OSCILLATOR ASSOCIATIVE MEMORIES FACILITATE HIGH-CAPACITY, COMPOSITIONAL INFERENCE

Christopher J. Kymn[1*], Connor Bybee[1], Zeyu Yun[1], Denis Kleyko[2,3], and Bruno A. Olshausen[1]

[1]Redwood Center for Theoretical Neuroscience, UC Berkeley, Berkeley, USA
[2]Centre for Applied Autonomous Sensor Systems, Örebro University, Örebro, Sweden
[3]Intelligent Systems Lab, Research Institutes of Sweden, Kista, Sweden
[*]`cjkymn@berkeley.edu`

ABSTRACT

We introduce a novel, high-capacity associative memory capable of factorizing compositional representations of variables. The approach is implemented as a continuous-time oscillator neural network. By performing factorization with a continuous-time dynamical system, the proposed Factorizing Oscillator Associative Memory (FOAM) provides efficient solutions to computationally hard problems such as inference in compositional representations and combinatorial optimization. We demonstrate favorable performance compared to existing approaches to factorization, efficient implementation, and relevance to standard tasks such as the subset sum problem. We also identify concrete reasons why our model exhibits improved capacity, by formulating methods to track its convergence.

1 INTRODUCTION

An important open problem for both computational neuroscience and machine learning is to maintain a memory of the world that can be used efficiently for future inference. Indeed, problems such as perception, reasoning, and combinatorial optimization need to use prior knowledge to analyze novel situations on-the-fly. Towards this end, associative memories have proven to be a simple yet powerful component. Hopfield (1984) proposed a continuous-time dynamical system formulation of a neural network model forming an associative memory. Such memories are capable of storing binary patterns in a distributed fashion and can efficiently recall a stored pattern in the presence of errors or erasures. It was then shown in Hopfield & Tank (1985) that networks with the same dynamics can be used to solve hard combinatorial problems. While these networks operated on binary patterns, Noest (1987) showed that real-valued patterns can be stored in an associative memory formed with phasor-valued representations and the corresponding continuous-time dynamics.

Still, a major ongoing challenge consists in scaling associative memories to handle the many complex and diverse sensory input patterns that arise from interacting with the natural environment. One strategy for addressing this challenge is to develop compositional representations, in which memories (new and old) can be factored into modular parts. For this purpose, Frady et al. (2020) proposed a *resonator network* that enables high-capacity storage and retrieval of patterns using compositional structure. Compositional representations are formed by a vector binding operation that combines multiple factors. The resonator network can then infer the factors from such a representation, thereby acting as a *compositional associative memory*. Prior work has shown that resonator networks exhibit favorable scaling in capacity (Kent et al., 2020) and can perform efficient inferences over large search spaces (Kymn et al., 2025). In this paper, we demonstrate that using oscillatory dynamics further increases the capacity.

While much prior work has examined circuit implementations of associative memories, it is not always known if the circuit will perform as well as the digital implementation. We show that even the discrete-time simulation of an oscillator network outperforms the original algorithm in the ability to recall stored patterns and factor representations with few iterations. We also demonstrate the usefulness of the network in representing both continuous and discrete phasor representations, providing insights for novel implementations of resonator networks in neuromorphic hardware.

2 FACTORIZING OSCILLATOR ASSOCIATIVE MEMORY (FOAM)

Attractor neural networks are simple neural circuits whose dynamics perform computations, such as pattern cleanup or retrieval. In the simplest form they use binary-valued patterns but this formulation can be generalized to complex-valued representations with unit amplitude ("phasors"). A seminal paper on phasor neural networks by Noest (1987) describes a simple continuous-time oscillator network.[1] This model consists of an D-dimensional state vector of phases, $\phi \in \mathbb{R}^D$, that is governed by the following equation:

$$\frac{d}{dt}\phi_j = |h_j| \sin\left(\arg(h_j) - \phi_j\right), \text{where } h_j = \sum_{k=1}^{D} W_{jk}\, e^{i\phi_j}, \tag{1}$$

and where $|h_j|$ and $\arg(h_j)$ denote the amplitude and phase of the complex number h_j, respectively. From these dynamics, it should be evident that we have a fixed point whenever $\forall j, \arg(h_j) = \phi_j$. Noest showed that this dynamical system has a Lyapunov function when the weights are Hermitian symmetric (i.e., $W_{jk} = W^*_{kj}$). We note that this formulation is similar in spirit to the well-studied Kuramoto model, except that the magnitude of the oscillator input, $|h_j|$, is included in the phase dynamics. This has the effect of increasing the rate of change of the oscillator phase as the magnitude of the input increases.

We now show how to adapt the dynamics of Noest (1987) to perform vector factorization. More concretely, the objective is to factorize an input vector $\mathbf{s} = \mathbf{x}^* \odot \mathbf{y}^*$, where $\mathbf{x}^*, \mathbf{y}^*, \mathbf{s}$ are phasor vectors of dimension D, and $\odot$ denotes element-wise multiplication (Hadamard product). Suppose also that $\mathbf{x}^*$ and $\mathbf{y}^*$ each come from a discrete set of candidates: $\mathbf{x}^* \in \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(m)}\}$, $\mathbf{y}^* \in \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, ..., \mathbf{y}^{(m)}\}$. Factorization amounts to recovering $\mathbf{x}^*$ and $\mathbf{y}^*$ given $\mathbf{s}$ and the sets of possible vectors for each factor; thus, the total search space is m^2. Here, we introduce a new model, the Factorizing Oscillator Associative Memory (FOAM), which extends prior oscillator-based associative memory models to support compositional factorization. We formulate the dynamics as follows:

$$\frac{d}{dt}\phi_j^{(x)} = |h_j^{(x)}| \sin\left(\arg(h_j^{(x)}) - \phi_j^{(x)}\right), \text{and } \frac{d}{dt}\phi_j^{(y)} = |h_j^{(y)}| \sin\left(\arg(h_j^{(y)}) - \phi_j^{(y)}\right), \tag{2}$$

where

$$h_j^{(y)} = \sum_{k=1}^{D} W_{jk}^{(y)}(\, s_j \odot e^{-i\phi_j^{(x)}}), \text{and } h_j^{(x)} = \sum_{k=1}^{D} W_{jk}^{(x)}\ (s_j \odot e^{-i\phi_j^{(y)}}). \tag{3}$$

Ideally, these dynamics will cause the states for each factor to become similar to the ground truth vectors that generated the vector $\mathbf{s}$. More concretely, the amplitude of the complex-valued inner product between the state vector $\exp(\phi^{(x)})$ and a candidate vector $\mathbf{x}^{(p)}$ should be highest for the ground truth vector $\mathbf{x}^*$.

Finally, the matrices $\mathbf{W}^{(x)}$ and $\mathbf{W}^{(y)}$ are constructed with an outer product learning rule based on the candidate vectors for each factor:

$$\mathbf{W}^{(x)} = \sum_{p=1}^{m} (\mathbf{x}^{(p)}\mathbf{x}^{(p)\dagger} - \mathbf{I}_D) \tag{4}$$

where $\dagger$ denotes the transpose conjugate, and $\mathbf{I}_D$ is the D-dimensional identity matrix. We consider the more general case, in which $\mathbf{s}$ is a product of $R > 2$ vectors, in Section 3.3 and Appendix A.

3 RESULTS

3.1 HIGHER CAPACITY WITH EFFICIENT IMPLEMENTATION

First, we analyze the performance of the new oscillator version compared to the baseline resonator network. We examine how well networks of varying dimensions (D) recover the original factors

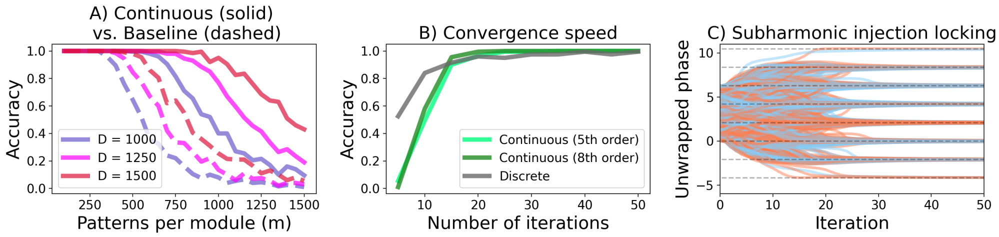

Figure 1: **Performance characteristics of factorizing oscillators. A)** The continuous update equations (FOAM model, solid lines) provide a strong increase in performance relative to the standard, discrete update resonator network baseline (dashed lines). Each line demonstrates the average accuracy (over 200 trials) of the factorization solution for a model with two factors and a particular dimension. Different colors indicate different dimension (D); higher dimensions have higher capacity as expected from the study of Hopfield networks. **B)** Continuous-time dynamics can be simulated efficiently. Plots show average accuracy (over 200 trials) with $D = 1000, m = 300$. Remarkably, with adaptive step sizes, the number of iterations required for convergence is comparable for discrete and continuous updates. In fact, the continuous case reaches 100 percent empirical accuracy faster. The results for different order stepsizes (light and dark green) are comparable. **C)** Subharmonic injection locking encourages phases to settle to particular values, which can be useful for readout and neuromorphic implementations in oscillators. Each line shows the time evolution of one phasor, and distinct colors denote distinct modules.

when m patterns are stored per module. The factorization is considered *correct* if and only if, for every factor, the network state is closer to the ground truth vector than any other item in the codebook. Here, the relevant notion of "closeness" is the amplitude of the inner product. The accuracy curves in Figure 1 show the proportion of correct trials for varying dimensions and number of patterns. As expected from prior studies of associative memory, the accuracy quickly falls off after a critical number of patterns is stored. The capacity refers to the largest pattern load with accuracy above a threshold (e.g., at least 0.99); its scaling laws can be estimated empirically (Kent et al., 2020; Kymn et al., 2024a). We find that across a variety of dimensions, the capacity is much higher for the continuous (ours) vs. discrete-dynamics (baseline) case. The analysis performed in the next section indicates that this is due to benign instabilities which adversely affect the discrete but not continuous case.

In addition, and perhaps surprisingly, we find that the continuous version can be implemented in approximately the same number of time-steps as the discrete version (Figure 1B). This works in part because we use adaptive step solver methods to simulate our dynamical systems, accelerating convergence (Kidger, 2021). These results show that the continuous-time nature of the equations can be efficiently implemented. Future work will examine the trade-offs between the improvement in performance and the additional computations required for integrating the differential equations.

We also consider how to enforce phase discretization with continuous dynamics. Such discrete states are known as Q-state phasors (Noest, 1988), and discretization can help reduce crosstalk between patterns in the final, converged states. To address this scenario, we consider the inclusion of an additional term to the dynamics (highlighted in blue):

$$\frac{d}{dt}\phi_j^{(x)} = |h_j^{(x)}| \sin\left(\arg(h_j^{(x)}) - \phi_j^{(x)}\right) - \sin(Q\phi_j)\frac{t^2}{\tau}, \tag{5}$$

where Q corresponds to the number of phasor states, t is the time, and τ functions as a time constant. The time-dependence of the term defines an annealing schedule, in which the constraint exerts stronger influence over time. This additional term successfully ensures that each component converges to one of these Q values (Figure 1C) and has previously been characterized in the study of associative memory (Nishikawa et al., 2004; Bybee & Sommer, 2022).

3.2 Increased capacity is due to "benign instability" with increased patterns

A critical question emerges from the results reported in Figure 1 : why is the capacity higher for the continuous dynamics vs. the discrete one? It is observed that for the continuous dynamics, we

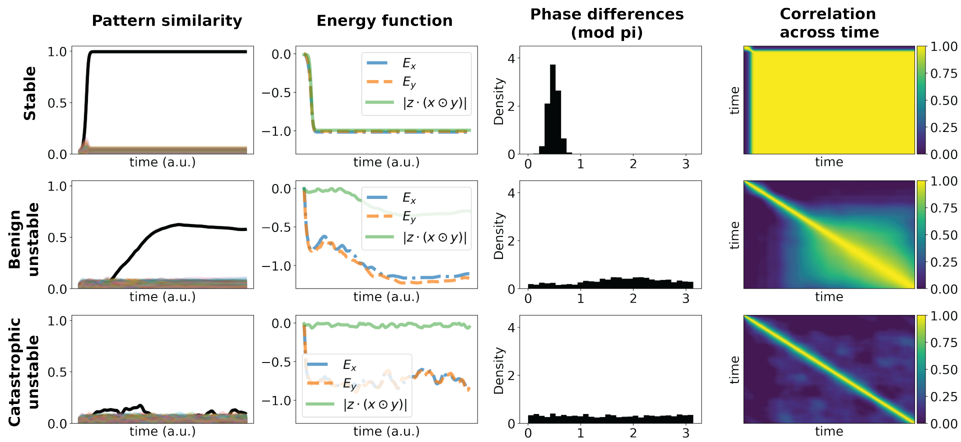

Figure 2: **Analysis of dynamics in three different regimes. Top row**: in the *stable* regime (few patterns per dimension), the dynamics behave 'as if' descending an energy function, system converges rapidly to one pattern, and does not drift. **Middle row**: In the *benign unstable* regime, many patterns converge to a spurious mixture, yet much more similar to the correct one than the rest. The presence of cross-talk noise causes patterns to drift, but does not affect which pattern is most similar. **Bottom row**: In the *catastrophic unstable* regime (too many patterns per dimension), no state stays similar to the correct factor, and the network activity is not similar across time steps.

discover an unstable regime in which network states are no longer stable but this does not impact the overall performance of the FOAM model. Therefore, we analyze its behavior by varying the number of stored patterns. To quantify this, we look at inner products of the system's states and stored patterns, "energy values"[2] of each module, phase differences, and the normalized inner products between states at different time points (temporal correlation). Figure 2 shows three distinct examples, with $D = 2000$ and $m \in \{40, 1300, 1900\}$; note that the effective search space for each example is m^2, which is much larger than D for the latter two cases.

We observe several different modes of activity, with typical behavior for each depicted on a row of Figure 2:

1. *Stable:* Convergence to the correct factors proceeds rapidly and recall is nearly perfect. Even though the weights are not symmetric, these points behave as if descending an energy function. Indeed, the final state converges to a stable pattern.

2. *Benign unstable:* This mode is benign in the sense that the best pattern is most similar to the correct factor compared to any other pattern. The correlation across time shows that the state, although fluctuating, stays self-similar over time. The network tends to decrease its values relative to the energy function, but not monotonically. Collectively, these results show that the states convergence to a mixture of the true pattern cross-talk contributions from other patterns.

3. *Catastrophic unstable:* In this mode, the correct factors can no longer be retrieved. The temporal correlation is also low except among neighboring timepoints, indicating large changes in the overall state.

Prior analysis of resonator networks by Kent et al. (2020) also observed the presence of accuracy-reducing limit cycles, which may be an effect of the large "jumps" in the space induced by the discrete-time dynamics. These jumps are doubly harmful, because the convergence criteria were based on fixed-point convergence, making drift a failure mode. The FOAM model dynamics, however, may not settle to a fixed point yet stay in the correct basin of attraction. In summary, the improved capacity can be explained by the existence of a "benign unstable" mode not present in the discrete-time dynamics.

3.3 Example application: subset sum problem

Finally, we evaluate the performance of the FOAM model on a combinatorial optimization problem – the subset sum problem. Concretely, the problem asks: if given a target integer T and a multi-set of integers $\mathcal{S}$, is there a subset $\mathcal{S}^* \subseteq \mathcal{S}$ that sums to T? The subset sum problem is simple to state but under mild restrictions (e.g., positive only integers) is NP-complete. In fact, it can be shown that there are interpretable computations that can map the subset sum problem onto the vector factorization problem. Figure 3 provides an example of FOAM finding a successful solution. The state vectors, corresponding to each item in the set, become similar over time to one of two vectors, indicating the decision to include or omit (Figure 3A). This can be visualized by the evolution of the "confidence" (Figure 3B), which shows the normalized difference in inner products between the two different factors. The dimension required to achieve successful factorization is relatively small ($D = 50$), which speaks to the relative computational efficiency of the approach. Additional information about the methods are provided in Appendix B

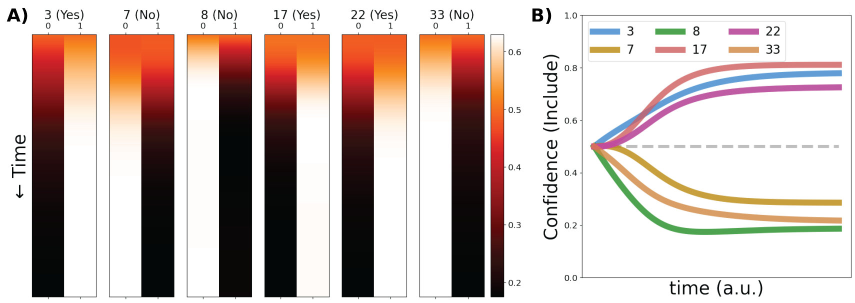

Figure 3: **Demonstration on the subset sum problem.** $T = 42, \mathcal{S} = \{3, 7, 8, 17, 22, 33\}, \mathcal{S}^* = 3, 17, 22$. FOAM model is able to factor out the three variables that should be summed together to obtain an exact match. **A)** We initialize one state vector per item in the set, with two possibilities: include (1) or omit (0). We find that over time the states converge to one variable, showing the "decision over the binary variable." **B)** Evolution of the "confidence" as the network involves in time. The confidence is the normalized difference in amplitudes of inner products. States are initialized to be halfway state. This particular demo is with vectors of dimension $D = 50$, which is intended to be smaller than the total search space, 2^6. If the dimension were higher, the final confidence for each member of $\mathcal{S}$ would also be higher.

4 Discussion

We propose a continuous-time dynamical system that implements a factorization mechanism based on interconnected associative memories. It solves the problem of combinatorial search over a large space of possible solutions by finding a set of factors that together must satisfy a number of constraints. The search is implemented through the non-linear dynamics of a network of coupled oscillators, where the dynamics act directly on the phases of individual components. The system exhibits higher memory capacity compared to existing methods, and we provide empirical evidence to support its success in large combinatorial search spaces. The model leads to practical optimization solutions in several settings, such as the subset sum problem.

There is a rich history of solving computational problems, including combinatorial optimization problems, using oscillators (Wang & Roychowdhury, 2019; Bybee et al., 2023) and resonator networks (Kleyko et al., 2022; Kymn et al., 2024b). More recently, there has also been increasing interest in using oscillations in artificial neural networks to extract semantic information from visual scenes and solve constraint satisfaction problems (e.g., Bybee et al. (2022); Keller & Welling (2023); Liboni et al. (2025); Löwe et al. (2024); Miyato et al. (2024)). Our work complements these approaches by showing that structured compositional representations, when combined with oscillatory dynamics, lead to efficient and transparent computations.

We envision that the present study can be extended in several promising ways. These include formal mathematical analysis of the dynamics introduced here, implementation in a neuromorphic circuit, and further evaluation on combinatorial optimization problems to quantitatively characterize scaling.

Acknowledgments

We thank our reviewers for their constructive and insightful feedback. We also thank members of the Redwood Center for Theoretical Neuroscience for helpful discussions. The work of CJK was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate (NDSEG) Fellowship Program. The work of CJK, CB, ZY, and BAO was supported by the Center for the Co-Design of Cognitive Systems (CoCoSys), one of seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, as well as NSF awards 2147640 and 2313149. DK has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 839179. CB is also supported by the Swartz Postdoctoral Fellowship for Theoretical Neuroscience provided by the Swartz foundation.

References

Connor Bybee and Friedrich Sommer. Optimal oscillator memory networks. In *Neuro-Inspired Computational Elements Conference*, pp. 81–83, 2022.

Connor Bybee, E Paxon Frady, and Friedrich T Sommer. Deep learning in spiking phasor neural networks. *arXiv preprint arXiv:2204.00507*, 2022.

Connor Bybee, Denis Kleyko, Dmitri E Nikonov, Amir Khosrowshahi, Bruno A Olshausen, and Friedrich T Sommer. Efficient optimization with higher-order Ising machines. *Nature Communications*, 14(1):6033, 2023.

E Paxon Frady, Spencer J Kent, Bruno A Olshausen, and Friedrich T Sommer. Resonator networks, 1: An efficient solution for factoring high-dimensional, distributed representations of data structures. *Neural Computation*, 32(12):2311–2331, 2020.

Benjamin Hoover, Duen Horng Chau, Hendrik Strobelt, Parikshit Ram, and Dmitry Krotov. Dense associative memory through the lens of random features. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 23549–23576. Curran Associates, Inc., 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/29ff36c8fbed10819b2e50267862a52a-Paper-Conference.pdf`.

John J Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, 81(10):3088–3092, 1984.

John J Hopfield and David W Tank. "Neural" computation of decisions in optimization problems. *Biological Cybernetics*, 52(3):141–152, 1985.

Silvio Kalaj, Clarissa Lauditi, Gabriele Perugini, Carlo Lucibello, Enrico M Malatesta, and Matteo Negri. Random features hopfield networks generalize retrieval to previously unseen examples. *arXiv preprint arXiv:2407.05658*, 2024.

T Anderson Keller and Max Welling. Neural wave machines: Learning spatiotemporally structured representations with locally coupled oscillatory recurrent neural networks. In *International Conference on Machine Learning (ICML)*, pp. 16168–16189, 2023.

Spencer J Kent, E Paxon Frady, Friedrich T Sommer, and Bruno A Olshausen. Resonator networks, 2: Factorization performance and capacity compared to optimization-based methods. *Neural Computation*, 32(12):2332–2388, 2020.

Patrick Kidger. *On Neural Differential Equations*. PhD thesis, University of Oxford, 2021.

Denis Kleyko, Connor Bybee, Christopher J Kymn, Bruno A Olshausen, Amir Khosrowshahi, Dmitri E Nikonov, Friedrich T Sommer, and E Paxon Frady. Integer factorization with compositional distributed representations. In *Neuro-Inspired Computational Elements Conference*, pp. 73–80, 2022.

Christopher Kymn, Sonia Mazelet, Anthony Thomas, Denis Kleyko, Edward Frady, Fritz Sommer, and Bruno Olshausen. Binding in hippocampal-entorhinal circuits enables compositionality in cognitive maps. *Advances in Neural Information Processing Systems*, 37:39128–39157, 2024a.

Christopher J Kymn, Sonia Mazelet, Annabel Ng, Denis Kleyko, and Bruno A Olshausen. Compositional factorization of visual scenes with convolutional sparse coding and resonator networks. In *2024 Neuro Inspired Computational Elements Conference (NICE)*, pp. 1–9. IEEE, 2024b.

Christopher J Kymn, Denis Kleyko, E Paxon Frady, Connor Bybee, Pentti Kanerva, Friedrich T Sommer, and Bruno A Olshausen. Computing with residue numbers in high-dimensional representation. *Neural Computation*, 37:1–37, 2025.

Luisa H B Liboni, Roberto C Budzinski, Alexandra N Busch, Sindy Löwe, T Anderson Keller, Max Welling, and Lyle E Muller. Image segmentation with traveling waves in an exactly solvable recurrent neural network. *Proceedings of the National Academy of Sciences*, 122(1):e2321319121, 2025.

Sindy Löwe, Phillip Lippe, Francesco Locatello, and Max Welling. Rotating features for object discovery. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Takeru Miyato, Sindy Löwe, Andreas Geiger, and Max Welling. Artificial Kuramoto oscillatory neurons. *arXiv:2410.13821*, 2024.

Matteo Negri, Clarissa Lauditi, Gabriele Perugini, Carlo Lucibello, and Enrico Malatesta. Storage and learning phase transitions in the random-features hopfield model. *Physical Review Letters*, 131(25):257301, 2023.

Takashi Nishikawa, Ying-Cheng Lai, and Frank C Hoppensteadt. Capacity of Oscillatory Associative-Memory Networks with Error-Free Retrieval. *Physical Review Letters*, 92(10): 108101, 3 2004. doi: 10.1103/PhysRevLett.92.108101. URL `https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.92.108101`.

Andre Noest. Phasor neural networks. In *Neural Information Processing Systems (NeurIPS)*, 1987.

Andre Noest. Discrete-state phasor neural networks. *Physical Review A*, 38(4):2196, 1988.

Tony Plate. Holographic recurrent networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 34–41, 1992.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 20, pp. 1–8, 2007.

Tianshi Wang and Jaijeet Roychowdhury. OIM: Oscillator-based Ising machines for solving combinatorial optimisation problems. In *International Conference on Unconventional Computation and Natural Computation (UCNC)*, pp. 232–256, 2019.

A FORMULATION FOR AN ARBITRARY NUMBER OF FACTORS

The two-module network described in Equation 2 is provided as a simple example. In general, the dynamical system provided here can support an arbitrary number of discrete modules. Some previous work has investigated the scaling capacities of resonator networks in the presence of multiple modules.

For the more general case, suppose we have a vector $\mathbf{s}$ that is the product of R different modules:

$$\mathbf{z} = \bigodot_{r=1}^{R} \mathbf{x}^{(r)} \tag{6}$$

where each $\mathbf{x}^{(r)}$ is a D-dimensional vector. Then the phase dynamics for each module $\phi^{(r)}$ are as follows:

$$\frac{d}{dt}\phi_j^{(r)} = |h_j^{(r)}| \sin\left(\arg(h_j^{(r)}) - \phi_j^{(x)}\right) \tag{7}$$

where

$$h_j^{(r)} = \sum_{k=1}^{D} W_{jk}^{(y)} \; (s_j \bigodot_{r' \neq r} e^{-i\phi_j^{(r')}}) \tag{8}$$

While equation 7 appears structurally similar to equation 2, the pre-activation term, $h_j^{(r)}$, now includes additional multiplication terms due to contributions from other factors.

B EXPERIMENTAL DETAILS: SUBSET SUM PROBLEM

To solve the subset sum problem, we make use of the fact that multiplying two complex numbers adds their arguments. The problem formulation of subset sum as a vector factorization problem mirrors the setup in Kymn et al. (2025), which did not use the FOAM architecture proposed here.

To formulate the subset sum problem as a vector factorization problem, we first define a randomized mapping from integers to phasor-valued vectors. We first draw a seed vector of random phases: $\mathbf{z} = [e^{i\phi_1}, e^{i\phi_2}, ..., e^{i\phi_D}]$, in order to define an encoding of integer values:

$$\mathbf{z}(c) = [e^{i\phi_1 c}, e^{i\phi_2 c}, ..., e^{i\phi_D c}] \tag{9}$$

This technique is known as random Fourier features (Rahimi & Recht, 2007) or trajectory association (Plate, 1992). This encoding has the useful algebraic property that $\mathbf{z}(c_1) \odot \mathbf{z}(c_2) = \mathbf{z}(c_1 + c_2)$.[3]

Therefore, to formulate the subset sum problem, we generate an encoding of our target, i.e., $\mathbf{z}(T)$. We then generate $|\mathcal{S}|$ factors, each with two patterns stored in the memory: $\mathbf{z}(0)$ and $\mathbf{z}(\mathcal{S}_i)$. These two patterns reflect the binary decision to include the item in the set (or not). We initialize the FOAM network states to be the average of these two vectors, although we observe strong convergence empirically when these states are initialized randomly.

NOTES

[1]Noest (1987) also introduces discrete-time dynamics for phasor neural networks, and that the asynchronous version is observed to have the same Lyapunov function as the original version.

[2]The energy values correspond to evaluations of the energy function from Noest (1987) according to the weights for each module. Although our *factorizing* dynamics do not in general monotonically descend this energy function, as we will show, measuring its changes over time helps us qualitatively characterize differences in behavior based on pattern load.

[3]The intersection of random features and associative memory has received much attention in recent years (e.g., Negri et al. (2023); Kalaj et al. (2024); Hoover et al. (2024)) and would be an exciting direction for future work.