10-2022

Multi-tasking Memcapacitive Networks

Dat Tran
Portland State University, datran@pdx.edu

Christof Teuscher
Portland State University, teuscher@pdx.edu

Citation Details

Published as: Tran, D., & Teuscher, C. (2023). Multi-tasking Memcapacitive Networks. IEEE Journal on Emerging and Selected Topics in Circuits and Systems.

Multi-tasking Memcapacitive Networks

Dat Tran, Electrical and Computer Engineering, Santa Clara University, *Member, IEEE* *
Christof Teuscher, Electrical and Computer Engineering, Portland State University, *Senior Member, IEEE*

***Abstract*—Recent studies have shown that networks of memcapacitive devices provide an ideal computing platform of low power consumption for reservoir computing systems. Random, crossbar, or small-world power-law (SWPL) structures are common topologies for reservoir substrates to compute single tasks. However, neurological studies have shown that the interconnections of cortical brain regions associated with different functions form a rich-club structure. This structure allows human brains to perform multiple activities simultaneously. So far, memcapacitive reservoirs can perform only single tasks. Here, we propose, for the first time, cluster networks functioning as memcapacitive reservoirs to perform multiple tasks simultaneously. Our results illustrate that cluster networks surpassed crossbar and SWPL networks by factors of $4.1\times, 5.2\times$, and $1.7\times$ on three tasks: Isolated Spoken Digits, MNIST, and CIFAR-10. Compared to single-task networks in our previous and published results, multitasking cluster networks could accomplish similar accuracies of 86%, 94.4%, and 27.9% for MNIST, Isolated Spoken Digits, and CIFAR-10. Our extended simulations reveal that both the input signal amplitudes and the inter-cluster connections contribute to the accuracy of cluster networks. Selecting optimal values for signal amplitudes and inter-cluster links is key to obtaining high classification accuracy and low power consumption. Our results illustrate the promise of memcapacitive brain-inspired cluster networks and their capability to solve multiple tasks simultaneously. Such novel computing architectures have the potential to make edge applications more efficient and allow systems that cannot be reconfigured to solve multiple tasks.**

***Index Terms*—memcapacitor, multi-tasking, network, topology, brain-inspired**

I. Introduction

THE emergence of new nano-devices has enabled the development of brain-inspired computing substrates with unique and promising properties. For example, memory devices, such as memristors and memcapacitors, allow for the direct implementation of synaptic connections, which are an essential part of *artificial neural networks* (ANNs) to mimic the function of a biological brain. Recent studies have shown that memcapacitive networks can solve neuromorphic tasks with high energy efficiency [1], [2]. The nonlinear characteristics of memcapacitive networks are ideal for neuromorphic computing, particularly in *reservoir computing* (RC) [1], [3], [4]. Reservoir computing is an alternative machine learning approach for brain-inspired computing systems that describes the higher-order cognitive functions and the short-term memory processes [5]. Results from different studies illustrate that random networks [6], regular as crossbar topologies [7], or *small-world power-law* (SWPL) topologies [4] have enough dynamics to function as reservoirs to perform simple tasks. In our previous work, we have demonstrated that the SWPL topologies provide a suitable structure for memcapacitive networks to perform tasks with high energy efficiency and low hardware cost [4].

Neurological studies have revealed that the human brain is organized in a rich-club structure, in which cortical brain regions (clusters) contain highly dense and inter-region connections [8]. Each brain region connects to a specific activity and has a small-world feature to minimize the cost of information processing while maximizing its capacity for growth and adaptation [9]. With the rich-club structure, the human brain can perform multiple tasks simultaneously.

So far, memcapacitive reservoirs with different topologies could only perform single tasks. Here, we propose, for the first time, cluster networks functioning as memcapacitive reservoirs to perform multiple tasks simultaneously. Even though our simulation results are lower than those of the state-of-art systems due to subsets of data in training and testing to avoid long simulation times, they illustrate a potential physical implementation of multitasking memcapacitive reservoirs beyond a theoretical concept.

II. Multitasking Reservoir Computing

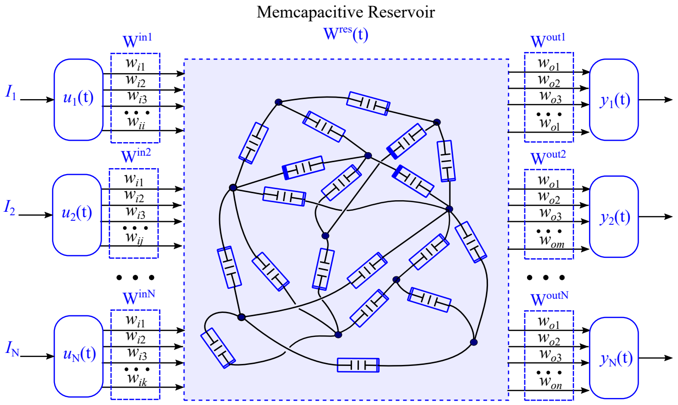


Fig. 1. Multi-tasking reservoir computing. Temporal and spatial information $(I_1, I_2, \ldots, I_N)$ from input data is transformed into temporal signals $[u_1(t), u_2(t), \ldots, u_N(t)]$ as input streams to the reservoir. The input vector W^{ini} limits the input signals I_i to prevent overstimulated reservoirs. The reservoir, composed of memcapacitive devices, translates the input signals $[u_1(t), u_2(t), \ldots, u_N(t)]$ into a high-dimensional space. Separate readout nodes $[y_1(t), y_2(t), \ldots, y_N(t)]$ are trained and tested for corresponding target functions. Readout nodes are trained by adjusting the output weight vectors $\left(W^{out1}, W^{out2}, \ldots, W^{outN}\right)$ using the ridge regression algorithm.

Feed-forward networks are a typical ANN architecture. However, the networks cannot retain information about previous timesteps from input streams. One of the solutions is to add recurrent connections to form *recurrent neural networks* (RNNs) so that the networks can retain information from the past. The recurrent connections, however, significantly

* This is the corresponding author: dtran3@scu.edu.

increase the training complexity of the networks. *Reservoir computing* (RC) is an alternative to avoid the training complexity of large recurrent networks. In an RC system, a reservoir (or, in our case, a memcapacitive network) remains untrained and maps input streams into high-dimensional internal states. A single readout layer extracts information from the reservoir and is trained with a simple gradient-descent algorithm.

Figure 1 shows a general architecture of multi-tasking memcapacitive reservoir systems. Temporal and spatial information of different tasks $(I_1, I_2, \ldots, I_N)$ is transformed into time signals as input streams $[u_1(t), u_2(t), \ldots, u_N(t)]$ to the memcapacitive reservoir. The scaling vector W^{inm}, whose values w_{im} are randomly selected from the set of $\{-\nu, +\nu\}$, limits input signal $u_m(t)$ to prevent overstimulated conditions in the reservoirs. The state matrix $W^{res}(t)$ represents the internal capacitance of memcapacitive devices as:

$$W^{res}(t) = [mc_1(t), mc_2(t), ..., mc_M(t)],$$

where $mc_i(t)$ is the capacitance of a memcapacitive device within the reservoir at time t. The time-dependent state $x(t)$ of the reservoir is defined as:

$$\begin{aligned} x(t+1) = & f\,\big[W^{res}(t)x(t) + W^{in1}u_1(t) + \\ & W^{in2}u_2(t) + \ldots + W^{inN}u_N(t)\big]\,, \end{aligned}$$

where $f()$ is the transforming function of a reservoir node, $u_i(t)$ is the time signal of temporal and spatial input I_i, and W^{ini} is the scaling input vector. The inner product of the reservoir state $x(t)$ and the output weight vector W^{outi} form a separate readout node $y_i(t)$ for each task:

$$y_i(t) = x(t)W^{outi},$$

where the output weight vector W^{outi} is trained with a simple technique such as the ridge regression algorithm. The description of the transforming function $f()$ is in Section II of the *Supplementary Materials*.

III. Network Structure for Multitasking Reservoirs

Our work considers three network structures for multitasking reservoirs: crossbar networks, SWPL networks, and our newly proposed SWPL cluster networks. Section III in the *Supplementary Materials* explains the formation of three network types in detail.

A. Fully Connected Networks using Crossbar Structure

Crossbars are typical networks for emergent nanodevices (memristors and memcapacitors). In crossbar networks, nanodevices are fabricated at the cross junctions of nanowires. Crossbar arrays can emulate the synaptic behavior of deep neural networks and perform pattern recognition tasks [2], [10].

Figure 2 illustrates an example of a 2-layer crossbar network that serves as a multi-tasking reservoir. The converted input streams from different tasks $[u_1(t), u_2(t), \ldots, u_N(t)]$ provide input signals to the reservoir through the input nodes I. The readout nodes $[y_1(t), y_2(t), \ldots, y_N(t)]$ extract information from the reservoir through the output nodes O for training and testing. Each readout node is trained with a ridge regression algorithm.

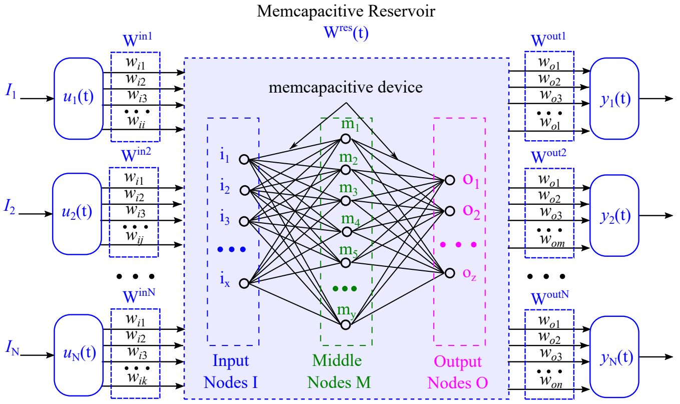


Fig. 2. A fully connected network for a multi-tasking reservoir. The input streams $[u_1(t), u_2(t), \ldots, u_N(t)]$ provide stimulus signals to the reservoir through the input nodes I. The readout nodes $[y_1(t), y_2(t), \ldots, y_N(t)]$ extract information from the reservoir through the output nodes O.

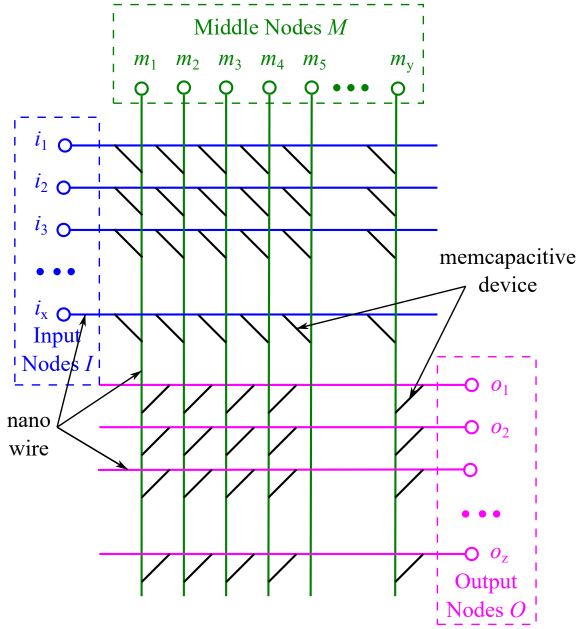


Fig. 3. The structure of 2-layer crossbar network. Memcapacitive devices are located at the cross junctions of nanowires. The input nodes (I) connect to the middle nodes (M) through memcapacitive devices at the cross junctions of nanowires. Similarly, the central nodes (M) attach to the output nodes (O) through nanowires.

Figure 3 shows an implementation of fully connected networks using a crossbar structure. The first group of horizontal nanowires (blue nanowires) functions as the input nodes (I). Each input node (i_i) links to all the nodes (vertical green nanowires) of the middle layer through memcapacitive devices located at the cross junctions of nanowires. The second group of horizontal nanowires (magenta nanowires) is output nodes (O) that provide output signals to readout nodes. Similar to a memristive crossbar structure, each node (m_i) in the middle layer (M) can connect to all output nodes.

B. Small-world Power-law Networks

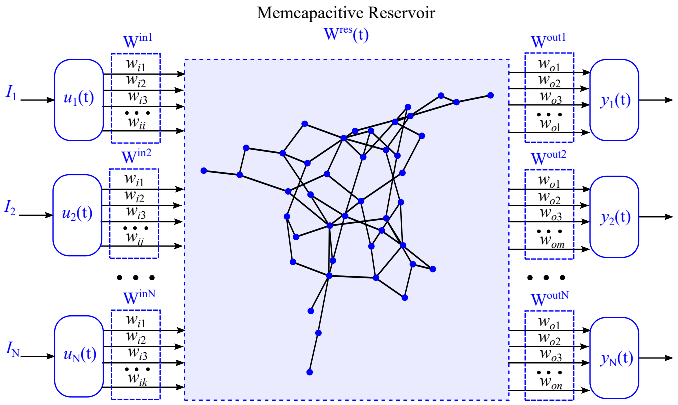


Fig. 4. Small-world power-law (SWPL) network for a multi-tasking reservoir. The dynamic response of SWPL reservoirs depends on locality α and randomness β [4]. Input stream I_i provides stimulus signals to the reservoir through a set of random input nodes. A readout node connects to the reservoir through another set of random output nodes.

Small-world (SW) networks enable a biological brain to improve the communication between different regions [11], its resilience against neural disease [12], and balances resources for growth and adaptation [9].

Real physical or biological SW networks tend to have more local connections than global links due to limited resources [9]. In our previous work, we showed that *small-world power-law* (SWPL) networks, the subset of SW networks, describe the true nature of biological and neural networks [4], [13]. The characteristics of SWPL networks depend on the locality α and the randomness β. Different combinations of α and β generate different responses from SWPL networks. The effect of α and β on the dynamic nature of SWPL networks is described in our previous study [4]. Although SW or SWPL topologies are a typical structure for reservoir computing [13], such as Echo State Networks [14], [15], memcapacitive SWPL networks offer a probable implementation of physical networks. Further explanations are in Section III of the *Supplementary Materials*.

As shown in Fig. 4, the time signal $u_i(t)$ from an input stream I_i stimulates the reservoir through a set of random input nodes. Another set of random nodes provides input signals to readout node $y_i(t)$ for a specific task. Similar to the crossbar multi-tasking reservoirs, the training stage of SWPL reservoirs utilizes a ridge regression algorithm.

C. Small-world Power-law Cluster Networks

fMRI scans revealed an internal structure of the human brain with segregation and integration where multiple activities are associated with different regions [16]. Such multitasking activities are possible due to the clustered structure of the brain. Here, we propose SWPL cluster networks as a structure for memcapacitive reservoirs to mimic the multiple activities in the human brain.

Figure 5 illustrates SWPL cluster networks for a multi-tasking memcapacitive reservoir. In this network, each cluster is responsible for a particular task. The input stream $u_i(t)$ from task I_i connects solely to a $Cluster_i$. The outputs of the $Cluster_i$ become the input signals of a readout node $y_i(t)$ to train and test for the task I_i. The in/out-degree ($k - degree$) describes the interconnections between clusters where information can travel from one cluster to another with the network.

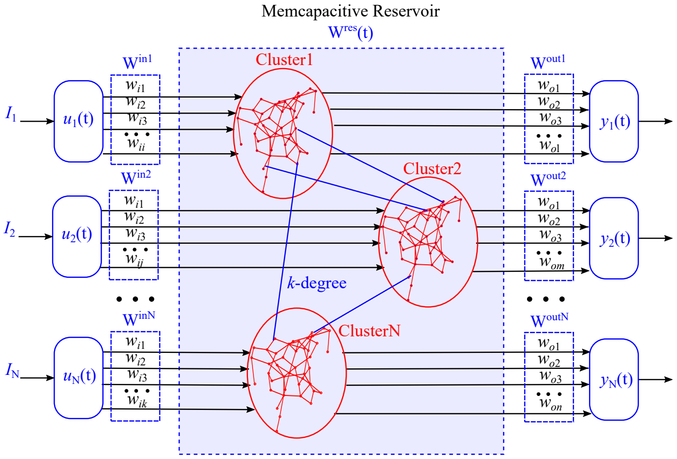


Fig. 5. Small-world power-law cluster network for a multi-tasking reservoir. Each cluster is responsible for a particular task. The input stream $u_i(t)$ from a particular task I_i connects to a separate cluster $Cluster_i$. This cluster provides the input signals for the readout node $y_i(t)$ of the task I_i. k-degree illustrates the interconnections between clusters within memcapacitive reservoirs.

IV. Methodology

A. Device Models

To establish a solid comparison, we selected four memcapacitive models for our multitasking networks: the *Biolek* behavior model, the *Mohamed* metal-oxide junction, the *Najem* biomimetic membranes, and the *MRMC* device that has both memristive and memcapacitive responses. We modified all models, except the*Najem* model, to express state volatility, which is an essential attribute of reservoir computing.

TABLE I
Memcapacitive Devices

Model †	Type	Values at w ◇		Phy. Dev.
		w_{max}	w_{min}	
Biolek [17] ⋆	NA	$100pF$	$1pF$	NA
Mohamed [18]	Metal-oxide	$6.5nF$	$1nF$	yes
Najem [19]	Biomimetic	$272pF$	$0.3pF$	yes
MRMC [1] ‡	$NdNiO_3$	$223.1M\Omega$ $370pF$	$14.3M\Omega$ $16pF$	yes

† These models were modified to exhibit state volatility, the essential property in reservoir computing, except the *Najem* model, which is the model of a volatile device.
◇ w represent a symbolic and internal state of a device between w_{min} and w_{max}. Different models have different variables to represent their internal state: C for the *Biolek* model, x and m for the *Mohamed* model, R and W for the *Najem* model, and x and m for the *MRMC* model (see Section I in *Supplementary Materials* for more detailed information).
* This model is not based on a physical device.
‡ This device has both memristive and memcapacitive behaviors.

Table I provides general characteristics of all memcapacitive models. Since experimental data for the *Mohamed* and *Najem* models are not available, we used the mathematical model equations published in studies [18], [19]. The *MRMC* model

This article has been accepted for publication in IEEE Journal on Emerging and Selected Topics in Circuits and Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JETCAS.2023.3235242

was from the experimental data of a physical device. The device modifications and descriptions are in Section I of the *Supplementary Materials*.

B. Power Calculation

Using the law of power conservation, we calculated the power consumption from the supplied power as follows:

$$p_s(t) = v_s(t)i_s(t),$$
$$P_{avg} = \int_0^T \sum_i^m p_k(\tau)d\tau,$$

where $v_s(t)$ and $i_s(t)$ are the voltage and the current of the supplied sources at a timestep t, $p_s(t)$ is the supplied power, T is the total simulation time, and P_{avg} is the average supplied power of the sources to networks.

C. Benchmark Tasks

We used three standard benchmarks to evaluate the classified accuracy of our multitasking memcapacitive reservoirs: MNIST, Isolated Spoken Digits, and CIFAR-10. Section IV in *Supplementary Materials* explains in detail the converting process of spatial and temporal information to input vectors for each dataset. The input vectors, through a set of random input nodes, provide signals to the reservoirs.

We performed a small experiment of collecting simulation times of training and testing SWPL cluster networks with a subset of data and the entire dataset. The results are in Section IV-D of *Supplementary Materials*. From the results, we observe that using the whole dataset for training and testing is an impractical option for finding optimal hyperparameters of networks. As a result, we used a subset of data as a feasible option for training and testing networks. We expect that using a subset of data for training and testing will impact the accuracy of networks.

The MNIST dataset is a collection of handwritten digit images (28×28) that includes 60,000 training and 10,000 testing images. The digit images extend into 784-pixel vectors as reservoir inputs for training and testing. Due to long simulation times, we only used a subset of data (1,000 images for training and 500 images for testing).

The spoken digit dataset contains the recordings of spoken digits in wave file format at 8 kHz [20]. The digit dataset holds 1,500 recordings of digits 0 to 9 from different English speakers. We divide the dataset into two non-overlapping sets: 1,000 digits for training and 500 digits for testing. Unlike the MNIST dataset, which contains only spatial information, the spoken digit dataset is more complex with spatial and temporal data. Through a preprocessing process of Mel-frequency cepstral coefficients, the spoken digits transform into 117-coefficient vectors as training and testing data [21].

CIFAR-10 is a collection of 60,000 color images (32×32) of ten object classes. Since CIFAR-10 images are highly dimensional due to the color representation of image features, we reduced the image dimension through a grayscale technique [22]. Since simulation times for CIFAR-10 images were quite long, we only used a subset of training and testing CIFAR-10 datasets (5000 training and 500 testing images).

D. Parallel and Sequential Mode for Training and Testing

We trained and tested SWPL cluster networks in sequential and parallel modes. For the sequential method, we trained each cluster with associated inputs successively. Unused inputs were connected to a grounding reference with high-value resistors. The description of the grounding resistors is in Section IV - F of the *supplementary materials*. In the parallel mode, all network clusters were simultaneously trained and tested. Since we only selected a subset of random training data due to simulation times, as mentioned in Section IV-C, the training subset could be an uneven combination of inputs from multiple datasets. For example, a random subset of data for the multiple tasks of Isolated Spoken Digits (task one) and MNIST (task 2) could have 200 data for digit one from task one and digit six from task 2, whereas only 10 data for digit zero and digit nine. The unbalanced input data could lead to a limitation in the learning stage where networks were trained many times with one target while only a few times for other targets. To avoid unbalanced data in a training process, we recombined input data using the technique in Fig. 6.

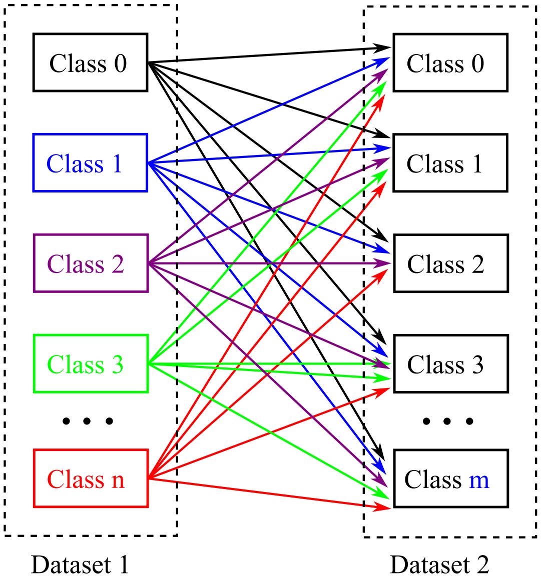


Fig. 6. Combining input data for parallel training and testing networks. Each $target_i$ or $(Class_i)$ in Dataset 1 combines all classes $(Class_0, Class_1, Class_2, \ldots, Class_m)$ in Dataset 2 to form input streams. The result is a balanced dataset for training and testing.

Figure 6 describes the logical procedure of generating training and testing input streams from two datasets. Each $target_i$ (or $Class_i$) in Dataset 1 combines with all classes $(Class_0, Class_1, Class_2, \ldots, Class_m)$ in Dataset 2 to form input data. The final input streams contain balanced input data of all classes from both datasets for training and testing networks in parallel. However, this technique has a drawback because a cluster is trained multiple times with a similar target. Finding a better approach for combining data will be the extension of our current study. Section IV - E in *Supplementary Materials* describes how to generate training and testing data for the parallel mode.

E. Simulation Environment

We built a software framework with Python and Ray. Ray is a comprehensive computing framework for distributed systems in machine learning and artificial intelligence applications [23]. Ray provides a friendly ecosystem that integrates easily with many state-of-the-art optimization algorithms, particularly Optuna, which is used for hyperparameter optimizations in data mining [24].

V. Simulation Results

A. Crossbar, SWPL, and SWPL Cluster Networks

In the first experiment, we simulated three memcapacitive reservoir networks (crossbar, SWPL, and SWPL cluster networks) for two multiple tasks: multiple tasks of MNIST and Spoken Digits, and multiple tasks of Spoken Digits and CIFAR-10. Due to the stochastic nature of reservoir networks, we ran simulations on five samples of each reservoir type to obtain average results. From our previous work, we learned that the dynamic response of networks depended on the nonlinear characteristics of memcapacitive devices [4]. Furthermore, according to Jaeger, the network performance (or accuracy) relies on how well the network translates input data into its internal and high-dimensional states [25]. Thus, some networks need more devices while others use fewer to achieve similar performance accuracy. With the Optuna algorithm, we derived the possible sets of network hyperparameters matching the nonlinear behaviors of devices for optimal classification accuracy. The optimal hyperparameters are in Section V of the *Supplementary Materials*. All classification accuracy is from the results of the testing dataset, and we independently simulated networks with multiple tasks.

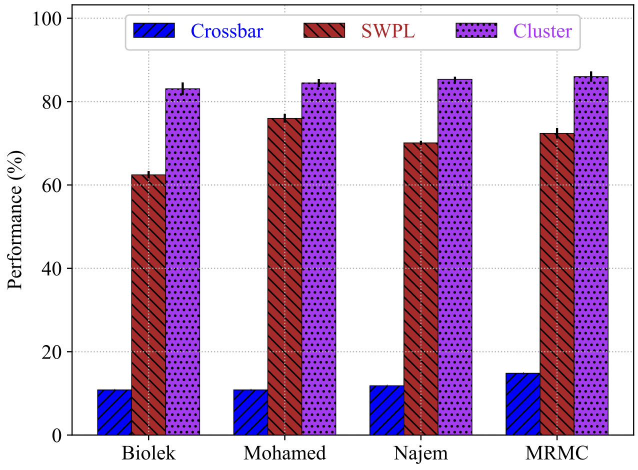


Fig. 7. Accuracy of the crossbar, SWPL, and SWPL cluster networks on MNIST task. All reservoir networks of a particular device type had similar nodes. The optimizing hyperparameters were from the Optuna algorithm. Due to the stochastic nature of networks, five samples of each reservoir type were simulated with the same hyperparameters to obtain average results.

Figures 7, 8, and 9 show the simulation results of three multitasking reservoir networks. We observe that the SWPL cluster networks in parallel mode accomplished higher accuracy than the crossbar and SWPL networks. As expected, the causal effect of input signals from different tasks within the crossbar and SWPL networks reduced the dynamic responses and, therefore, affected the performance accuracy of the systems. On the other hand, this effect was less severe with the cluster structure. The SWPL cluster networks could maintain distinctive internal states for different inputs and preserve their performance accuracy. The separation measurements of network states in Section VII of the *Supplementary Materials* confirmed our observations on the causal effect of input signals. On average, the SWPL cluster networks outperformed the crossbar and SWPL networks by factors of $4.1\times, 5.2\times$, and $1.7\times$ on MNIST, Spoken Digit, and CIFAR-10 tasks.

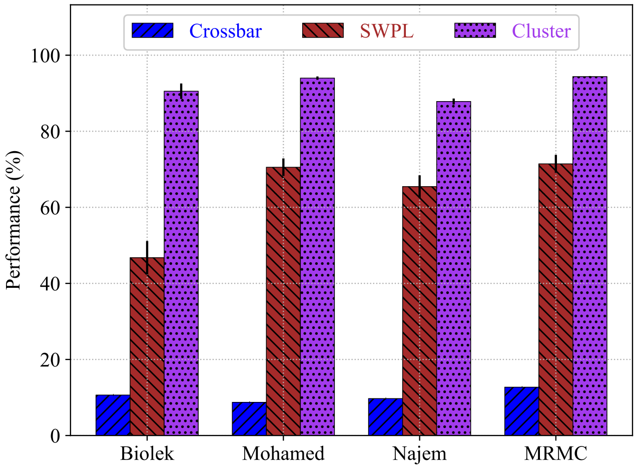


Fig. 8. Accuracy of the crossbar, SWPL, and SWPL cluster networks on Spoken Digit task. All reservoir networks of a particular device type had similar nodes. The optimizing hyperparameters were from the Optuna algorithm. Due to the stochastic nature of networks, five samples of each reservoir type were simulated with the same hyperparameters to obtain average results.

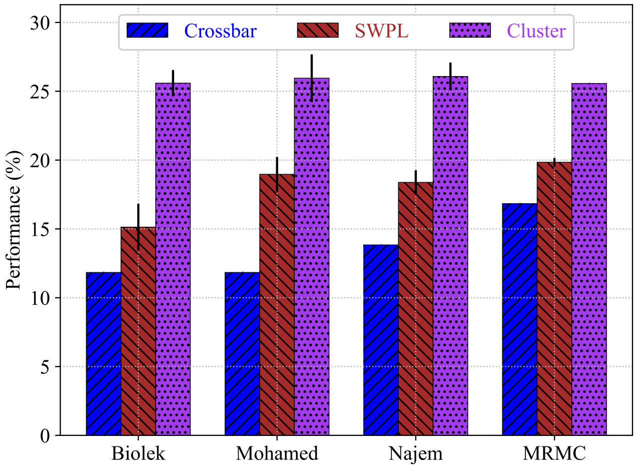


Fig. 9. Accuracy of the crossbar, SWPL, and SWPL cluster networks on CIFAR-10 task. All reservoir networks of a particular device type had similar nodes. The optimizing hyperparameters were from the Optuna algorithm. Due to the stochastic nature of networks, five samples of each reservoir type were simulated with the same hyperparameters to obtain average results.

B. Single-task SWPL, and multitasking SWPL Cluster Networks

In the second experiment, we investigated how SWPL cluster networks of different memcapacitive devices perform on three tasks (MNIST, Spoken Digits, and CIFAR-10) in parallel and sequential modes, compared to single-task SWPL networks. Since there was a stochastic factor in the formation of networks, we simulated three instances of reservoirs with similar hyperparameters to obtain average results. The accuracy only included the results of testing data.

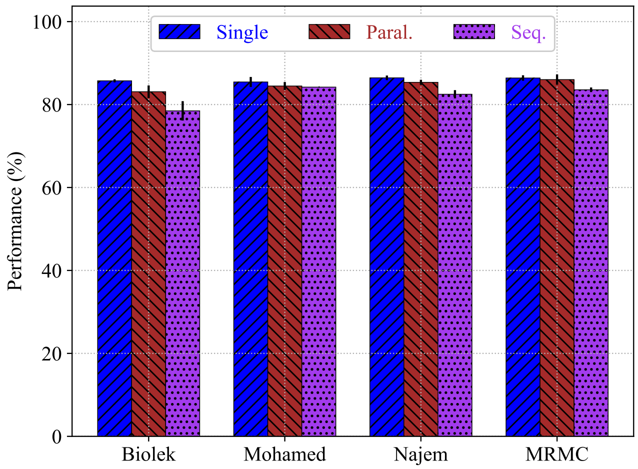


Fig. 10. Accuracy of single-task SWPL and SWPL cluster networks on MNIST. In general, both single SWPL and SWPL cluster networks with two training and testing modes accomplished similar classification accuracy.

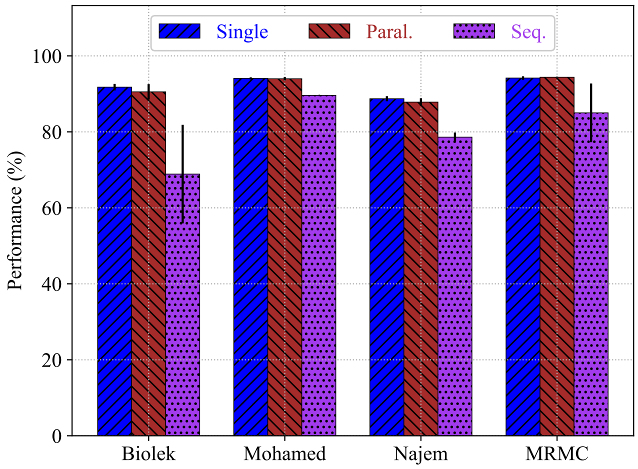


Fig. 11. Accuracy of single-task SWPL and SWPL cluster networks on Spoken Digits.

Fig. 10 shows the simulation results of networks on the MNIST task. In general, the SWPL cluster networks accomplished comparable accuracy in parallel ($Paral.$) and sequential ($Seq.$) modes compared to single-task SWPL networks. However, the *Biolek* SWPL cluster networks did not achieve comparative accuracy. From our previous study [4], we noticed that the internal state of the *Biolek* model was significantly sensitive to applied voltages (changes in mV). Finding the most optimal setting for the *Biolek* networks requires an extensive simulation time during optimization. In our current simulations, the hyperparameters for the *Biolek* networks were not optimal due to limited simulation times. Therefore, the performance accuracy of *Biolek* SWPL cluster networks was lower than others.

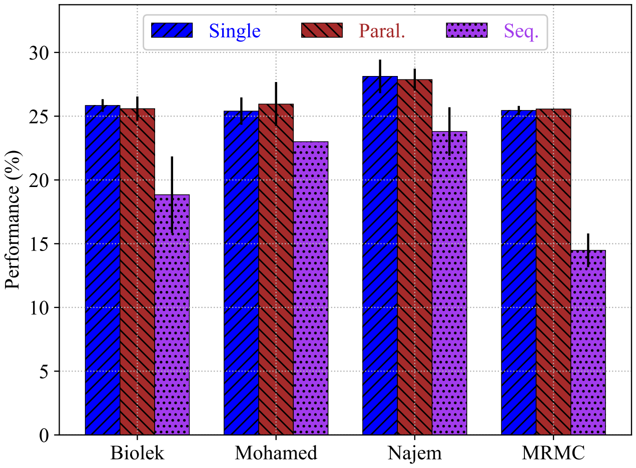
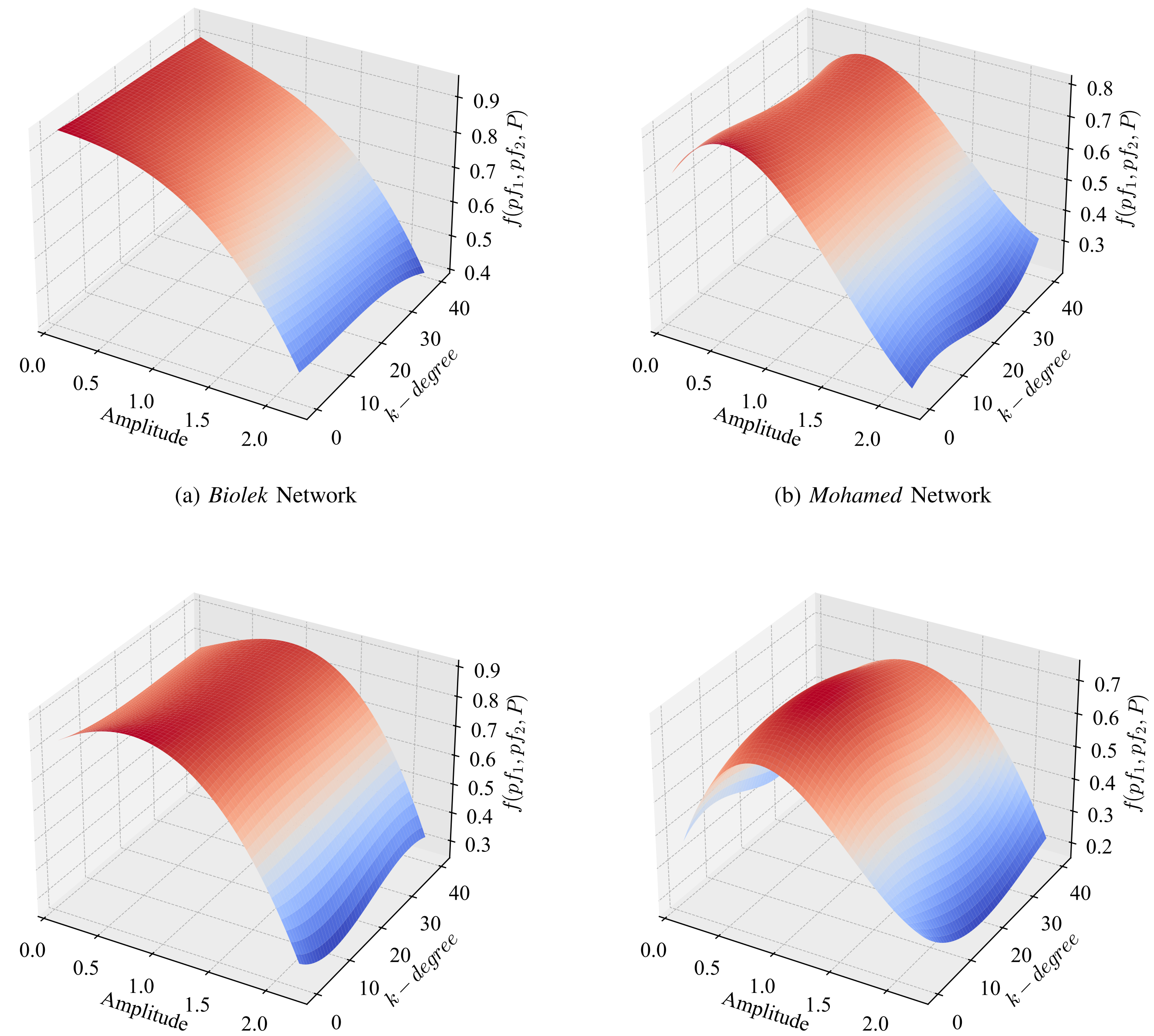


Fig. 12. Accuracy of single-task SWPL and SWPL cluster networks on CIFAR-10.

For a complex task such as clarifying spoken digits in Fig. 11, the SWPL cluster networks with the sequential mode consistently failed in their accuracy compared to single-task SWPL reservoirs on similar jobs. In the parallel mode, however, the SWPL cluster reservoirs could maintain comparable accuracy. We believe that the inter-cluster links created causal effects between clusters within reservoirs. A disturbance within one cluster triggered a dynamic change in other clusters through interconnections. In the sequential mode, during the training phase of reservoirs, the input data to readout nodes from clusters did not account for this causal effect, and reservoirs achieved low accuracy during the testing phase. However, in the parallel mode, the causal impact was part of the input data, so training readout nodes could account for this effect. As a result, the reservoirs could retain their classification accuracy. In Section VIII of the *Supplementary Materials*, we performed additional experiments to measure the Euclidean distance of network states. The results affirmed what we saw in the performance accuracy of the SWPL cluster networks for parallel and sequential modes.

We observe a similar tendency of SWPL cluster networks on high-dimensional tasks such as CIFAR-10, as shown in Fig. 12. In the parallel mode, training readout nodes could accommodate the correlation effects due to inter-cluster links. Therefore, the networks could maintain their equivalent accuracy compared to single-task networks. In the sequential mode, on the other hand, the training phase did not consider the correlated effect, resulting in low accuracy.

Table II contains the percentage accuracies of SWPL cluster networks in a parallel mode for two multiple tasks: Isolated Spoken Digits and MNIST, and Isolated Spoken Digits and



(a) *Biolek* Network

(b) *Mohamed* Network

(c) *Najem* Network

(d) *MRMC* Network

Fig. 13. Trade-off function $f\left(pf_1, pf_2, P\right)$ of cluster networks with different values of amplitudes (*Ampl*) and inter-cluster connections (*k-degree*) on classifying spoken digits and MNIST images. The interpolating values were from the two functions $bisplrep()$ and $bisplev()$. Compared to the inter-cluster connections, the amplitudes contributed significantly to the maximum values of the trade-off functions. The maximum values of the trade-off function $f()$ occurred at different values of $Ampl$: [a] for *Biolek* network $Ampl \in [0.1, 0.15]$, [b] for *Mohamed* network $Ampl \in [0.45, 0.6]$, [c] for *Najem* networks $Ampl \in [0.7, 0.8]$, and [d] for *MRMC* network $Ampl \in [0.8, 0.86]$.

CIFAR-10. In general, the percentage accuracy of SWPL cluster networks was not as high as the state-of-the-art systems due to limited simulation times. However, our results (Cr.) were similar to the previous results (Pr.) in our published work for single-tasks SWPL memcapacitive networks [4].

C. Trade-off for Overall Performance

The simulation results show that both the signal amplitudes of input streams ($Amplitude$) and the inter-cluster connections ($k - degree$) contribute to the overall classification accuracy of SWPL cluster networks. In the third experiment, we want to analyze how these factors affect the overall performance of the cluster networks in various settings of the input signal amplitudes and inter-cluster links on the two tasks: Spoken Digits and MNIST. We simulated five instances of each cluster network with similar hyperparameters to acquire average results.

We see that the classification accuracy of SWPL cluster networks improved with high settings of signal amplitude and $k - degree$. However, with the high settings, SWPL cluster

TABLE II
PERCENTAGE ACCURACY

Model	Percentage Accuracy †					
	MNIST		Digits		CIFAR-10 ‡	
	Cr.	Pr.	**Cr.**	Pr.	**Cr.**	Pr.
Biolek [17]	**83.1**	85.7	**90.5**	91.7	**25.6**	25.8
Mohamed [18]	**84.4**	85.4	**94.0**	94.1	**26.0**	25.4
Najem [19]	**85.3**	86.4	**87.8**	88.7	**27.9**	28.1
MRMC [1]	**86.0**	86.4	**94.4**	94.2	**25.6**	25.5

† "**Cr.**" represents the results in our current work (bold values). "Pr." refers to the results in our previous work [4]. The performance accuracy of SWPL cluster networks was lower than the state-of-the-art systems due to the subset data in training and testing to avoid long simulation times in optimizing processes.
‡ To reduce simulation time for CIFAR-10, we decreased image dimensions with grayscale images and only used a subset of data for training and testing. As a result, the accuracy was even lower than the state-of-the-art systems.

networks also consumed more power than those with low settings. We defined function $f()$ as a trade-off function of SWPL cluster networks: $f(pf_1, pf_2, P) = pf_1 + pf_2 - P$. In this trade-off function, pf_1 and pf_2 were the classification accuracy of SWPL cluster networks on two tasks(Spoken Digits and MNIST), and P was the overall power consumption of SWPL networks. The missing points (or the interpolated values) were from the bivariate B-spline interpolation functions $bisplrep()$ and $bisplev()$, the two common functions for finding spreading points across a surface [26]. With the positive effect of pf_1 and pf_2 and the negative influence of P, the maximum values of $f(pf_1, pf_2, P)$ were where the networks achieved their optimal accuracy of high classification accuracy (pf_1 and pf_2) and low power consumption (P). The simulation results of the trade-off function are in Fig 13.

As shown in Fig. 13, the trade-off functions of different device cluster networks reached maximum values at the different values of amplitudes (*Ampl*) and inter-cluster connections (*k-degree*). Each memcapacitive device responded differently to input signals due to its physical property. We expect that the optimal values of signal amplitudes would be different for memcapacitive cluster networks. Furthermore, we observe that the change in signal amplitudes had a significant impact on the trade-off functions compared to inter-cluster links. The maximum values of the trade-off functions happened at different values of *Ampl*: for *Biolek* network $Ampl \in [0.1, 0.15]$ (Fig. 13a), for *Mohamed* network $Ampl \in [0.45, 0.6]$ (Fig. 13b), for *Najem* networks $Ampl \in [0.7, 0.8]$ (Fig. 13c), and for *MRMC* network $Ampl \in [0.8, 0.86]$ (Fig. 13d). The results suggest that selecting appropriate values of signal amplitudes matching the physical property of memcapacitive devices will further improve the overall performance of cluster networks in maintaining high classification accuracy and low power consumption.

Optimizing hyperparameters, including network nodes, was based on the Optuna algorithm to find possible network settings matching the characteristics of memcapacitive devices. Since networks with a higher number of network nodes (or devices) improved the classification accuracy but consumed more power, we define another trade-off function f to compare the overall performance of SWPL cluster networks between different device types: $fp = Perf_1 + Perf_2 - Power$. In this function, $Perf_1$ and $Perf_2$ are normalized accuracy performances of $task_1$ and $task_2$, and $Power$ is the overall and normalized power consumption of networks during training and testing processes. As shown in Fig. 14, both the *BiolekC4* and *Najem* SWPL cluster networks offered higher trade-off values than other cluster networks. As a result, selecting a suitable device for SWPL cluster networks, such as the *BiolekC4* and *Najem* devices, will further improve the potential of multitasking networks.

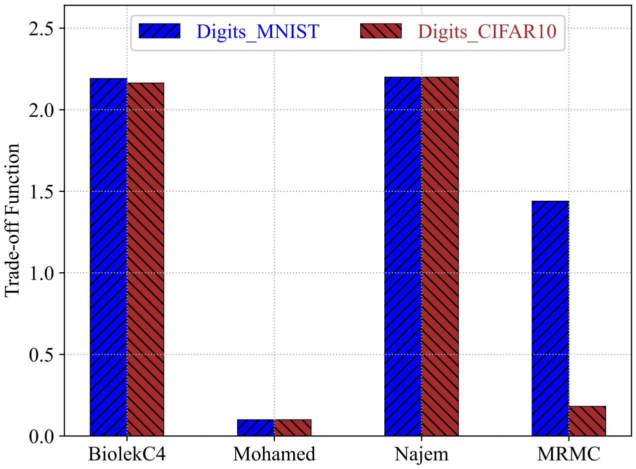


Fig. 14. Power-accuracy trade-off between memcapacitive models for SWPL cluster networks in parallel mode. The SWPL cluster networks were trained and tested in a parallel mode for two multiple tasks: Isolated Spoken Digits and MNIST, and Isolated Spoken Digits and CIFAR-10.

VI. CONCLUSION

We have shown for the first time that memcapacitive rich-club cluster networks can solve multiple tasks simultaneously with high accuracy and low power. On average, the cluster networks surpassed crossbar and SWPL networks by factors of $4.1\times, 5.2\times$, and $1.7\times$ on the three benchmark tasks we used (Isolated Spoken Digits, MNIST, and CIFAR-10). Even though the performance accuracy of SWPL cluster networks in parallel mode was lower than the state-of-the-art systems for the three benchmarks, the accuracy results were similar to our previous results in the published work for single-task memcapacitive networks [4]. Further experiments might increase the accuracy using the whole datasets for training cluster networks instead of subsets to avoid long simulation times in optimizing processes. The trade-off results also revealed that selecting suitable devices for tasks could further improve the potential benefit of SWPL cluster networks. Our simulation data can be used for future hardware implementations. Our results illustrate the promise of memcapacitive brain-inspired cluster networks and their capability to solve multiple tasks simultaneously. Such novel computing architectures have the potential to make edge applications more efficient and would allow systems that cannot reconfigure to solve concurrent tasks.

REFERENCES

[1] H.-T. Zhang, T. J. Park, A. N. Islam, D. Tran S.J., S. Manna, Q. Wang, S. Mondal, H. Yu, S. Banik, S. Cheng *et al.*, “Reconfigurable perovskite nickelate electronics for artificial intelligence,” *Science*, vol. 375, no. 6580, pp. 533–539, 2022.

[2] K.-U. Demasius, A. Kirschen, and S. Parkin, “Energy-efficient memcapacitor devices for neuromorphic computing,” *Nature Electronics*, vol. 4, no. 10, pp. 748–756, 2021.

[3] D. Tran S.J. and C. Teuscher, “Memcapacitive reservoir computing,” in *2017 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*. Newport, RI, USA: IEEE, 2017, pp. 115–116.

[4] ——, “Computational Capacity of Complex Memcapacitive Networks,” *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 17, no. 2, pp. 1–25, 2021.

[5] M. Rigotti, D. D. Ben Dayan Rubin, X.-J. Wang, and S. Fusi, “Internal representation of task rules by recurrent dynamics: the importance of the diversity of neural responses,” *Frontiers in Computational Neuroscience*, vol. 4, p. 24, 2010.

[6] G. Milano, G. Pedretti, K. Montano, S. Ricci, S. Hashemkhani, L. Boarino, D. Ielmini, and C. Ricciardi, “In materia reservoir computing with a fully memristive architecture based on self-organizing nanowire networks,” *Nature Materials*, vol. 21, no. 2, pp. 195–202, 2022.

[7] S.-O. Park, H. Jeong, J. Park, J. Bae, and S. Choi, “Experimental demonstration of highly reliable dynamic memristor for artificial neuron and neuromorphic computing,” *Nature Communications*, vol. 13, no. 1, pp. 1–13, 2022.

[8] M. P. Van Den Heuvel and O. Sporns, “Rich-club organization of the human connectome,” *Journal of Neuroscience*, vol. 31, no. 44, pp. 15 775–15 786, 2011.

[9] A. K. Barbey, “Network Neuroscience Theory of Human Intelligence,” *Trends in Cognitive Sciences*, vol. 22, no. 1, pp. 8–20, 2018.

[10] M. Naqi, M. S. Kang, T. Kim, S. Baek, A. Bala, C. Moon, J. Park, S. Kim *et al.*, “Multilevel artificial electronic synaptic device of direct grown robust MoS_2 based memristor array for in-memory deep neural network,” *npj 2D Materials and Applications*, vol. 6, no. 1, pp. 1–9, 2022.

[11] X. Ma, G. Jiang, S. Fu, J. Fang, Y. Wu, M. Liu, G. Xu, and T. Wang, “Enhanced network efficiency of Functional Brain networks in Primary insomnia Patients,” *Frontiers in Psychiatry*, vol. 9, p. 46, 2018.

[12] U. Chockanathan, A. Z. Abidin, A. M. DSouza, G. Schifitto, and A. Wismüller, “Resilient modular small-world directed brain networks in healthy subjects with large-scale Granger causality analysis of resting-state functional MRI,” in *Medical Imaging 2018: Biomedical Applications in Molecular, Structural, and Functional Imaging*, vol. 10578, International Society for Optics and Photonics. Houston, TX: SPIE, 2018, p. 105780B.

[13] K.-i. Kitayama, “Guiding principle of reservoir computing based on “small-world” network,” *Scientific reports*, vol. 12, no. 1, pp. 1–10, 2022.

[14] Y. Kawai, T. Tokuno, J. Park, and M. Asada, “Echo in a small-world reservoir: Time-series prediction using an economical recurrent neural network,” in *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE, 2017, pp. 126–131.

[15] G. Sha, P. H. Yu, D. Qi, X. HanLiang, C. ZhenKai, and H. TianLu, “An improved small-world topology for optimizing the performance of echo state network,” in *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. IEEE, 2020, pp. 1413–1419.

[16] F. Parente and A. Colosimo, “Functional connections between and within brain subnetworks under resting-state,” *Scientific Reports*, vol. 10, no. 1, pp. 1–13, 2020.

[17] D. Biolek, M. Di Ventra, and Y. V. Pershin, “Reliable SPICE simulations of memristors, memcapacitors and meminductors,” *Radioengineering*, vol. 22, no. 4, pp. 945–968, July 2013.

[18] M. G. A. Mohamed, H. Kim, and T. Cho, “Modeling of Memristive and Memcapacitive Behaviors in Metal-Oxide Junctions,” *The Scientific World Journal*, vol. 2015, p. 910126, 2014.

[19] J. S. Najem, M. S. Hasan, R. S. Williams, R. J. Weiss, G. S. Rose, G. J. Taylor, S. A. Sarles, and C. P. Collier, “Dynamical nonlinear memory capacitance in biomimetic membranes,” *Nature Communications*, vol. 10, no. 1, pp. 1–11, 2019.

[20] Z. Jackson, “Free Spoken Digit Dataset (FSDD),” https://github.com/Jakobovski/free-spoken-digit-dataset, 2016, accessed: 2018-04-09.

[21] N. Soures, L. Hays, and D. Kudithipudi, “Robustness of a memristor-based liquid state machine,” in *Neural Networks (IJCNN), 2017 International Joint Conference on*. Anchorage, AK: IEEE, 2017, pp. 2414–2420.

[22] L. Chaudhary, F. Hussain, and U. Gillani, “Utilization of DCT Coefficients for the Classification of Standard Datasets in Cloud/Edge Computing Environment,” in *2022 2nd International Conference on Digital Futures and Transformative Technologies (ICoDT2)*. IEEE, 2022, pp. 1–7.

[23] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan *et al.*, “Ray: A distributed framework for emerging {AI} applications,” in *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, 2018, pp. 561–577.

[24] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[25] H. Jaeger, “The ”echo state” approach to analyzing and training recurrent neural networks-with an erratum note,” *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, p. 13, 2001.

[26] M. Gentile, F. Courbin, and G. Meylan, “Interpolating point spread function anisotropy,” *Astronomy & Astrophysics*, vol. 549, p. A1, 2013.