https://doi.org/10.1038/s44335-024-00017-x

Leveraging stochasticity in memristive synapses for efficient and reliable neuromorphic systems

Hritom Das[1] ✉, Karan P. Patel[2], Rocco D. Febbo[2], Catherine D. Schuman[2] & Garrett S. Rose[2]

Neuromorphic computing is inspired by the human brain's architecture to develop power-efficient and optimized neural networks. Different characteristics of synapses and neurons are emulated in the neuromorphic hardware or software model to mimic the behavior of the synaptic brain. Noise in the neuromorphic system is an important constraint to evaluate its overall performance. A reliable *READ* operation is essential to secure an acceptable performance from a neuromorphic system. In addition, a memristive synapse is very prone to stochastic behavior. Moreover, a dynamically reconfigurable *READ* operation is proposed for our 3T1R synapse to explore the effect of stochasticity on neuromorphic applications. Our proposed method allows for relevant applications in neuromorphic computing to utilize the stochastic behavior of the memristive 3T1R synapse. Performance evaluations show ~8.9x energy optimization with our proposed device. Optimization algorithm (EONS) is utilized for training and a hardware framework (RAVENS) is used for hardware simulation during training.

Memristive synapses are very popular building blocks for neuromorphic computing. Synapses are used to construct circuits and systems such as neuroprocessors, dot product engines[1,2], spike-timing-dependent plasticity (STDP)[3–7], homeostatic plasticity[8], reservoir computer (RC)[9,10] and so on. All of these applications require a reliable synapse to confirm performance of the system. However, synapses are very sensitive to its *FORMING*, *RESET*, *SET*, and *READ* variations[11]. However, *READ* variation can be utilized as a feature of these kinds of devices. The distribution of the *READ* current can be used for Stochastic computing (SC), which is a popular approach to unite algorithms with stochastic features of hardware.

SC is a branch of computing that dates back to the late 1960s[12]. It explores the potential benefits of performing computations with probabilities rather than explicit values[13–16]. One of the main reasons for the popularity of this field is due to the inherent unpredictability of the devices used for traditional computers. These otherwise undesired properties are mitigated in traditional computers through robust digital designs. SC instead proposes to utilize these random properties by exploring their potential for application in mathematical models that require random distributions[17,18]. Previously, SC relied on the randomness observed in signal noise as well as the randomness due to the fabrication of the devices[19–21]. Meaning each device had a unique, fixed behavior. Recently with the rise of new devices and methods of operation, this behavior can be tuned at runtime as opposed to being fixed over the lifetime of the device. This allows for much more flexibility in terms of the types of computations that can be performed and the amount of hardware required.

One such example of a device is that of a metal-insulator-metal (MIM) memristor. These devices operate by supplying a voltage potential across the insulator using the metal contacts. If this voltage is high enough, the insulator becomes a conductor through the formation of a filament connecting the metals. Interestingly for nano-scale fabrication of specific insulator materials, if this potential is reversed the filament can be broken. As you might imagine, the formation and destruction of this filament path is very stochastic. However, if the current during these events is limited, the conductivity of the filament path can be varied with some predictability. The final conductance of the filament path can be modeled quite reliably through a normal distribution whose mean value is directly proportional to the current during the formation of the filament.

Another potential application for SC is that of machine learning (ML) accelerator circuits. In many ML algorithms, randomness has been shown to improve performance. For instance, during the initialization of weights in a neural network. As well as its use in stochastic gradient descent. Furthermore, SC has been utilized in Bayesian neural network circuits which utilize MIM devices[22]. Another domain of ML that utilizes stochasticity is neuromorphic computing. Neuromorphic computation explores the potential benefits of a more biologically plausible ML approach. For instance, often utilizing the much more biologically plausible spiking neuron models as

[1]Department of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK, USA. [2]Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN, USA. ✉e-mail: hritom.das@okstate.edu

opposed to the McCulloch-Pitts neuron used extensively in deep learning. These methods have been shown to provide large performance benefits in terms of power consumption which is useful in edge applications. Additionally, neuromorphic computing could benefit greatly in terms of increasing the accuracy of the stochastic nature of biological systems such as those seen in neurons in the brain[23].

At the same time, a reliable memory component is necessary to build a stable system for data storage and neuromorphic computing. The *Forming* operation of memristive synapse needs a higher voltage (~3 V) to set the resistance level from higher (HRS) to lower (LRS), which is also challenging. It can be lowered by proper transistor sizing. The voltage-controlled synaptic device is sensitive to pulse width and supply voltage variation[2]. A current-controlled synapse shows higher reliability in *SET* operation than the voltage-controlled synapse. LRS shows lower inherent variation for *SET* and *READ* operations compared to HRS. In addition, 3T1R synapses show lower *READ* power consumption compared to 1T1R configuration.

Moreover, various memristive synapses such as 1T1R[24],1D1R[25], 1S1R[26], and 2T1R[27] are proposed by different research groups. However, most of the proposed solutions are not fully compatible with CMOS technology, and expensive pieces of equipment are required to operate the synapse. On the other hand, our proposed synapse is fully CMOS-compatible and can be operated by digital signals. Due to that, our proposed design is economic and lower power. In addition, our proposed design shows enhanced stability compared to a 1T1R synapse.

In this work, 3T1R memristive synapses are utilized for a runtime adaptation between a fully deterministic device and a stochastic device. Figure 1 shows adaptation and its data distribution with energy benefit. It also illustrates the different application opportunities with stochastic devices.

The key contributions of this work are as follows.

- A dynamically reconfigurable *READ* operation is proposed to take advantage of the stochasticity of *READ* data.
- This synaptic device also can be used as a reliable memory and crossbar array for spiking neural networks.
- A complete *READ* data distribution and their biasing comparability are presented with proper data distribution analysis, which can be utilized for various applications.
- Power vs. performance evaluation of data distribution and applications are illustrated.
- Stochasticity is evaluated on Neuromorphic applications such as classification, control, and reservoir computing.

Our proposed synapse is constructed with three thick oxide transistors and a HfO_2 based memristor device. Figure 2a shows our proposed synaptic circuitry with a 8 × 8 synaptic array. A digital-to-analog converter (DAC) is utilized to program the synapse at a particular resistance level. This memristive array is used to *READ* the weight value of memristor as "Final Read Current". Finally, a winner-take-all (WTA) circuitry is utilized to solve a classification problem. A brief explanation of our proposed synapse is presented here. Our proposed synapse needs a one-time forming operation to create a filament from the top electrode (TE) to the bottom electrode (BE). A brief explanation of the memristive device and a Verilog-A model will be presented in the next subsection, which is used to simulate our synaptic circuit in Cadence Virtuoso. A pair of CMOS *MP*1 and *MN*2 from IBM 65 nm are utilized to complete the forming operation of our proposed synapse, which is current-controlled. Current controlled operations are more stable than a voltage-controlled operation[2]. After a successful forming operation the resistance level of the memristor will be around a few kΩ. Hence, a *RESET* operation will be initiated to break the filament and set the

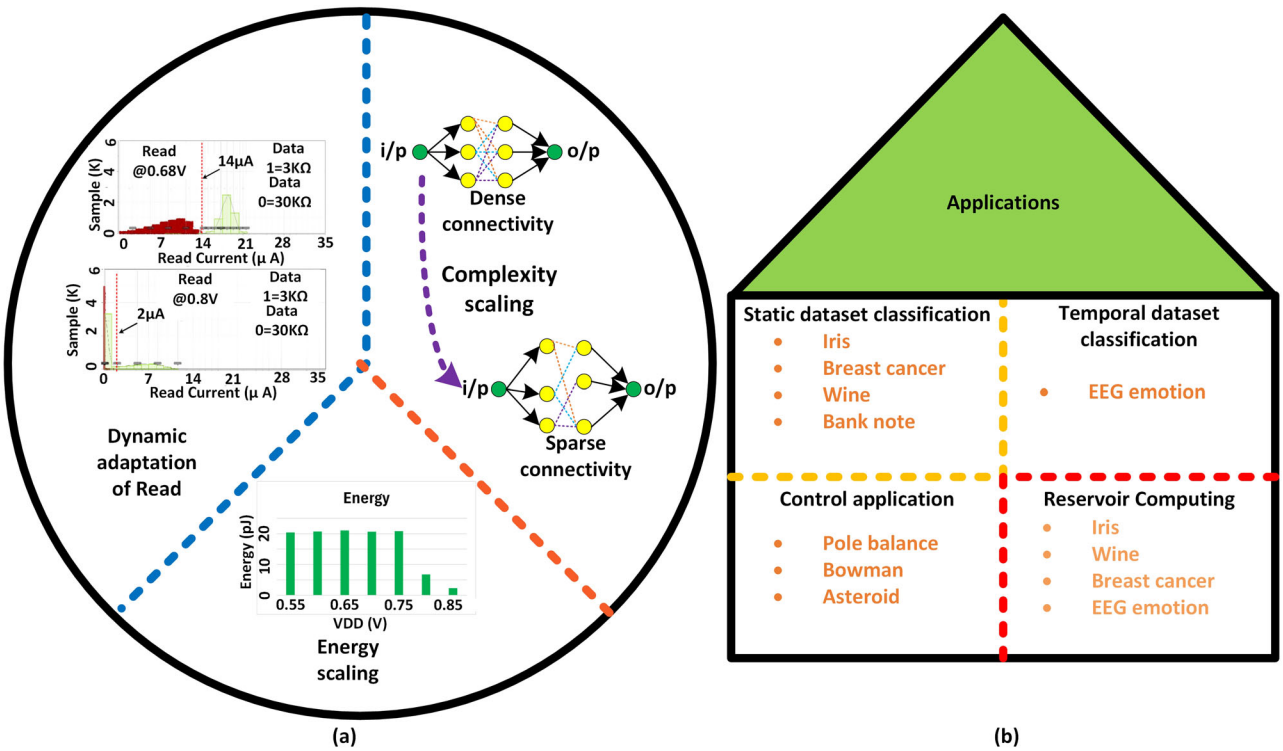

Fig. 1 | The opportunities of neuromorphic computing and possible applications are illustrated. a Shows the dynamic adaptation between conventional and stochastic synapses with proper data distribution. It also shows the possibility of utilizing our proposed synapses with a regular SNN and reservoir computing with optimized operation costs. Finally, with stochastic synapses, the energy of our proposed architecture gets optimized at run-time. **b** Performance of various applications is observed with regular and stochastic synapses. Static and temporal datasets are considered for the performance evaluation[2,29,31,32]. In addition, control applications are considered to observe the perforation at different setups[2]. Reservoir computing is also checked with stochastic synapses, which can be a design option with optimized power[10].

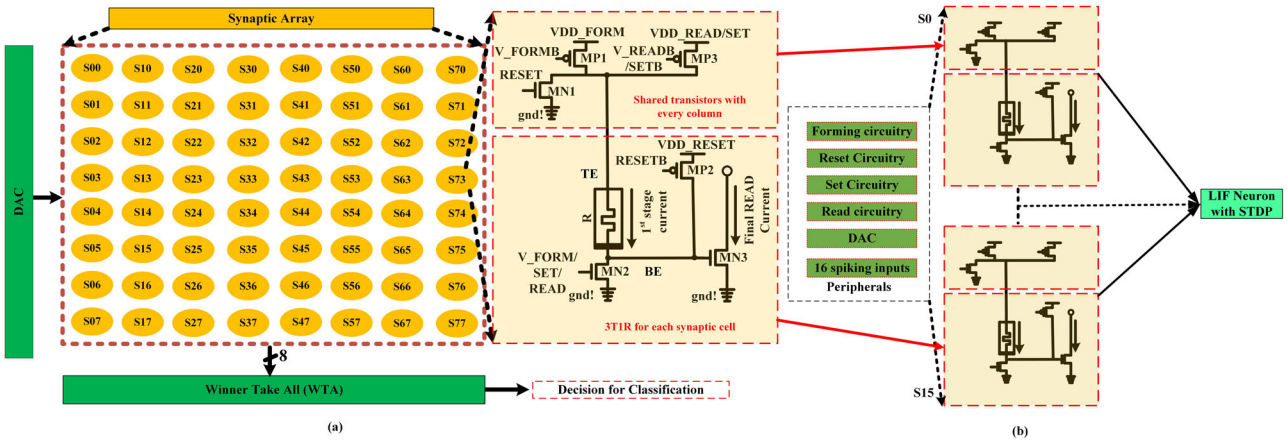

Fig. 2 | An 8 × 8 synaptic array is illustrated with peripheral circuitry. a Digital to Analog Circuit (DAC) is utilized to provide *SET* and *READ* voltage to the synapse. A Winner-Take-All (WTA) circuit is used to determine the maximum current contained columns from all active columns. A current-controlled synaptic circuit is presented at the right of the synaptic array. Three thick oxide transistors and a HfO_2 based memristor are utilized to construct the synapse. **b** Shows the configuration for a neuroprocessor. Peripheral circuits control the gate of the synapse and read currents are sent to the LIF neuron.

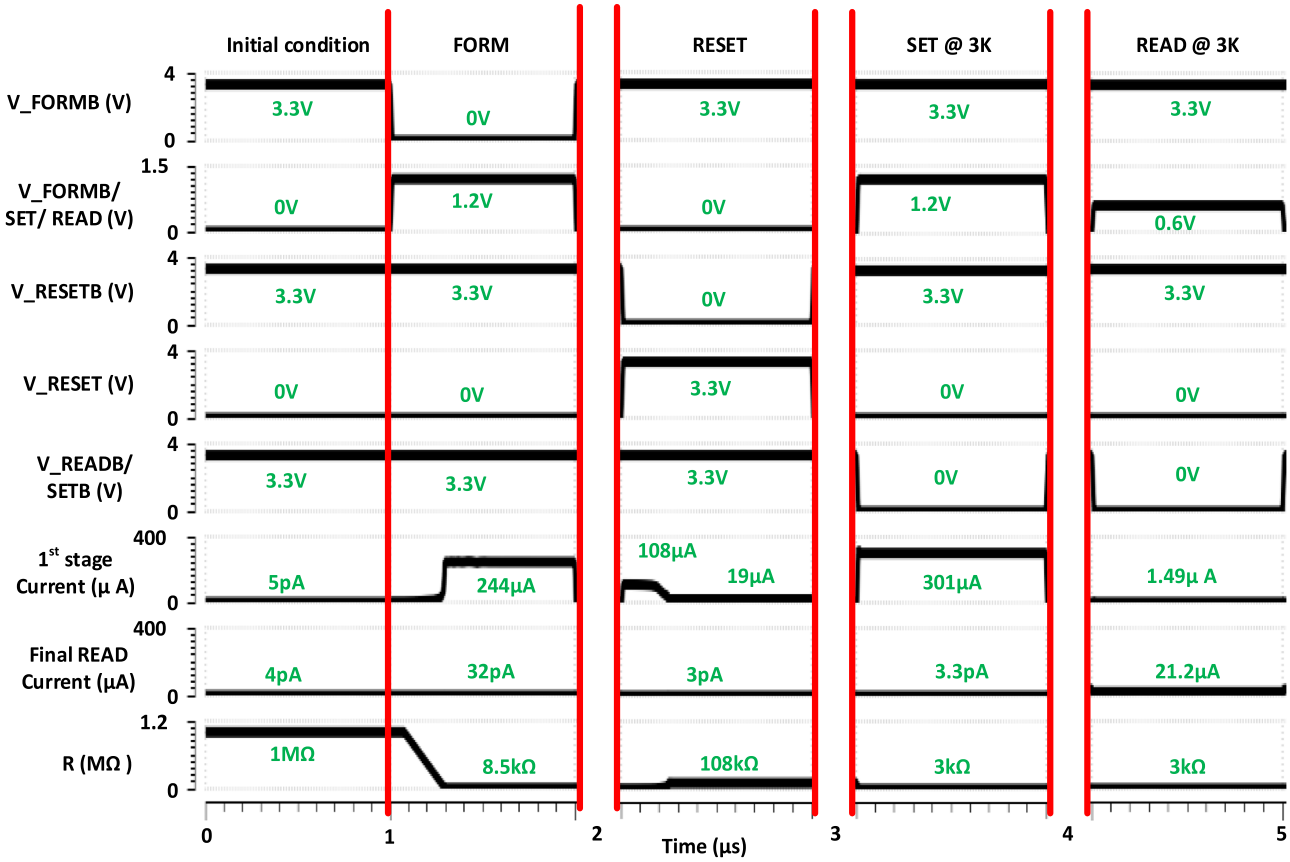

Fig. 3 | Cadence Spectre simulation is performed with 65 nm CMOS 10LPE process from IBM. A Verilog-A model is used to simulate the memristor devices. Initially, the memristor of the synapse is unformed. The initial condition shows the synapse is in a standby condition, where all the signals are disabled. Then a one-time form operation is applied to the synapse. Later a *RESET* operation is applied to do the following *SET* operation. Finally, a *READ* operation is applied to sense the *READ* current of the synapse with a neuron or a WTA circuitry.

resistance level at around a few hundred kΩ. Here, a pair of CMOS transistors *MP*2 and *MN*1 are used to do the *RESET* operation, which is voltage controlled. A *SET or programming* operation is applied to our proposed synapse to set the memristor at a specific resistance level. Here, a low resistance state (LRS) is utilized to mitigate the inherent process variation of the memristor device. In our simulation, we have considered a range of set variations at different resistance levels. This variation information is utilized from measured result[1]. In this work, two resistance values are targeted, which are 3 kΩ and 30 kΩ. *SET* operation requires *MP*3 and *MN*2 to program the memristor value at our targeted level. Finally, a *READ* operation occurs with *MP*3, *MN*2 and *MN*3. The *READ* operation is folded into two parts. The *READ* operation will be explained in detail in the next subsection.

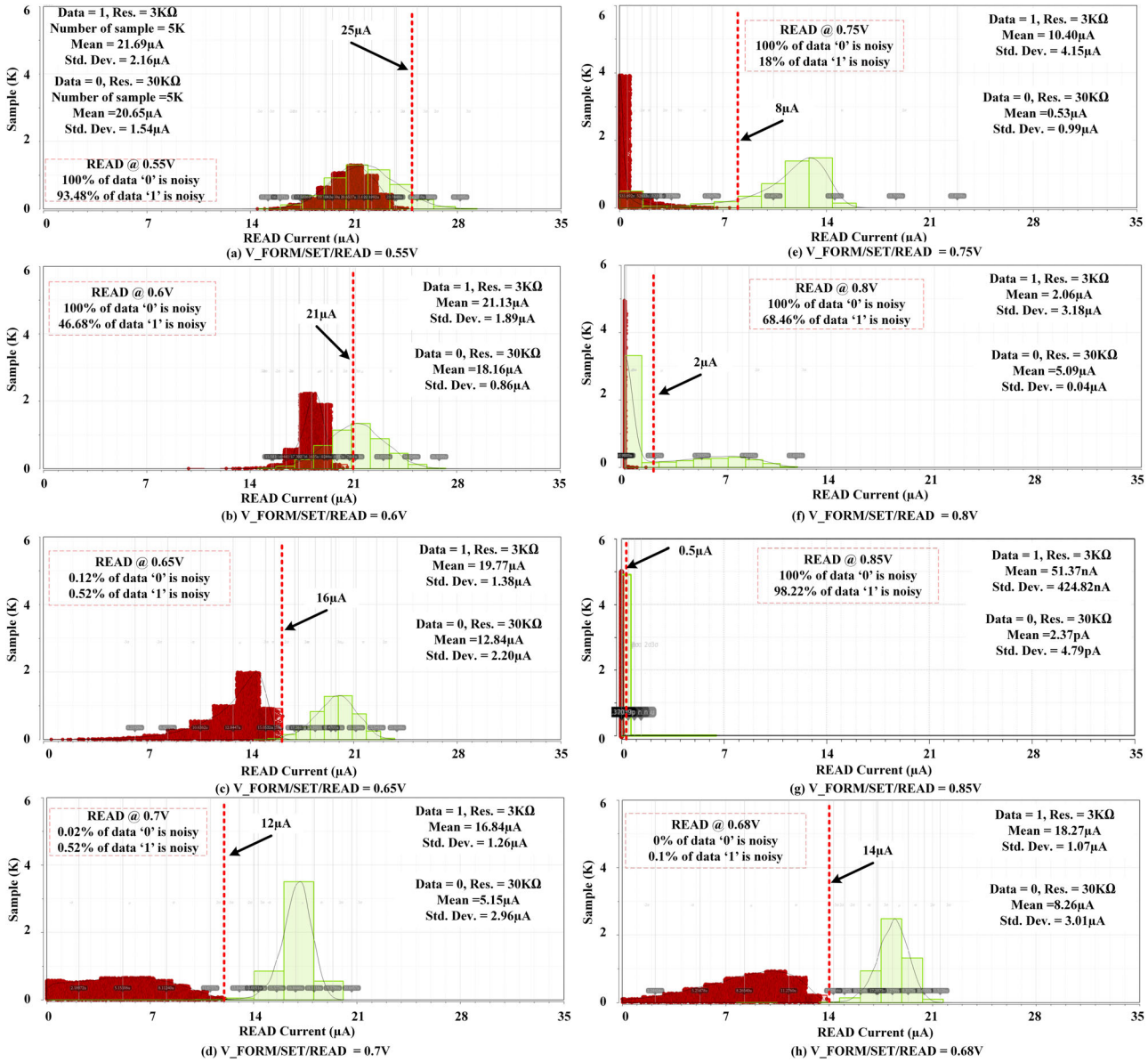

Fig. 4 | Monte Carlo simulation of our proposed synapse is illustrated to observe the final *READ* distribution at different *READ* voltage. The noise distribution of our synapse is illustrated with a 3σ range. **a** Shows the *READ* current distribution at 0.55 V, where the current is distributed from ~14 μA to ~29 μA. About 100% and 93.48% of data '0' and '1' are overlapped with each other. **b** Exhibits about 100% of data '0' and 46.68% of data '1' are contained noise. **c** Shows much lower noisy data for both '1' & 0'. Finally, (**d**) Shows the data noise is minimal at 0.7 V. **e** Exhibits the data noise for '0' is increased to 100% from 18% compared to the last test case (**d**). Hence, (**f**) illustrates higher noisy data at 0.8 V. **g** Shows the noise pattern of data '0' and '1' are 100% and 98.22% respectively. A sensing line is utilized to differentiate the *READ* current between data '0' to '1'. Finally, an optimal operating region is found at 0.68 V. **h** Shows an optimal operating region for *READ* operation, where data distribution illustrates a reliable *READ* operation for a neuromorphic system. Traditionally, highly noisy data is not useful, but data distributions like (**a**, **b**, **e**, **f**, **g**) can be utilized for applications where a biased dataset can enhance the performance. Moreover, this kind of stochasticity might be useful for neuromorphic applications. In addition, (**c**), (**d**), and (**h**) can be used to develop a reliable neuromorphic system.

Figure 2b shows the neuromorphic design with our proposed synapse. There are different peripheral circuits to control the gates of different transistors. The forming peripheral circuit will control the gate of the forming PMOS and gate of the the NMOS for form operation will be controlled with a DAC. In the same way, the other peripherals also operate the gate for each operation. The incoming spikes are going to be stored in the synaptic devices. When the read signal is turned on, each synapse will produce a read current based on the stored weight. All sixteen synapses will produce their corresponding read current and send it to the LIF neuron. Finally, based on the membrane potential and reference voltage the neuron will produce a spike or wait for the next set of read signals.

When a spike hits the NMOS below the memristor the circuit will then drive a small current relative to the memristor's resistance into an integrate and fire neuron. If there is no spike, there is no current and thus no further integrated charge on the neuron. The spike amplitude is small enough such that it does not lead to a large enough current to change the resistance of the memristor in any significant way. Larger currents are used to (re)write the memristor's resistance value, which is useful for programming weights and online learning.

A Verilog-A model is developed based on testing results from our probe station. There are many different parameters to model the behavior of HfO_2 based memristor such as forming voltage, *RESET* voltage, boundary

resistance states for LRS, HRS & *RESET* failure, curve fitting parameters, std. dev. at LRS & HRS and so on[28]. These parameters of the Verilog-A model help the system-level simulation to illustrate the behavior of the memristor as closely as possible with the inherent variation of the memristor. At LRS, the inherent variation of our proposed device is from ~9% to ~31%. This model can also detect unwanted scenarios like *RESET* failure. A detailed explanation of the Verilog-A model and I-V curve are illustrated in our prior work[28]. The HfO_2 based memristive device is constructed with TiN-HfO_2-TiN materials. The oxygen vacancies or defects are utilized to create the filaments from the top to bottom electrode of the device.

Results

In this work, our main focus is to explore the *READ* variability at two resistance levels, which are 3 kΩ and 30 kΩ. Our proposed *READ* operation is current-controlled and a low-power approach. According to Fig. 2, at first *MP*3 and *MN*2 will be turned on to create an approximately constant compliance current as 1st stage current from TE to BE. This current will be converted to a voltage and applied to the gate of *MN*3. Hence, a *Final-READCurrent* will be sensed from the drain of *MN*3. A timing diagram is illustrated in Fig. 3 to exhibit the *READ* operation with other basic operations of our proposed synapse. At first, the proposed synapse will be in its initial stage, where the memristor is unformed and all the controlling signals such as *V_FORMB*, *V_FORM/SET/READ*, and so on will be in inactive mode. The resistance level of our unformed memristor is about 1 MΩ. One-time *FORM* operation is applied to set the resistance level 1 MΩ to ~8 kΩ. *V_FORM* and *V_FORM/SET/READ* are set to low and high to do a *FORM* operation. About 244 μA current will be illustrated through 1st stage current.

After a successful *FORM* operation, the synapse needs *RESET* operation to set the resistance at a few hundred kΩ. *V/_RESET* and *V/_RESETB* are set at high and low respectively to conduct a *RESET* operation. Initially, the 1st stage current is 108 μA during a *RESET* operation. This results in a resistance level of ~108 kΩ. Now, the synapse is ready to do a program at a new resistance value. *V_FORM/SET/READ* and *V_SET/READ* are set at high and low to do a program operation. The programmed resistance value will be 3 kΩ at 1.2 V. Here, the 1st stage current draws 301 μA current to set the synapse at 1.2 V. Finally, a *READ* operation is initiated by setting the *V_FORM/SET/READ* and *V_READ* at 0.6 V and 0 V. The 1st and *Final-READCurrent* will be 1.49 μA and 21.2 μA respectively. In this work, two resistance levels are targeted for programming and reading. Moreover, the synapse will be dynamically tuned to introduce noise inside the dataset. The next section will focus on the noisy or stochastic data distribution of a readout dataset at different reading scenarios.

Performance evaluation based on cadence simulation

Figure 4 shows the read data distribution at different read voltages. Additional information will be discussed in the method section. Table 1 shows the performance evaluation of our proposed dynamic *READ* technique. In this work, the Verilog-A model is utilized to check the simulation results with CMOS variation. The process variation of memristive devices is considered based on the measured data in our probe station. The raw wafer is tested with DC and multi-Z probes with a power supply and signal generator. At the same time, the process variation of CMOS is considered based on the 65 nm CMOS process from IBM 10LPe, which is also integrated with the fabricated circuits. Moreover, the process variation of the thick oxide transistor (dgxfet) and inherent variation of memristive devices are responsible for the current distribution. At first, the synapse is *READ* at 0.55 V to observe the *READ* current distribution and its energy consumption for 1 μs pulse width. Here, both LRS and HRS levels are considered to calculate the average power and energy of our proposed synapse at different *READ* voltages. At 0.55 V the average *READ* power and energy of our memristive synapse are 20.45 μW and 20.45 pJ respectively. Here, both 1st stage and *Final READ Current* are considered to calculate the power and energy. This operating region or *READ* voltages are a good fit for a random number generator with a slight data biasing. Hence, *READ* voltage is scaled

Table 1 | Targeted applications based on power and data distribution requirement

READ voltage (V)	READ Power (μW)		Avg. Power (μW)	Avg. Energy (pJ)	Data '0' (% of noise)	Data '1' (% of noise)	Targeted Application	Comment
	3 kΩ	30 kΩ						
0.55	20.71	20.19	20.45	20.45	~100%	~93.48%	Biased coin flip	Better data distribution with bias and high power
0.6	21.63	19.93	20.78	20.78	~100%	~48.68%	Biased coin flip	Better data distribution with bias and high power
0.65	23.74	18.40	21.07	21.07	~0.12%	~0.52%	Reliable SNN	Almost perfect READ operation
0.7	27.52	13.88	20.7	20.7	~0.02%	~0.52%	Reliable SNN	Almost perfect READ operation
0.75	31.58	10.14	20.86	20.86	~100%	~18%	Biased coin flip	Poor data distribution with bias and low power
0.8	8.71	4.83	6.77	6.77	~100%	~68.46%	Biased coin flip	Poor data distribution with bias and low power
0.85	2.54	2.18	2.36	2.36	~100%	~98.22%	Biased coin flip	Poor data distribution with bias and low power

from 0.55 V to 0.6 V to observe the data distributions and power consumption. Data '1' biasing is drooped from ~93% to ~49% in this operation region with an energy consumption is increased from 20.45 pJ to 20.78 pJ respectively. This type of data distribution is a good fit for applications like biased coin flips and neuromorphic applications where the biased dataset is helpful for testing.

After that, *READ* voltage is varied from 0.60 V to 0.65 V, where the energy consumption is varied from 21.78 pJ to 21.07 pj. In addition, a negligible amount of data has noise in the distribution. Almost 100% of the data can be read perfectly with a sense amplifier, WTA, or a neuron. This operating region can be utilized to design a reliable SNN. Hence, the *READ* voltage is varied from 0.65 V to 0.70 V. Here, the data distribution shows more optimized data distribution than the last test case at 0.65 V with 20.7 pJ as an average energy per *READ* operation. After that, the *READ* voltage scaled up to 0.75 V, where the data '1' is more biased toward data '0'. In the next test case, the *READ* voltage is scaled up to 0.8 V, where ~68% of data '1' is biased towards data '0'. Here, the *READ* energy dropped drastically. When the *READ* voltage is scaled up, the *READ* energy is dominated by the 1st current. In addition, after 0.75 V, both LRS and HRS show lower read power. Due to that, the overall read power optimized a lot. At the same time, the read current distribution shows a higher data-biasing probability, which can be considered as a trade-off with power savings. The power in Table 1 is only considered the synapse's power. At 0.85 V, the *READ* energy shows about 3x lower value than the last test case with a poor data distribution.

Here, we can define three operation regions for *READ*, which are dynamically reconfigurable. The operating regions are: (i) a variable of data '1' is biased toward data '0' with high energy consumption and better data distribution, (ii) reliable *READ* operation where almost 100% data '0' and '1' can be differentiated with a higher energy consumption compared with (i), and (iii) a variable data '1' is biased towards data '0' with lower energy and poor data distribution compared to (i) and (ii).

Performance evaluation based on neuromorphic applications

Figure 5 showcases the classification results for the Iris, Wine, and Breast Cancer datasets[29,30]. These are popular datasets from SciKit Learn used within the machine learning field which provides a strong baseline of understanding regarding the performance of our implementation. Focusing on the testing accuracies of the various datasets, the first notable observation is that default RAVENS behavior is more consistent and results in fewer outliers as opposed to the stochastic 3T1R devices. This is to be expected since stochasticity inherently introduces a probabilistic component that experiences more inconsistencies in the performance of SNNs. Another key observation is that pairing STDP with stochasticity results in better performance than without STDP for certain situations. This is most prominently seen in the Iris dataset where STDP resulted in better performance for all device versions including default RAVENS. Another notable example involves 0.80 V, and 0.85 V versions of the Wine dataset, where the introduction of STDP resulted in a much larger improvement in testing accuracy. It is also worth mentioning that although default RAVENS (without stochasticity) typically yields better testing accuracies, certain combinations of stochastic devices and STDP results in similar, and in some cases even better performance than standard SNN behavior (without STDP and stochasticity). Moreover, if the synaptic system can run at *stochastic_*850 with 0.85 V and achieve the same accuracy as the synapse was run at 0.65 V then the testing process can save up to 8.93x energy compared to *normal_RAVENS*.

Additionally, Fig. 6 showcases the timeseries classification results for the EEG Emotion dataset[31,32]. This dataset captures a subject's electroencephalogram (EEG) signals while watching video clips to classify their emotion. This requires the SNNs to handle timeseries data in order to gather enough information over time to make a classification. Note, that the fitness result for the normal RAVENS implementation has a static value and this is due to it being deterministic and lacking stochasticity. As seen by the results, there is a large amount of variation in accuracy with similar performance across the board. However, the 0.85 V implementation without STDP is far more consistent and performs the best overall, suggesting some level of stochasticity may be helpful in timeseries applications.

Figure 7 depicts the control results for the Cartpole-v1, Bipedalwalker-v3, and Bowman applications from OpenAI. These types of applications control elements of the environment in a time-dependent manner which lends well to the temporal nature of SNNs. Focusing on the fitness scores of these applications, we can see a similar trend to that of the classification results: the default RAVENS behavior typically yields better fitness scores with fewer outliers compared to the stochastic devices. Typically, the more stochastic the device, the worse the performance, but slight stochastic behavior performs similarly or better than default RAVENS as seen by the 0.55 V and 0.85 V devices for Cartpole-v1 and Bowman. On the other hand, STDP didn't improve fitness scores as it did for the classification tasks except for Bipedalwalker-v3. Surprisingly, Bidpedalwalker-v3 had outliers for the stochastic devices that performed significantly better than the more consistent default RAVENS behavior.

Additionally, the stochastic devices have the added benefit of improved power efficiency since not only will the neurons be firing less often, but the circuitry of the stochastic behavior has fewer power demands compared to accurate *READ* distributions. To maintain similar performance under stochastic firing patterns, the network training introduces certain network structures that make them more resilient to these miss-fires. However, this may require larger networks as seen by the boxplots depicting the number of edges and nodes of the classification and control networks. Certain combinations of the stochastic devices and STDP may require more edges and nodes to compensate for the miss-firing of the neurons compared to default RAVENS. Still, certain combinations have relatively similar or even smaller network sizes compared to default RAVENS; however, they may increase depending on the complexity of the application. Although these various characteristics need to be specifically tailored for each application, these devices provide more options that allow engineers to weigh the pros and cons of certain attributes (energy vs. accuracy vs. size) to best fit their design needs for their specific applications.

Reservoir computer evaluation

Additional experiments were also performed on reservoir computers (RC). Reservoir computers are used to map inputs to a higher-dimensional space through their highly recurrent network structures. These, usually randomly initialized RC networks, require the inputs to map to unique outputs while preventing infinite signal propagation in the network by demonstrating "fading memory" characteristics causing the signals to eventually die out. This is performed in SNN-based RCs by utilizing inhibitory synaptic connections to prevent explosive activity while leveraging sparse recurrent connections with synaptic delays to elevate the inputs to a spatially and temporally higher dimensional space. The RCs also consist of a readout layer that is trained, i.e., linear regression, to map the reservoir state to the application output. This composition creates a non-linear system suitable for modeling dynamical systems. Furthermore, the SNN structure isn't trained which makes it computationally efficient since the readout mechanism is the only component requiring training within the entire system.

With that being said, Fig. 8 displays the results for the previously mentioned classification datasets and Fig. 9 shows the results for the timeseries application. From the results, we can see that default RAVENS typically performs better than the stochastic counterparts, with less stochasticity performing better than the more stochastic devices. This makes sense because the SNN structure isn't being modified to account for the stochastic nature. However, like before, paring stochasticity with STDP improves performance in some situations as seen by the Breast Cancer and Iris dataset.

We also explored RCs on control applications as well, but training such systems is more difficult. Reinforcement learning (RL) was utilized to train the readout layer to translate the SNN outputs to optimum actions to take for the control application. Proximal policy optimization (PPO) was the training approach selected due to its ability to incrementally improve

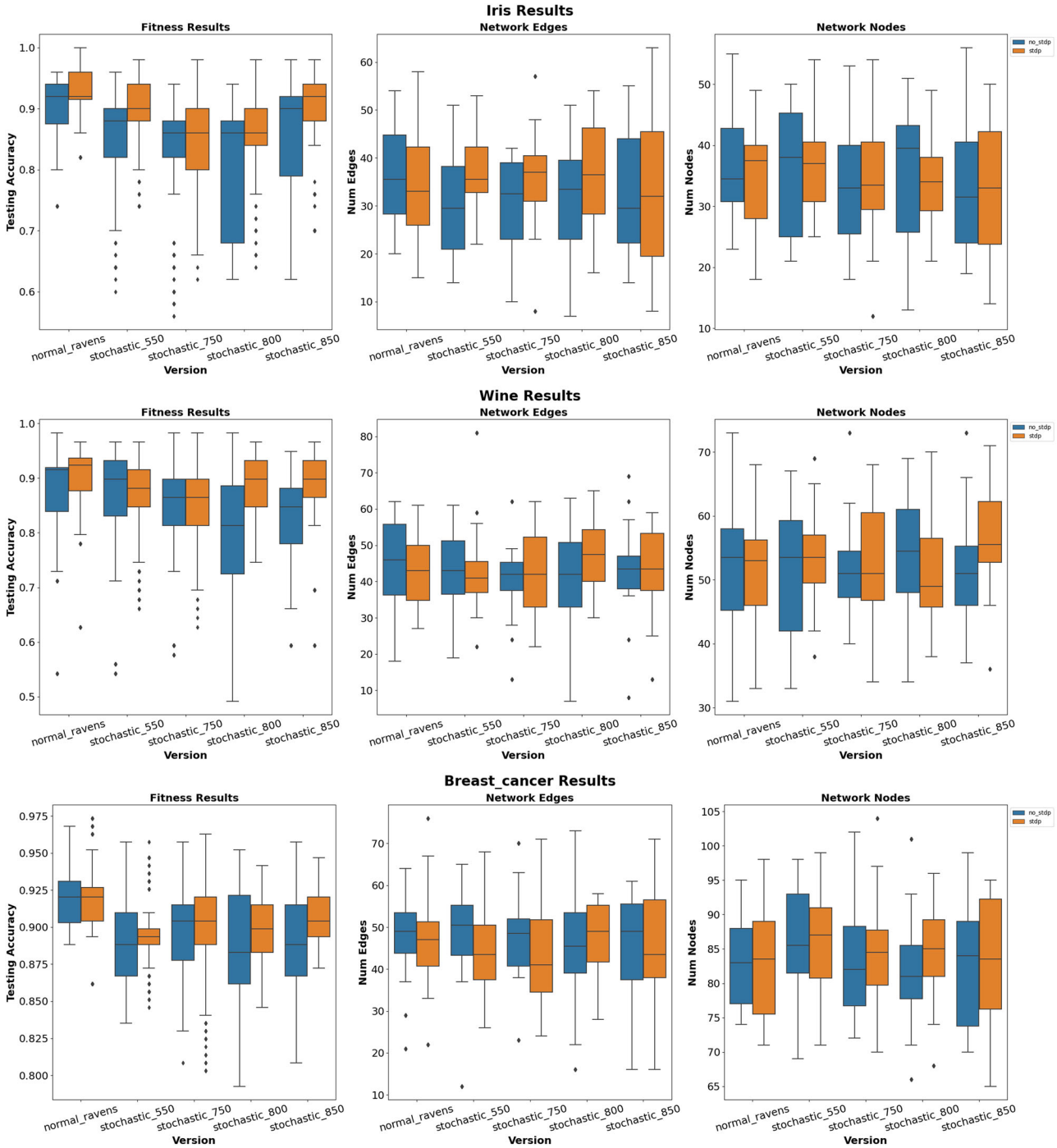

Fig. 5 | Classification results for Iris, Wine, and Breast Cancer datasets. The first, second, and third rows display results for the Iris, Wine, and Breast Cancer datasets respectively including the fitness results, the number of network edges, and the number of network nodes.

performance without destructively modifying parameter weights[33]. This approach exponentially increases training time and the network size of the readout layer. The results are depicted in Fig. 10. Note, the Bowman application was omitted since it wasn't able to learn the environment and gain rewards. From the results, we can see that default RAVENS performed the best with slight stochasticity having comparable performance as seen by the 0.85 V device in Bipedal Walker and 0.85 V and 0.80 V in Cart Pole. STDP slightly helped or drastically decreased performance for both applications. However, STDP with the 0.80 V device performed the best for CartPole. This volatility suggests that the training approach, initialization of the RCs, and STDP results in inconsistent performance since the RL approach needs to account for both the RC structure, stochasticity, and synaptic weight changes, which is a more difficult problem. However, stochastic devices show a great possibility for a significant amount of energy savings with similar performance as normal synaptic devices, where STDP can boost the overall performance a bit with extra energy and design footprint.

Discussion

Neuromorphic devices and systems are gaining popularity for their energy-efficient and bio-inspired features. Digital neuromorphic processors are more reliable in functionality and use CMOS nodes only. However, a digital system is not area and power-efficient like an analog neuromorphic processor. Analog processor uses emerging materials and can store multi-bits as

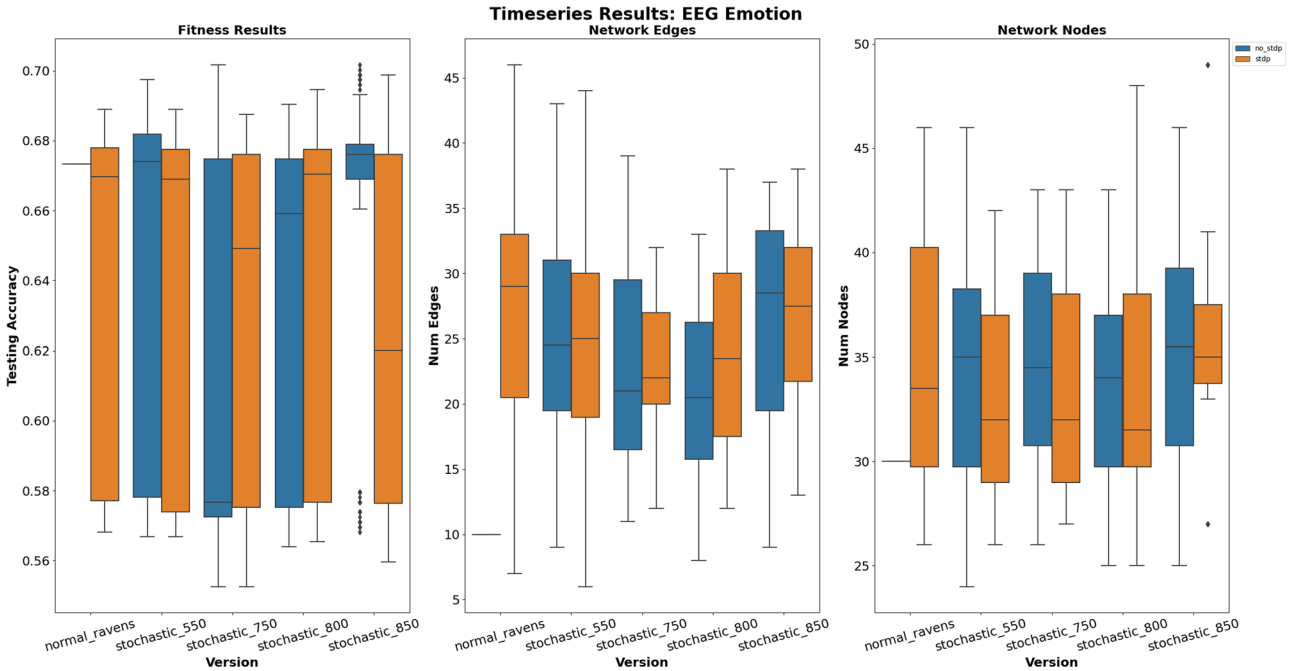

Fig. 6 | Timeseries results for EEG Emotion. The results include the testing accuracy, network edges, and network nodes for the EEG Emotion application.

well. Analog neuroprocessors can do in-memory and near-memory computation, which reduces peripherals and computation power. Emerging material-based synapses show higher inherent process variation. However, the synapse shows reliable operation with a lower number of memory states. In this work, two memory states are targeted for write and read. At the same time, the read power of synapses is dynamically scaled and the read distribution is monitored with Monte Carlo simulations. Read voltage scaling shows read power optimization with stochasticity in the data distribution, which can be utilized in neuromorphic computation without significant accuracy degradation. This kind of data distribution is also suitable for approximate computation, where the system can tolerate a portion of noisy data to improve the power saving with a reasonable accuracy degradation.

TABLE 2 shows a research group presented the work on stochastic memory devices for security and neuromorphic computing[19]. A NbO_2 based-stochastic device shows power savings with different applications like security and neuromorphic computing. Compared to our work, there was no analysis on the dynamic adaptation of stochasticity of the device, STDP, network optimization and so on. However, this work shows a lower power consumption compared to our proposed work. This work uses an isolated device based design, whereas our proposed design shows full integration with CMOS devices. In[20], the authors presented probabilistic neural computing with scholastic devices. Various device mechanisms like MTJ and TD are evaluated for this work. Probabilistic algorithms are developed to evaluate the performance of the device in the neuromorphic domain. However, there was no information on STDP adaptation, reservoir computing, and power savings with the proposed devices. In ref. 34, the author proposed an implementation process of a stochastic neuro-fuzzy system with a memristor array. they utilized 180 nm CMOS process with HfOx material to implement the array. They use the stochasticity of the device to implement the system for robotics and image processing. However, there was no evaluation of STDP and reservoir computing with their proposed system. Another research group claimed work on stochastic devices for computation and neuromorphic application[13]. Here, the author presented a device based on boron material and included the behavior of STDP. However, there was no indication of reservoir computing. They utilized the stochasticity of devices for error-tolerant application and analog behavior for neuromorphic computation. Moreover, this work shows about 5.08x higher power consumption compared to our proposed work.

In ref. 35, authors proposed a HfO_2-based memristive synapse with 65 nm CMOS technology. This device's stochasticity shows efficient brain-inspired learning opportunities. Their work focused on the stochastic data distribution of the device, which is useful for power-efficient operation for various applications. However, there was no information about the STDP and reservoir computing with the proposed design. In refs. 1,2,11, author presented a memristive device, which is built with HfO_2 and 65 nm CMOS process. These devices show higher stochasticity at high resistance levels and lower stochasticity at low resistance levels. They utilized EONS for network optimization and RAVENS as a hardware framework. They also covered the inclusion of STDP, reservoir computing, and applications like classification & control. In addition, these work show about 3.49x, 2.27x, and 7.7x higher power consumption compared to our proposed work. Another two research works also show higher power consumption compared to our proposed work[16,36].One of the research work utilized HfO_2 as a memristive device with 250 nm CMOS process[36] and other one used MIM configuration with 65 nm CMOS process[16]. On the other hand, our proposed work shows significant power savings with stochasticity. In addition, our work shows the trade-off between performance and cost of STDP inclusion.

In this work, a HfO_2-based memristive device is utilized to build a CMOS-compatible synapse, which shows the stochasticity with dynamic read operation. This device shows the opportunity for a fully functional device and at the same time, it enables controllable stochasticity with dynamic read operation. A Monte Carlo analysis showed the different operating points during the read operation. In addition the stochastic device shows up to 8.9x energy saving when compared to normal operation. It was shown that the stochasticity of the device can provide better learning for neuromorphic computing. Various applications were tested using neuromorphic software frameworks EONS and RAVENS while utilizing our memristive synapse with configurable read operation. These applications showed performance improvement in some cases while in others the results remained similar to non-stochastic. Specifically, the STDP learning method was analyzed with our synapse in these applications as well. Enabling STDP allowed for the neuromorphic optimization process to utilize the stochasticity more efficiently in some cases. Finally, a reservoir computing analysis was performed. Since reservoir computing is inherently stochastic, allowing a tunable stochastic feature in the form of the synapse allowed it to increase performance in some cases. However, in many cases with RC the

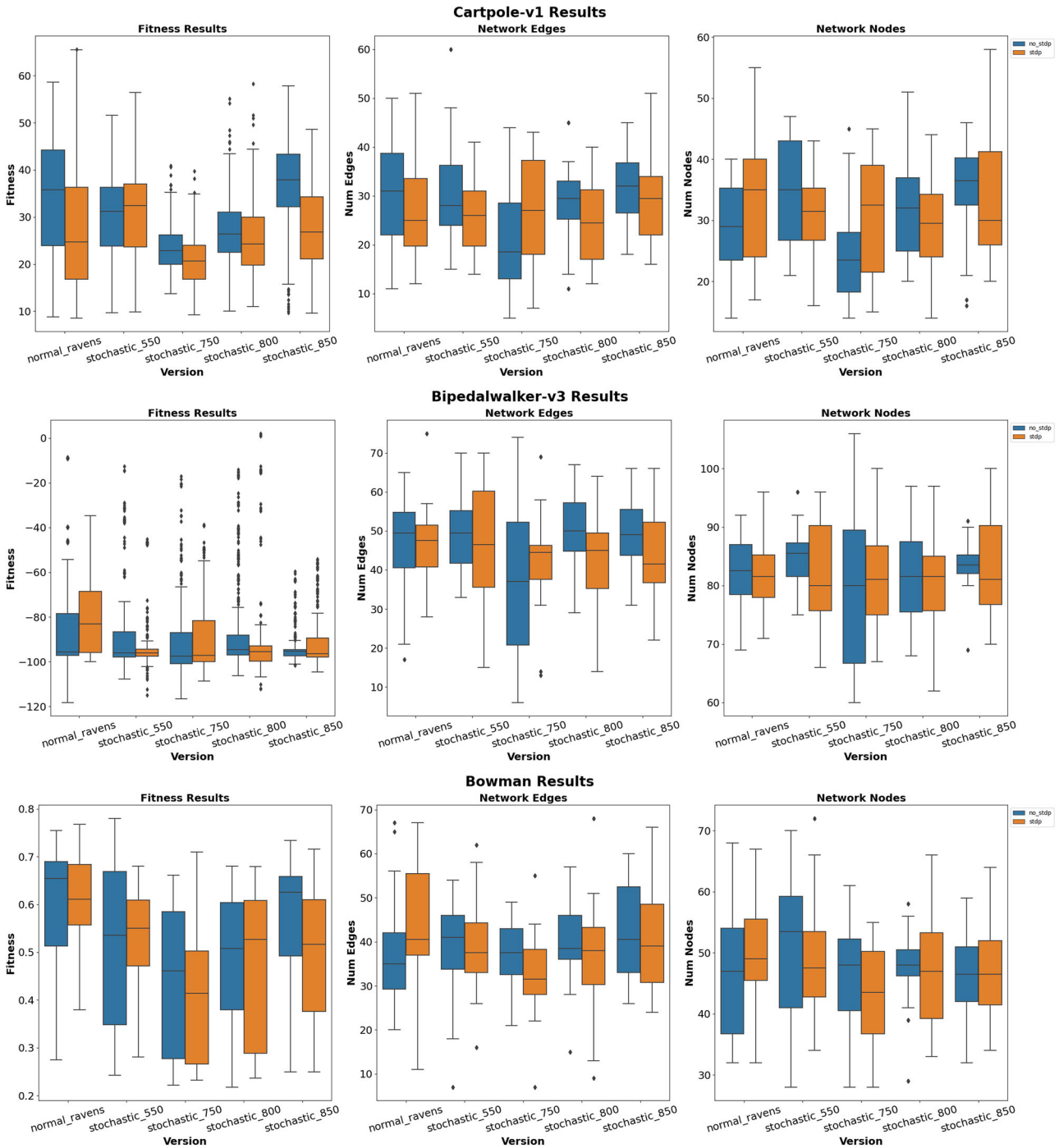

Fig. 7 | Control results for Cart Pole, Bipedal Walker, and Bowman control applications. The first, second, and third rows display results for the CartPole-v1, BipedalWalker-v3, and Bowman applications respectively including the fitness results, the number of network edges, and the number of network nodes.

performance dropped. Further exploration in RC with STDP and the synapse is needed to get the full picture. Additional future work will include other neuron properties such as leak and refractory which could be beneficial when combined with our 3T1R synapse.

Method

Stochasticity mapping of proposed synapse

In this work, our proposed synapse is targeted to program at 3 kΩ and 30 kΩ with programming voltage at 1.2 V and 0.77 V respectively. Hence, the synapse will be *READ* at different voltages to observe the *READ* current distribution. Here, *FinalREADcurrent* is denoted as *READCurrent* in Fig. 4. In addition, Fig. 4 shows the Monte Carlo simulation of *READ* operation, where 5000 samples are considered. Here, the *READ* voltage is varied from 0.55 V to 0.85 V to observe the stochastic pattern of read-out data. All the simulations are observed in Cadence Spectre with a Verilog-A model for memristive devices and 65 nm CMOS process by IBM. Figure 4a shows the *READ* operation at 0.55 V. The red and green colored distribution shows the data samples for 30 kΩ and 3 kΩ respectively. The final *READ* current of a 3 kΩ resistance will be higher than at 30 kΩ. Due to that, 3 kΩ is represented as data '1' and 30 kΩ is represented as data '0' in Fig. 4. In this test case, data '0' is 100% overlapped with part of data '1'. In addition, 93.48% of data '1' is overlapped with data '0'. If a reference current of a sense amplifier or a neuron is set at 25 μA, then 6.52% data '1' can be read without any noise. This kind of data distribution can be useful for weighted coin flip or

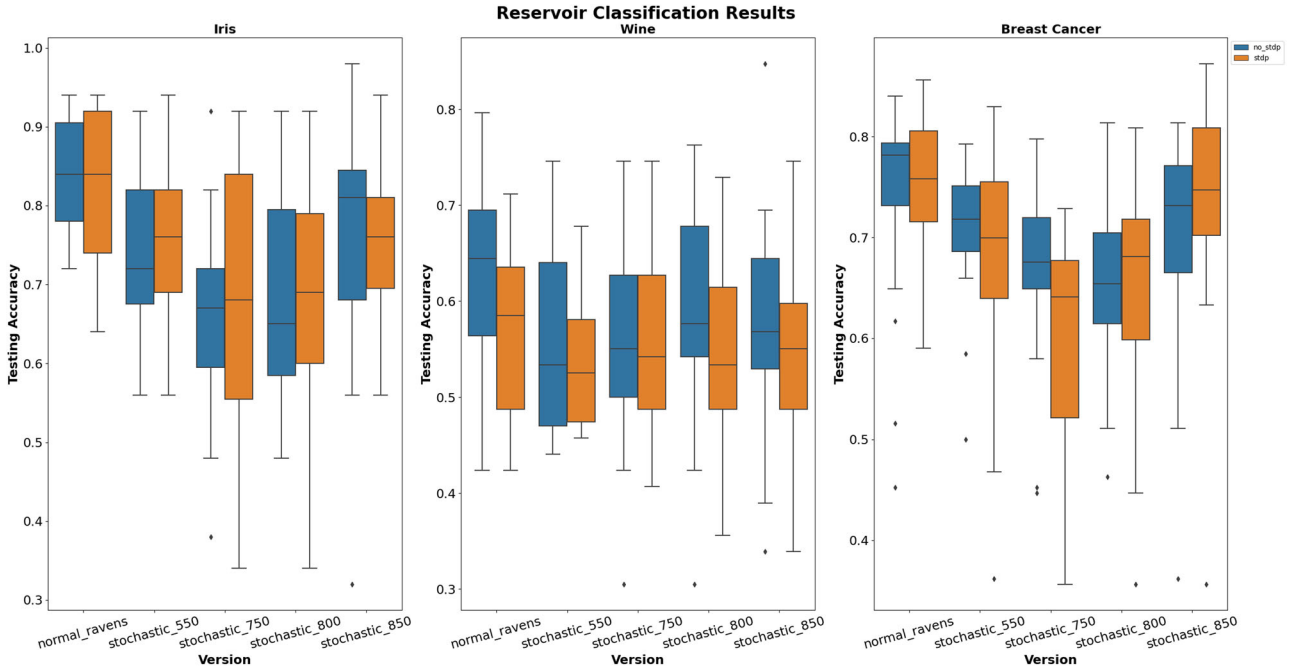

Fig. 8 | Reservoir classification results for Iris, Wine, and Breast Cancer datasets. The results include the testing accuracy for the Iris, Wine, and Breast Cancer datasets using the Reservoir Computing approach.

Fig. 9 | Reservoir timeseries results for EEG Emotion. The results display the testing accuracy for the EEG Emotion dataset using the Reservoir Computing approach.

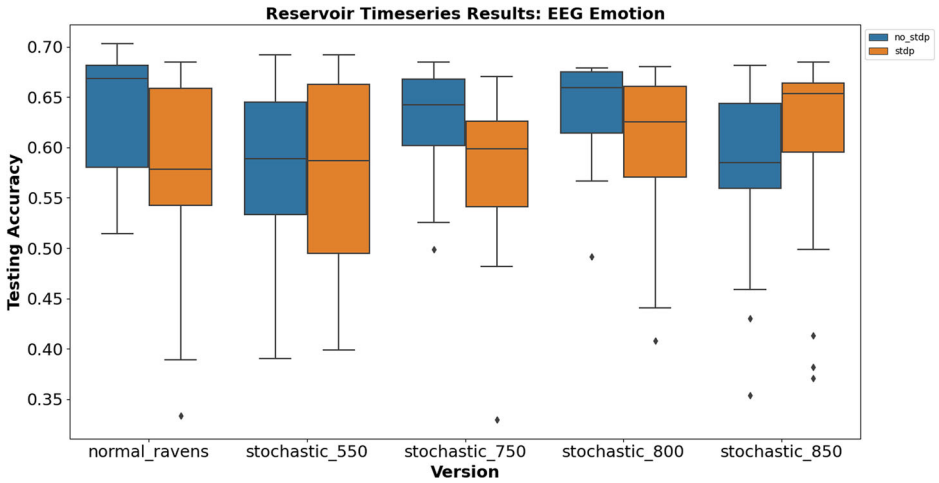

Fig. 10 | Reservoir control results for Cart Pole and Bipedal Walker control applications. The results include the fitness scores of the BipedalWalker-v3 and CartPole-v1 applications using the Reservoir Computing approach.

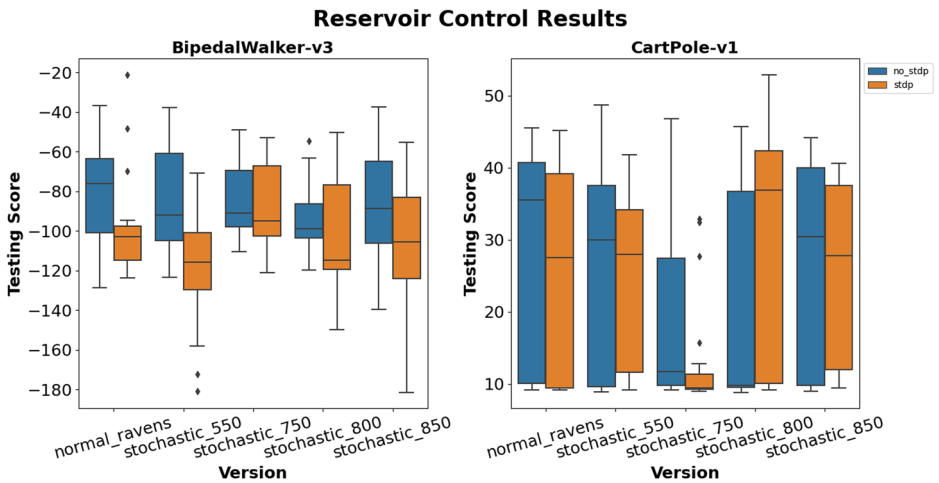

Table 2 | Comparison with prior works

Reference	Adv. Electron. Mater 2019[19]	Adv. Mater. 2023[20]	Science 2024[34]	Nanoscale 2013[13]	Frontiers 2016[35]	JETCAS 2023[1]	TCAS I 2023[2]	Scientific Reports 2024[11]	Scientific Reports 2020[36]	TNANO 2014[16]	This work
Process	-	-	180 nm	-	65 nm	65 nm	65 nm	65 nm	250 nm	65 nm	65 nm
Synaptic material	NbO_2	-	HfOx	Boron	HfO_2	HfO_2	HfO_2	HfO_2	HfO_2	MIM	HfO_2
Stochasticity in read data	Yes	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Dynamic stochastic adaptation	No	No	No	No	No	No	No	No	No	No	Yes
STDP	No	No	No	Yes		No	No	No	Yes	No	Yes
Classification	No	No	No	No	Yes	Yes	Yes	Yes	Yes	No	Yes
Control	No	No	Yes	No	-	Yes	Yes	Yes	No	No	Yes
Reservoir computing	No	No	No		No	Yes	Yes	Yes	No	No	Yes
Optimization algorithm	-	-	Yes	No	-	EONS	EONS	EONS	-	-	EONS
Hardware framework	-	Yes	Yes	No	-	RAVENS	RAVENS	RAVENS	-	-	RAVENS
Read Power	2 µW	-	-	12 µW	-	8.24 µW	5.35 µW	18.19 µW	4 µW	5 µW	2.36 µW
Power savings	0.85x lower	-	-	5.08x higher	-	3.49x higher	2.27x higher	7.7x higher	1.69x higher	2.11x higher	1x

neuromorphic applications[20]. Figure 4b exhibits *READ* current distribution at 0.6 V. In this test case, about 100% and 46.68% data of '0' and '1' are noisy, respectively. The mean current of *READ* operation is decreased with the increment of *READ* voltage. Next, Fig. 4c shows about 0.12% and 0.52% of data '0' and '1' are noisy accordingly, which is approximately stable condition for a *READ* operation with 5000 operations.

To achieve a more optimized *READ* current pattern, higher *READ* voltages are observed. Figure 4d shows at 0.7 V, the *READ* noise has reduced compared to the previous test case. Here, about 0.02% and 0.52% of data '0' and '1' are noisy respectively. Figure 4e shows the Monte Carlo analysis at 0.75 V, where the *READ* noise for data '0' is increased drastically. In addition, about 1/5 of the data '1' is noisy at this *READ* voltage. The next test case at 0.8 V will show that data '1' can be biased more towards data '0'. According to Fig. 4f shows about 100% and 68.46% of data '0' and '1' are noisy. Here, the *READ* current reduced is drastically. At 0.85 V the *READ* current is reduced drastically with a poor *READ* current distribution. About 100% and 98.22% of data '0' and '1' are overlapped with each other. Finally, a more reliable *READ* voltage is identified at 0.68 V for high-performance and error-free applications. Figure 4h shows a most reliable *READ* operation region, where the data noise is almost zero for both data '0' and '1'.

It can be concluded from the above observation, that there are three operating regions for this proposed synapse such as (i) higher *READ* current and stochastic *READ* data, which falls between 0.55 V and 0.6 V, (ii) optimum *READ* operation region with lower *READ* current and negligible *READ* noise, which falls between 0.61 V and 0.7 V, and (iii) lower *READ* current and poor *READ* data distribution region, which falls between 0.71 V and 0.85 V.

Spiking neural network evaluation

Utilizing the Monte Carlo simulation results from Fig. 4, probabilistic models were created for 0.55 V, 0.75 V, 0.80 V, and 0.85 V device versions that were then integrated into the TENNLab neuroprocessor called Reconfigurable and Very Efficient Neuromorphic System (RAVENS)[37]. The RAVENS neuroprocessor is a spiking neuromorphic processor that implements integrate-and-fire neurons with each neuron having a threshold, whether or not it will leak, or if it will have refractory periods while synapses have a weight and delay parameter. All these aspects are customizable to the application at hand with additional support for STDP, which dynamically modifies the synapse weights based on firing activity. To explore the impacts of the stochastic behaviors of the 3T1R devices on spiking neural networks (SNN), the probabilistic models of the devices were integrated into the RAVENS processor to simulate stochasticity in the hidden neurons. When a neuron exceeds its threshold value, the 3T1R device model is sampled to determine if the neuron will fire or not: a value of 0 means the neuron will fire while a value of 1 means the neuron will not fire. The various device models were evaluated on both classification and control tasks and compared to default RAVENS behavior to establish a baseline. Each task was also evaluated with and without STDP to better explore the dynamics of STDP and stochasticity in conjunction with one another.

Conventional means to introduce stochasticity is to leverage pseudo-random number generators (PRNG) such as linear-feedback-shift-registers (LFSR). These RNG units are deterministic, computationally expensive, and can require more hardware real estate to produce quality numbers. However, the proposed 3T1R devices utilize physics-based interactions from the material and device parameters to produce better-quality RNG that can be tuned for a desired distribution. Additionally, memristors are extremely popular and well-researched for neuromorphic applications which lend well for easy adoption.

For our training approach, we leveraged Evolutionary Optimization for Neuromorphic Systems (EONS)[38]. EONS optimizes the number of neurons, synapses, network structure, and other parameters through evolution. It begins with a random initial population of networks with the appropriate number of input and output neurons depending on the application. The networks are evaluated on the task and ranked based on their performance, with top-performing networks used to produce the next

generation of networks through crossover and mutation. This process is repeated for a given number of generations with the best performing individual of the last generation being the solution.

Data Availability

The data (circuit simulation) generated during this study are available from the corresponding author (H.D.) upon reasonable request.

Code availability

The codes for application evaluation are available from the author (C.D.S. and K.P.P.) upon reasonable request.

Received: 15 May 2024; Accepted: 30 October 2024;
Published online: 21 February 2025

References

1. Das, H. et al. Optimizations for a current-controlled memristor- based neuromorphic synapse design. *IEEE J. Emerg. Sel. Top. Circ. Syst.* **13**, 889–900 (2023).
2. Das, H. et al. An efficient and accurate memristive memory for array-based spiking neural networks. *IEEE Trans. Circ. Syst. I: Regul. Pap.* **70**, 4804–4815 (2023).
3. Weiss, R., Das, H., Chakraborty, N. N. & Rose, G. S. Stdp based online learning for a current-controlled memristive synapse. In *2022 IEEE 65th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 1–4, https://doi.org/10.1109/MWSCAS54063.2022.9859294 (2022).
4. Chakraborty, N. N., Das, H. & Rose, G. S. A mixed-signal short-term plasticity implementation for a current-controlled memristive synapse. In *Proc. Great Lakes Symposium on VLSI 2023*, GLSVLSI '23, 179–182, https://doi.org/10.1145/3583781.3590283. (Association for Computing Machinery, New York, NY, USA, 2023).
5. Chakraborty, N. N., Das, H. & Rose, G. S. Spike-timing-dependent plasticity for a hafnium-oxide memristive synapse. In *2023 IEEE 66th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 463–467, https://doi.org/10.1109/MWSCAS57524.2023.10406099 (2023).
6. Chakraborty, N. N., Das, H. & Rose, G. S. Spike-driven synaptic plasticity for a memristive neuromorphic core. In *2023 IEEE 66th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 644–648, https://doi.org/10.1109/MWSCAS57524.2023.10406136 (2023).
7. Chakraborty, N. N., Ameli, S. O., Das, H., Schuman, C. & Rose, G. S. Hardware software co-design for leveraging stdp in a memristive neuroprocessor. *Neuromorphic Comput. Eng.* **4**, 024010 (2024).
8. Chakraborty, N. N., Das, H. & Rose, G. S. Homeostatic plasticity in a leaky integrate and fire neuron using tunable leak. In *2023 IEEE 66th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 738–742, https://doi.org/10.1109/MWSCAS57524.2023.10406066 (2023).
9. Rathore, M. et al. Reliability analysis of memristive reservoir computing architecture. In *Proc. Great Lakes Symposium on VLSI 2023*, GLSVLSI '23, 131–136, https://doi.org/10.1145/3583781.3590210 (Association for Computing Machinery, New York, NY, USA, 2023).
10. Schuman, C. D., Das, H., Plank, J. S., Aziz, A. & Rose, G. S. Evaluating neuron models through application-hardware co-design. In *2023 57th Asilomar Conference on Signals, Systems, and Computers*, 537–542 (IEEE, 2023).
11. Das, H., Schuman, C., Chakraborty, N. N. & Rose, G. S. Enhanced read resolution in reconfigurable memristive synapses for spiking neural networks. *Sci. Rep.* **14**, 8897 (2024).
12. Gaines, B. R. Stochastic computing systems. *Adv. Inform. Syst. Sci.* **2**, 37–172 (1969).
13. Gaba, S., Sheridan, P., Zhou, J., Choi, S. & Lu, W. Stochastic memristive devices for computing and neuromorphic applications. *Nanoscale* **5**, 5872–5878 (2013).
14. Yu, S. et al. Stochastic learning in oxide binary synaptic device for neuromorphic computing. *Front. Neurosci.* **7**, 186 (2013).
15. Burr, G. W. et al. Neuromorphic computing using non-volatile memory. *Adv. Phys. X* **2**, 89–124 (2017).
16. Knag, P., Lu, W. & Zhang, Z. A native stochastic computing architecture enabled by memristors. *IEEE Trans. Nanotechnol.* **13**, 283–293 (2014).
17. Edstrom, J., Das, H., Xu, Y. & Gong, N. Memory optimization for energy-efficient differentially private deep learning. *IEEE Trans. Very Large Scale Integr. Syst.* **28**, 307–316 (2020).
18. Liu, J., Gong, N. & Das, H. Two birds with one stone: Differential privacy by low-power sram memory. *IEEE Transact. Dependable Secure Comput.* 1–14, https://doi.org/10.1109/TDSC.2024.3382630 (2024).
19. Carboni, R. & Ielmini, D. Stochastic memory devices for security and computing. *Adv. Electron. Mater.* **5**, 1900198 (2019).
20. Misra, S. et al. Probabilistic neural computing with stochastic devices. *Adv. Mater.* **35**, 2204569 (2023).
21. Zhang, H., Zhu, D., Kang, W., Zhang, Y. & Zhao, W. Stochastic computing implemented by skyrmionic logic devices. *Phys. Rev. Appl.* **13**, 054049 (2020).
22. Harabi, K.-E. et al. A memristor-based bayesian machine. *Nat. Electron.* **6**, 52–63 (2023).
23. Ramezani, H. & Akan, O. B. Importance of vesicle release stochasticity in neuro-spike communication. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 3343–3347, https://doi.org/10.1109/EMBC.2017.8037572 (2017).
24. Solanki, J., Beckmann, K., Pelton, J., Cady, N. & Liehr, M. Effect of resistance variability in vector matrix multiplication operations of 1t1r reram crossbar arrays using an embedded test platform. In *2023 IEEE 32nd Microelectronics Design & Test Symposium (MDTS)*, 1–5, https://doi.org/10.1109/MDTS58049.2023.10168152 (2023).
25. Sung, S., Wu, C., Jung, H. S. & Kim, T. W. Highly-stable write-once-read-many-times switching behaviors of 1d–1r memristive devices based on graphene quantum dot nanocomposites. *Sci. Rep.* **8**, 12081 (2018).
26. Im, I. H., Kim, S. J. & Jang, H. W. Memristive devices for new computing paradigms. *Adv. Intell. Syst.* **2**, 2000105 (2020).
27. Ielmini, D., Ambrogio, S., Milo, V., Balatti, S. & Wang, Z.-Q. Neuromorphic computing with hybrid memristive/cmos synapses for real-time learning. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1386–1389 (IEEE, 2016).
28. Das, H. et al. Rfam: Reset-failure-aware-model for hfo2-based memristor to enhance the reliability of neuromorphic design. In *Proc. Great Lakes Symposium on VLSI 2023*, GLSVLSI '23, 281–286, https://doi.org/10.1145/3583781.3590211 (Association for Computing Machinery, New York, NY, USA, 2023).
29. Asuncion, A. & Newman, D. Uci machine learning repository (2007).
30. Pedregosa, F. et al. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
31. Bird, J. J., Faria, D. R., Manso, L. J., Ekárt, A. & Buckingham, C. D. A deep evolutionary approach to bioinspired classifier optimisation for brain-machine interaction. *Complexity* **2019**, 4316548 (2019).
32. Bird, J. J., Ekart, A., Buckingham, C. D. & Faria, D. R. Mental emotional sentiment classification with an eeg-based brain-machine interface. In *Proc. International Conference on Digital Image and Signal Processing (DISP'19)* (2019).
33. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

34. Shi, T. et al. Stochastic neuro-fuzzy system implemented in memristor crossbar arrays. *Sci. Adv.* **10**, eadl3135 (2024).
35. Neftci, E. O., Pedroni, B. U., Joshi, S., Al-Shedivat, M. & Cauwenberghs, G. Stochastic synapses enable efficient brain-inspired learning machines. *Front. Neurosci.* **10**, 185771 (2016).
36. Zahari, F. et al. Analogue pattern recognition with stochastic switching binary cmos-integrated memristive devices. *Sci. Rep.* **10**, 14450 (2020).
37. Foshie, A. Z., Plank, J. S., Rose, G. S. & Schuman, C. D. Functional specification of the ravens neuroprocessor (2023).
38. Schuman, C. D., Mitchell, J. P., Patton, R. M., Potok, T. E. & Plank, J. S. Evolutionary optimization for neuromorphic systems. In *NICE: Neuro-Inspired Computational Elements Workshop* (2020).

Acknowledgements

This material is based in part on research sponsored by Air Force Research Laboratory under agreement FA8750-21-1-1018. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under award number DE-SC0022566.

Author contributions

H.D. proposed the dynamically reconfigurable synaptic device to take advantage of the stochasticity, enhanced performance, and power savings features of the proposed device. He prepared the initial draft for all the circuit simulations and analysis. K.P.P. helps with the application evaluation of the proposed architecture. R.D.F. helps with the writing. G.S.R. and C.D.S. were helping with overall supervision.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Hritom Das.