

University of Mississippi

eGrove

Faculty and Student Publications

Engineering, School of

10-17-2023

Biomembrane-Based Memcapacitive Reservoir Computing System for Energy-Efficient Temporal Data Processing

Md Razuan Hossain
University of Mississippi

Ahmed S. Mohamed
The Pennsylvania State University

Nicholas X. Armendarez
The Pennsylvania State University

Joseph S. Najem
The Pennsylvania State University, jsn5211@psu.edu

Md Sakib Hasan
University of Mississippi, mhasan5@olemiss.edu

Follow this and additional works at: https://egrove.olemiss.edu/engineering_facpubs



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Hossain, M.R., Mohamed, A.S., Armendarez, N.X., Najem, J.S. and Hasan, M.S. (2023), Biomembrane-Based Memcapacitive Reservoir Computing System for Energy-Efficient Temporal Data Processing. *Adv. Intell. Syst.*, 5: 2300346. <https://doi.org/10.1002/aisy.202300346>

This Article is brought to you for free and open access by the Engineering, School of at eGrove. It has been accepted for inclusion in Faculty and Student Publications by an authorized administrator of eGrove. For more information, please contact egrove@olemiss.edu.

Biomembrane-Based Memcapacitive Reservoir Computing System for Energy-Efficient Temporal Data Processing

Md Razuan Hossain, Ahmed Salah Mohamed, Nicholas X. Armendarez, Joseph S. Najem,* and Md Sakib Hasan*

Reservoir computing is a highly efficient machine learning framework for processing temporal data by extracting input features and mapping them into higher dimensional spaces. Physical reservoirs have been realized using spintronic oscillators, atomic switch networks, volatile memristors, etc. However, these devices are intrinsically energy-dissipative due to their resistive nature, increasing their power consumption. Therefore, memcapacitive devices can provide a more energy-efficient approach. Herein, volatile biomembrane-based memcapacitors are leveraged as reservoirs to solve classification tasks and process time series in simulation and experimentally. This system achieves a 99.6% accuracy for spoken-digit classification and a normalized mean square error of 7.81×10^{-4} in a second-order nonlinear regression task. Furthermore, to showcase the device's real-time temporal data processing capability, a 100% accuracy for an epilepsy detection problem is achieved. Most importantly, it is demonstrated that each memcapacitor consumes an average of 41.5 fJ of energy per spike, regardless of the selected input voltage pulse width, while maintaining an average power of 415 fW for a pulse width of 100 ms, orders of magnitude lower than those achieved by state-of-the-art devices. Lastly, it is believed that the biocompatible, soft nature of our memcapacitor renders it highly suitable for computing applications in biological environments.

1. Introduction


Temporal data processing and time series prediction have recently gained increasing interest due to their ubiquitous utility in various fields such as speech recognition,^[1,2] language modeling, text generation,^[3,4] trend forecasting,^[5] traffic forecasting,^[6] and financial forecasting.^[7,8] Temporal series prediction requires recurrent neural network (RNN) paradigms capable of history-dependent, multilayered input mapping to an output layer.^[9] This means that the output is influenced by both the current and the prior inputs as well as the network states. Therefore, to achieve history-based computing, nodes in the hidden layers of conventional RNNs are recurrently or cyclically connected to themselves.^[9] These cyclical connections increase the RNNs' computational cost and complexity of training as cyclic dependencies can exhibit bifurcations, and thereby nonconvergence in training.^[10,11] To circumvent the training bottleneck, the concept of reservoir computing (RC)

was independently introduced by Jaeger et al.^[12] and Maass et al.^[13] in their RNN models, namely, the echo state network and the liquid state machines, respectively.

RC is a brain-inspired emerging machine learning architecture.^[14] As described in **Figure 1C**, an RC incorporates an input layer that feeds the input signal to an RNN of fixed random weights called the reservoir.^[12] The reservoir, in most cases, nonlinearly maps the input signal to a high-dimensional state space that dynamically evolves with the time-varying input.^[12] The reservoir states, corresponding to all training inputs, then get projected to the output layer via the "memory-less" (i.e., time-independent) readout layer.^[12] Unlike conventional RNNs which necessitate weight training between every two subsequent layers,^[9] the training of RC systems is constrained to the readout layer,^[12] which can be done, only once, using linear or logistic regression.^[15,16] Limiting the number of training layers to only one static readout layer offers a drastic reduction in the overall computational cost of the network when compared to conventional RNNs.^[12] This nonlinear mapping from low-dimensional to high-dimensional space increases input separability, thereby facilitating the classification of different output classes.^[17]

M. R. Hossain, M. S. Hasan
Department of Electrical and Computer Engineering
University of Mississippi
Oxford, MS, USA
E-mail: mhasan5@olemiss.edu

A. S. Mohamed, N. X. Armendarez, J. S. Najem
Department of Mechanical Engineering
The Pennsylvania State University
University Park, PA, USA
E-mail: jsn5211@psu.edu

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202300346>.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202300346

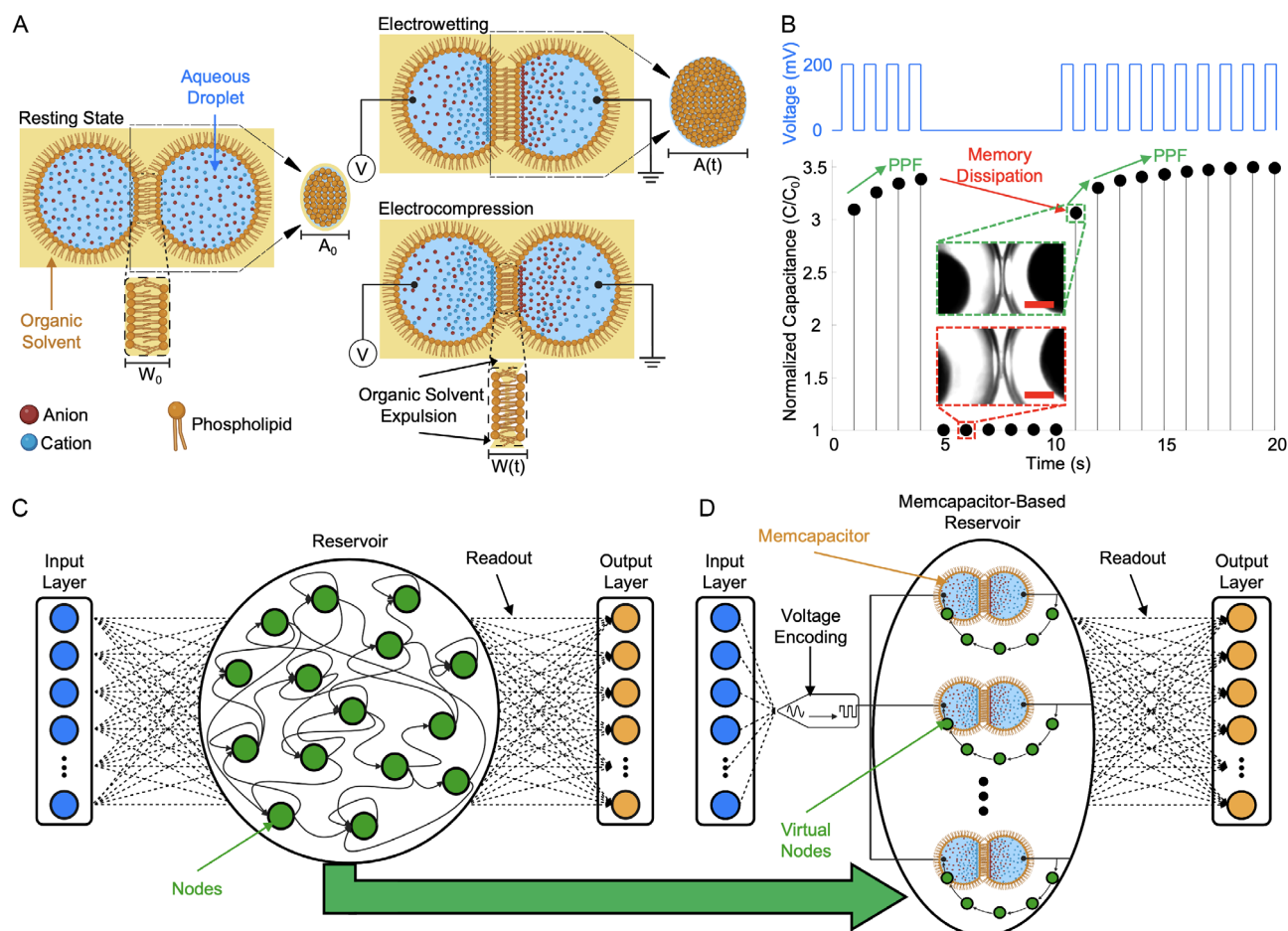


Figure 1. A memcapacitive reservoir computing scheme. A) A schematic describing a biomembrane-based memcapacitor in the resting state and in response to voltage. In response to voltage, this device exhibits hysteretic changes in membrane area (i.e., electro-wetting) and thickness (i.e., electro-compression) which give rise to dynamical memcapacitance.^[54] B) In response to a train of pulses, the memcapacitor outputs an increased relative capacitance from pulse to pulse (i.e., paired-pulse facilitation) due to its volatile memory. The inset enclosed in green-dashed displays an inverted microscope-obtained bottom view of the memcapacitor exhibiting an almost two-fold increase in interfacial area compared to the interfacial area at rest (the inset enclosed in dashed red). The scale bars in red in both insets are equivalent to 200 μm . C) A representation of a conventional reservoir computing system entailing an input layer, the reservoir, a readout layer, and an output layer. D) An equivalent system can be built by replacing the reservoir layer with memcapacitors as reservoirs. The random spatial nodes in a conventional reservoir in (C) are replaced by serial temporal virtual nodes.

In practice, RC systems were initially implemented and studied in silico with marked temporal signal prediction accuracies.^[18–20] However, soon after Appeltant et al.^[21] theorized the equivalence of any dynamically-rich time-delay system to a dynamical reservoir, a spurt in more efficient hardware realizations of the reservoir occurred.^[22,23] These hardware implementations included spintronic oscillators,^[24,25] atomic switch networks,^[26,27] silicon photonic modules,^[28,29] ferroelectric transistors,^[30,31] and most notably memristors.^[32–38] Memristors, short for “memory resistors”, are two-terminal, state-dependent resistive elements that colocate volatile memory and, in many cases, complex nonlinear dynamics,^[39] hence the prevalence of memristor-based RC systems in the literature. Memristor-based RC systems achieved remarkable performance in various applications including, yet not limited to,

hand-written^[32,34] and spoken digit recognition,^[33,37] chaotic time series prediction,^[33,37,40] and real-time neural firing pattern classification.^[35]

In theory, any dynamical system with sufficient short-term memory and nonlinearity can act as a reservoir.^[21] Therefore, reservoirs can be generalized to encompass a whole class of nonlinear electronics called memelements. The term “memelements” was coined by Chua^[41] to refer to any passive circuit element with a response dependent on input history, such as memristors, memcapacitors, meminductors, as well as other higher-order memelements.^[42,43] Memcapacitors, in particular, have garnered attention in computing applications^[44,45] that require very low power consumption due to their energy-storing nature as opposed to the energy-dissipative nature of memristors, in addition to the pulse-width-independent energy per spike

consumption. Although various research groups have proposed theoretical models for memcapacitive devices,^[46–49] only a few physical implementations have been demonstrated.^[50–53] In terms of in silico neuromorphic computing applications, memcapacitor models^[48,54,55] have been employed in the simulation of memcapacitive networks, where the memcapacitors serve only as variable weights between the network's nodes.^[56–59] Such architectures utilize nodal activation functions for the nonlinear transformation, as the memcapacitor itself does not perform the nonlinear mapping. To the best of our knowledge, there have been no reports of memcapacitor-based RC system architectures where the memcapacitor nonlinearly maps an input to a higher dimensional feature space with fading memory, thereby fully replacing the network-based reservoir layer. For such a memcapacitor-based RC system, it is imperative that the memcapacitor exhibits both nonlinearity with respect to input excitation and volatility (i.e., the fading memory property).^[13,32] The utility of physical memcapacitors in neuromorphic computing applications has been limited to artificial neural networks due to the constraint of nonvolatile memcapacitors.^[45] Additionally, to the best of our knowledge, there have been no reports on the physical implementation of a memcapacitor-based RC system.^[51,52]

Here, we experimentally realize a memcapacitor-based RC system that leverages two-terminal, volatile, scalable, memcapacitors, first introduced by Najem et al.,^[54] as reservoirs that fully replace conventional network-based RC architectures due to their intrinsic nonlinearity and state volatility. The memcapacitor consists of a synthetic lipid bilayer formed between two lipid-encased aqueous droplets submerged in an oil phase.^[54] At the interface of both droplets, an elliptical, planar lipid bilayer ($\approx 100 \mu\text{m}$ in radius) spontaneously forms with a highly insulating ($> 100 \text{M}\Omega \text{cm}^2$) core consisting of a mixture of hydrophobic lipid tails and residual entrapped oil (Figure 1A). Upon transmembrane voltage application, the ionically charged lipid bilayer manifests geometrical changes due to electrowetting (EW) and electrocompression (EC) (Figure 1A), leading to an increase in bilayer area and a decrease in the hydrophobic, respectively (see Section S1 and Figure S1–S4, Supporting Information for more details). The bilayer exhibits dynamical voltage-controlled capacitance with paired-pulse-facilitation (Figure 1B) via geometric reconfigurability of its interfacial area and hydrophobic thickness,^[54] enabling the high-dimensional temporal transformation with minimal power and energy consumption. First, we demonstrate the device's computational quality by conducting spoken-digits recognition as well as predicting a second-order dynamical time series and compare the achieved accuracies one-to-one with another memristor-based RC report.^[32,33] Then, taking advantage of the device's biological synapse-like timescale ($\approx 102 \text{ms}$ ^[60]), we detect epilepsy from an electroencephalography (EEG) signal to demonstrate the device's real-time temporal processing. Furthermore, for completion, we solve an IRIS dataset classification problem to confirm that the device's short-term dynamics can solve a static classification problem (Section S5, Supporting Information). Finally, we present the device's power and energy per spike consumption and compare them with the consumption of other state-of-the-art memristors that were deployed as reservoirs.^[32,34,35,37,61,62]

2. Results

2.1. Memcapacitor-Based Reservoir Computing Architecture

The structure of RC is based on a three-layered architecture comprising an input layer, a reservoir, and an output layer,^[16] as shown in Figure 1C. The input layer is responsible for encoding and distributing the input data to the reservoir, a large, fixed, and sparsely connected network of nonlinear dynamic nodes.^[12] These nodes, often referred to as neurons or perceptrons, are interconnected through random and fixed weights, enabling the reservoir to serve as a rich temporal memory.^[12,13] The output layer consolidates the high-dimensional reservoir states into a meaningful output, with only the connections between the reservoir and output layer being adjusted during training.^[19,63] This unique structure allows RC to efficiently handle many temporal tasks, outperforming traditional RNNs in terms of computational complexity and training time and solidifying its position as a powerful and versatile approach to processing time series data.^[19,64]

Taking advantage of the memcapacitor's second-order nonlinear dynamics in which EW and EC operate at two distinct time-scales,^[54] short-term memory (Figure 1B and S2–S4, Supporting Information) and innate stochasticity (Figure S5, Supporting Information), we replaced the conventional RNN-based RC with a memcapacitor-based RC, as shown in Figure 1C,D. Unlike memristor-based reservoirs, where the resistance or conductance of the memristor is commonly representative of the reservoir state,^[32,33,37,65] in memcapacitor-based reservoirs, the reservoir state is reflected by capacitance, or, as more often used in this work, capacitance normalized to its resting capacitance state (C/C_0). It is worth noting that there are nontrivial dissimilarities between a conventional RC and a memcapacitor-based RC in terms of their computing architecture. A conventional RC typically incorporates hundreds, often thousands, of interconnected cyclic nodes.^[22] Contrarily, a memcapacitor-based reservoir exploits the memcapacitor's inherent time-delay aspect (i.e., memory) to account for the nodes' recursion.^[21] Furthermore, the nonlinearity of a single memcapacitor can compensate for the nonlinearity offered by a number of interconnected nodes, depending on the device's degree of nonlinearity and quality of fading memory (i.e., volatility). Accordingly, as hypothesized by Appeltant et al.,^[21] a single dynamical system with sufficient nonlinearity and time delay, such as our memcapacitor, can act as a full reservoir, which promotes practical implementations of physical RCs. It should be noted that it has been mathematically proven that a reservoir state cannot concurrently render high degrees of nonlinearity and memory capacity because nonlinearity degrades memory.^[66,67]

For a memcapacitor-based RC with a few memcapacitors connected in parallel, as depicted in Figure 1D, an effective mapping of the input signal to a high-dimensional state space can be accomplished by means of virtual nodes.^[21] When first introduced,^[21] virtual nodes were adapted by sampling many points along the reservoir's time-delayed response to every input time step. In other words, every input value sent to the reservoir is sampled-and-held via a voltage encoder (Figure 4A), yielding time-varying reservoir states that span the holding time. Each sampled point along the time-delayed response sequence can

be deemed an independent node and state-space dimension, hence the name “virtual node” (Figure 1D).^[21] However, for high-frequency sampling, consecutive nodes are close enough to be considered linearly dependent, which is redundant for high-dimensional mapping; thus, it is more functional to choose an adequately spaced sample from the obtained array to designate as the reservoir’s virtual nodes. Interestingly, for input sequences with negligible variations, as seen in cochleogram channels fully populated with 0-bits (channels 3–20 in Figure 2), to avoid redundantly large feature spaces, it is more advisable to assign a virtual node once every few inputs rather than more than once per input. To be comprehensive, in practical implementations, one can choose to elect as many or as few virtual nodes as demanded by the task of interest from the full array of measured reservoir states. Furthermore, for some applications, applying a form of postprocessing on elected virtual nodes before transmitting them to the readout layer can also improve the overall system’s performance.

In this study, we discuss the employment of the memcapacitor-based reservoir for solving three distinct problems, where the input encoding and the feature space

definition for each problem were executed differently. We start with a benchmark spoken-digit classification problem, where the input signal is binary (either a 0 or a 1) and temporally history-dependent (Figure 2). For this problem, a virtual node was elected for every five equally spaced inputs. Then, we discuss how we used the memcapacitor-based reservoir for predicting a second-order regression problem, where the input signal is random, continuous (nonbinary), and history-dependent. Unlike the spoken-digit problem, the input signal was encoded in 10 different timescales to effectively increase the dimensionality of the reservoir. We then present an epilepsy detection problem using an EEG input signal as a real-time temporal signal processing problem. Leveraging the device’s biological fading memory (100 ms),^[60] we incorporate a feature modification postprocess that integrates 60 short-timescale (5.8 ms) features into one long-timescale (348 ms) cumulative feature (i.e., virtual node) to be passed to the readout layer. As a supplementary problem (Section S5, Supporting Information), we solve a static classification problem, namely the IRIS dataset problem, where the input is history-independent to demonstrate the device’s exemplary performance across tasks with dissimilar inputs.

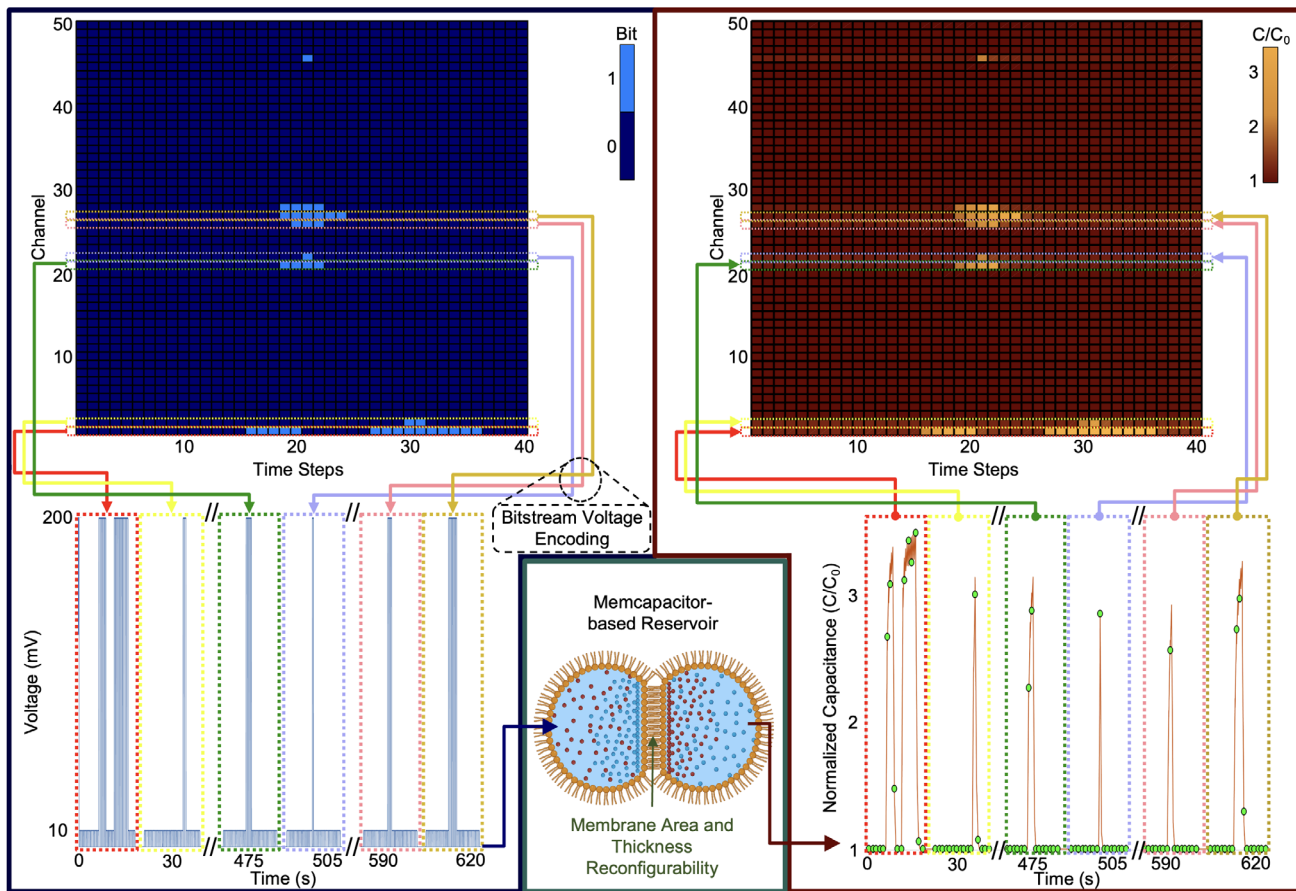


Figure 2. The encoding process flow of the spoken digit classification problem. The binary 2D cochleogram (top-left corner) represents neural spike trains in different human cochlear channels. Each input was converted into voltage pulses (bottom-left corner), where the 10 and 200 mV pulses correspond to a 0 (resting neuron) bit and a 1 (firing neuron) bit, respectively. The input voltage train was fed to the memcapacitor-based reservoir, where the normalized capacitance response is recorded (bottom-right corner). The dynamic normalized capacitance was then mapped to a 2D matrix (top-right corner) and, for every channel or row, one virtual node was selected for every five-time steps as depicted by the green circles on the capacitance plot (bottom-right corner).

2.2. Spoken Digit Classification

Here, we demonstrate the performance of our memcapacitor-based RC system for a benchmark speech recognition task. The dataset we used, NIST TI46, comprises binary cochleograms of isolated spoken digit waveforms, 0-9 in spoken English (Figure S6, Supporting Information). The dataset was provided to us as a courtesy by the authors of Moon et al.^[33] It consists of 500 binary 2D cochleograms, with 450 used for training and the remaining 50 for testing. Each cochleogram, an example of spoken digit 6 is shown in Figure 2, digitally represents the response of 50 human cochlear channels to sound waves captured over 40 timesteps. For each channel in the vertical axis, a 40-bit binary sequence renders a neuron firing event along 40 timesteps, where the light blue area denotes a firing event or 1-bit while the dark blue area signifies no firing or 0-bit for the corresponding time step. Subsequently, 1-bit and 0-bit time steps were encoded, respectively, as 200 and 10 mV square pulses (500 ms pulse width), as shown in the voltage versus time plot at the bottom-left corner of Figure 2.

Our analysis of the 500 datasets revealed 25 000 channels in total, with 795 being distinct. To streamline the experimental setup, we routed the 795 unique bitstreams (Figure S7, Supporting Information) that comprise all 25 000 channels to a single memcapacitor. The corresponding capacitance responses were then estimated from the recorded current and normalized to the resting state at every time step using the method described in the Experimental Section. The upper right inset in Figure 2 depicts a 2D heat map of the normalized capacitance response (Figure S8, Supporting Information) resulting from inputting the example spoken digit 6 cochleogram in the

upper left inset of Figure 2. As seen in Figure 2, the shown cochleogram indicated a predominance of 0 bits, suggesting that the reservoir state was mostly inactive, which was also true for all 500 cochleograms comprising the training and testing dataset. Accordingly, selecting a virtual node after each time step led to an unreasonably large feature space (31 800 dimensions for 795 channels and 40 timesteps), as explained in Section 2.1. Such a large feature space could result in states' linear dependence, and therefore information redundancy. To overcome this problem, we divided the input sequence into n equal intervals (every five-time steps in length, where capacitance data were recorded at the end of each interval, yielding a total of eight virtual nodes per channel). These elected virtual nodes are represented by the green circles in the bottom-right inset of Figure 2. The reservoir is comprised of a total of $50n$ virtual nodes, given that there are 50 frequency channels, each containing n virtual nodes. These nodes are used in a one-vs-all readout layer ($50n \times 10$) to classify among ten digits. The fitcecoc function in MATLAB was utilized with a logistic regression learner for training the readout network. It is important to note that while only one memcapacitor is required to process all channels, we could have used multiple memcapacitors in parallel to speed up the encoding process with the same accuracy and energy consumption.

In summary, as an average of ten runs, our memcapacitive RC system attained a 99.6% success rate in experiments and 100% in simulations for all inputs. Further, we have also evaluated the performance of our system at 75%, 50%, and 25% of the input completion, meaning that our RC system predicts the spoken digit before the utterance is fully delivered. The results are summarized in confusion matrices plotted in Figure 3. In addition,

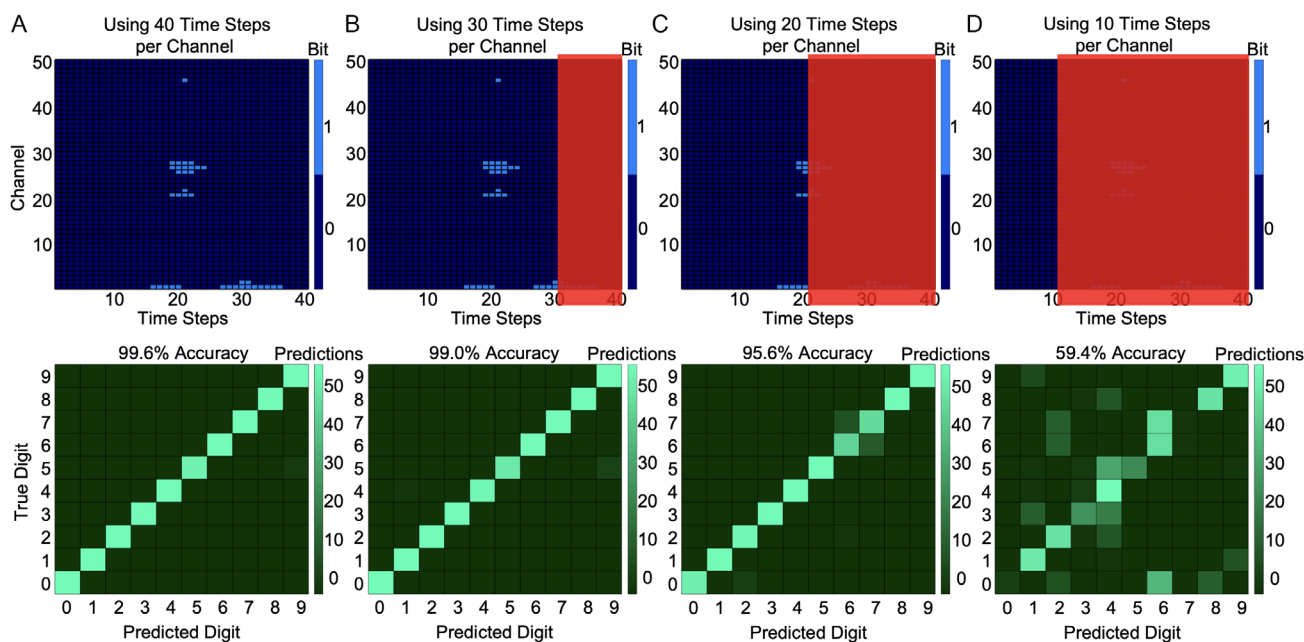


Figure 3. Prediction results for the spoken digit classification problem using a memcapacitor-based RC system. A) Confusion Matrix for spoken digits using all timesteps in every channel (100% of the utterance). The overall testing dataset accuracy is 99.6%. B) Confusion Matrix for spoken digits using 30 out of 40 timesteps per channel (75% of the utterance). The overall testing dataset accuracy is 99.0%. C) Confusion Matrix for spoken digits using all time steps 20 out of 40 timesteps per channel (50% of the utterance). The overall testing dataset accuracy is 95.6%. D) Confusion Matrix for spoken digits using only 10 out of 40 timesteps per channel (25% of the utterance). The overall testing dataset accuracy is 59.4%.

Table 1. Recognition rate and energy comparison of the spoken digit classification in an experimental framework. (Note: the comparison of energy has solely been conducted on the reservoir of the test dataset).

Time steps	This work		Moon et al. ^[33]	
	Accuracy [%]	Energy [J]	Accuracy [%]	Energy [J]
10	59.4	4.22×10^{-9}	57.8	2.68×10^{-6}
25	97.0	1.05×10^{-7}	98.2	1.41×10^{-5}
40	99.6	1.19×10^{-7}	99.2	2.55×10^{-5}

Table 1 presents a comparison of recent work^[32] for experimental accuracy and the reservoir energy (Figure S9, Supporting Information).

2.3. Solving a Second-Order Nonlinear Dynamic Task

Nonlinear dynamical systems are mathematical models that describe the complex behavior of natural and engineered systems that are characterized by time-dependent interactions among their constituent elements.^[68] The study of these nonlinear systems has proven invaluable in elucidating a broad spectrum of phenomena in various disciplines, such as fluid dynamics,^[69] population biology,^[70] climatic systems,^[71] etc. The governing equations for second-order nonlinear dynamical systems encompass time derivatives of the second order. Some prominent examples of second-order nonlinear dynamics within these domains include electrical system converters^[72] as well as damping properties in mechanical systems,^[73] among others.^[74–76] In this section, we use a memcapacitive reservoir to predict a second-order dynamic nonlinear transfer function.^[32] The transfer function is described as follows:

$$\gamma(t) = 0.4\gamma(t-1) + 0.4\gamma(t-1)\gamma(t-2) + 0.6u^3(t) + 0.1 \quad (1)$$

The output signal, $\gamma(t)$, depends on the present input, $u(t)$, as well as the previous two inputs, $\gamma(t-1)$ and $\gamma(t-2)$ (i.e., a time lag of two-time steps), as shown in Equation (1). In this study, we trained the memcapacitor-based RC system to map a random input onto a higher-dimensional space, thereby enabling the generation of an accurate second-order dynamic nonlinear transfer function output from the input after training without prior knowledge of the underlying mathematical relationship between input and output.

Figure 4A presents the schematic of the RC system developed for solving a second-order nonlinear dynamic task. We initially channel the input signal into the voltage encoder, which converts the input into voltage pulses. The voltage pulses are then directed to the memcapacitive reservoir, generating different reservoir states. These states are subsequently employed by the readout function to derive the anticipated output.

We chose a random input signal sequence within the range of 0–0.5 and transformed it into a voltage amplitude between 50 and 200 mV. We employ a random sequence of 300-time frames as inputs for training the memcapacitor-based RC system. The reservoir is composed of 50 memcapacitive virtual nodes within the employment of five physical memcapacitive devices. For each memcapacitor, the input voltages are provided via pulse streams

of 50% duty cycle with 10 different time frame widths (equally spaced between 200 and 600 ms) applied to each of the five memcapacitors in the reservoir throughout the experiment. It is worth noting that there exists a slight variation across the five memcapacitors due to inherent slight differences in bilayer area and hydrophobic thickness (denoted A_0 and W_0 , respectively, in Figure 1A). We found that including five devices with slightly varying properties enhances reservoir performance, as inherent device-to-device variations contribute to more separable reservoir outputs. Comparable performance improvement was observed for inputs with 10 distinct time frames or pulse widths (PW). We found that having 50 reservoir states (comprising five devices, each receiving 10 input encodings) was optimal since further increasing the number of reservoir states increased the computational overhead without significantly improving the prediction accuracy. In this instance, the readout layer is a 50×1 feedforward layer and serves to convert the reservoir state into a single output. A simple linear regression model with gradient descent is utilized to train the weights in the readout layer. The process flow of solving the second-order dynamic task is demonstrated in Figure 4A.

Panels B and C in Figure 4 present the graphical representation of the original and predicted signals for 100-time frames of training and testing data in the experiment, respectively. Notably, the readout function is not retrained during the testing phase. The normalized mean square error (NMSE) values (refer to Section S3.1, Supporting Information) attained for training and testing data amount to 5.28×10^{-4} and 6.32×10^{-4} in simulation (Figure S10, Supporting Information), and 5.75×10^{-4} and 7.81×10^{-4} in experiment, respectively. **Table 2** summarizes a comparison with recent work.^[32] Furthermore, it is important to note that solving this problem using a reservoir based on a conventional linear network yielded a larger NMSE than that of the memcapacitor-based reservoir (Figure S11 and Section S3.2, Supporting Information). This highlights the indispensability of the intrinsic nonlinear dynamics of the memcapacitor device for higher dimensional transformation.

2.4. Real-Time Epilepsy Detection from EEG Signal

EEG signals, which are time series data, serve as a valuable tool for examining abnormal brain activity occurring during seizure episodes, as these signals are captured from diverse areas within the brain.^[77] Representing temporal patterns of brain activation, EEG signals contain essential information about brain functionality.^[78] Thus, by extracting pertinent features, researchers can gain a more profound understanding of the brain's underlying activities.^[79] The primary objective of this study is to demonstrate the effectiveness of our memcapacitor-based RC system capable to solve classification problems in real-time.

In this study, we have used a dataset from the University of Bonn, Germany.^[80] This dataset is composed of five distinct classes of EEG signals, specifically F, N, S, Z, and O. Each class contains 100 EEG signals with a duration of 23.6 s, sampled at a frequency of 173.67 Hz (5.8 ms) and exhibiting voltage amplitudes below and above of $\pm 1000 \mu\text{V}$. Class S EEG signals are captured during seizure activity, while class Z EEG signals are obtained from healthy individuals.^[81] The primary focus of

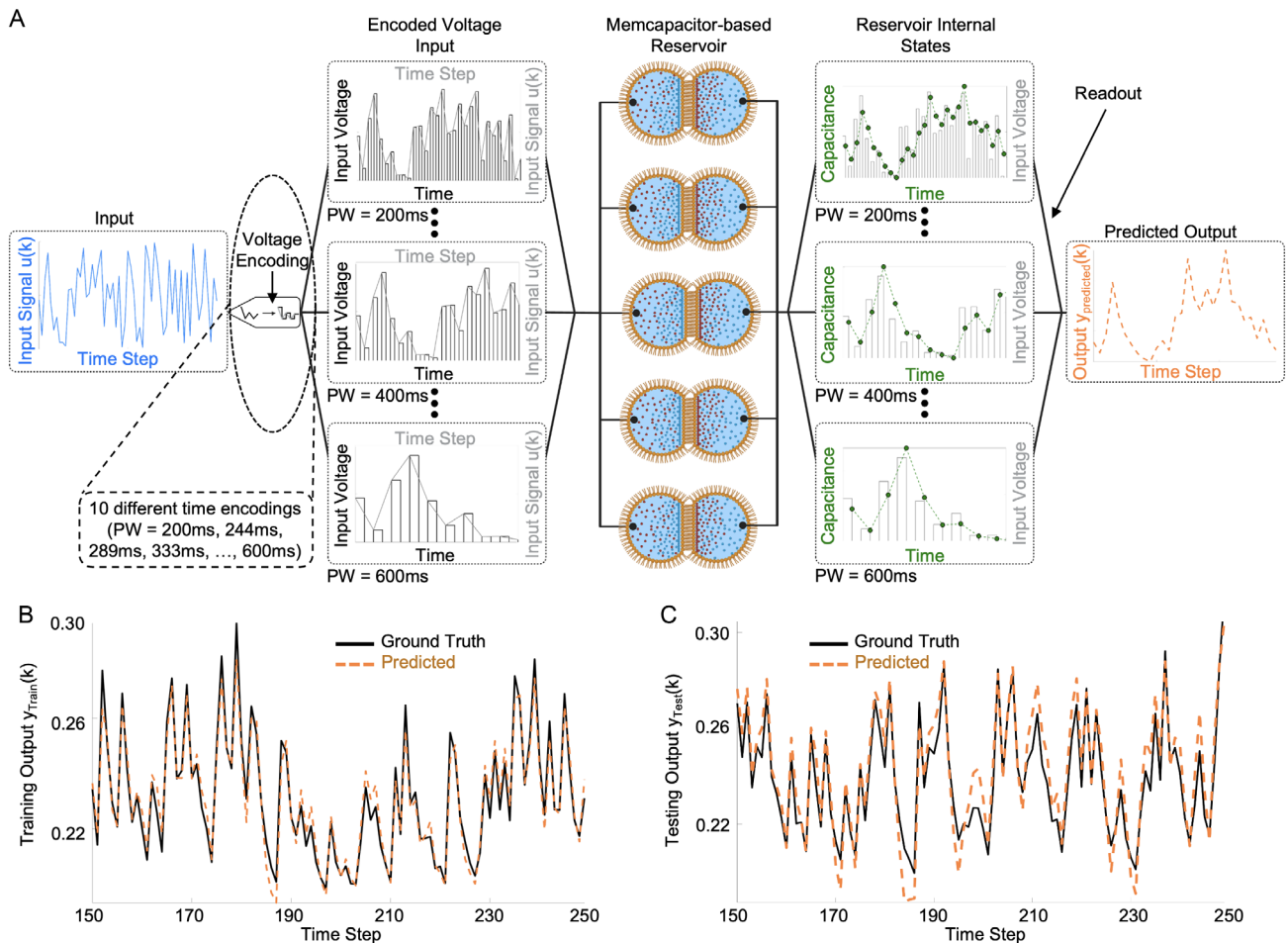


Figure 4. Solving a second-order nonlinear dynamic task using a memcapacitor-based RC system. A) The process flow of solving second-order dynamic tasks. The random input signal gets encoded (i.e., sampled and held) with ten different pulse widths (PW) ranging from 200 to 600 ms. The encoded voltage was then fed to five memcapacitors and the corresponding capacitance was measured at the end of every voltage pulse for all ten different encodings. We assigned a virtual node to every capacitance measured, resulting in 50 virtual nodes per time step (five memcapacitors by ten-time encodings). Subsequently, a 50-by-300 state matrix is passed to the readout layer for output prediction using linear regression. B) Experimentally obtained output prediction compared to the ground truth output using the training dataset. The estimated NMSE is 5.75×10^{-4} . C) Experimentally obtained output prediction compared to the ground truth output using the testing dataset. The estimated NMSE is 7.81×10^{-4} .

Table 2. Comparison table of solving second-order nonlinear dynamic system. (Note: here, reservoir energy has been compared only for the test dataset).

Work	Train [NMSE]	Test [NMSE]	Reservoir states	Reservoir energy [J]
This work	5.75×10^{-4}	7.81×10^{-4}	50	2.72×10^{-8}
Du et al. ^[32]	3.61×10^{-3}	3.13×10^{-3}	90	3.34×10^{-4}

this research is the real-time classification of class Z (healthy) and S (epileptic) EEG signals.

Figure 5 presents the process flow for EEG dataset classification. To assess the classification performance of our system using minimal information from raw data, we took the absolute value of the signal and clipped it to the highest value of 300 μ V. The signal is then converted into a voltage pulse train ranging between 100 and 200 mV, with a pulse width of ≈ 6 ms. This

pulse train is then introduced to a memcapacitive reservoir, which causes alterations in the reservoir's dynamics. To optimize feature size, a feature modification layer is added after the reservoir, where integration occurs.

For instance, each EEG data point consists of a voltage signal extending over 4097-time steps. We divided the entire capacitance value into 68 virtual nodes. As illustrated in Figure 5, we integrated the capacitance between virtual nodes rather than capturing it at the virtual node itself, recording the value at the (60n)th step (with $n = 1, 2, 3, \dots, 68$). In this manner, the capacitance value for 60-time steps generates a single feature, leading to a total of 68 features for each data point, thereby enabling real-time classification. While the virtual node technique reduces feature sizes, it retains less past information. Conversely, the virtual node method with the integration technique preserves past information from previous steps, ultimately enhancing performance. In the output layer, logistic regression is performed.

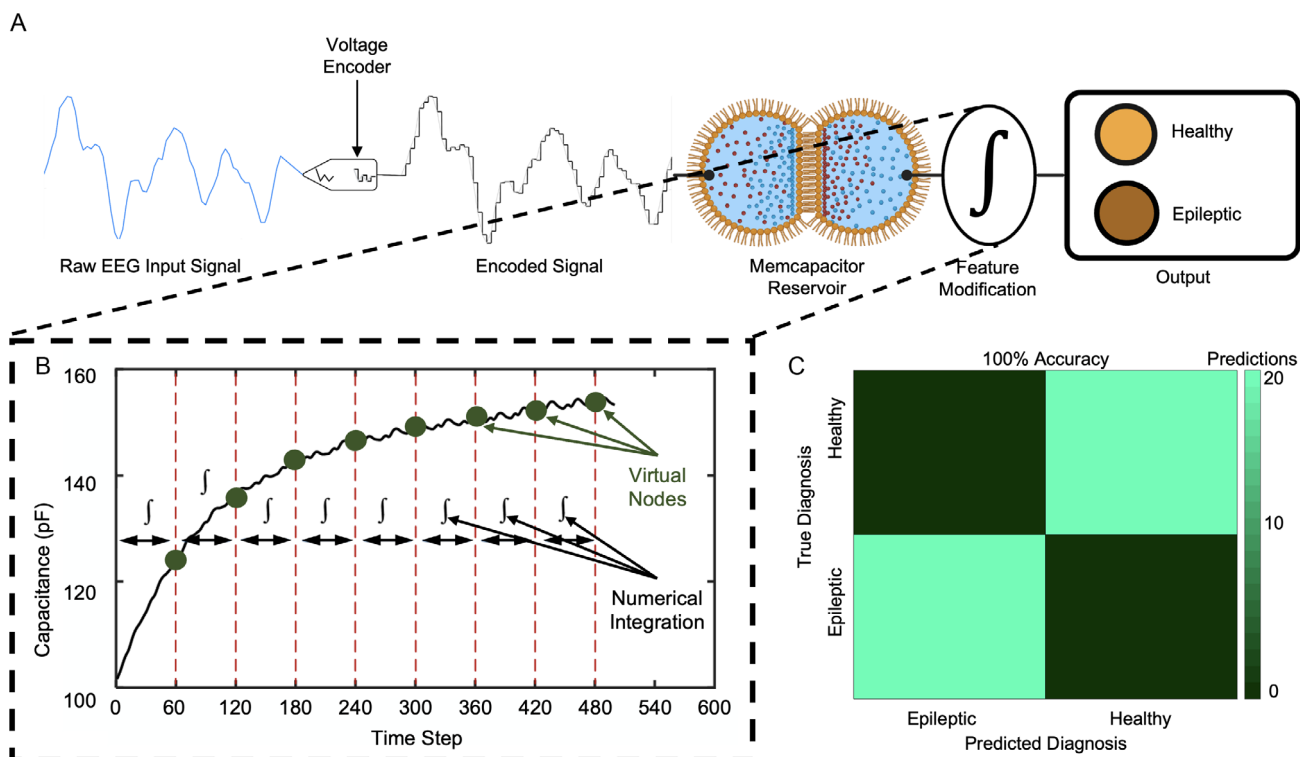


Figure 5. Epilepsy detection task from an input EEG signal using a memcapacitor-based RC system. A) The process flow of EEG dataset classification using a memcapacitor-based RC system. Note that the additional feature modification step precedes the readout. B) The feature modification postmapping step numerically integrates the memcapacitor states every 60-time step and combines these 60 features into one combined feature. C) A Confusion Matrix summarizing the prediction results for the EEG signal classification using the memcapacitor-based RC system. The results convey a 100% testing accuracy.

By employing the memcapacitor-based RC system, we achieved 100% accuracy for both training and testing EEG data.

2.5. Energy Consumption of Memcapacitor-Based RC

To evaluate the energy efficiency of our memcapacitor, we computed the mean power as well as energy consumed per spike via simulation and compared it to six state-of-the-art memristors that were previously employed for RC (Du et al.,^[32] Midya et al.,^[34] Zhu et al.,^[35] Zhong et al.,^[37] Najem et al.,^[61] Maraj et al.^[62]). We transmitted a uniform distribution random signal for each device model ranging from 0 to 1, spanning 1000 data points with appropriate voltage encoding parameters. That is, for every memristor device model, the random input signal was encoded into a voltage pulse train with a pulse width and voltage range that match the values reported in their corresponding articles. Then, the resulting unique 1000-pulse trains were fed to their corresponding devices, and the current responses were simulated via the reported device models. The point-by-point products of the simulated currents and input voltages were then averaged to yield power consumption, numerically integrated, and then averaged per pulse width basis to yield the mean energy consumption per spike. In contrast, the energy per spike calculation of the memcapacitor was done using the product of simulated capacitance and the square of the voltage difference for

increasing voltage amplitudes only. Meaning that we only consider the energy involved in charging the memcapacitor and disregard the discharging energy. As a result, the memcapacitor's energy per spike and mean power are almost five and three orders of magnitude, respectively, lower than that of the most energy-efficient memristor^[35] included in this study. In addition, the energy per spike of the memcapacitor is independent of the chosen input voltage pulse width, in contrast to memristors and, more generically, energy-dissipative devices in which energy consumption scales with the voltage pulse width. This is due to the fundamental dependence of capacitive current on the dynamic voltage transience as opposed to the dependence of ohmic current on the static voltage magnitude and on-time. As a result, memcapacitors only consume dynamic energy during pulse transition and do not consume any static energy during the pulse duration making the energy consumption completely independent of the pulse width. In the context of real-time memristor-based RC, the task of interest dictates the input voltage pulse width, and therefore the energy required to solve the problem. For instance, to classify distinct neural activity patterns in real-time, Zhu et al.^[35] had to select a pulse width of 2 ms for their input voltage to mimic the spike time of a biological action potential even though their memristor is able to operate at shorter timescales (500 μ s). This problem-related imposition of pulse width resulted in an increase in their energy consumption per spike from 50 fJ to almost 10 pJ. Unlike memristor-based

reservoirs, the energy consumption of real-time memcapacitor-based RC is solely dependent on the device's capacitance and encoded input voltage magnitude, which are both dictated by the device's properties. For our memcapacitor, the energy consumption is a function of capacitance, which is proportional to the bilayer area and, consequently, the droplets' sizes (Figure 1A and S1, Supporting Information). For this analysis, we chose 50 nL-sized droplets, which ultimately yield a mean capacitance of ≈ 20 pF, 41.5 fJ of energy per spike, and a mean power consumption of 415 fW for a 100 ms pulse width. A log-scale bar graph is indicated in Figure 6 as a visual summary of the results.

3. Discussion

In this study, we present a physical memcapacitive RC system that takes advantage of the inherent short-term memory capabilities of biomembrane-based memcapacitive devices to fully replace network-based RC systems. Our approach utilizes the concept of virtual nodes for prediction tasks in the context of spoken digit classification problems. Notably, our reservoir system demonstrates the ability to classify data even when faced with incomplete inputs. When tackling regression such as second-order nonlinear dynamic problems, our system can accurately predict the actual output, even in the absence of knowledge about the transfer function. To emphasize the importance of the fading memory property of the device for this specific issue, we compared our memcapacitive RC system with a conventional linear network. We observed a significant difference in prediction NMSE (see Section S3.2, Supporting Information), indicating the superiority of our approach. Furthermore, the versatility of our system is demonstrated through its ability to process not only temporal data but also static, real-time data classification. In this regard, we have successfully addressed the epilepsy detection problem. Additionally, we solved IRIS dataset classification problem (Section S5 and Figure S12, Supporting Information) in

simulation to demonstrate the device's capability in solving problems with a static input. Finally, we demonstrated, in simulation, that the energy per spike consumption of the memcapacitor is not only orders of magnitude less than state-of-the-art memristors but also is independent of the pulse width of the input voltage, which is of particular interest in real-time physical RC.

The memcapacitor-based RC system presented in this work is expected to pave the way for ongoing advancements targeting the optimization of network performance across a diverse array of applications, such as action recognition, prediction, and classification. This method holds particular promise for applications that prioritize network size and energy efficiency over rapid processing speeds. In our current implementation, the reservoir is constructed in hardware, while input and output layers reside within a simulation platform. As part of our future research endeavors, we aim to build the entire system in hardware, thereby addressing real-time application challenges more effectively. Additionally, we plan to incorporate algorithmic advancements, such as improving the input encoding and new memcapacitive reservoir architecture, into our work. These improvements in simulation and experimental contexts will establish the groundwork for the application of memcapacitive RC systems in machine learning tasks tailored to neuromorphic computing applications.

We note that while solid-state memristors have achieved notable integration into VLSI circuits and are pervasive in industrial and commercial applications, in many different classes of applications, it is more appropriate to implement computation directly with devices that compute with biological mechanisms. We underscore that the goal of such biomolecular-device-based computing is not to compete with electronic circuits in terms of computational speed or size. The main advantage of biomolecular-device-based computing systems is their environment of application along with energy efficiency.

A primary motivation behind our work is directly linked to the objective of physical RC, where the main aim is to harness the intrinsic computational capabilities of specific materials while

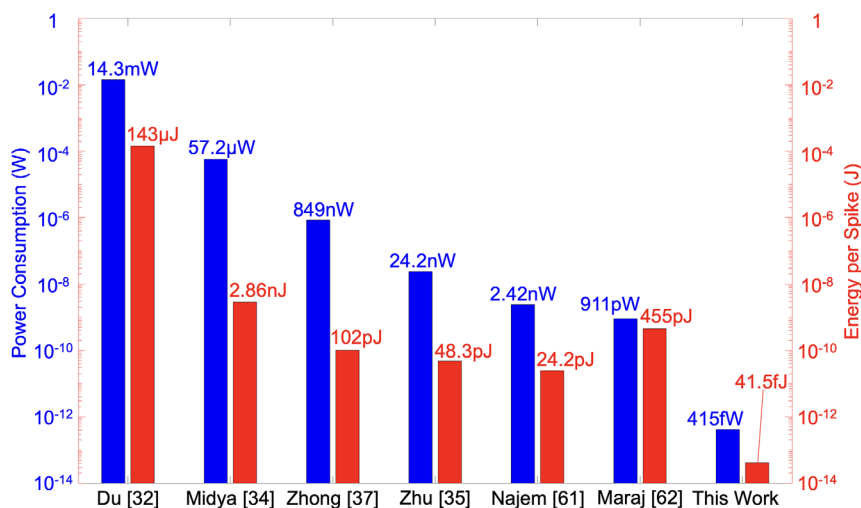


Figure 6. A comparison of power and energy per spike consumption is plotted on a log scale, showcasing the significant difference in power and energy between the memcapacitor used in this study and state-of-the-art memristors employed as reservoirs. The memcapacitor exhibits orders of magnitude lower power and energy per spike consumption compared to the memristor.^[32,34,35,37,61,62]

possessing physical properties desirable for applications that are not directly connected to their computational purpose.^[22] Previous research indicates the potential of this approach, particularly in edge computing^[22,35] for biomedical and brain-computing interfaces, where the inherent attributes of complementary metal–oxide–semiconductor (CMOS)-based devices may be less advantageous in the biological milieu. However, iontronic, biocompatible tools, such as our memcapacitor, are more likely to be suited. Though memcapacitors may not match the scalability of traditional CMOS-based systems, strides have been^[82–91] and are still being made to advance their scalability. Such neuromorphic devices and architectures exhibit a profound potential to offer rich dynamics and computing paradigms that reduce the need for extensive nanoscale devices, as evident from our and past studies.^[32,35,37,62]

Despite scalability limitations compared to solid-state devices, by exploring and harnessing the unique properties of volatile memcapacitive devices, we seek to revolutionize the field of RC and contribute to the development of more efficient and versatile neuromorphic systems. As the demand for intelligent and energy-efficient computing solutions grows, the findings of this study hold significant potential for transforming a wide range of industries and applications. Ultimately, the memcapacitive RC system presented here is a crucial step forward in the pursuit of next-generation, energy-efficient neuromorphic computing systems.

4. Experimental Section

Lipid Solutions Preparation and Membrane Assembly: An aqueous stock solution containing 500 mM potassium chloride (KCl, Sigma), 10 mM 3-morpholino propane-1-sulfonic acid (MOPS, Sigma) with a measured pH of 5.8, and 2 mg mL⁻¹ 1,2-Diphytanoyl-sn-glycero-3-phosphocholine (DPhPC, Avanti) liposomes in deionized water (18.2 MΩ cm) were prepared and stored. To prepare this aqueous stock solution, 160 μL of 25 mg mL⁻¹ DPhPC lipids dissolved in chloroform solution are acquired and evaporated under clean dry air, leaving 4 mg of residual lipid cake at the bottom of a 4 mL vial. The vial is then left under vacuum for a minimum of 2 hours and then hydrated with 2 mL of 500 mM KCl, 10 mM MOPS buffer solution, resulting in an aqueous solution containing 2 mg mL⁻¹ multilamellar DPhPC liposomes. To convert the multilamellar liposomes to unilamellar liposomes, the lipid solution is first subjected to six freeze/thaw cycles, and then extruded by forcing it, in 11 immediately successive passes, through a 100 nm-pore polycarbonate membranes (Whatman) using an Avanti Mini Extruder. Finally, the extruded solution is sonicated for 5 min and vortex mixed for 60 s.^[92–94] This solution can be stored for weeks at 4 °C or directly used for experimentation. A micropipette was used to pipette two 200 nL droplets from the prepared lipid stock solution onto two 125 μm-diameter, ball-end silver/silver chloride (Ag/AgCl 99.99%, Goodfellow) wires submerged in a decane (≥99%, Sigma-Aldrich) oil-filled, transparent acrylic reservoir. Prior to droplet deposition, the wires were coated with 1% agarose gel to avoid droplets detaching from the wires due to the decreasing surface tension associated with the lipid monolayer formation.

Initially, the droplets were suspended on the wires free of contact with each other, the acrylic substrate, and the oil/air interface for 5–7 min, allowing for a packed lipid monolayer to form at the droplets' water/oil interface. The monolayer formation was monitored visually via a 4× objective lens on an Olympus IX73 inverted microscope. Once the monolayers were formed, which were detected by observing the droplets leave the lens' focal plane as they vertically sagged from the wires, the droplets were brought in contact with each other by moving the wires using 3-axis micromanipulators to spontaneously form a bilayer at the contact interface.

Electrical Measurements Setup: Prior to implementing any problem-specific transmembrane voltage signal, we ensured a successful interfacial bilayer formation by supplying a triangular 10 Hz, 10 mV voltage signal to the electrodes via a Tektronix AFG31022 function generator. As a result of the membrane's high insulation (>100 MΩ cm) and capacitive interfacial area, a small 10 Hz square (≈20 pA) current response is expected as an output from a nonleaking bilayer. To obtain dynamic and steady-state changes in capacitance as a function of voltage (Figure S2, S3, and S5, Supporting Information), a voltage waveform consisting of a 50 mHz, 150 mV amplitude sinusoidal waveform superimposed on a 20 Hz, 10 mV triangular waveform was supplied to the membrane. The slow-frequency component of the waveform drove the geometric reconfiguration of the membrane (i.e., EW and EC) while the fast-frequency component was used to obtain the capacitance magnitude at every semiperiod. For the capacitance step response in Figure S4, Supporting Information, the low-frequency component was a 10 s, 150 mV square wave. A custom MATLAB script (available upon request) was used to compute the capacitance at every semiperiod by fitting the analytical solution of a parallel RC circuit current response to the measured current. Simultaneously, the bilayer area changes (Figure S1, Supporting Information) were monitored at 30 fps using a camera attached to the inverted microscope. The corresponding videos, obtained via Olympus CellSens software, were then postprocessed using a custom MATLAB script (available upon request) to extract the bilayer's interfacial minor axis radius, which was used to compute the interfacial area changes (Figure S2, Supporting Information) and subsequently, the bilayer hydrophobic thickness changes (Figure S2, Supporting Information). A custom MATLAB script (available upon request) was used to control an NI 9264 voltage output module to send an arbitrary transmembrane voltage signal. The capacitance magnitude at every pulse was computed from the obtained current responses using a custom MATLAB script (available upon request).

All the current measurements were recorded and digitized at 50 000 samples s⁻¹ (to avoid capacitive-spike aliasing) using a patch-clamp amplifier Axopatch 200B and Digidata 1440 A data acquisition system (Molecular Devices), respectively. All current recordings are conducted on an active vibration isolation table and under appropriate shielding, using a lab-made Faraday cage, to reduce the noise to less than 2 pA.

All model simulations and energy consumption calculations were implemented using a custom MATLAB script (available upon request).

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

M.R.H. and A.S.M are both first co-authors and contributed equally to this work.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords

lipid bilayer, memcapacitors, neuromorphic computing, nonlinearity, physical reservoir computing, short-term plasticity, volatile memory

Received: June 20, 2023

Revised: September 25, 2023

Published online: October 17, 2023

- [1] A. Graves, A. r. Mohamed, G. Hinton, in *2013 IEEE Int. Conf. Acoustics, Speech and Signal Processing*, IEEE **2013**, pp. 6645–6649.
- [2] H. Sak, A. Senior, K. Rao, F. Beaufays, arXiv preprint, arXiv:1507.06947, **2015**.
- [3] T. Mikolov, M. Karařát, L. Burget, J. Āernocký, S. Khudanpur, *Interspeech*, Vol. 2, Makuhari **2010**, pp. 1045–1048.
- [4] S. Wiseman, S. M. Shieber, A. M. Rush, in *Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Brussels, Belgium **2018**, pp. 3174–3187.
- [5] A. G. Salman, B. Kanigoro, Y. Heryadi, in *Proc. 2015 Int. Conf. on Advanced Computer Science and Information Systems (ICACSIS 2015)*, IEEE, Depok, Indonesia **2016**, pp. 281–285.
- [6] Y. Li, R. Yu, C. Shahabi, Y. Liu, in *6th Int. Conf. on Learning Representations, (ICLR 2018) – Conf. Track Proc.*, Vancouver, BC, Canada **2018**.
- [7] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, K. P. Soman, in *Int. Conf. on Advances in Computing, Communications and Informatics, (ICACCI 2017)*, IEEE, Udupi, India **2017**, pp. 1643–1647.
- [8] A. K. Rout, P. K. Dash, R. Dash, R. Bisoi, *J. King Saud Univ. Comput. Inform. Sci.* **2017**, *29*, 536.
- [9] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, L. Farhan, *J. Big Data* **2021**, *8*, 53.
- [10] K. Doya, *Proc. IEEE Int. Symp. Circuits Syst.* **1992**, *5*, 2777.
- [11] M. Lukoševičius, H. Jaeger, B. Schrauwen, *KI - Künstliche Intelligenz* **2012**, *26*, 365.
- [12] H. Jaeger, *The “echo state” approach to analysing and training recurrent neural networks*, GMD Technical Report No. 148, German National Research Center for Information Technology, Bonn, Germany **2001**, p. 13.
- [13] W. Maass, T. Natschläger, H. Markram, *Neural Comput.* **2002**, *14*, 2531.
- [14] M. Rabinovich, R. Huerta, G. Laurent, *Science* **2008**, *321*, 48.
- [15] D. Verstraeten, B. Schrauwen, M. D’Haene, D. Stroobandt, *Neural Networks* **2007**, *20*, 391.
- [16] M. Lukoševičius, H. Jaeger, *Comput. Sci. Rev.* **2009**, *3*, 127.
- [17] T. M. Cover, *IEEE Trans. Electron. Comput.* **1965**, *EC-14*, 326.
- [18] M. C. Oztuik, D. Xu, J. C. Principe, *Neural Comput.* **2007**, *19*, 111.
- [19] B. Schrauwen, D. Verstraeten, J. Van Campenhout, *IIOimport* **2007**, 471.
- [20] M. Lukoševičius, *Neural Networks: Tricks of the Trade*, 2nd ed., Springer **2012**, pp. 659–686.
- [21] L. Appeltant, M. C. Soriano, G. Van Der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, I. Fischer, *Nat. Commun.* **2011**, *2*, 468.
- [22] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, A. Hirose, *Neural Networks* **2019**, *115*, 100.
- [23] C. S. Dunham, S. Lilak, J. a. Hochstetter, M. Cucchi, S. Abreu, K. Nakajima, *Jpn. J. Appl. Phys.* **2020**, *59*, 060501.
- [24] J. Torrejon, M. Riou, F. A. Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. D. Stiles, J. Grollier, *Nature* **2017**, *547*, 7664.
- [25] W. Jiang, L. Chen, K. Zhou, L. Li, Q. Fu, Y. Du, R. Liu, *Appl. Phys. Lett.* **2019**, *115*.
- [26] H. O. Sillin, R. Aguilera, H. H. Shieh, A. V. Avizienis, M. Aono, A. Z. Stieg, J. K. Gimzewski, *Nanotechnology* **2013**, *24*, 384004.
- [27] S. Lilak, W. Woods, K. Scharnhorst, C. Dunham, C. Teuscher, A. Z. Stieg, J. K. Gimzewski, *Front. Nanotechnol.* **2021**, *3*, 38.
- [28] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, P. Bienstman, *Nat. Commun.* **2014**, *5*, 3541.
- [29] G. Van Der Sande, D. Brunner, M. C. Soriano, *Nanophotonics* **2017**, *6*, 561.
- [30] K. Toprasertpong, E. Nako, Z. Wang, R. Nakane, M. Takenaka, S. Takagi, *Commun. Eng.* **2022**, *1*, 21.
- [31] N. T. Duong, Y.-C. Chien, H. Xiang, S. Li, H. Zheng, Y. Shi, K.-W. Ang, *Adv. Intell. Syst.* **2023**, 2300009.
- [32] C. Du, F. Cai, M. A. Zidan, W. Ma, S. H. Lee, W. D. Lu, *Nat. Commun.* **2017**, *8*, 2204.
- [33] J. Moon, W. Ma, J. H. Shin, F. Cai, C. Du, S. H. Lee, W. D. Lu, *Nat. Electron.* **2019**, *2*, 480.
- [34] R. Midya, Z. Wang, S. Asapu, X. Zhang, M. Rao, W. Song, Y. Zhuo, N. Upadhyay, Q. Xia, J. J. Yang, R. Midya, Z. Wang, S. Asapu, X. Zhang, M. Rao, W. Song, Y. Zhuo, N. Upadhyay, Q. Xia, J. J. Yang, *Adv. Intell. Syst.* **2019**, *1*, 1900084.
- [35] X. Zhu, Q. Wang, W. D. Lu, *Nat. Commun.* **2020**, *11*, 2439.
- [36] M. R. Hossain, J. S. Najem, T. Rahman, M. S. Hasan, in *Proc. of the IEEE Conf. on Nanotechnology*, IEEE, Piscataway, NJ **2021**, pp. 116–119.
- [37] Y. Zhong, J. Tang, X. Li, B. Gao, H. Qian, H. Wu, *Nat. Commun.* **2021**, *12*, article no. 408.
- [38] J. Cao, X. Zhang, H. Cheng, J. Qiu, X. Liu, M. Wang, Q. Liu, *Nanoscale* **2022**, *14*, 289.
- [39] L. O. Chua, *IEEE Trans. Circuit Theory* **1971**, *18*, 507.
- [40] M. R. Hossain, P. S. Paul, M. Sadia, A. Dhungel, J. S. Najem, M. S. Hasan, in *Int. Conf. on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME 2022)*, IEEE, Maldives **2022**.
- [41] L. O. Chua, *IEEE Trans. Circuits Syst.* **1980**, *27*, 1014.
- [42] M. S. Abdelouahab, R. Lozi, L. Chua, *Int. J. Bifurc. Chaos* **2014**, *24*, 1430023.
- [43] D. Bialek, Z. Bialek, V. Biolkova, *Electron. Lett.* **2016**, *52*, 1744.
- [44] Y. V. Pershin, M. D. Ventra, *Electron. Lett.* **2014**, *50*, 141.
- [45] K. U. Demasius, A. Kirschen, S. Parkin, *Nat. Electron.* **2021**, *4*, 748.
- [46] M. D. Ventra, Y. V. Pershin, L. O. Chua, *Proc. IEEE* **2009**, *97*, 1717.
- [47] J. Martinez-Rincon, M. D. Ventra, Y. V. Pershin, *Phys. Rev. B: Condens. Matter Mater. Phys.* **2010**, *81*, 195430.
- [48] M. G. A. Mohamed, H. Kim, T. W. Cho, *Sci. World J.* **2015**, *2015*, 910126, <https://doi.org/10.1155/2015/910126>.
- [49] A. K. Khan, B. H. Lee, *AIP Adv.* **2016**, *6*, 095022.
- [50] T. You, L. P. Selvaraj, H. Zeng, W. Luo, N. Du, D. Bürger, I. Skorupa, S. Prucnal, A. Lawrenz, T. Mikolajick, O. G. Schmidt, H. Schmidt, T. You, L. P. Selvaraj, N. Du, D. Bürger, I. Skorupa, O. G. Schmidt, H. Schmidt, H. Zeng, W. Luo, S. Prucnal, A. Lawrenz, T. Mikolajick, *Adv. Electron. Mater.* **2016**, *2*, 1500352.
- [51] Z. Wang, M. Rao, J. W. Han, J. Zhang, P. Lin, Y. Li, C. Li, W. Song, S. Asapu, R. Midya, Y. Zhuo, H. Jiang, J. H. Yoon, N. K. Upadhyay, S. Joshi, M. Hu, J. P. Strachan, M. Barnell, Q. Wu, H. Wu, Q. Qiu, R. S. Williams, Q. Xia, J. J. Yang, *Nat. Commun.* **2018**, *9*, 1.
- [52] Q. Zheng, Z. Wang, N. Gong, Z. Yu, C. Chen, Y. Cai, Q. Huang, H. Jiang, Q. Xia, R. Huang, *IEEE Electron Device Lett.* **2019**, *40*, 1309.
- [53] D. Kwon, I. Y. Chung, *IEEE Electron Device Lett.* **2020**, *41*, 493.

- [54] J. S. Najem, M. S. Hasan, R. S. Williams, R. J. Weiss, G. S. Rose, G. J. Taylor, S. A. Sarles, C. P. Collier, *Nat. Commun.* **2019**, *10*, article no. 3239.
- [55] D. Bielek, M. D. Ventra, Y. V. Pershin, *Radioengineering* **2013**, *22*, 945.
- [56] S. J. D. Tran, C. Teuscher, in *Proc. of the IEEE/ACM Inter. Symp. on Nanoscale Architectures (NANOARCH 2017)*, IEEE, Piscataway, NJ **2017**, pp. 115–116.
- [57] S. J. D. Tran, C. Teuscher, in *15th IEEE Int. Conf. on Nano/Micro Engineered and Molecular System (NEMS)*, IEEE, Piscataway, NJ **2020**, pp. 200–205.
- [58] D. Tran, C. Teuscher, *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2023**, *13*, 323.
- [59] S. J. D. Tran, C. Teuscher, in *Proc. of the 4th IEEE Int. Conf. on Rebooting Computing*, IEEE, Piscataway, NJ **2019**.
- [60] R. S. Zucker, W. G. Regehr, *Annu. Rev. Physiol.* **2003**, *64*, 355.
- [61] J. S. Najem, G. J. Taylor, R. J. Weiss, M. S. Hasan, G. Rose, C. D. Schuman, A. Belianinov, C. P. Collier, S. A. Sarles, *ACS Nano* **2018**, *12*, 4702.
- [62] J. J. Maraj, K. P. T. Haughn, D. J. Inman, S. A. Sarles, *Adv. Intell. Syst.* **2023**, *5*, 2300049.
- [63] C. Gallicchio, A. Micheli, L. Pedrelli, *Neurocomputing* **2017**, *268*, 87.
- [64] J. B. Butcher, D. Verstraeten, B. Schrauwen, C. R. Day, P. W. Haycock, *Neural Networks* **2013**, *38*, 76.
- [65] J. J. Yang, D. B. Strukov, D. R. Stewart, *Nat. Nanotechnol.* **2013**, *8*, 13.
- [66] J. Dambre, D. Verstraeten, B. Schrauwen, S. Massar, *Sci. Rep.* **2012**, *2*, 514.
- [67] M. Inubushi, K. Yoshimura, *Sci. Rep.* **2017**, *7*, 10199.
- [68] L. A. Aguirre, C. Letellier, *Math. Probl. Eng.* **2009**, *2009*, 238960, <https://doi.org/10.1155/2009/238960>.
- [69] A. K. Alomari, N. R. Anakira, A. S. Bataineh, I. Hashim, *Math. Probl. Eng.* **2013**, *2013*, 136043, <https://doi.org/10.1155/2013/136043>.
- [70] Z. Varga, *Period. Math. Hung.* **2008**, *56*, 157.
- [71] T. N. Palmer, *J. Clim.* **1999**, *12*, 575.
- [72] Y. H. Li, X. J. Tian, X. Q. Li, *ISRN Appl. Math.* **2013**, *2013*, 981692.
- [73] M. Ruderman, *J. Franklin Inst.* **2021**, *358*, 4292.
- [74] Y. Xu, J. M. Hollerbach, D. Ma, *IEEE Control Syst. Mag.* **1995**, *15*, 15.
- [75] A. T. Vo, T. N. Truong, H. J. Kang, *IEEE Access* **2021**, *9*, 31373.
- [76] J. Mei, W. Ren, B. Li, G. Ma, *IEEE Trans. Neural Networks Learn. Syst.* **2015**, *26*, 1885.
- [77] A. T. Zallas, M. G. Tsipouras, D. I. Fotiadis, *IEEE Trans. Inform. Technol. Biomed.* **2009**, *13*, 703.
- [78] E. Başar, *Int. J. Psychophysiol.* **2012**, *86*, 1.
- [79] J. J. M. Reynolds, J. S. Plank, C. D. Schuman, G. R. Bruer, A. W. Disney, M. E. Dean, G. S. Rose, *ICONS '18: Proc. Inter. Conf. on Neuromorphic Systems*, Association for Computing Machinery, New York **2018**, article no. 12, <https://doi.org/10.1145/3229884.3229896>.
- [80] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, C. E. Elger, *Phys. Rev. E* **2001**, *64*, 061907.
- [81] A. Nishad, A. Upadhyay, G. R. S. Reddy, V. Bajaj, *Electron. Lett.* **2020**, *56*, 1370.
- [82] T. M. Schimmel, M. A. Nguyen, S. A. Sarles, S. C. Lenaghan, *Microfluid. Nanofluid.* **2021**, *25*, 78.
- [83] M. A. Nguyen, S. A. Sarles, in *ASME 2016 Conf. on Smart Materials, Adaptive Structures and Intelligent Systems (SMASIS 2016)*, American Society of Mechanical Engineers (ASME), Stowe, Vermont **2016**, Vol. 2, <https://doi.org/10.1115/SMASIS2016-9034>.
- [84] M. A. Nguyen, B. Srijanto, C. P. Collier, S. T. Retterer, S. A. Sarles, *Lab Chip* **2016**, *16*, 3576.
- [85] M. A. Nguyen, G. Taylor, S. A. Sarles, in *ASME 2017 Conf. on Smart Materials, Adaptive Structures and Intelligent Systems (SMASIS 2017)*, American Society of Mechanical Engineers (ASME), Snowbird, UT **2017**, Vol. 1, <https://doi.org/10.1115/SMASIS2017-3878>.
- [86] G. Villar, A. D. Graham, H. Bayley, *Science* **2013**, *340*, 48.
- [87] A. Alcinesio, O. J. Meacock, R. G. Allan, C. Monico, V. R. Schild, I. Cazimoglu, M. T. Cornall, R. K. Kumar, H. Bayley, *Nat. Commun.* **2020**, *11*, 2105.
- [88] A. Alcinesio, R. K. Kumar, H. Bayley, *ChemSystChem* **2022**, *4*, 202100036.
- [89] A. D. Graham, S. N. Olof, M. J. Burke, J. P. K. Armstrong, E. A. Mikhailova, J. G. Nicholson, S. J. Box, F. G. Szele, A. W. Perriman, H. Bayley, *Sci. Rep.* **2017**, *7*, 7004.
- [90] S. A. Sarles, D. J. Leo, *Lab Chip* **2010**, *10*, 710.
- [91] E. J. Challita, J. S. Najem, R. Monroe, D. J. Leo, E. C. Freeman, *Sci. Rep.* **2018**, *8*, 6494.
- [92] G. J. Taylor, S. A. Sarles, *Langmuir* **2015**, *31*, 325.
- [93] G. J. Taylor, G. A. Venkatesan, C. P. Collier, S. A. Sarles, *Soft Matter* **2015**, *11*, 7592.
- [94] J. S. Najem, G. J. Taylor, N. Armendarez, R. J. Weiss, M. S. Hasan, G. S. Rose, C. D. Schuman, A. Belianinov, S. A. Sarles, C. P. Collier, *JoVE* **2019**, <https://doi.org/10.3791/58998>.