

Spider web-inspired sensing and computation with fiber network physical reservoirs

Journal of Intelligent Material Systems and Structures

1–12

© The Author(s) 2026

Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/1045389X261470871

journals.sagepub.com/home/jim



Apoorva Khairnar¹, Yogesh Phalak¹, Jun Wang¹, Ziyang Zhou¹, Benjamin Jantzen^{2,3}, Suyi Li¹  and Noel Naughton¹ 

Abstract

Physical reservoir computing leverages the intrinsic dynamics of mechanical systems to perform computation through their natural responses to input signals. Here, we study a compliant fiber network inspired by orb-weaving spider webs and investigate how its mechanical design and operating conditions shape its computational capability. Using Cosserat rod-based simulations, we identify how network topology, geometry, actuation, and axial tension impact the nonlinear computation and memory capacity of the network. We further evaluate several readout reduction strategies to assess how computational performance varies with the number and placement of measured outputs. We then experimentally validate these results using a physical fiber-network prototype. Overall, results provide insights and guidance on design, actuation, and sensing choices to enable fiber networks for mechano-intelligent computation. They demonstrate the ability of structured compliant fiber networks to serve as physical reservoirs capable of nonlinear transformation and input-history retention.

Keywords

physical reservoir computing, mechanical intelligence, bio-inspiration, fiber network

Introduction

Spider webs, in particular the well-structured webs built by orb-weaving spiders, consist of networks of slender, flexible fibers. These webs serve a function critical to the spider's survival, allowing the spider both to capture prey and sense its local environment. Spiders have evolved a unique ability to interface with and extract information about their environment by sensing vibrations that travel along the web via hairs located on their legs that are highly sensitive to local fiber vibrations. Interfacing with the mechanical information embedded in the web enables spiders to localize prey location, identify structural damage, and distinguish between different types of web deformation (e.g. caught prey vs external environmental disturbance) (Landolf and Barth, 1996; Masters, 1984; Mortimer, 2019; Wu et al., 2023).

In this regard, the web serves as an information processing and message passing substrate, mechanically transforming input deformations as a form of mechano-computation (Wang et al., 2023), which the spider uses to offload complex sensing tasks. Inspired by such webs, we consider the information processing

capability of an engineered network of compliant fibers through the lens of physical reservoir computing.

Physical reservoir computing leverages the nonlinear dynamic properties of physical systems to perform complex computations (Tanaka et al., 2019). Compliant mechanical systems in particular have attracted significant interest due to the ability of their nonlinear dynamics to be leveraged to exhibit 'mechanically intelligent' behavior (Bhovad and Li, 2021; Degraeve et al., 2015; Khairnar et al., 2025; Liu et al., 2023; Nakajima et al., 2013, 2014, 2018; Wang and Li, 2023). By offloading taxing computational tasks to the nonlinear physical dynamics, such systems can simplify sensing

¹Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA, USA

²Department of Philosophy, Virginia Tech, Blacksburg, VA, USA

³Department of Computer Science, Virginia Tech, Blacksburg, VA, USA

Corresponding author:

Noel Naughton, Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA 24061, USA.

Email: nnaughton@vt.edu

Data Availability Statement included at the end of the article

and control tasks (Pfeifer and Bongard, 2006; Ulrich, 1988).

Reservoir computing was originally inspired by recurrent neural structures in the brain and focused on how the nonlinear dynamics of recurrent neural networks with fixed internal weights serve to separate input streams into a high-dimensional state space from which arbitrary functions can be learned via a simple linear output layer of weights (Jaeger, 2001; Maass, 2011; Nakajima and Fischer, 2021; Schrauwen et al., 2007). In time, this insight has been generalized to the realization that any nonlinear dynamical system similarly performs this transformation on the input, meaning that any sufficiently complex, nonlinear dynamical system can serve as a reservoir computer.

Physical reservoir computing has applied this insight to physical systems, using the nonlinear dynamics of complex systems to perform computation (Konkoli et al., 2018; Nakajima, 2020; Tanaka et al., 2019). This approach has achieved notable recent interest due to its potential to provide a scalable pathway toward embedding computation into material and structural systems, laying the groundwork for new forms of distributed, embodied intelligence. Notable examples include nonlinear spring-mass networks capable of emulation and pattern generation tasks (Dion et al., 2018; Du et al., 2017; Hauser et al., 2011, 2012; Urbain et al., 2017; Yamanaka et al., 2018; Yamane et al., 2015), architected materials that respond predictably to input stimuli (Caluwaerts and Schrauwen, 2011; Paul, 2004, 2006; Zhang and Wang, 2022), origami-based mechanical systems that extract complex information from its structural dynamics (Bhovad and Li, 2021; Liu et al., 2023; Wang and Li, 2023, 2025), and soft robotic systems that integrate sensing and actuation to process information (Degraeve et al., 2015; Li et al., 2012; Nakajima et al., 2013, 2014, 2018; Wang et al., 2025).

Critically, the performance of a physical reservoir depends strongly on the design and operating conditions of its mechanical substrate. In our prior work, we identified that networks of compliant, overlapping fibers inspired by spiderwebs are capable of serving as physical reservoir computers (Khairnar et al., 2025). Here, through both simulations and experiments, we investigate the effects of key topological, geometric and actuation features of such networks to define fundamental design principles and operational ranges for their ability to serve as physical reservoir computers. Further, we consider methods to reduce the dimensionality of the fiber network readouts with minimal loss of performance in order to reduce data redundancy and simplify the fiber network's implementation.

Overall, our findings indicate that networks of flexible fibers inspired by spider webs can effectively serve as physical reservoir computers. This positions such fiber network reservoirs to potentially serve as unique, physically intelligent structures capable of serving as a

low-cost and easy to manufacture compliant sensing technology with potential applications in robotics, autonomous systems, and structural monitoring.

Methods

Fiber network setup

Inspired by the structured webs of orb-weaving spiders (Figure 1(a)), two fiber network topologies are considered: crosshatch and polygonal. Fiber networks consist of slender, compliant fibers arranged in overlapping architectures with intersecting fibers bonded together. Crosshatch topologies consist of an $N \times N$ fiber arrangement of horizontal and vertical fibers (Figure 1(b)) while polygonal topologies consist of a N -sided polygon where fibers connect all non-neighborly vertices (Figure 1(c)). Each fiber spans the network with one end fixed in place while the other end is pre-tensioned and attached to a spring to facilitate fiber deformations.

To actuate the fiber networks, a lateral mechanical deformation input $u(t)$ is applied to the midpoint of a single fiber (green arrows in Figure 1(b) and (c)). The input $u(t)$, which in general may be any continuously varying function, consists of a timeseries constructed by fitting a cubic spline to random points sampled at regular frequency from a uniform distribution. This approach allows generation of a continuous and smooth input that is compatible with the elastic dynamics of the fiber while still ensuring that any structure observed in the output arises from the reservoir itself rather than from correlations in the input (Dambre et al., 2012). The x-y displacements at m readout locations on the fiber network are tracked as the physical reservoir output $\mathbf{x}(t) = \{x_i(t), y_i(t) \mid i \in 1 \dots m\}$ where $x_i(t), y_i(t) \mapsto \mathbb{R}$ and m is determined based on the number of fibers in the network (i.e. $\mathbf{x}(t)$ has $2m$ entries). The readout locations are generally selected to be any location where two fibers cross (nodes) as well as midpoints on fibers between any two nodes.

Quantifying physical reservoir capacity

Reservoir computers are universal filters (Nakajima and Fischer, 2021) such that, for an arbitrary, real-valued nonlinear functional $z[u(t)]$, there exists a mapping, $\mathcal{W}: \mathbf{x}(t) \mapsto z[u(t)]$ where $\mathcal{W}$ is linear in $\mathbf{x}(t)$. An approximation of such a map,

$$\hat{z}(t) = \mathbf{W} \cdot \mathbf{x}(t) \quad (1)$$

can be identified by estimating the linear weights $\mathbf{W} \in \mathbb{R}^{2m}$ through ridge regression. Here ridge regression with a train/test split of 75/25 is used in all cases with splits performed chronologically (i.e. no data shuffling) to prevent information leakage from future samples into training.

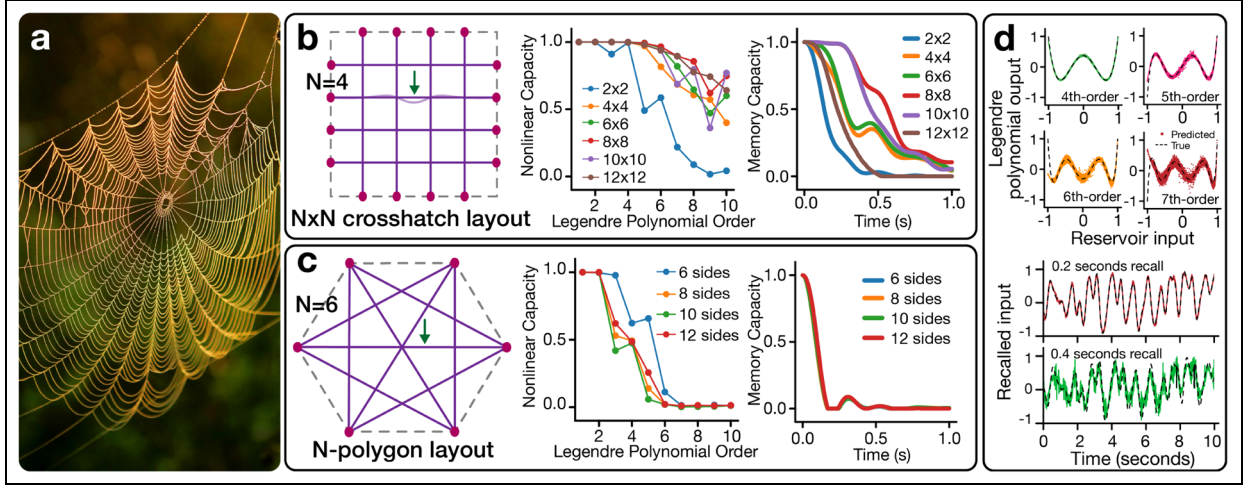


Figure 1. Spider web-inspired fiber network physical reservoirs. (a) Spider webs transmit mechanical perturbations from captured prey and the surrounding environment to spiders who process the mechanical information via sensitive hairs on their legs. (b) Cross hatch networks consisting of a $N \times N$ network of horizontal and vertical fibers were examined. Simulation results for an increasing number of N fibers show improvement in both nonlinear and memory capacity, with larger networks generally exhibiting better performance than smaller networks, though occasional performance degradation (e.g. memory capacity of 12×12 network) is observed. (c) Polygonal networks, where fibers connect all non-neighboring vertices on a N -sided polygon were also considered. Polygons with more vertices, and so more fibers in the network, did not produce improved performance like in the crosshatch networks. The hexagon ($N = 6$) network exhibits the best nonlinear capacity while all network topologies had similar memory capacity. (d) Example computations of computing nonlinear Legendre polynomials and recalling past inputs for the 8×8 crosshatch network, which exhibits the best all-around performance of the network topologies investigated. Black dashed lines denote true values.

In practice, the types of functions that can be learned are limited by the dynamics of the particular reservoir. Our goal then is to probe the capability of a fiber network to serve as a physical reservoir computer. In line with our prior work (Khairnar et al., 2025), we do so by defining a general capacity metric capable of quantifying the reservoir’s ability to accurately learn a linear map $\mathbf{W}$ that provides an approximation $\hat{z}(t) = \mathbf{W} \cdot \mathbf{x}(t)$ of the functional $z[u(t)]$ over the time interval $[a, b]$ (Dambre et al., 2012).

$$c[z] = 1 - \frac{\int_a^b (\hat{z}(t) - z[u(t)])^2 dt}{\int_a^b (z[u(t)] - \bar{z})^2 dt} \quad (2)$$

where $\bar{z}$ is the mean of $z[u(t)]$ over the time interval $[a, b]$. We then define two specific metrics that quantify the computational capability of the reservoir: nonlinear capacity and memory capacity.

Nonlinear capacity measures the reservoir’s ability to perform nonlinear computations on the current input $u(t)$ into the network. Here, we consider the ability of the reservoir to learn Legendre polynomials $P_k(u(t))$ of order k , such that $z_k[u(t)] = P_k(u(t))$ for $k \in \{1, \dots, N\}$ where $N = 10$. Legendre polynomials provide a complete and orthogonal basis set that spans the interval $[-1, 1]$ and so provide not only a challenging test of a fiber network’s nonlinear capability, but also indicate the ability of the reservoir to approximate any

smooth function over this interval due to the efficiency with which Legendre polynomials represent such functions. We additionally define an total nonlinear capacity of the reservoir as the sum of the individual capacity metrics $C_{nl} = 1/N \sum_{k=1}^N c[z_k]$.

Memory capacity measures the reservoir’s ability to generate linear mappings that recall previous inputs. In this case, $z_\tau[u(t)] = u(t - \tau)$ where $\tau \in [0, T]$ is a time lag. We consider recall up to 1 s in the past ($T = 1.0$ s). Finally, we define the total memory capacity of the reservoir as the integration of the individual capacity metrics over the recall interval $C_m = 1/T \int_{\tau=0}^T c[z_\tau]$.

Simulation experiments

To investigate the role of different fiber network parameters, we perform numerical simulations of fiber network dynamics. We model fiber networks as assemblies of Cosserat rods, which provide a computationally efficient description of elastic fibers as one-dimensional filaments while still allowing the full range of deformation modes: bending, twisting, stretching, and shearing. Each fiber is modeled as an individual Cosserat rod with a diameter of 2 mm, Young’s modulus of 100 MPa, and a density of 1000 kg/m³. The diameter and Young’s modulus were selected to ensure numerical stability in the simulations. At every crossing, fibers are coupled using a zero-displacement spring-damper

boundary condition (Zhang et al., 2019), which enforces connection at the intersection points. Each fiber is fixed at one end, while the other end is free to translate but loaded with a constant tensile force F_t to apply a constant tension to the fiber. This force F_t is independent of displacement. Rather, it is applied as a constant force to the fiber end even as the fiber translates axially and rotates in the fiber network plane.

To actuate the network, we apply a time-dependent external force $F_{ex}(t) = F_{max}u(t)$ at the midpoint of the central horizontal fiber where $u(t)$ is generated via 5 Hz random sampling. For all simulations, the timestamp sampling uses a fixed seed such that every network is actuated using the same $u(t)$ with only the force magnitude varied between cases.

Each simulation was 100 s long and sampled at 250 Hz with recorded measurements consisting of the planar x-y displacements at fiber intersection points and with the midpoints of fiber segments between intersections. These measurements define the physical reservoir state $\mathbf{x}(t) \in \mathbb{R}^{2m}$. The resulting input-output data is then used within the physical reservoir computing procedure described earlier to evaluate the capacity metrics. The ridge regularization parameter α is 0.01 for all simulations except for output feature analysis where it is 0.1 as empirical testing found the best generalization agreement (similar train and test scores) at this level for the output feature analysis. Additional cross-validation results supporting these choices are provided in the Supplemental Information.

We implement this numerical modeling approach using *Elastica* (Gazzola et al., 2018; Naughton et al., 2021; Zhang et al., 2019), an open-source Cosserat-rod framework in Python that has been applied to and validated against a wide set of fibrous-dynamics problems, including animal locomotion and manipulation (Tekinalp et al., 2024; Zhang et al., 2021), fibrous structures (Bhosale et al., 2022; Khairnar et al., 2025), and soft robotic control (Chang et al., 2023; Naughton et al., 2021, 2026; Shih et al., 2023).

Physical experiments

Physical experiments were performed for the cross-hatch network topology. Fibers consisted of nylon monofilament with lengths of 520 mm and diameters of 0.45 mm. One end of each fiber is fixed, while the other is connected to linear springs ($k = 52.5$ N/m) to maintain consistent tension (Figure 5(a)). Fiber intersections are bonded using a silicone rubber adhesive (Sil-Poxy).

The network was actuated at the midpoint of the central horizontal fiber using a Dynamixel AX-18A servomotor. A lever attached to the servomotor was then attached to the horizontal fiber via a thread, allowing rotation of the servomotor to produce controlled actuation of the fiber network with minimal out-of-plane

displacement (Figure 5(b)). For these experiments, the input $u(t)$ is defined as the normalized vertical displacement of the actuation point rather than the applied force time-series used in simulation since the applied force at the input location is not directly measured. The input actuation was generated by modulating the motor joint angle using a cubic spline fitted to random inputs at regularly spaced intervals. The same input profile type was used to modify the motor joint angle as was used to generate the force profile for the simulation experiment (e.g. same random seed), but adjusted such that the profile was generated by fitting a spline to uniform random data sampled at a 2 Hz rate. The resulting spline was then sampled at 100 Hz to generate the input $u(t)$.

Each trial ran for 150 s and was recorded at 120 FPS in HD (1920×1080) using a Sony $\alpha 7$ III mirrorless camera. To measure the reservoir state, red optical tracking markers were attached to fiber intersection points and at the midpoints between intersections, and motion tracked using custom Python code to extract marker trajectories over time. The x-y displacements of all tracked marker locations were then used to form the reservoir output state $\mathbf{x}(t)$, which was used to compute the same capacity metrics as in the simulation experiments. For all experiments, the ridge regularization parameter α is fixed at 1 based on empirical testing of generalization performance.

Results

To identify key fiber architectural features, we first use simulation to explore a wide range of parameter combinations. This allows us to efficiently identify a feature's impact on the mechanical computation and physical reservoir capability of the fiber network setup.

Fiber network topology

We first consider the role of network topology on reservoir capacity. We compare crosshatch and polygonal networks by sweeping $N \in \{2, 4, 6, 8, 10, 12\}$ for the crosshatch case and $N \in \{6, 8, 10, 12\}$ for the polygon case while keeping the nominal network length (outer edge distance) constant. As fiber density increases, the spacing between fiber intersection nodes decreases. This effectively stiffens the network, making it more resistant to deformation for the same input force. To ensure consistent levels of deformation across networks, we scale the actuation force to produce a consistent lateral deflection $\delta = 13.3$ mm based on a model of a simply supported Euler-Bernoulli beam

$$F_{max} = \frac{48\delta EI}{s^3} \quad (3)$$

where E is the fiber Young's modulus, I is the second area moment of inertia, and s is the distance between the two connection nodes of the fiber section where the input force is applied.

For crosshatch networks (Figure 1(b)), nonlinear capacity increases as networks become denser up to 8×8 networks before saturating, with limited to no improvement for the densest networks. Nonlinear capacity improvement is achieved through improved estimation of Legendre polynomials of all orders. In particular, very high capacity (>0.99) is achieved for progressively higher Legendre polynomials as network density increases. Similar improvement is observed for memory capacity, which rises steadily as fiber density increases up to the 8×8 network before plateauing for the 10×10 network and decaying substantially for the 12×12 network. Larger networks generally support longer recall horizons (up to 0.5 s), indicating improved memory of larger delays. This improved performance is a coupled effect of both an increasing number of read-outs as well as increased dynamical richness as fiber network density increases (see SI). The computational performance of the 8×8 network is illustrated in Figure 1(d). It shows the performance of the network when challenged to estimate Legendre polynomials of orders four ($c = 0.999$) through seven ($c = 0.896$) as well as recall past inputs with time delays of 200 ms ($c = 0.997$) and 400 ms ($c = 0.713$).

In contrast, polygonal networks do not exhibit the same scaling with fiber density (Figure 1(c)) indicating that network structure, and not just size, plays an important role in reservoir computational ability. All polygonal networks exhibit similar nonlinear and memory performance (though the hexagonal network moderately outperforms the others in nonlinear capacity). Notably, both nonlinear and memory capacity is on par with or below the worst performing crosshatch (2×2) network. Together, these results indicate the crosshatch topology consistently outperforms a polygonal topology under the conditions tested. As such, our remaining analyses focus on crosshatch networks.

Input force scaling and fiber spacing

Matching input and reservoir dynamics is well established to be critical to physical reservoir performance. In Figure 1, we heuristically scaled the input force to produce comparable lateral deflections at the input location. To investigate how variations in the input force impact reservoir performance, in Figure 2(a) we vary input force scaling and fiber spacing in a 4×4 crosshatch network to identify its impact on total nonlinear capacity C_{nl} and total memory capacity C_m (the sum of the individual capacities). The spacing between nodes was varied from 20 to 150 mm. For each spacing, a reference actuation force was determined via equation

(3) (black dashed diagonal lines) and all spacing-force combinations were then simulated.

For a fixed spacing, increasing actuation force tends to gradually improve both nonlinear and memory capacity up to a point before a strong drop off occurs (Figure 2(a)). This drop off forms a ridge in the heat map that is consistent for both nonlinear and memory capacity. Notably, the maximum performance does not follow the constant displacement scaling but rather appears clustered right at the edge of the performance ridge. This is broadly true for nonlinear capacity while memory capacity appears to also be bounded by fiber spacing, with a strong peak in performance occurring for intermediate spacings of 60–100 mm.

It has been commonly observed that reservoir performance is often maximal when the reservoir operates at the ‘edge of chaos’ (Boedeker et al., 2012; Legenstein and Maass, 2007). Our results appear to be consistent with this, as the best reservoir performance is found right before a strong drop beyond which performance is permanently decayed. To identify the source of this performance drop, we consider the role of fiber buckling. Specifically, we consider buckling in the two vertical fibers on either side of the horizontal fiber where the input force is applied. In this context, the fibers can be viewed as columns, each subjected to a maximum compressive load $P = F_{max}/2$ that splits the maximum actuation force. Based on this, we define a non-dimensional buckling number $\mathcal{B}$ that describes the onset of buckling in the fiber ($\mathcal{B} \geq 1$)

$$\mathcal{B} = \frac{Ps^2}{\pi^2 EI}. \quad (4)$$

In Figure 2(b), the total capacity results of Figure 2(a) are replotted against this buckling number, showing the onset of the performance ridge seen in the heatmaps of Figure 2(a) corresponds with the onset of buckling $\mathcal{B} = 1$ in these vertical fibers. The existence of buckling is further visually confirmed in the renderings of fiber dynamics, clearly showing visibly buckled configurations once this threshold is crossed.

We briefly note that, post-buckling, there is a region in the memory capacity heatmap of Figure 2(a) for large fiber spacing and large force magnitudes where memory performance appears to recover. This suggests there may be a post-buckling region of large fiber spacing where memory capacity can be recovered. However, the uniformly poor nonlinear capacity of post-buckling networks limits the utility of such a regime and so it is not considered in detail here. Post-buckling configurations can be highly sensitive to their initial positions but can also propagate their buckling behavior through the network in consistent ways. The partial recovery of memory but no recovery of nonlinear computation suggests that information related to these two metrics may

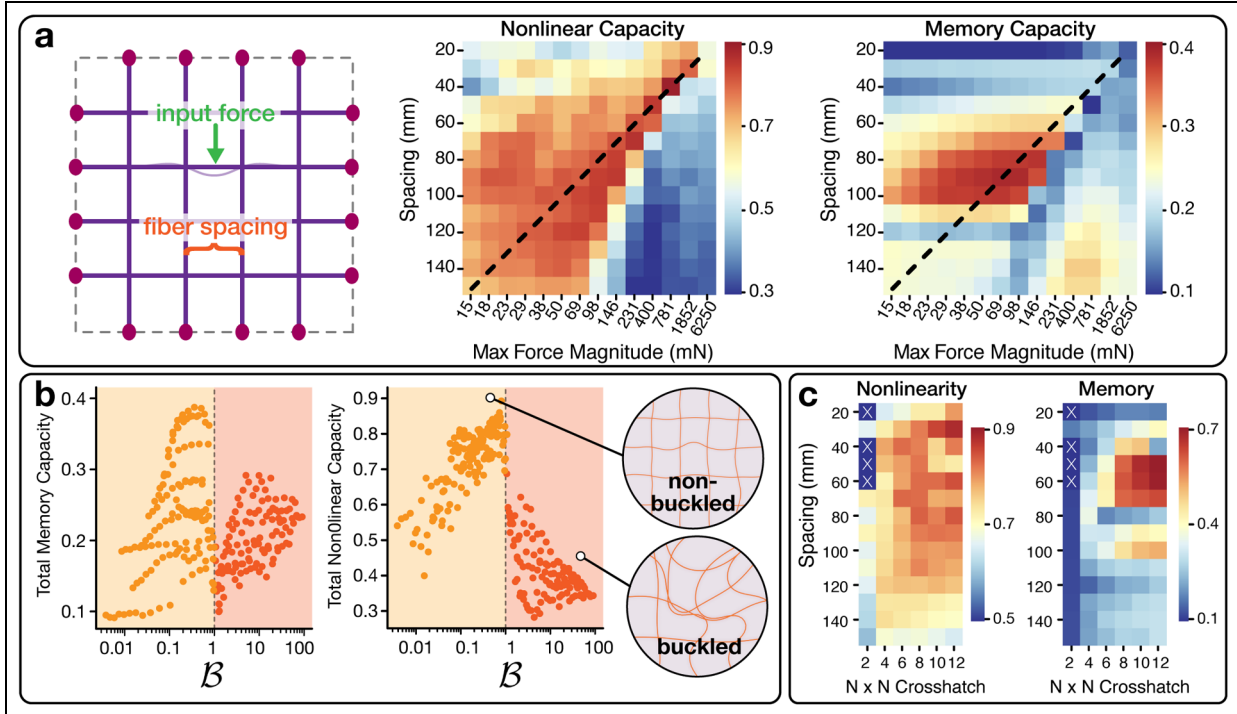


Figure 2. Crosshatch performance depends on input dynamics matching fiber dynamics. (a) For a 4×4 crosshatch network, the input force and spacing between fibers was varied. For each fiber spacing, an input force was selected to produce a constant lateral deflection of 13.3 mm for a Euler-Bernoulli simply supported beam (dashed diagonal lines). These input forces were also applied for all other fiber spacings and the total nonlinear capacity (left) and total memory capacity (right) were computed for each case. For each spacing, nonlinear and memory capacity tends to increase for increasing input force before exhibiting a rapid deterioration. (b) Plot of the total memory capacity (left) and total nonlinear capacity (right) versus the Euler-Bernoulli buckling criteria where values greater than unity predict buckling onset. The strong performance deterioration in computational capacity observed in (a) is found to be associated with the onset of buckling ($B > 1$) in the vertical fibers. (c) Computational performance of networks with increasing number of fibers using the input force associated with the best performance for that fiber spacing in (a). X's indicate numerically unstable simulations that failed to solve. Overall, improved performance is exhibited as the fiber network size increases, with a broad improvement in nonlinear capacity for nearly all fiber spacing while memory performance improvement is more concentrated in fiber spacings between 50 and 70 mm apart.

be located in these different aspects of the fiber network's dynamics.

Based on these results, we identified the unbuckled ($B < 1$) force-spacing pairs that yielded the best total nonlinear capacity and best total memory capacity and used them to investigate the impact of fiber network size for a fixed spacing between nodes (the results of Figure 1 are for a fixed fiber length with increasingly smaller spacing). $N \times N$ networks of $N \in \{2, 4, 6, 8, 10, 12\}$ were considered (Figure 2(c)).

Generally, both nonlinear capacity and memory capacity increase with network size, though with some notable caveats. While nonlinear capacity increased with number of fibers for spacings less than 120 mm, for spacings greater than this, there is instead a decrease in nonlinear capacity. Improvement in nonlinear capacity is most rapid up to networks of 8×8 before plateauing. Further, the improvement in nonlinear capacity is more broad-based than the improvement in memory

capacity, with improvement in memory capacity restricted to spacings between 50 and 70 mm, similar to the higher performance band between spacings of 60 and 100 mm identified in Figure 2(a).

Fiber pretension

Prior experimental results have indicated that the axial tension applied to each fiber, F_t , can strongly influence memory capacity (Khairnar et al., 2025). In our setup, this pretension is applied at one end of every fiber along its axial direction. To understand the impact of this pretension, we investigated the force-spacing pairs that produced the highest memory capacity within the 50–100 mm spacing range and simulated their performance under pretension loads between 0.01 and 5 N on a 4×4 network (Figure 3). Nonlinear capacity (Figure 3(a)) monotonically decreases as both F_t and fiber spacing increase, with the best nonlinear performance

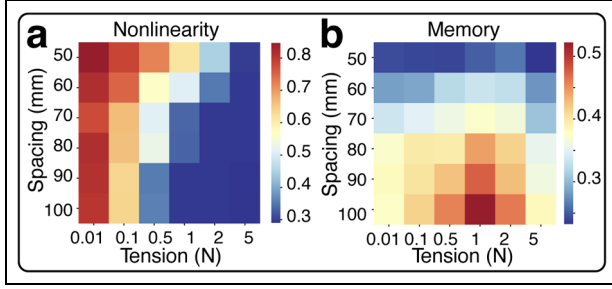


Figure 3. Fiber pretension and computational capacity. Total capacity results of a 4×4 crosshatch network under different spacing and fiber pretension levels. (a) Total nonlinear capacity declines with pretension, achieving best performance when there is no pretension applied. (b) Total memory capacity increases, generally peaking for 1 N of pretension applied to each fiber before decreasing.

occurring for minimal (0.01 N) pretension. In contrast, memory capacity (Figure 3(b)) increases with spacing (for a fixed F_t) and also initially increases with F_t before peaking at 1 N and then decreasing. Pretensioning fibers increases their resonance frequency, potentially suggesting a connection between how memory is stored in the network and fiber dynamics.

Output feature down-selection

To this point, fiber network reservoir performance has been evaluated by using the (x, y) output of fiber crossing points and the midpoints between crossings. For a $N \times N$ network, this yield $6N^2 + 4N$ total features that make up the reservoir readout layer. The use of these features as readout locations has been heuristic, focused on ensuring consistent and uniform coverage of the network. However, these readout locations exhibit large amounts of co-linearity, indicating redundant information is contained in the reservoir output vector. It is natural then to consider if a compact set of readout features can be identified to yield a more optimized reservoir design.

To do so, we adopt an empirical strategy whereby we group the readout points into three groups based on locations at either a fiber crossing point, the midpoint of a horizontal fiber section, or the midpoint of a vertical fiber section. We further consider the x- and y-displacements of each group separately, yielding 6 total subgroups of reservoir outputs. All tests were performed on a 6×6 crosshatch network with an input force magnitude of 0.85 N, a fiber length of 500 mm, and a tension force of 0.01 N.

In Figure 4(a), we examine the nonlinear and memory capacity of each output feature subgroup compared to the computational performance achieved using all of the reservoir outputs. Interestingly, fiber connection points exhibited relatively lower computational

performance compared to midpoint groups. For nonlinear capacity, performance remained relatively uniform across subgroups. Note that due to co-linearity, this does not indicate that each subgroup has unique information, simply that each subgroup contains some nonlinear information within it. The y-component of the horizontal midpoint group and the x-component of the vertical midpoint group, which together are the lateral midpoint components, were the two highest performing subgroups. For memory capacity, performance was notably highest in the y-component of the horizontal midpoint group with all other subgroups exhibiting substantially lower memory.

Based on these results, we create a feature group consisting of the lateral midpoints (y-component of the horizontal midpoint group and the x-component of the vertical midpoint group) with a total of $2N(N + 1)$ features to compare with the full output feature set. We also created two additional *ad hoc* feature groups based on spatial localization of features within the network (Figure 4(c)). The first group, termed ‘near actuation’ consists of readout locations located near the actuation point under the theory that strong dynamics occur in this region of the network. The second group consists of readouts located between the actuation points and the ends of fibers where the pretension force is applied as this region will be most free to deform compared to the opposite side where the fibers are fixed in place. This group is termed as ‘near springs’, though in these simulations the fiber ends are not attached to physical springs; they are loaded through a constant axial pretension, and the term is used only to indicate proximity to the tensioned boundary.

Of these three feature groups (Figure 4(c)), the lateral midpoint and ‘near springs’ group performs best. The lateral midpoint group achieved $>95\%$ the nonlinear computational performance and $>85\%$ the memory performance of when all the output features were used but with a two-third reduction in output features. This result indicates the majority of the mechanical computation of the fiber network is captured by the lateral bending deformations of the fibers. The ‘near springs’ group performed similarly, with slightly worse nonlinear and slightly better memory performance compared to the lateral midpoint group. The near-actuation group performed worse, achieving $\sim 85\%$ the nonlinear computation performance of when all outputs were used but with $\sim 60\%$ the memory performance.

Experimental validation

In Figure 5 we seek to confirm our simulation results via physical experiments on 4×4 and 6×6 crosshatch networks across a range of input magnitudes for a fixed total fiber length (Figure 5(a)). Motor joint angle was used as a proxy for input force, with the maximum

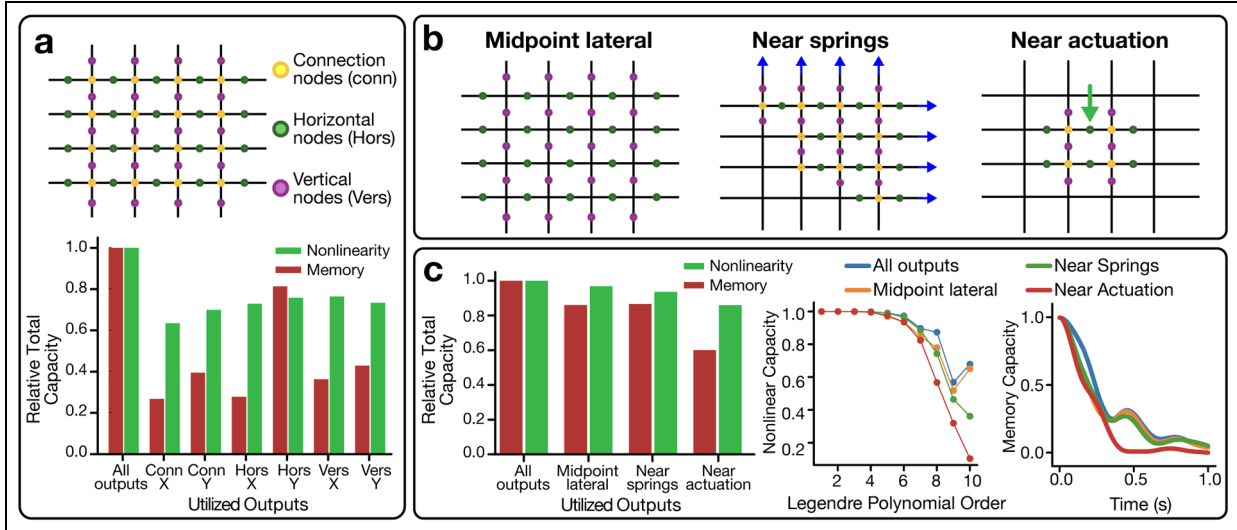


Figure 4. Readout feature reduction (a) Readout points are grouped into three groups: fiber intersection points, midpoints on horizontal fibers, and midpoints on vertical fibers. For each group, the x- and y-displacement of the fiber is tracked, yielding 6 total groups. The total nonlinear and memory capacity of each group alone is then computed and compared with the capacity achieved when all outputs are used. (b) Schematics depicting the readout locations of the midpoint lateral, near springs, and near actuation feature groups. For midpoint lateral, only the lateral direction (y-direction on horizontal fibers; x-direction on vertical fibers) is used while near springs and near actuation groups use both x- and y-displacement data for each node in the group. (c) Computational capacity of the three feature sub-groups. Overall, use of midpoint lateral features (Hors-Y and Vers-X) achieves greater than 85% the performance of all the outputs using only 35% the number of features.

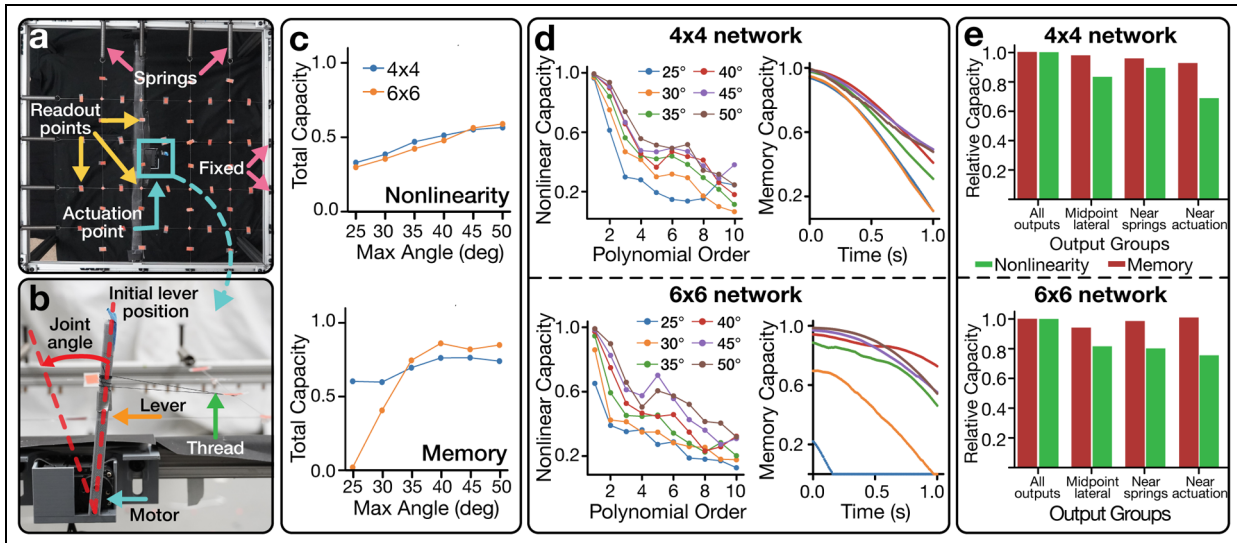


Figure 5. Physical reservoir fiber network experimental validation. (a) Overview of experimental setup for 4x4 and 6x6 crosshatch geometries investigated. (b) Schematic illustrating how the maximum motor angle corresponds to the maximum fiber deflection. (c) Total nonlinear capacity (Top) and memory capacity (Bottom) experimental results of the 4x4 and 6x6 network show both networks exhibit similar total nonlinear capacity, which increases as the maximum deflection angle increases. The 6x6 network initially exhibits poor memory capacity for small deflection angles but rapidly improves as the angle increases, eventually outperforming the 4x4 network. (d) Individual nonlinear and memory capacity results of 4x4 (Top) and 6x6 (Bottom) networks for different motor joint angles. (e) Output feature down selection of 4x4 (Top) and 6x6 (Bottom) networks. Results broadly match simulation results of Figure 4(c). All three groups showed strong retention of memory performance. 'Midpoint lateral' and 'near springs' groups show similarly strong retention of nonlinear capacity for both networks, with the 'near actuation' group showing weaker relative performance.

motor angle varied from 25° to 50° in steps of 5° (Figure 5(b)).

As actuation magnitude increases, both network sizes show improved nonlinear performance (Figure 5(c)), consistent with simulations. Both networks generally exhibit similar total nonlinear capacity for a given maximum motor joint angle. When considering the specific capacities for different Legendre polynomial orders (Figure 5(d)), as maximum joint angle increases, the networks become progressively better at approximating individual Legendre polynomials.

Notable differences in memory performance are observed between the 4×4 and 6×6 networks. For the 4×4 network, the total memory capacity increases modestly with actuation before largely plateauing for larger angles (Figure 5(c)). The 6×6 network displays a different progression. At 25° , its total memory capacity is close to zero. As actuation increases from 25° to 40° , memory increases sharply before similarly plateauing for higher angles. Notably, for these higher angles, memory capacity is substantially better than the simulated networks, with the 6×6 network able to reconstruct inputs from up to 0.5 s ago with a capacity above 0.8 (Figure 5(d)). Together, these experimental results confirm key observation from simulations that indicate that crosshatch networks of compliant fibers can effectively serve a physical reservoirs capable of both nonlinear computation and memory recall.

We further evaluate if the feature-selection insights from Figure 4 apply to physical networks. We defined the same three output feature groupings as in Figure 4(b) and computed the total nonlinear and memory capacities of the network using only readouts from these subsets (Figure 5(e)). Results for both network sizes broadly agree with the simulation results of Figure 4, with the lateral midpoint group and the ‘near springs’ group performing strongly and the ‘near actuation’ group performing worse. Notably, memory performance was retained to a higher degree than nonlinear capacity, a flip of the simulation results of Figure 4.

Finally, we tested the performance of our physical 6×6 network with a 50° maximum motor joint angle on the NARMA- n benchmark tasks, which have become a common benchmark for evaluating physical reservoir performance (Wringe et al., 2025). Experimental trajectories were down-sampled from 120 to 10 Hz and the input signal $u(t)$ was normalized to lie between 0 and 0.2. NARMA- n task trajectories for $n = 2, 5$, and 10 were then generated using the standard benchmark coefficients and update rules suggested by Wringe et al. (2025).

Figure 6(a) shows violin plots of the test set error distribution for the three different NARMA- n tasks while Figure 6(b) shows a direct comparison between the benchmark timeseries and the reservoir estimate over the test set. Both plots indicate strong, unbiased estimation performance by the fiber reservoir on all three

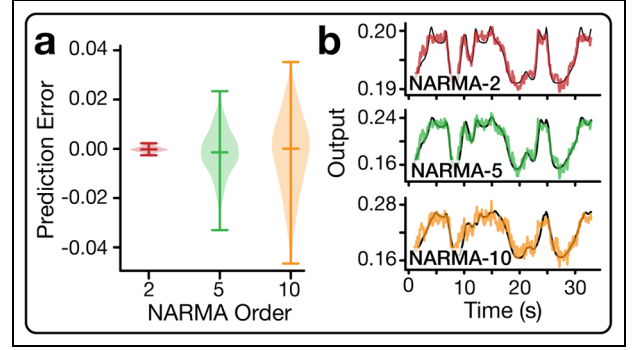


Figure 6. NARMA- n benchmark performance. (a) Violin plots of error ($\hat{y}(t) - y(t)$ where $\hat{y}(t)$ is the fiber reservoir prediction of the timeseries at the next timestep) of NARMA- n benchmark tasks for $n = 2, 5$, and 10. The fiber reservoir provides unbiased estimation for all tasks, however, its error tails increase for higher values of n , consistent with the increased nonlinearity and memory difficulty of higher order NARMA tasks. The root mean square error (RMSE) of the $n = 2, 5$, and 10 tasks is 0.00096, 0.00924, and 0.01573, respectively, while the interdecile range (IDR) of the error is 0.00246, 0.02383, and 0.04082, respectively. (b) Reservoir output estimates of the three NARMA- n timeseries.

NARMA- n benchmarks, with error bands increasing for NARMA-5 and NARMA-10, reflective of its stronger nonlinear and memory processing requirements.

Discussion

This work explores the mechano-computational power of a spider web-inspired physical reservoir. By quantifying physical reservoir capability through the joint metrics of nonlinear and memory capacity, we identified how network topology, actuation magnitude, fiber spacing, network size, and axial pretension impact the ability of the fiber network to serve as a physically computing substrate.

Overall, the fiber network was sensitive to changes in its input and structure. In particular, the crosshatch fiber topology strongly outperformed a polygonal topology, despite the polygonal topology being more visually similar to a spider web. The exact reasons for the crosshatch’s better performance are not clear. It is possible the crosshatch’s highly uniform organization may help information propagate further through the network before being damped out, while the polygonal topology’s central connection node may damp out information due to the multiple fibers acting to stiffen the mechanical response of that node. Further, the fiber network topologies explored here are symmetric, likely causing redundancy in the mechanical response of different parts of the network. Incorporation of random fiber organization and optimization of the fiber’s topological motif may enable further improvement in computational performance.

Reservoir performance was also found to be maximal immediately before the onset of buckling. Buckling injects a level of chaos into the fiber reservoir dynamics due to its extreme sensitivity to small variations in the fiber position at the onset of buckling. As such, it is not surprising that it destroys computational performance in the reservoir. The ability to predict buckling in fiber networks may provide a useful design tool for future reservoirs, as maximizing the portion of the network that is near, but not undergoing, a buckling response may increase the network's computational performance.

The physical experimental implementation of the fiber network broadly agreed with the Cosserat rod simulation results, demonstrating that, indeed, fiber networks can effectively serve as physical reservoir computers. To ensure numerical stability, the Cosserat rod simulations used in this paper used a softer elastic modulus than the nylon monofilament (fishing line) used in the experiments. Additionally, the fibers in simulation are subjected to a constant tensile force applied at their ends while in the experimental setup the reaction tensile force exerted by the springs attached to the ends of the fibers is proportional to the fiber's axial displacement. These discrepancies may explain some of the limited disagreement that is observed as the combination of stiffer fibers and springs used in the physical implementation may effectively be pretensioning the network, in which case the better memory and worse nonlinear performance is consistent with the pretensioning simulation results of Figure 3.

Finally, we note that the fiber network design used in this work was focused on demonstrating the computational ability of a network made of compliant fibers, which was successfully accomplished. However, further refinements to the network design, such as non-optical reservoir readouts, enabling environmental signal inputs rather than relying on a dedicated motor, and improving fiber network endurance to sustained loadings are necessary to advance fiber network physical reservoirs toward a deployable state where they could be used as a sensor. Incorporation of such features would then yield fiber network reservoirs that are inherently compliant, lightweight, low-cost, easily manufacturable, and perform computation directly through the distributed mechanical deformation of the sensing structure itself. This makes them especially attractive for applications in which the substrate is already mechanical, such as soft robotic proprioceptive sensing or infrastructure monitoring and proximity monitoring. In such applications, it may be possible to classify or infer the location of a disturbance through readout of the fiber network, similar in spirit to how spiders localize disturbances on a web.

Conclusion

Inspired by the ability of spiders to use their webs to sense their environment, this work combined simulations and physical experiments to investigate the ability of a network of compliant fibers to serve as a physical reservoir computer. The impact of network topology, geometry, input scaling, and feature outputs were identified and the ability of a fiber network to perform nonlinear transformations in an input signal and recall prior inputs was demonstrated. These basic hallmarks of physical reservoir computing demonstrate the ability of a fiber network to serve as a mechanically intelligent, embedded computational substrate, with potential applications in robotics, autonomous systems, and structural health monitoring.

ORCID iDs

Suyi Li  <https://orcid.org/0000-0002-0355-1655>

Noel Naughton  <https://orcid.org/0000-0002-5553-4718>

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: Funding for this work was provided by NSF EFRI BEGIN OI #2422340 (B.J., S.L., N.N.) and NSF DCSD #2328522 (S.L.). Computational support was provided by Virginia Tech's Advanced Research Computing through use of its Tinkercliffs and OWL cluster.

Declaration of conflicting interests

The authors (Khairnar, Phalak, Wang, Jantzen, Li, and Naughton) and Virginia Tech have filed a provisional patent application relating to the fiber network sensor described in this work.

Data availability statement

Simulations described in this manuscript used the open source Python-based *Elastica* software available online at <https://github.com/GazzolaLab/PyElastica>. Code for the setup and running of the reported fiber network simulations is available on GitHub: <https://github.com/NaughtonLab/ReservoirFiberNetworks>.

Supplemental material

Supplemental material for this article is available online.

References

- Bhosale Y, Weiner N, Butler A, et al. (2022) Micromechanical origin of plasticity and hysteresis in nestlike packings. *Physical Review Letters* 128(19): 198003. <https://doi.org/10.1103/PhysRevLett.128.198003>
- Bhovad P and Li S (2021) Physical reservoir computing with origami and its application to robotic crawling. *Scientific*

- Reports 11(1): 13002. <https://doi.org/10.1038/s41598-021-92257-1>
- Boedecker J, Obst O, Lizier JT, et al. (2012) Information processing in echo state networks at the edge of chaos. *Theory in Biosciences* 131(3): 205–213. <https://doi.org/10.1007/s12064-011-0146-8>
- Caluwaerts K and Schrauwen B (2011) The body as a reservoir: locomotion and sensing with linear feedback. In: *2nd International conference on Morphological Computation (ICMC 2011)*.
- Chang H-S, Halder U, Shih C-H, et al. (2023) Energy-shaping control of a muscular octopus arm moving in three dimensions. *Proceedings of the Royal Society A* 479(2270): 20220593. <https://doi.org/10.1098/rspa.2022.0593>
- Dambre J, Verstraeten D, Schrauwen B, et al. (2012) Information processing capacity of dynamical systems. *Scientific Reports* 2(1): 514. <https://doi.org/10.1038/srep00514>
- Degrave J, Caluwaerts K, Dambre J, et al. (2015) Developing an embodied gait on a compliant quadrupedal robot. In: *2015 IEEE/RSJ international conference on Intelligent Robots and Systems (IROS)*, pp.4486–4491. IEEE.
- Dion G, Mejaouri S and Sylvestre J (2018) Reservoir computing with a single delay-coupled non-linear mechanical oscillator. *Journal of Applied Physics* 124(15): 152–132. <https://doi.org/10.1063/1.5038038>
- Du C, Cai F, Zidan MA, et al. (2017) Reservoir computing using dynamic memristors for temporal information processing. *Nature Communications* 8(1): 2204. <https://doi.org/10.1038/s41467-017-02337-y>
- Gazzola M, Dudte LH, McCormick AG, et al. (2018) Forward and inverse problems in the mechanics of soft filaments. *Royal Society Open Science* 5(6): 171628. <https://doi.org/10.1098/rsos.171628>
- Hauser H, Ijspeert AJ, Fuchslin RM, et al. (2011) Towards a theoretical foundation for morphological computation with compliant bodies. *Biological Cybernetics* 105: 355–370. <https://doi.org/10.1007/s00422-012-0471-0>
- Hauser H, Ijspeert AJ, Fuchslin RM, et al. (2012) The role of feedback in morphological computation with compliant bodies. *Biological Cybernetics* 106: 595–613. <https://doi.org/10.1007/s00422-012-0516-4>
- Jaeger H (2001) The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *German National Research Center for Information Technology GMD Technical Report* 148(34): 13.
- Khairnar A, Phalak Y, Wang J, et al. (2025) Physical reservoir computing with compliant fiber networks. In: *Smart materials, adaptive structures and intelligent systems*, Vol. 89275, p.V001T08A002. American Society of Mechanical Engineers.
- Konkoli Z, Nichele S, Dale M, et al. (2018) Reservoir computing with computational matter. In: *Computational Matter*. pp.269–293. Springer International Publishing.
- Landolf MA and Barth FG (1996) Vibrations in the orb web of the spider nephila clavipes: Cues for discrimination and orientation. *Journal of Comparative Physiology A* 179: 493–508. <https://doi.org/10.1007/bf00192316>
- Legenstein R and Maass W (2007) Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks* 20(3): 323–334.
- Li T, Nakajima K, Cianchetti M, et al. (2012) Behavior switching using reservoir computing for a soft robotic arm. In: *2012 IEEE international conference on robotics and automation*, pp.4918–4924. IEEE.
- Liu Z, Fang H, Xu J, et al. (2023) Discriminative transition sequences of origami metamaterials for mechanologic. *Advanced Intelligent Systems* 5(1): 2200146. <https://doi.org/10.1002/aisy.202200146>
- Maass W (2011) Motivation, theory, and applications of liquid state machines. In: *Computability in Context: Computation and Logic in the Real World*. pp.275–296. Imperial College Press.
- Masters WM (1984) Vibrations in the orbwebs of *Nuctenea sclopeteria* (Araneidae) ii. prey and wind signals and the spider’s response threshold. *Behavioral Ecology and Sociobiology* 15: 217–223. <https://doi.org/10.1007/bf00292978>
- Mortimer B (2019) A spider’s vibration landscape: adaptations to promote vibrational information transfer in orb webs. *Integrative and Comparative Biology* 59(6): 1636–1645. <https://doi.org/10.1093/icb/icz043>
- Nakajima K (2020) Physical reservoir computing—an introductory perspective. *Japanese Journal of Applied Physics* 59(6): 060501. <https://doi.org/10.35848/1347-4065/ab8d4f>
- Nakajima K and Fischer I (2021) *Reservoir computing*. Springer.
- Nakajima K, Hauser H, Kang R, et al. (2013) A soft body as a reservoir: Case studies in a dynamic model of octopus-inspired soft robotic arm. *Frontiers in Computational Neuroscience* 7: 91. <https://doi.org/10.3389/fncom.2013.00091>
- Nakajima K, Hauser H, Li T, et al. (2018) Exploiting the dynamics of soft materials for machine learning. *Soft Robotics* 5(3): 339–347. <https://doi.org/10.1089/soro.2017.0075>
- Nakajima K, Li T, Hauser H, et al. (2014) Exploiting short-term memory in soft body dynamics as a computational resource. *Journal of The Royal Society Interface* 11(100): 20140437. <https://doi.org/10.1098/rsif.2014.0437>
- Naughton N, Sun J, Tekinalp A, et al. (2021) Elastica: A compliant mechanics environment for soft robotic control. *IEEE Robotics and Automation Letters* 6(2): 3389–3396.
- Naughton N, Tekinalp A, Shivam K, et al. (2026) Neural reservoir control of a bio-hybrid soft arm. *Proceedings of the National Academy of Sciences* 123(17): e2522094123. <https://doi.org/10.1073/pnas.2522094123>
- Paul C (2004) *Investigation of morphology and control in biped locomotion*. PhD thesis, Verlag nicht ermittelbar.
- Paul C (2006) Morphological computation: A basis for the analysis of morphology and control requirements. *Robotics and Autonomous Systems* 54(8): 619–630.
- Pfeifer R and Bongard J (2006) *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press.
- Schrauwen B, Verstraeten D and Van Campenhout J (2007) An overview of reservoir computing: theory, applications and implementations. In: *Proceedings of the 15th European symposium on artificial neural networks*, pp.471–482.
- Shih CH, Naughton N, Halder U, et al. (2023) Hierarchical control and learning of a foraging cyboctopus. *Advanced Intelligent Systems* 5(9): 2300088. <https://doi.org/10.1002/aisy.202370039>
- Tanaka G, Yamane T, Héroux JB, et al. (2019) Recent advances in physical reservoir computing: A review. *Neural Networks* 115: 100–123. <https://doi.org/10.1016/j.neunet.2019.03.005>

- Tekinalp A, Naughton N, Kim SH, et al. (2024) Topology, dynamics, and control of a muscle-architected soft arm. *Proceedings of the National Academy of Sciences* 121(41): e2318769121. <https://doi.org/10.1073/pnas.2318769121>
- Ulrich NT (1988) *Grasping with mechanical intelligence*. Master's thesis, University of Pennsylvania.
- Urbain G, Degraeve J, Carette B, et al. (2017) Morphological properties of mass-spring networks for optimal locomotion learning. *Frontiers in Neurorobotics* 11: 16. <https://doi.org/10.3389/fnbot.2017.00016>
- Wang J and Li S (2023) Building intelligence in the mechanical domain—harvesting the reservoir computing power in origami to achieve information perception tasks. *Advanced Intelligent Systems* 5(9): 2300086. <https://doi.org/10.1002/aisy.202300086>
- Wang J and Li S (2025) Re-purposing a modular origami manipulator into an adaptive physical computer for machine learning and robotic perception. *Advanced Science* 12(45): e09389. <https://doi.org/10.1002/advs.202509389>
- Wang J, Qiao Z, Zhang W, et al. (2025) Proprioceptive and exteroceptive information perception in a fabric soft robotic arm via physical reservoir computing with minimal training data. *Advanced Intelligent Systems* 7(4): 2400534. <https://doi.org/10.1002/aisy.202400534>
- Wang T, Pierce C, Kojouharov V, et al. (2023) Mechanical intelligence simplifies control in terrestrial limbless locomotion. *Science Robotics* 8(85): eadi2243. <https://doi.org/10.1126/scirobotics.adi2243>
- Wringe C, Trefzer M and Stepney S (2025) Reservoir computing benchmarks: A tutorial review and critique. *International Journal of Parallel Emergent and Distributed Systems* 40: 313–351. <https://doi.org/10.1080/17445760.2025.2472211>
- Wu J, Miller TE, Cicirello A, et al. (2023) Spider dynamics under vertical vibration and its implications for biological vibration sensing. *Journal of The Royal Society Interface* 20(206): 20230365. <https://doi.org/10.1098/rsif.2023.0365>
- Yamanaka Y, Yaguchi T, Nakajima K, et al. (2018) Mass-spring damper array as a mechanical medium for computation. In: *International conference on artificial neural networks*, pp.781–794. Springer.
- Yamane T, Katayama Y, Nakane R, et al. (2015) Wave-based reservoir computing by synchronization of coupled oscillators. In: *Neural information processing: 22nd international conference, ICONIP 2015, Istanbul, Turkey, 9–12 November 2015, Proceedings Part III* 22, pp.198–205. Springer.
- Zhang X, Chan FK, Parthasarathy T, et al. (2019) Modeling and simulation of complex dynamic musculoskeletal architectures. *Nature Communications* 10(1): 4825. <https://doi.org/10.1038/s41467-019-12759-5>
- Zhang X, Naughton N, Parthasarathy T, et al. (2021) Friction modulation in limbless, three-dimensional gaits and heterogeneous terrains. *Nature Communications* 12(1): 6076. <https://doi.org/10.1038/s41467-021-26276-x>
- Zhang Y and Wang K-W (2022) Harnessing physical reservoir computing in nonlinear mechanical metastructures. In: *AIAA Scitech 2022 Forum*, pp.1464.