spiking network did not reach the performance of the optimized complex TPAM. Some of the approximations in the spiking network added noise, which prevented it from reaching the full capacity of the ideal complex model. Nonetheless, these experiments show empirically that the spiking model behaves similarly to the complex TPAM model.

Sequence-Associative Memories and Complex Attractor Networks. Last, we investigated how complex fixed-point attractor networks can help to understand sequence-associative memories: simple networks with binary threshold neurons and parallel, time-discrete update dynamics for storing sequences of patterns of fixed length (Background). Consider the storage of closed sequences or limit cycles of fixed length L: $\boldsymbol{\xi}^1 \to \boldsymbol{\xi}^2 \to \ldots \to \boldsymbol{\xi}^L \to \boldsymbol{\xi}^1 \to \ldots$, with $\boldsymbol{\xi}^l \in \mathbb{R}^N \; \forall l = 1, \ldots, L$. In the case of storing multiple sequences, an index is added to label the different sequences: $\{\boldsymbol{\xi}^{\mu,l}, \mu = 1, \ldots, M\}$. The learning in these models is also described by a Hebbian outer-product learning scheme (45). Here, we use a combination of Hebbian and anti-Hebbian learning to produce a skew-symmetric interaction matrix:

$$\mathbf{J} = \sum_{\mu=1}^{M} \sum_{l=1}^{L} \boldsymbol{\xi}^{\mu,l} \left(\boldsymbol{\xi}^{(\mu,l-1) \bmod L} - \boldsymbol{\xi}^{(\mu,l+1) \bmod L} \right)^{\top}. \qquad [13]$$

Since the matrix $\mathbf{J}$ is skew-symmetric, there is no Lyapunov function describing the dynamics in the network. However, we can use the spectral properties of the weights to construct an equivalent fixed-point attractor network.

Consider [**13**] for the simple example with $L = N = 3$, $M = 1$ and the stored patterns $\boldsymbol{\xi}$ being the cardinal basis vectors of $\mathbb{R}^3$. One of the complex eigenvectors of $\mathbf{J}$ is $\mathbf{v} = \left(e^{\mathrm{i}\frac{2\pi}{3}}, e^{\mathrm{i}\frac{4\pi}{3}}, 1 \right)^{\top} =: \left(e^{\mathrm{i}\phi_1}, e^{\mathrm{i}\phi_2}, e^{\mathrm{i}\phi_3} \right)^{\top}$, which is the (equidistant) phasor pattern that represents the entire stored limit cycle in complex space. One can now form a complex matrix $\mathbf{W}'$ that possesses $\mathbf{v}$ as a fixed point—i.e., has eigenvalue of 1—simply by dividing $\mathbf{J}$ by the eigenvalue associated with $\mathbf{v}$, which is $\lambda = \mathrm{i}\sqrt{3}$:

$$\mathbf{W}' = \frac{1}{\mathrm{i}\sqrt{3}} \mathbf{J} = \frac{1}{\mathrm{i}\sqrt{3}} \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}. \qquad [14]$$

Since the eigenvalues of any skew-symmetric matrix have zero real part (71), the interaction matrix $\mathbf{W}'$ is always Hermitian in general. Thus, the described construction is a recipe to translate sequence memory networks into complex neural networks governed by a Lyapunov dynamics. In the resulting networks, the synaptic matrix is $\mathbf{W}'$, the neural nonlinearity is $g(u_i) = u_i / |u_i|$, and the Lyapunov function is [**5**].

One could now suspect that storing the pattern $\mathbf{v}$ in a phasor network (24) would result in the same network-interaction matrix $\mathbf{W}'$. However, this is not the case. The weight matrix resulting from learning the phase vector $\mathbf{v}$ with the conjugate outer product learning rule [**1**] is

$$\mathbf{W} = \mathbf{v}\mathbf{v}^{*\top} - \mathbf{I} = \begin{bmatrix} 0 & e^{\mathrm{i}(\phi_1 - \phi_2)} & e^{\mathrm{i}(\phi_1 - \phi_3)} \\ e^{\mathrm{i}(\phi_2 - \phi_1)} & 0 & e^{\mathrm{i}(\phi_2 - \phi_3)} \\ e^{\mathrm{i}(\phi_3 - \phi_1)} & e^{\mathrm{i}(\phi_3 - \phi_2)} & 0 \end{bmatrix}. \qquad [15]$$

The phase vector $\mathbf{v}$ is again an eigenvector of $\mathbf{W}$.

The takeaways from this example are the following points:

1) Outer-product sequence memories with skew-symmetric weights can be mapped to continuous-valued phasor networks with Hermitian weights $\mathbf{W}'$, whose dynamics is described by the Lyapunov function [**5**]. A similar idea of deriving a Lyapunov function for M-periodic sequences of binary patterns was proposed in refs. 52 and 53. Specifically, these authors proposed to embed sequence trajectories in a real-valued space, consisting of M copies of the original state space.
2) Continuous-valued phasor networks derived from sequence memories with outer-product rule [**13**] are different from phasor networks using the conjugate complex outer-product learning rule [**1**], such as TPAM networks.
3) Phasor networks derived from outer-product sequence memories have 2 severe restrictions. First, since the synaptic weights are imaginary without real part, they only can rotate the presynaptic signals by 90°. Second, the first-order Markov property of sequence memory networks translates into phasor networks with interactions only between direct phase-angle neighbors.
4) Phasor networks derived from sequence memories can, by construction, only store phase patterns whose phase angles are equidistantly dividing 2π. Because of symmetry reasons, such equidistant phase patterns can be stabilized, despite the restrictions described in the previous point.

Discussion

We present a theory framework for temporal coding with spiking neurons that is based on fixed-point attractor dynamics in complex state space. Although a coding scheme that uses spike timing would seem to leverage the benefits of spiking neurons (12), its use in neuroscience and neuromorphic applications is still rather limited. Based on our results, we suggest that problems with spike-timing codes (72, 73) are not inherent, but are due to our lack of understanding error-correcting properties of spiking neural circuits.

Threshold Phasor Networks. A type of complex attractor network is introduced, TPAM. TPAM inherits from previous phasor networks (24) the Lyapunov dynamics and the capability to store and error-correct arbitrary continuous-valued phasor patterns.

The neural threshold mechanism added in TPAM significantly increases the synaptic memory capacity and suppresses the meaningless representation of flat or very noisy postsynaptic phase distributions. The capability of TPAM to store patterns with arbitrary continuous-valued phases as attractor states permits the processing of analog data directly. In contrast, the fixed points of most attractor networks are discrete, and the patterns to store have to first be discretized or even binarized. Even for real-valued attractor networks, the stable states lie at saturation points and are still discrete. Data correlations in real-world sensor data pose a problem for all associative memory models, including TPAM. Thus, for images, we propose a 3-layer architecture for data indexing and storage. The architecture includes a TPAM for error correction and improves on similar previous models, such as SDM (62).

We show that the dynamics of prominent traditional models for sequence memory (45–47) can be described by a TPAM-like model with a different learning rule. This phasor description of sequence memory is similar to earlier work, which constructs a Lyapunov function in an enhanced (real-valued) state space (53), but it has interesting implications. Specifically, the TPAM learning rule can overcome limitations of traditional Hebbian sequence learning—i.e., how to simultaneously memorize distinct sequences that share some same states.

Mapping Complex Attractors into Periodic Temporal Codes. A state of TPAM at a discrete time step can be translated by a simple phase-to-timing mapping into a spike-timing pattern within a time interval T. A T-periodic spike raster then corresponds to a fixed point in TPAM. By creating spiking neural networks that mirror the fixed-point dynamics of TPAM, the stored T-periodic limit cycles exhibit robustness to perturbation and pattern completion.

Earlier suggestions to compute with periodic spike patterns have inspired our work. For example, Herz (citation details available from the authors upon request)* used the enhanced state-space approach to derive a Lyapunov function for networks of nonleaky integrate-and-fire neurons with synaptic delays. As in their earlier model without synaptic delays (74), such networks enable rapid (although not very robust) convergence to stored periodic spike patterns. Further, networks were proposed (75) for transducing rate-coded signals into periodic spike-timing patterns. Other examples include "synfire braids" (76) and "polychronization" (63) in networks with synaptic delays and neurons that detect synchrony.

Complex Algebra with Spiking Neural Circuits. We describe 2 concrete models of spiking networks for implementing TPAM. The simplest one, which provides direct insight, is a network with resonate-and-fire neurons (64). Using the complex synapses and a modified spiking mechanism, the resulting network dynamics computes the complex dot product needed for TPAM quite naturally. However, the update scheme becomes time-continuous and event-driven—triggered by threshold crossings in individual neurons. This fundamentally differs from the parallel update in the time-discrete TPAM. However, even though they are not equivalent in the entire state space, they become equivalent at the fixed points.

From our model with oscillatory neurons, we were able to derive a second network model with stronger resemblance to neurobiological circuits, featuring ordinary integrate-and-fire neurons and circuit connections that obey Dale's principle (69). The complex-valued synapses are replaced with real-valued synapses that have time delays. Network effects are used to generate oscillatory postsynaptic currents in neurons that do not oscillate by themselves. The approximately correct shaping of the convolution kernels of synaptic transmission is achieved by using a combination of standard biologically plausible mechanisms: dendritic filtering, synaptic time constants, synaptic delays, and additional inhibitory neurons that balance excitation. Interestingly, the model implements complex algebra relying on a number of mechanisms observed in neuroscience. "Periodic firing and synchrony": Action-potentials in the brain are often periodic, synchronized with intrinsic rhythms visible in local field potentials (14, 77, 78). "Delayed synaptic transmission": Due to variability in axon length and myelination, the distribution in measured delay times in monosynaptic transmission is broad, easily spanning the full cycle length of gamma and even theta oscillations (79). However, our model does not exclude other circuit mechanisms (for example, a thalamo-cortical loop) as possible mechanisms for longer delays. "Balance between excitation and inhibition": Excitation/inhibition balance is widely seen throughout cortex (80–82), and inhibitory feedback onto pyramidal cells is a major feature of the canonical cortical microcircuit (83, 84).

Note that the parameter setting in our biophysical network is generic; we did not attempt to model a particular brain region, and other alternative mechanisms are possible. In the simulations demonstrated, $T = 200$ ms was chosen as the cycle period,

*A. V. Herz, Do signal delays change rapid phase locking of pulse-coupled oscillators? Preprint (25 January 1996).

but a frequency in the gamma range with T = 20–25 ms may be more appropriate for directly comparing to biological measurements. We discuss ideas for biological realism further in *SI Appendix*.

Learning Limit Cycles in a Spiking Network. The conjugate outer-product learning rule in TPAM requires either complex synaptic connections or tunable synaptic delays. If complex synaptic connections are possible, then Hebbian learning requires cotuning the real and imaginary weights. Delay learning has been suggested in previous models (85), but without compelling biological evidence. Alternatively, spike-timing–dependent plasticity (86) could be used to shape synapses for TPAM, by strengthening and pruning synapses from a pool with fixed synaptic delays.

Model Extensions. The presented TPAM models can be extended to combine aspects of spike timing and rate coding. Complex numbers with variable magnitudes can be represented by rapid bursts of multiple spikes or by probabilistic spiking. As in the real-valued Hopfield model (1), the hard threshold in the transfer function of TPAM can be exchanged by a sigmoidal transfer function. In the spiking models, this means removing the refractory mechanism preventing more than 1 spike per cycle. Attractors would lie near saturation points in the magnitudes, as in the real-valued Hopfield model (1), but phase relationships could still be arbitrary. In such a model, spike patterns would have periodically modulated spike rates, as seen in hippocampal place cells (13).

Further, TPAM-like networks with modified learning rules and threshold strategy could be used to construct line attractors with spike timing—useful for understanding place coding in hippocampus (87, 88) or for modeling/predicting dynamical systems (4, 89).

Neuromorphic Computing. The presented theory has direct impact on "neuromorphic computing," which has recently been rediscovered as a promising paradigm for computation—for example, *Braindrop* (90) and IBM's *True North* (91). Recently, Intel released the neuromorphic chip *Loihi* (16), which features individual synaptic delays and on-chip learning. Our theory offers a principled way of "programming" spiking-neuron hardware, leveraging the speed of temporal codes, and providing straightforward connections to complex matrix algebra and error correction.

Materials and Methods

Detailed methods can be found in *SI Appendix*.

ACKNOWLEDGMENTS. This work was supported by NSF Grant IIS1718991, NIH Grant 1R01EB026955-01, the Intel Strategic Research Alliance program, and Berkeley DeepDrive. We thank Pentti Kanerva and members of the Redwood Center for valuable feedback.

1. J. J. Hopfield, Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. U.S.A.* **81**, 3088–3092 (1984).
2. D. J. Amit, *Modeling Brain Function: The World of Attractor Neural Networks* (Cambridge Univ Press, Cambridge, UK, 1992).
3. A. Pouget, P. Dayan, R. S. Zemel, Inference and computation with population codes. *Annu. Rev. Neurosci.* **26**, 381–410 (2003).
4. C. Eliasmith *et al.*, A large-scale model of the functioning brain. *Science* **338**, 1202–1205 (2012).
5. A. M. Bruckstein, M. Morf, Y. Y. Zeevi, Demodulation methods for an adaptive neural encoder model. *Biol. Cybernetics* **49**, 45–53 (1983).
6. W. Bialek, F. Rieke, R. R. de Ruyter van Steveninck, D. Warland, Reading a neural code. *Science* **252**, 1854–1857 (1991).
7. B. Cessac, H. Paugam-Moisy, T. Viéville, Overview of facts and issues about neural coding by spikes. *J. Physiol. Paris* **104**, 5–18 (2010).
8. M. Abeles, *Local Cortical Circuits: An Electrophysiological Study* (Springer, Berlin, Germany, 1982).
9. P. Reinagel, R. C. Reid, Temporal coding of visual information in the thalamus. *J. Neurosci.* **20**, 5392–5400 (2000).
10. S. Panzeri, M. E. Diamond, Information carried by population spike times in the whisker sensory cortex can be decoded without knowledge of stimulus time. *Front. Synaptic Neurosci.* **2**, 1–14 (2010).
11. P. Tiesinga, J. M. Fellous, T. J. Sejnowski, Regulation of spike timing in visual cortical circuits. *Nat. Rev. Neurosci.* **9**, 97–107 (2008).
12. S. Thorpe, D. Fize, C. Marlot, Speed of processing in the human visual system. *Nature* 381, 520–522 (1996).
13. J. O'Keefe, M. L. Recce, Phase relationship between hippocampal place units and the hippocampal theta rhythm. *Hippocampus* **3**, 317–330 (1993).
14. G. Buzsaki, A. Draguhn, Neuronal oscillations in cortical networks. *Science* **304**, 1926–1929 (2004).
15. J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U.S.A.* **79**, 2554–2558 (1982).
16. M. Davies *et al.*, Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* **38**, 82–99 (2018).
17. D. H. Ackley, G. E. Hinton, T. J. Sejnowski, A learning algorithm for Boltzmann machines. *Cognit. Sci.* **9**, 147–169 (1985).
18. J. J. Hopfield, D. W. Tank, "Neural" computation of decisions in optimization problems. *Biol. Cybernetics* **52**, 141–152 (1985).
19. W. A. Little, "The existence of persistent states in the brain" in *From High-Temperature Superconductivity to Microminiature Refrigeration*, B. Cabrera, H. Gutfreund, V. Z. Kresin, Eds. (Springer, Berlin, Germany, 1974), pp. 145–164.
20. N. H. Farhat, D. Psaltis, A. Prata, E. Paek, Optical implementation of the Hopfield model. *Appl. Opt.* **24**, 1469–1475 (1985).
21. M. V. Tsodyks, M. V. Feigel'man, The enhanced storage capacity in neural networks with low activity level. *Europhys. Lett.* **6**, 101–105 (1988).
22. C. Meunier, D. Hansel, A. Verga, Information processing in three-state neural networks. *J. Stat. Phys.* **55**, 859–901 (1989).
23. A. Treves, Graded-response neurons and information encodings in autoassociative memories. *Phys. Rev. A* **42**, 2418–2430 (1990).
24. A. J. Noest, "Phasor neural networks" in *Neural Information Processing Systems 1987*, D. Z. Anderson, Ed. (Neural Information Processing Systems, San Diego, CA, 1988) pp. 584–591.
25. M. A. Cohen, S. Grossberg, Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Trans. Syst. Man Cybernetics* **SMC-13**, 815–826 (1983).
26. E. Ising, Beitrag zur theorie des ferromagnetismus. *Z. Phys.* **31**, 253–258 (1925).
27. R. Kühn, S. Bös, J. L. van Hemmen, Statistical mechanics for networks of graded-response neurons. *Phys. Rev. A* **43**, 2084–2087 (1991).
28. D. J. Willshaw, O. P. Buneman, H. C. Longuet-Higgins, Non-holographic associative memory. *Nature* **222**, 960–962 (1969).
29. J. Buhmann, R. Divko, K. Schulten, Associative memory with high information content. *Phys. Rev. A* **39**, 2689–2692 (1989).
30. G. Palm, F. T. Sommer, Information capacity in recurrent McCulloch–Pitts networks with sparsely coded memory states. *Network Comput. Neural Syst.* **3**, 177–186 (1992).
31. F. Schwenker, F. T. Sommer, G. Palm, Iterative retrieval of sparsely coded associative memory patterns. *Neural Networks* **9**, 445–455 (1996).
32. D. Gross, I. Kanter, H. Sompolinsky, Mean-field theory of the Potts glass. *Phys. Rev. Lett.* **55**, 304–307 (1985).
33. J. Yedidia, Neural networks that use three-state neurons. *J. Phys. A Math. Gen.* **22**, 2265–2273 (1989).
34. M. Bouten, A. Engel, Basin of attraction in networks of multistate neurons. *Phys. Rev. E* **47**, 1397–1400 (1993).
35. J. Cook, The mean-field theory of a q-state neural network model. *J. Phys. A Math. Gen.* **22**, 2057–2067 (1989).
36. F. Gerl, K. Bauer, U. Krey, Learning with q-state clock neurons. *Z. Phys. B Condens. Matter* **88**, 339–347 (1992).
37. A. Hirose, Dynamics of fully complex-valued neural networks. *Electron. Lett.* **28**, 1492–1494 (1992).
38. S. Jankowski, A. Lozowski, J. M. Zurada, Complex-valued multistate neural associative memory. *IEEE Trans. Neural Networks* **7**, 1491–1496 (1996).
39. L. Donq-Liang, W. Wen-June, A multivalued bidirectional associative memory operating on a complex domain. *Neural Networks* **11**, 1623–1635 (1998).
40. H. Aoki, Y. Kosugi, "An image storage system using complex-valued associative memories" in Proceedings of the 15th International Conference on Pattern Recognition, A. Sanfeliu et al., Eds. (IEEE, Piscataway, NJ, 2000), pp. 626–629.
41. I. Aizenberg, "Why we need complex-valued neural networks?" in *Complex-Valued Neural Networks with Multi-Valued Neurons* (Springer, Berlin, Germany, 2011), pp. 1–53.
42. M. Kobayashi, Fast recall for complex-valued Hopfield neural networks with projection rules. *Comput. Intell. Neurosci.* **2017**, 1–6 (2017).
43. G. Y. Sirat, A. D. Maruani, R. C. Chevallier, Grey level neural networks. *Appl. Opt.* **28**, 414–415 (1989).
44. F. C. Hoppensteadt, E. M. Izhikevich, Synaptic organizations and dynamical properties of weakly connected neural oscillators: I. Analysis of a canonical model. *Biol. Cybernetics* **75**, 117–127 (1996).
45. S. I. Amari, Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Trans. Comput.* **C-21**, 1197–1206 (1972).
46. G. Willwacher, Capabilities of an associative storage system compared with the function of the brain. *Biol. Cybernetics* **24**, 181–198 (1976).

47. G. Willwacher, Storage of a temporal pattern sequence in a network. *Biol. Cybernetics* **43**, 115–126 (1982).
48. D. Kleinfeld, Sequential state generation by model neural networks. *Biophysics* **83**, 9469–9473 (1986).
49. H. Sompolinsky, I. Kanter, Temporal association in asymmetric neural networks. *Phys. Rev. Lett.* **57**, 2861–2864 (1986).
50. D. J. Amit, Neural networks counting chimes. *Proc. Natl. Acad. Sci. U.S.A.* **85**, 2141–2145 (1988).
51. U. Riedel, R. Kühn, J. Van Hemmen, Temporal sequences and chaos in neural nets. *Phys. Rev. A* **38**, 1105–1108 (1988).
52. A. V. Herz, Global analysis of parallel analog networks with retarded feedback. *Phys. Rev. A* **44**, 1415–1418 (1991).
53. A. V. Herz, Z. Li, J. Van Hemmen, Statistical mechanics of temporal association in neural networks with transmission delays. *Phys. Rev. Lett.* **66**, 1370–1373 (1991).
54. T. Kohonen, Correlation matrix memories. *IEEE Trans. Comput.* **100**, 353–359 (1972).
55. T. Wennekers, F. T. Sommer, G. Palm, "Iterative retrieval in associative memories by threshold control of different neural models" in *Supercomputing in Brain Research: From Tomography to Neural Networks*, H. J. Herrmann, D. E. Wolf, E. Pöppel, Eds. (World Scientific, Hackensack, NJ, 1995), pp. 301–319.
56. Y. Kuramoto, "Self-entrainment of a population of coupled non-linear oscillators" in *International Symposium on Mathematical Problems in Theoretical Physics, H.* Araki, Ed. (Lecture Notes in Physics, Springer, Berlin, Germany, 1975), vol. 39, pp. 420–422.
57. J. M. Kosterlitz, D. J. Thouless, Ordering, metastability and phase transitions in two-dimensional systems. *J. Phys. C Solid State Phys.* **6**, 1181–1203 (1973).
58. H. Schuster, P. Wagner, A model for neuronal oscillations in the visual cortex. *Biol. Cybernetics* **64**, 77–82 (1990).
59. H. Sompolinsky, D. Golomb, D. Kleinfeld, Global processing of visual stimuli in a neural network of coupled oscillators. *Proc. Natl. Acad. Sci. U.S.A.* **87**, 7200–7204 (1990).
60. E. Niebur, H. G. Schuster, D. M. Kammen, C. Koch, Oscillator-phase coupling for different two-dimensional network connectivities. *Phys. Rev. A* **44**, 6895–6904 (1991).
61. J. L. Van Hemmen, W. F. Wreszinski, Lyapunov function for the Kuramoto model of nonlinearly coupled oscillators. *J. Stat. Phys.* **72**, 145–166 (1993).
62. P. Kanerva, *Sparse Distributed Memory* (MIT Press, Cambridge, MA, 1988).
63. E. M. Izhikevich, Polychronization: Computation with spikes. *Neural Comput.* **18**, 245–282 (2006).
64. E. M. Izhikevich, Resonate-and-fire neurons. *Neural Networks* **14**, 883–894 (2001).
65. J. Veit, R. Hakim, M. P. Jadi, T. J. Sejnowski, H. Adesnik, Cortical gamma band synchronization through somatostatin interneurons. *Nat. Neurosci.* **20**, 951–959 (2017).
66. M. A. Whittington, R. Traub, N. Kopell, B. Ermentrout, E. Buhl, Inhibition-based rhythms: Experimental and mathematical observations on network dynamics. *Int. J. Psychophysiol.* **38**, 315–336 (2000).
67. D. F. M. Goodman, R. Brette, The brain simulator. *Front. Neurosci.* **3**, 192–197 (2009).
68. C. Eliasmith, C. H. Anderson, *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems* (MIT Press, Cambridge, MA, 2003).
69. J. C. Eccles, P. Fatt, K. Koketsu, Cholinergic and inhibitory synapses in a pathway from motor-axon collaterals to motoneurones. *J. Physiol.* **126**, 524–562 (1954).
70. B. Da *et al.*, Temporal precision in the neural code and the timescales of natural vision. *Nature* **449**, 92–96 (2007).
71. H. W. Eves, *Elementary Matrix Theory* (Courier Corporation, North Chelmsford, MA, 1980).
72. M. N. Shadlen, J. A. Movshon, Synchrony unbound: A critical evaluation of the temporal binding hypothesis. *Neuron* **24**, 67–77 (1999).
73. M. London, A. Roth, L. Beeren, M. Häusser, P. E. Latham, Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. *Nature* **466**, 123–127 (2010).
74. J. J. Hopfield, A. V. Herz, Rapid local synchronization of action potentials: Toward computation with coupled integrate-and-fire neurons. *Proc. Natl. Acad. Sci. U.S.A.* **92**, 6655–6662 (1995).
75. J. J. Hopfield, Pattern recognition computation using action potential timing for stimulus representation. *Nature* **376**, 33–36 (1995).
76. E. Bienenstock, A model of neocortex. *Network Comput. Neural Syst.* **6**, 179–224 (1995).
77. A. Riehle *et al.*, Spike synchronization and rate modulation differentially involved in motor cortical function. *Science* **278**, 1950–1953 (1997).
78. G. F. Lynch, T. S. Okubo, A. Hanuschkin, R. H. Hahnloser, M. S. Fee, Rhythmic continuous-time coding in the songbird analog of vocal motor cortex. *Neuron* **90**, 877–892 (2016).
79. H. A. Swadlow, "Information flow along neocortical axons" in *Time and the Brain*, R. Miller, Ed. (Harwood Academic Publishers, London, 2000).
80. J. Mariño *et al.*, Invariant computations in local cortical networks with balanced excitation and inhibition. *Nat. Neurosci.* **8**, 194–201 (2005).
81. B. Haider, A. Duque, A. Hasenstaub, D. A. McCormick, Neocortical network activity in vivo is generated through a dynamic balance of excitation and inhibition. *J. Neurosci.* **26**, 4535–4545 (2006).
82. B. V. Atallah, M. Scanziani, Instantaneous modulation of gamma oscillation frequency by balancing excitation with inhibition. *Neuron* **62**, 566–577 (2009).
83. R. J. Douglas, K. A. Martin, D. Whitteridge, A canonical microcircuit for neocortex. *Neural Comput.* **1**, 480–488 (1989).
84. J. S. Isaacson, M. Scanziani, How inhibition shapes cortical activity. *Neuron* **72**, 231–243 (2011).
85. H. Hüning, H. Glünder, G. Palm, Synaptic delay learning in pulse-coupled neurons. *Neural Comput.* **10**, 555–565 (1998).
86. B. Szatmáry, E. M. Izhikevich, Spike-timing theory of working memory. *PLoS Comput. Biol.* **6**, e1000879 (2010).
87. P. E. Welinder, Y. Burak, I. R. Fiete, Grid cells: The position code, neural network models of activity, and the problem of learning. *Hippocampus* **18**, 1283–1300 (2008).
88. M. G. Campbell *et al.*, Principles governing the integration of landmark and self-motion cues in entorhinal cortical codes for navigation. *Nat. Neurosci.* **21**, 1096–1106 (2018).
89. M. Boerlin, C. K. Machens, S. Denève, Predictive coding of dynamical variables in balanced spiking networks. *PLoS Comput. Biol.* **9**, e1003258 (2013).
90. A. Neckar *et al.*, Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model. *Proc. IEEE* **107**, 144–164 (2019).
91. P. A. Merolla *et al.*, A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **345**, 668–673 (2014).