communications materials

A Nature Portfolio journal

Review article

https://doi.org/10.1038/s43246-025-00876-2

Transforming memristor noises into computational innovations

Chenchen Ding[1,2], Yuan Ren[1,2], Zhengwu Liu[1] ✉ & Ngai Wong[1]

Memristor-based compute-in-memory (CIM) systems show promise in accelerating various computing tasks with high energy efficiency, while various inherent noises in memristors, generally viewed as non-ideal characteristics, are detrimental to system performances. However, recent studies reveal that these noises can be harnessed to enable advanced computational functionalities, transforming challenges into opportunities. In this work, we systematically review the noise utilization strategies for these functionalities by categorizing them into two main types: 'noise-based perturbators' and 'noise-based generators'. The former utilize noise to help systems escape local minima and improve global convergence, as seen in combinatorial optimization, and to enrich feature spaces, as seen in reservoir computing (RC). The latter employ noise to produce random numbers or distributions, as used in physical unclonable functions (PUF), stochastic computing (SC) with true random number generator (TRNG), and Bayesian neural network (BNN). By examining these approaches, we highlight the potential of memristor noises to enable functionalities that are challenging to achieve with conventional precise computing systems. Finally, we discuss the challenges ahead and provide an outlook for future research. This review aims to pave the way for memristor-based energy-efficient and resilient computing technologies.

As big data processing demands increase and Moore's Law[1] is approaching its physical limits, the limitations of the traditional von Neumann architecture[2] have become increasingly apparent. In response, compute-in-memory (CIM) has emerged as a promising paradigm, performing calculations at the data storage location to overcome data movement bottlenecks. Memristive device-based CIM[3,4], owing to its excellent electronic properties, simple manufacturing, and high-density integration potential, has attracted significant attention from both academia and industry[5–8]. In recent years, a range of memristive devices have demonstrated excellent application potential, including oxide-based resistive memory, phase-change memory (PCM), magnetic random-access memory (MRAM), and ferroelectric memory. Each technology offers unique advantages for CIM systems. Among these, oxide-based resistive memory has emerged as particularly noteworthy due to its reliable data storage, high integration density, low energy consumption, and high speeds, making it a significant driver for CIM technologies. In this paper, unless noted otherwise, the term "memristor" primarily refers to oxide-based resistive memory.

The conductive properties of memristors stem from the movement of internal ions within their metal-insulator-metal (MIM) structure under voltage bias, resulting in the formation of conductive filaments[9]. However, memristors exhibit a range of inherent noises during operations[7,10,11], such as the noises in write and read operations of memristors and stochasticity of switching and switching delays. Meanwhile, memristors display significant variability both between individual devices and across successive write and read cycles of the same device, i.e., device-to-device variations and cycle-to-cycle variations, respectively. These characteristics and behaviors originate from random formation and rupture of conductive filaments, stochastic ion migration, trap state dynamics, material inhomogeneities, and thermal fluctuations of charge carriers[10,12–14]. Nevertheless, these stochastic phenomena often present significant challenges for the reliable integration of memristors into complex systems. Researchers have explored methods to address the noise issue, including advanced memristor programming techniques[15], optimized high-precision data mapping techniques[16], redundant data representation[6], and noise-aware training methods[17]. Additionally, hybrid precision solutions have been used, by deploying memristor-based noise-sensitive components within digital complementary metal–oxide–semiconductor (CMOS) circuits-based frameworks while relegating less sensitive computational tasks to analog memristors[18,19]. This approach aims to minimize noise impact by leveraging the strengths of both traditional and emerging technologies.

Recent years have witnessed a paradigm shift in the perception of intrinsic noise within memristor-based systems. What was once viewed

[1]Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong SAR, China. [2]These authors contributed equally: Chenchen Ding, Yuan Ren. ✉e-mail: zwliu@eee.hku.hk

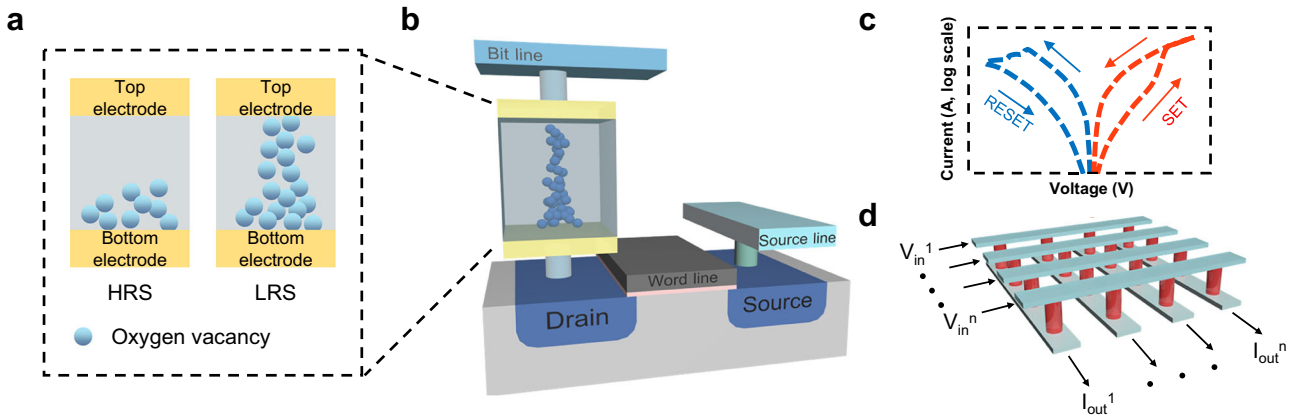

Fig. 1 | Memristor and memristor-based CIM principles. a Schematic of oxygen vacancy distributions of a memristor in its HRS and LRS states. **b** Common 1T1R cell used in CIM applications. **c** Resistive switching characteristics of the memristor. **d** Memristor array utilized in CIM for vector-matrix multiplication. HRS high resistance state, LRS low resistance state, 1T1R one-transistor-one-resistor.

exclusively as a disruptive phenomenon is now being re-evaluated for its potential to confer computational advantages[20–23]. This evolving understanding has led to a shift from minimizing the negative effects of noises to exploring ways to exploit these inherent characteristics, fostering innovative applications across multiple computing domains, such as combinatorial optimization[24], reservoir computing (RC)[25,26], physical unclonable function (PUF)[27,28], stochastic computing (SC)[28], and Bayesian neural network (BNN)[2,9]. Previous reviews have analyzed these specific fields by focusing on summarizing various implementation methods and devices used separately. However, a comprehensive review from the perspective of memristor noise utilization strategies across multiple applications is still lacking. Furthermore, some reviews do not encompass latest field advancements, particularly those published post-2020[24,27,28]. These gaps underscore the need for a systematic exploration of noise leveraging in diverse computational contexts, potentially unlocking new possibilities in emerging computing paradigms.

This paper aims to address these gaps by providing a systematic review of noise utilizations in memristor-based systems. We first introduce typical concepts and noise behaviors of memristors. Then, we categorize noise utilization strategies into two types based on their specific working mechanisms: noise-based perturbators and noise-based generators. Noise-based perturbators utilize loosely controlled analog noise as a system perturbation, aimed at enhancing system dynamics, such as altering system states and giving diverse responses. Examples in this type include combinatorial optimization and RC. Conversely, noise-based generators employ selective control and processing of noise sources with specific distributions to produce desired numerical values or probability distributions. Examples in this type include PUF, SC, and BNN. Finally, we discuss the challenges ahead and also point out the potential future research directions.

Memristors and noise behaviors

Figure 1a illustrates the working principle of a memristor, showing its high resistance state (HRS) and low resistance state (LRS). These states are controlled by the migration of defects, such as oxygen vacancies or metal ions, within the conductive filaments. Generally, in SET operation, a positive voltage applied to the top electrode drives defects towards the bottom electrode, resulting in an LRS. Conversely, in RESET operation, a negative voltage causes defect migration back to the top electrode, leading to an HRS due to conductive filaments rupture.

Figure 1b shows a typical one-transistor-one-resistor (1T1R) configuration, where a two-terminal metal-insulator-metal memristor structure is connected with a transistor for switch control. Other configurations, such as one-resistor (1 R) and two-transistor-two-resistor (2T2R), are also used in various computational applications[8,10]. Figure 1c plots schematic of a typical direct-current I-V curve of a memristor, showing the conductance switching characteristics. When many memristors are integrated into a crossbar array, vector-matrix multiplications can be in situ realized in a highly efficient manner through physical laws[7,11,30] (Fig. 1d).

The material properties and manufacturing process variations of memristors not only result in unstable conductance behaviors during operation, such as write or programming variation (Fig. 2a), read fluctuations (Fig. 2b), and nonlinear current-voltage relationship (Fig. 2c), but also affect the switching characteristics of memristors, including switching delay (Fig. 2d) and switching voltage (Fig. 2e). These non-ideal characteristics ultimately lead to device-to-device and cycle-to-cycle variations in memristor arrays.

Table 1 showcases common noises in memristors, which are origins for various variations in electrical behaviors. These characteristics include:

- Flicker noise (1/f noise)[31–34]: A common phenomenon in nanodevices, with its power spectral density (PSD) exhibiting an inverse relationship with frequency. It primarily arises from defects and traps in the material, affecting device performance at low frequencies.
- Shot noise[35,36]: Stems from the discrete nature of charge carriers and their random arrival at conduction channels. Its PSD is frequency-independent and often modeled using a Poisson process.
- Thermal noise[37]: Results from the random thermal motion of carriers in the materials forming memristors and peripheral circuits, setting the lower bound for system noise levels and detectable signals.
- Random telegraph noise (RTN)[38–41]: Manifests as sudden, step-like transitions between discrete voltage or current levels at random intervals. It is attributed to the activation and deactivation of electron traps within or near the filament.

In practice, flicker noise and RTN often significantly impact device behavior, reducing the readout margin and causing significant read noises[14,21,42].

Recent research has explored ways to integrate these non-ideal behaviors into computational processes, achieving positive effects that are challenging to realize in conventional CMOS-based systems. As illustrated in Fig. 3, these efforts can be categorized into two types: noise-based perturbators (Fig. 3a) and noise-based generators (Fig. 3b). Perturbators utilize the non-ideal characteristics of memristors as system perturbations, which help to overcome local optima in combinatorial optimization and improve the dynamics in RC systems, ensuring continuous progression and achieving feature dimension enhancement in these tasks. Generators, on the other hand, leverage constrained or selected memristor non-idealities to produce various useful outputs. These include generating random bits for cryptographic and security applications, creating PUFs for device authentication and secure key generation, and generating random distributions for statistical and simulation applications.

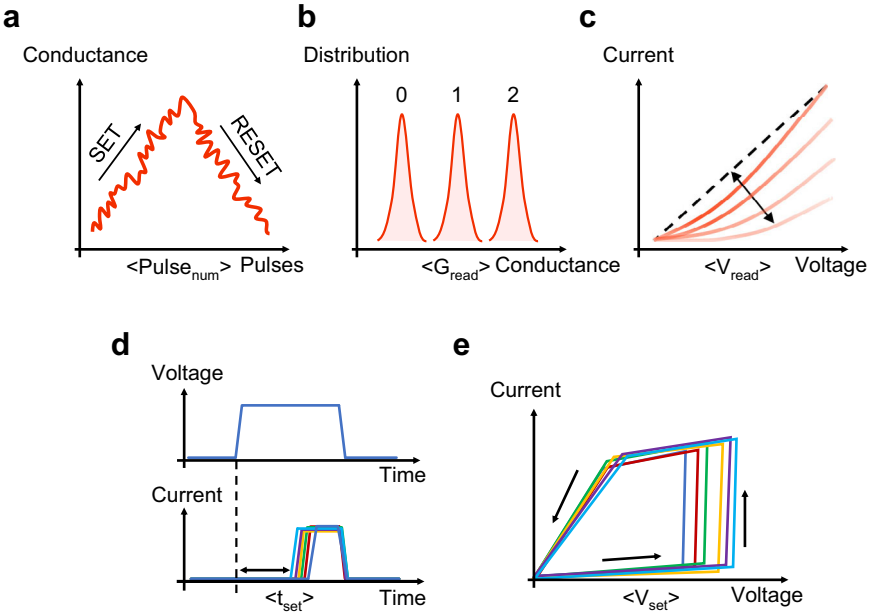

Fig. 2 | Several electrical characteristics reflecting the device non-ideality of memristors. a Write variation in memristor conductance. **b** Read fluctuation of memristor conductance. **c** Nonlinearity between read voltage and read current. **d** Randomness in switching delay. **e** Variability in memristor switching voltage.

Noise-based perturbators

Systems for solving combinatorial optimization problems and constructing RC are susceptible to local optima entrapment and monotonous feature extraction during their optimization and utilization phases. Moreover, these types of algorithms inherently exhibit higher noise tolerance compared to other computational tasks[26,43,44]. This characteristic allows the exploitation of memristor noise as a natural perturbation source, which can influence the system's original state or generate diverse responses to inputs, facilitating continuous optimization and enhancing feature dimensions in these applications.

Combinatorial optimization

Combinatorial optimization has widespread applications in engineering[45–48]. Paradigmatic algorithms solving combinatorial optimization problems such as the Traveling Salesman, Knapsack, and Graph Coloring problems often exhibit deceptively simple formulations[49–55] and can be converted into vector matrix multiplications. However, these algorithms show a computational complexity explosion when the problem sizes grow. Consequently, traditional CMOS-based von Neumann architectures struggle to solve them efficiently at large scales. Thankfully, in these years, memristor arrays have shown excellent potential for solving these problems[22,56–61] with the excellent ability to handle vector-matrix multiplications.

Hopfield networks, Ising machines, and Boltzmann machines are typical approaches[45] for solving combinatorial optimization problems on memristor arrays. Taking the Boltzmann machine as an example[60], combinatorial optimization problems are initially mapped to energy-related models, where the system's state is defined by an energy function. To implement the optimization process on memristor arrays, problem-related weights are mapped onto the array, while binary inputs represent the states of individual nodes in model. This configuration allows the gradients associated with node states to be efficiently computed through simple vector-matrix multiplications. The optimization process proceeds by iteratively switching inputs to progressively reduce the system's energy until convergence is achieved.

However, algorithms like Hopfield networks and restricted Boltzmann machines generally lack self-feedback, making them prone to converging to local optima. A typical approach is to introduce simulated annealing[57,62], which adds a dynamically adjustable perturbation to the system, allowing optimization in the opposite direction of the current gradient to escape local optima. Introducing this perturbation has been a research focus. One method (Fig. 4b) is to use dynamically adjustable noise sources from peripheral circuits as system perturbations[63]. However, this design requires auxiliary circuit overhead, consuming additional resources. Another method utilizes the intrinsic noise of memristors for system perturbation. As shown in Fig. 4c, in a Hopfield network for combinatorial optimization, this manifests as adding transient self-feedback[56], represented by the write noise of devices used for diagonal values. To ensure gradual convergence, the perturbation is decreased as optimization progresses by changing the resistance of memristors from low to high resistance states. This method naturally introduces perturbation through the special mapping structure of Hopfield networks. However, for larger problems, single device self-feedback may not provide sufficient perturbation. Besides, another approach is to combine inherent noise from memristors and peripheral circuits (Fig. 4d)[22]. A scheme based on hysteresis threshold functions can reflect noise in early optimization stages and partially eliminate it later. Experiments have analyzed the relationship between inherent system noise and optimal perturbation demand. Although magnitudes of noises in memristor array are generally lower than optimal magnitudes for small tasks, the gap narrows as problem scale increases and optimization progresses. This method does not require additional array operations but may impact overall optimization speed due to post-digital processing.

Furthermore, Yi et al. investigated the role of transient perturbations in diagonal elements of Hopfield networks for accelerating optimization[64]. Lee et al. leveraged inter-layer interference in vertical memristors to implement effective simulated annealing for combinatorial optimization problems[61]. In current research, optimization parameters, particularly in simulated annealing processes, are predominantly artificially set or controlled by software[22,56]. Increasing attention is being directed towards developing self-adaptive mechanisms responsive to the optimization problems, system states during optimization[65]. Moreover, as traditional simulated annealing can only assess the impact of single-node changes per iteration, the development of efficient parallel annealing techniques based on memristors represents a significant research direction[57].

Reservoir Computing (RC)

RC, evolved from recurrent neural networks (RNNs)[66–69], addresses training difficulties related to vanishing and exploding gradients in RNNs[70].Typical RCs are formed by reservoir layers and a readout layer[70,71]. The reservoir layer consisting of nonlinear connections represented by transient or fixed

Table 1 | Common noises in memristors

	Flicker noise[31–34]	Shot noise[35,36]	Thermal noise[37]	Random telegraph noise[38–41]
Features	Noise PSD almost inversely proportional to frequency (significant at low frequencies)	Intensity correlated with current magnitude	Constant PSD across frequency range	Random transitions of conductance between discrete states
Main causes	Energy state fluctuations caused by defects and traps	Discreteness of charge carriers	Thermal motion of carriers	Random trapping and detrapping of carriers by defects/traps
Influencing factors	Defect distribution, carrier concentration, temperature, bias voltage	Current magnitude, potential energy barrier properties	Temperature, resistance value	Defect density, device size, temperature, applied voltage
Main impacts	Affect analog computing accuracy with low-frequency signal	Cause instability in HRS readout	Affect read accuracy and minimum detectable signal	Cause instability in read/write operations

random weights, maps low-dimensional inputs to high-dimensional feature spaces. The readout layer is typically formed by a simple linear projection. Since the reservoir layers are designed to provide random perturbations to the system, physical systems capable of providing rich dynamics are appropriate to perform effective reservoir computing[72–75]. As one of these devices, memristors can serve as a source of random perturbations in the system[23,76–80] with their dynamic enhanced by implicit non-ideal characteristics like stochastic conductance distributions and nonlinearity between current and voltage. Memristor-based RC can be categorized into dynamic memristor-based RC and static memristor-based RC, based on the constituent elements of the reservoir layer.

As shown in Fig. 5a, b, dynamic memristor-based RC[23,81,82] forms the reservoir layer with a single physical node using dynamic memristor and a sequence of virtual nodes created by the evolution of the physical node output over time[25,83]. This paradigm can be extended to arrays of multiple dynamic memristors, allowing parallel formation of nonlinear connections to process input signals concurrently (Fig. 5c)[23,76]. Recent research has leveraged the inherent nonlinearity and short-term memory characteristics of dynamic memristors[78,84,85]. During the inference phase of the RC, the current state of a memristor depends on both the immediate input pulse and the influence of previous pulses within a certain time range. This temporal dependency is weighted, with more recent pulses having a stronger impact. Interestingly, this approach utilizes cycle-to-cycle noises of single memristors and device-to-device variations, to enhance input data feature dimensionality. Based on this characteristic, Zhong et al. proposed a parallel dynamic memristor-based reservoir computing system[86] by applying a controllable mask process, then introduced a fully analog RC system[76] (Fig. 5d), achieving exceptionally low power consumption and hardware cost. Furthermore, the multi-functional memristor array based on 3D integration[87] (Fig. 5e) achieves an effective combination of dynamic memristors and non-volatile memristors, further constructing a brain-like system with high computational density composed of neural networks and RCs.

RC systems based on dynamic memristors have limitations in processing speed and integration density and can hardly capture spatial relationships. Static memristor-based RC (Fig. 5f, g) overcomes these limitations by utilizing parallel processing in static memristor arrays, which provide higher density weights, allowing for faster computation and better capture of spatial relationships. One type of static memristor-based RC employs echo state layers, which generate internal "echoes" of the input to influence the system's state[88]. These layers are characterized by sparse and invariant random connections. Specifically, the inconsistent breakdown voltages between these memristors can be used to implement this type of reservoir pools with massive random connections[77,89]. By applying a fixed voltage to a TaO_x memristor array, the conductance follows a normal distribution, realizing fixed random weights of the echo state layer and improving energy efficiency through computation in memory (Fig. 5h). Wang & Li further explored the RC using the array based on self-selecting memristors and significantly reduced leakage current power consumption. Although their validation was limited to small-scale implementations, transistor-less array designs demonstrate higher scalability and potential for three-dimensional integration (Fig. 5i)[90]. Another typical paradigm of static memristor-based RC[91,92] is formed by liquid state layers which employ spiking neurons as internal computational nodes to emulate the accumulation and firing behavior of biological neurons. Additionally, this type of RCs often incorporates separate leaky integrate-and-fire (LIF) neurons as outputs following the reservoir layers (Fig. 5f). Soures et al.[93] proposed a memristor-based liquid state RC that achieved high robustness against noise and faulty components in classification tasks. Building upon this concept, Sayyaparaju et al.[94] introduced a reconfigurable architecture based on memristor arrays, capable of implementing any given reservoir topology. Notably, this design eschews complex analog-to-digital converters and digital-to-analog converters, thereby further reducing power consumption and area overhead and resulting in an efficient and compact design. Recently, Lin et al., inspired by the zero-shot learning manner of the human

brain, designed a zero-shot liquid state machine, achieving more complex multimodal event data learning[95].

Current research priorities include employing novel memristors to create more tunable and efficient RC systems[96–98], as well as achieving more complex and diverse internal dynamics through 3D integration[25,87,99,100]. These advancements in memristor-based RC designs continue to push the boundaries of efficient temporal data processing in hardware, opening up new possibilities for neuromorphic computing and edge AI applications.

Applications of other memristive devices working as perturbator

Apart from oxide-based resistive memories, significant advancements have been made in harnessing non-ideal characteristics as perturbation mechanisms in various types of memristive devices. For instance, spintronic devices—such as magnetic tunnel junctions (MTJs); ferroelectric devices like ferroelectric field-effect transistors (FeFETs) and ferroelectric diodes (FDs); and PCMs—have all demonstrated promising progress in this area.

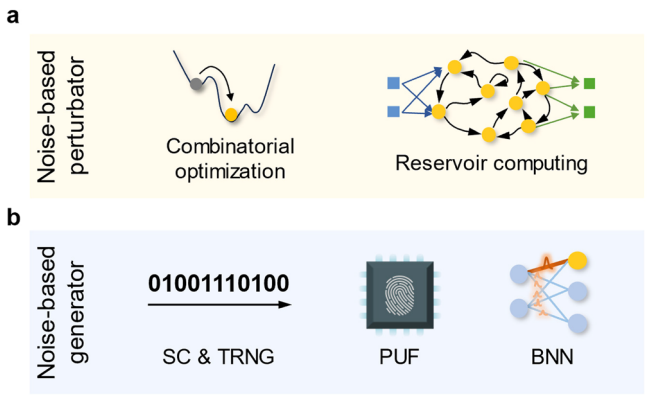

Fig. 3 | Perturbator and generator based on memristor noise. a Applications of noise-based perturbators. Noise serves as system perturbations to ensure continuous progression in combinatorial optimization and to enhance feature dimensionality in reservoir computing. **b** Applications of noise-based generators. Noise functions as a random source for SC and TRNG, as well as for generating PUFs and random distributions for BNNs.

Si et al. (Fig. 6a) proposed an Ising machine based on spin devices. They utilized the inherent random behavior in superparamagnetic tunnel junctions to perturb the solving progress, thereby meeting the randomness requirements in simulated annealing without introducing complex circuits[101]. However, these works are limited by relatively low operating frequencies and small scale, presenting challenges to efficiently solve larger-scale problems. Additionally, Kang et al. (Fig. 6b) proposed a spiking Boltzmann machine based on PCM[102]. They implemented an effective random mechanism for simulated annealing using the random walk noise of their LIF implementation, ultimately achieving effective solutions to the Max-Cut problem.

Toprasertpong et al. leveraged temporal nonlinear dynamics and the spatial distribution of currents flowing from multi-terminal FeFETs and implemented RC to process time series information[103]. Nako et al. used a parallelized design and researched various optimization methods[104], ultimately achieving good results in handling higher-dimensional temporal information. Furthermore, Chen et al. (Fig. 6c) first implemented RC using a fully ferroelectric design[105]. They utilized volatile FDs with imprint field which exhibits short-term memory and nonlinearity to form the dynamics for reservoir and utilized non-volatile FDs for the readout network, ultimately enabling the entire system to successfully handle various temporal tasks.

Meanwhile, Zhu et al. utilized halide perovskite (HP) memristors to form a reservoir and implemented real-time biological neural signal analysis[81]. Subsequently, John et al. further designed reconfigurable memristors with HP nanocrystals[97] (Fig. 6d). By actively regulating the compliance current, the devices achieve short-term volatility for reservoir layers and long-term stability for readout layers, ultimately supporting RC while demonstrating excellent durability. Taniguchi et al. leveraged voltage-induced anisotropy changes to control spontaneous magnetization relaxation MTJs, thereby realizing the short-term memory and nonlinear transformation required for RC without driving current or magnetic field[106].

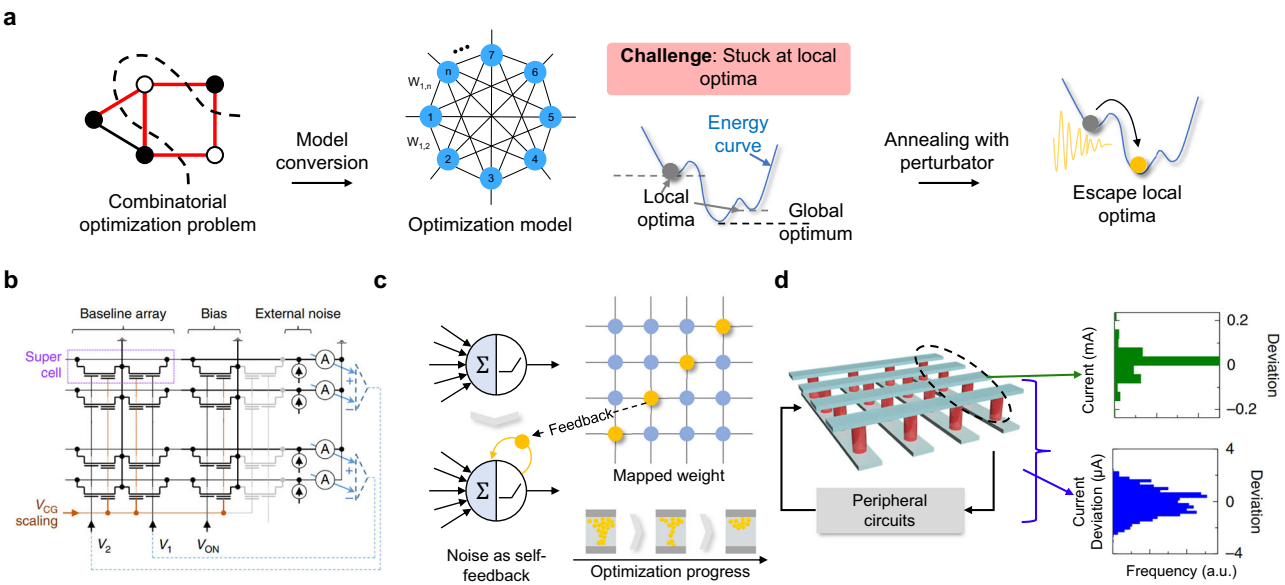

Fig. 4 | Continuous optimization using memristor noise perturbation. a Basic combinatorial optimization problem formed by the max-cut task. The optimization model can be constructed from the problem but always suffers converging at local optima. Memristor-based perturbator aids in escaping local optima during optimization. **b** Use external noise to perturb system states and help optimization[63]. **c** Noise as self-feedback in the Hopfield network. Weights are mapped with self-feedback on the memristor array. During optimization, the state of feedback memristors gradually changes from LRS to HRS[56]. **d** Use intrinsic noise to perturb the optimization progress. A hysteresis threshold method is designed to adjust the magnitude of used noise[22]. Panel **b** is reproduced with permission from ref. 63, copyright (Springer Nature, 2019). Panel **c** is reproduced with permission from ref. 56, copyright (AAAS, 2020). Panel **d** is reproduced with permission from ref. 22, copyright (Springer Nature, 2020).

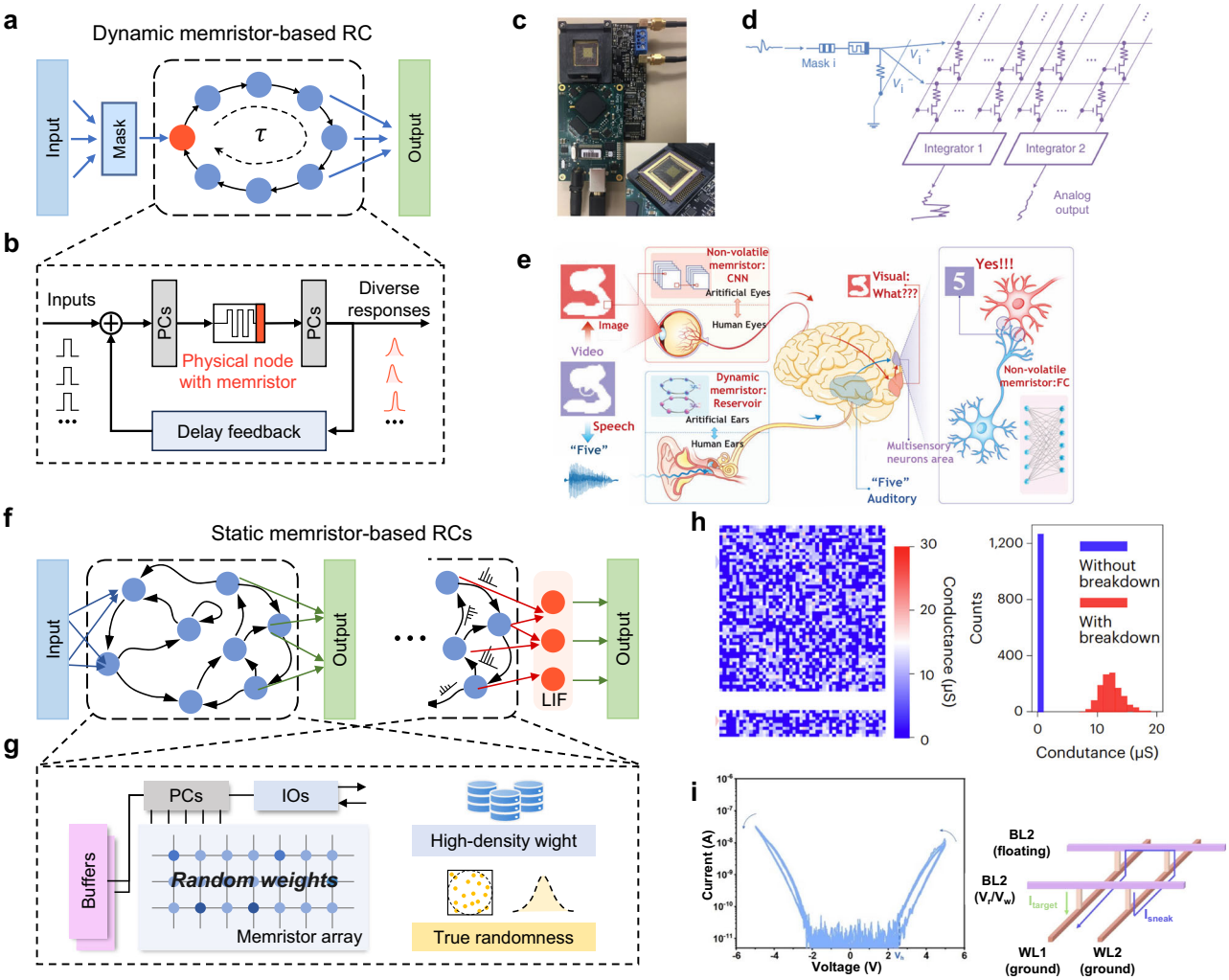

Fig. 5 | Formation of a reservoir using memristor noise. Dynamic memristor-based RC (**a**) with the reservoir layer formed by one physical node and a sequential of virtual nodes formed by the delay feedback (**b**). **c** System designed for implementing dynamic memristor-based RC[23]. **d** Circuit diagram of the fully analogue RC system[76]. **e** Combination of non-volatile memristor-based convolution neural network and dynamic memristor-based RC for more compact neuromorphic computing[87]. Architecture of static memristor-based RCs (**f**) with the echo state layer and liquid state layer, respectively. The former typically employs time-independent activation functions as internal nodes, while the latter commonly uses spiking neurons as internal units, often requiring leaky integrate-and-fire neurons as outputs. In terms of reservoir layer implementation, they generally share similar architectures. The memristor array-based implementation (**g**) can provide high-density weights and true randomness for these types of RCs. **h** Random resistor arrays used for echo state layers[77]. **i** Leakage suppression in RC with self-selective memristors[90]. Panel **c** is reproduced with permission from ref. 23, copyright (Springer Nature, 2019). Panel **d** is reproduced with permission from ref. 76, copyright (Springer Nature, 2022). Panel **e** is reproduced with permission from ref. 87, copyright (IEEE, 2023). Panel **h** is reproduced from ref. 77, copyright (Springer Nature, 2023). Panel **i** is reproduced with the permission from ref. 90, copyright (IOP Publishing, 2024).

Table 2 provides a comparative analysis of perturbators implemented across different types of memristive devices. Several kinds of memristive devices have demonstrated good progress in harnessing non-ideal characteristics as effective perturbators. Among them, some implementations with oxide-based resistive memory have showcased relatively larger scales while achieving more comprehensive performance evaluations[22,76,77]. Meanwhile, constraints in testing scale and testing methodologies have posed challenges for some studies in the evaluation of standardized system efficiency metrics[56,97,102]. Additionally, efficient design strategies such as fully analogue architectures and 3D integration have garnered considerable research interest.

Noise-based generators

In CMOS-based circuits, collecting noise to generate randomness is relatively power-intensive, and these circuits are vulnerable to noise and cryogenic attacks[107,108]. In contrast, memristors offer a more energy-efficient and secure alternative as random sources due to their inherent switching randomness and operational noise, which are more easily captured and resistant to modeling attacks.

As random source generators, certain noises of memristors are exploited and controlled to produce noise signals with desired distributions. These signals can be utilized for various applications, including the generation of PUFs, random bit streams for SC, and the production of probability distributions.

Stochastic Computing (SC) and True Random Number Generation (TRNG)

SC encodes numerical values using random sequences of 1 s and 0 s, transforming binary computations into serial bit operations based on probability distributions[109]. As shown in Fig. 7a, SC converts floating-point multiplication into an AND operation of probability bit streams. This simplifies complex operations, reducing hardware resource consumption. SC reliance on bit stream probabilities provides interference resistance, as individual bit flips minimally impact overall results. Consequently,

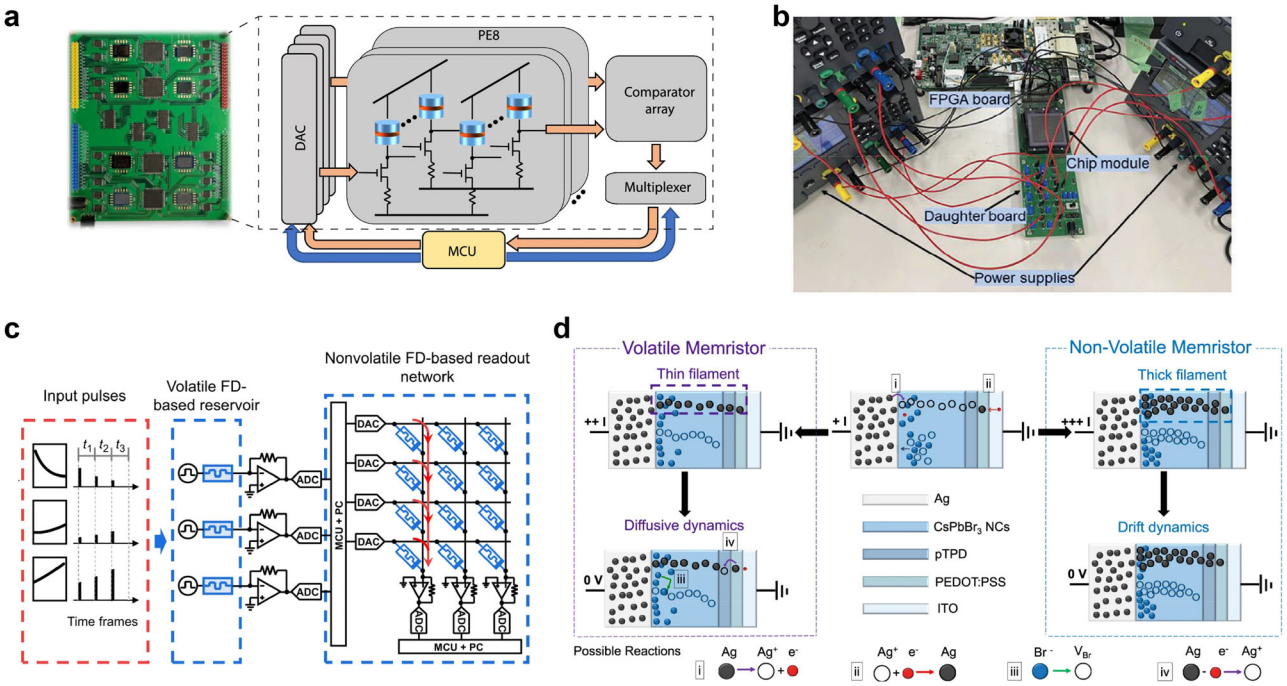

Fig. 6 | Demonstrations with other memristive devices as perturbators. a A spintronic device-based system that supporting the implementation of simulated annealing for combinatorial optimization tasks[101]. **b** PCM-based Max-Cut solver with random walk noise enhanced simulated annealing[102]. **c** Fully ferroelectric RC with volatile ferroelectric diodes as reservoir and non-volatile ferroelectric diodes as readout network[105]. **d** Reconfigurable HP memristors for efficient RC[97]. Panel **a** is reproduced with permission from ref. 101, copyright (Springer Nature, 2024). Panel **b** is reproduced with permission from ref. 102, copyright (Wiley, 2024). Panel **c** is reproduced with permission from ref. 105, copyright (Springer Nature, 2023). Panel **d** is reproduced with permission from ref. 97, copyright (Springer Nature, 2022).

designing efficient TRNGs for continuous random bit streams is crucial in SC.

Memristor switching exhibits good randomness distribution, modeled as a Poisson distribution[110]. This randomness is a true random source, unlike digital systems pseudo-random sources. As illustrated in Fig. 7b, the device switching probability during SET can be controlled by voltage and pulse width[110]. During readout, this probability converts to 0/1 bit streams through a comparison circuit, enabling random number generation with appropriate probabilities[111,112]. Furthermore, by implementing row-parallel operations on the array, the high parallelism inherent in memristor arrays can be fully leveraged for efficient random bit stream generation[113].

In SCs, the random number generator is a frequently working component, and its low energy consumption and streamlined design have a crucial impact on the energy efficiency and area of the whole SC system[114]. At the same time, the difficulty in monitoring slight noise can cause the complex and energy-consuming designs for noise collection and error code corrections[115,116]. To address these issues, researchers applied back-and-forth soft pulses on constant memristors[117] (Fig. 7c) and utilized noise-enhanced forming method with low forming current and high target resistance[118] (Fig. 7d), to generate more random and detectable noises. Equbal et al. suggested using memristor switching variability and race conditions of CMOS set-reset latches as coupled entropy sources, enhancing randomness without complex post-processing[119].

RTN in memristors, caused by charge trapping or detrapping, can be easily captured, compared to the thermal noise in CMOS-based circuits[120,121]. This characteristic enables random number generation at lower voltages and currents[7,122]. Huang et al.[123] demonstrated an ultra-low power TRNG using contact memristors. Furthermore, Song et al.[124] enhanced RTN current characteristics by optimizing device conductance and read voltage (Fig. 7e), achieving a stable TRNG that passed all National Institute of Standards and Technology randomness tests. However, in these implementations, the speed of electron capture and emission in memristive materials limits the sampling frequency for random number generation. Additionally, controlling RTN frequency and amplitude remains challenging, often necessitating complex circuits and post-processing[124]. In response to this, Gong et al. investigated a TRNG based on another significant noise source, flicker noise (Fig. 7f), and obtained a random bit stream with near-zero autocorrelation[125]. Besides, some dynamic memristors leverage inherent volatile threshold switching behavior to observe volatile resistive switching at low current ranges, thereby enabling high-endurance, low-power TRNGs[126,127].

In future works, improving throughput[128,129] and reliability[118,128,130] continues to be focal points of ongoing research of SC and TRNG. The inherent variability and noise-driven stochastic nature of memristor behaviors offer unique opportunities for generating high-quality random numbers and usage in SCs. This also demonstrates the potential of utilizing the non-idealities of memristors as generators in constructing more efficient, compact, and robust probabilistic computing systems.

Physical Unclonable Functions (PUFs)

PUFs are hardware security primitives that leverage physical characteristics of devices, such as manufacturing variations in integrated circuits or impurity distributions in optical systems. PUFs can generate hardware-specific responses by inputting fixed challenges. These responses can then form hardware-specific keys enabling applications like identity verification[131–133] and data encryption/decryption[134–136]. While traditional PUF architectures are CMOS transistor-based, utilizing manufacturing variations, they face reliability issues and vulnerability to modeling attacks[137]. Memristor-based PUFs have emerged as a promising alternative due to their inherent randomness, high density, and notable variation[138].

Memristor PUFs typically choose memristor pairs in an array or use crossbar arrays for implementation (Fig. 8a), exploiting randomness in conductance distribution[20,134,135,139], switching delay[140–143], and probabilistic switching[144–147] of memristors.

The randomness in conductance distributions is a crucial random source for memristor-based PUFs. Exploiting this characteristic, Nili et al. leveraged analog tuning and nonlinear conductance variations to realize a

Table 2 | Comparison of perturbators with different types of memristive devices

Memristive device	Year	Non-ideality used	Application	Scale (device number)	Integration approach	Operation type	Ref.
Oxide	2019	Thermal noise with extrinsic noise sources	Combinatorial optimization	400	0T1R 2D array	Hybrid	63
Oxide	2020	Write variation and nonlinearity	Combinatorial optimization	64	0T1R 2D array	Analog	56
Oxide	2020	Random telegraph noise	Combinatorial optimization	3600	1T1R 2D array	Hybrid	22
MTJ	2024	magnetization state random fluctuation	Combinatorial optimization	80	1T1MTJ cells	Hybrid	101
PCM	2024	Random walk noise	Combinatorial optimization	>10	6T2R array	Hybrid	102
Oxide	2019	Short-term memory, device-to-device variation	Reservoir computing	1024	0T1R array	Hybrid	23
Oxide	2022	Short-term memory, read nonlinearity	Reservoir computing	24 + 2048	Cells + 1T1R array	Analog	76
Oxide	2023	Short-term memory	Reservoir computing & NN	--	3D array	Hybrid	87
Oxide	2023	Dielectric breakdown voltage variation	Reservoir computing	256k	1T1R array	Hybrid	77
FD	2023	Short-term memory, read nonlinearity	Reservoir computing	7	Fe-diode cells	Hybrid	105
HP	2022	Short-term memory	Reservoir computing	25	0T1M cells	Hybrid	97

Oxide oxide-based resistive memory, *MTJ* magnetic tunnel junction, *FD* ferroelectric diodes, *PCM* phase-change memory, *HP* halide perovskite memristor, *NN* neural network.

dense, fast, and energy-efficient PUF[20]. Jiang et al. designed a unique physical fingerprint capable of provable key destruction (Fig. 8b)[148]. However, for PUFs, changes in operating conditions may increase the native bit error rate, requiring stability strategies like error correcting code, majority voting, and masking to avoid. To address this, Pang et al. proposed a split-resistance technique and enlarged the sensing window of compared cells, thereby generating stable PUF bits[139]. Recent research[134,135,149] has utilized randomly shaped conductive filaments formed after FORMING-RESET initialization as PUF features. By manipulating the more stable resident part of the conductive filament through RESET operations (Fig. 8c)[134], researchers have implemented reliable PUFs that can be hidden and restored as needed.

Switching delay characteristics offer another PUF generation method. PUFs based on memristor device pair's setting time mismatch show improved tolerance to process, voltage, temperature, and aging variations[141,143]. Cao et al. proposed reconfigurable PUF with 1T2R cells and achieved compact chip size and low bit error rate(Fig. 8d)[140]. More complex architectures, such as 2T2R PUF units with interlaced cell mirroring arrays[142], have been proposed to enhance randomness by mitigating system bias.

The probabilistic switching behavior caused by randomness of conductive filament formation and rupture[150] can also be used for PUF, with switching success rates related to applied voltage pulse characteristics[151]. Applying pulses with 50% switching probability, device related random patterns can be generated[144]. A notable aspect of this type of PUF is that it is implemented during the switching process, potentially allowing the corresponding device to retain its normal storage or other computational functions while generating the PUF. Building on this concept, Zhao et al. proposed a reconfigurable memristor PUF in the SET phase, enabling related memristor arrays to switch to normal storage function, and achieving a low bit error rate[145]. Additionally, Lin et al. stored PUF ID with the probabilistic switching characteristic of memristors under different forming conditions and implemented a TRNG based on the parity check of pulse counts required for writing LRS/HRS states in the meanwhile[147].

The non-ideal characteristics of memristors have provided an ideal source of randomness for PUF generation, thereby enabling energy-efficient designs. However, there still remains several key challenges for future researches: miniaturization[140,152], reliability enhancement[140,153], and the repurposing of PUF modules for their original storage or computational functions[140,149,154]. These investigations will contribute significantly to advancing the application of memristor-based PUFs in Internet of Things tasks.

Bayesian Neural Networks (BNNs)

BNNs based on memory devices, particularly filamentary memristors, offer a novel approach to neural network architecture. This section explores how memristors can be treated as Bayesian variables and programmed to match BNN probability distributions. BNNs incorporate Bayesian inference principles, providing probabilistic interpretations of model parameters. This approach offers several advantages, including uncertainty quantification in predictions and more informed decision-making based on prior knowledge and observed data[155]. Bayesian inference involves updating parameter probability distributions as new evidence becomes available. Traditional BNN implementations are computationally expensive and do not naturally align with memristor-based architectures.

Traditional deterministic neural networks and BNNs are different, especially in weight properties. Specifically, in a deterministic neural network model, the goal is to learn a set of weights that can accurately classify or predict the target variable based on the input data. This process can be formulated as an optimization problem, where the objective is to find the set of deterministic weights that maximize the likelihood of observing the given dataset, as shown in Fig. 9a. In contrast, BNNs offer an alternative perspective by calculating the posterior distribution for weights, which means unlike traditional neural networks that provide point estimates for parameters, the Bayesian approach considers the uncertainty associated with these parameters by modeling them as probability distributions[156] like Gaussian distribution or Laplace distribution (Fig. 9b).

Table 3 | Comparison of generators with different types of memristive devices

Memristive device	Year	Non-ideality used	Application	Scale (device number)	Integration approach	Multi-functional	Ref.
Oxide	2020	Switching asymmetry & nonlinearity	TRNG	4096	1T1R 2D array	No	117
Oxide	2021	Flicker noise	SC	128k	2T2R 2D array	No	125
Oxide	2022	Read cycle to cycle variation & device-to device variation	TRNG & PUF & NN	144k	2T2R 2D array	Yes	118
MTJ	2019	Interfacial magnetic anisotropy	TRNG & SC	8	1T1MTJ cells	Yes	173
MTJ	2024	Magnetization state random fluctuation	TRNG	1	1T1MTJ cells	No	174
FD	2023	Shot noise	TRNG & Bayesian machine	8192	3D array	Yes	177
FeFET	2024	Polarization domain fluctuation	TRNG & HE	100	FeFET 2D array	Yes	176
Oxide	2018	Read nonlinearity	PUF	200	0T1R 3D array	No	20
Oxide	2022	Write variation	PUF	8192	1T1R 2D array	No	134
Oxide	2024	Switching delay variation	PUF	1M	1T2R 2D array	No	140
MTJ	2023	Switching delay variation	PUF & NN	6.6 M	1T1MTJ 2D array	Yes	179
FeFET	2025	Random polarization switching	PUF	63	2FeFET1C 2D array	No	180
Oxide	2021	Write variation	MCMC sampling & BNN	16k	1T1R 2D array	Yes	162
Oxide	2023	Read cycle-to-cycle variation	BNN	4k	1T1R 2D array	No	160
Oxide	2023	Write variation	BNN	1024	1T1R 2D array	No	161

Oxide oxide-based resistive memory, *MTJ* magnetic tunnel junction, *FD* ferroelectric diodes, *FeFET* ferroelectric field-effect transistors, *PCM* phase-change memory, *HP* halide perovskite memristor, *NN* neural network.

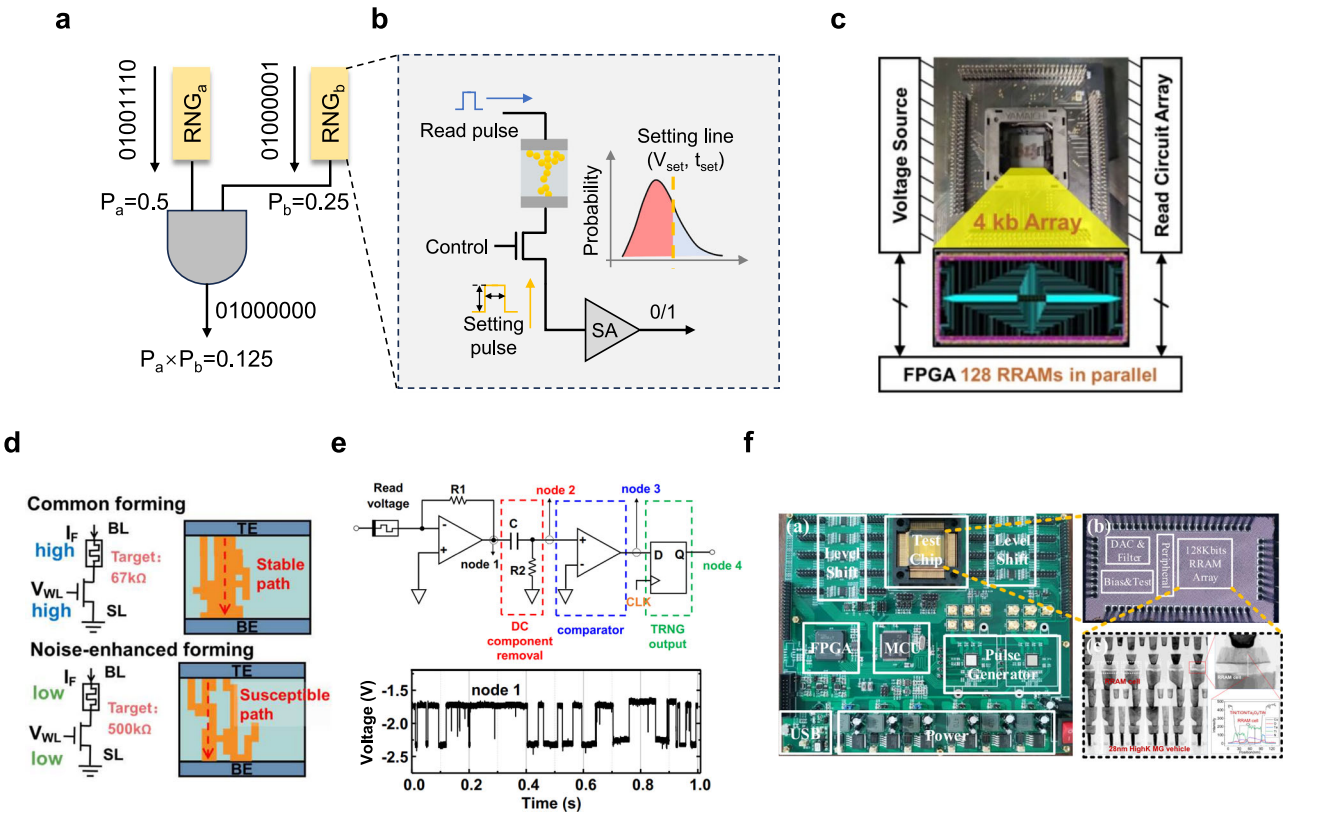

Fig. 7 | Memristor noise-based generative SC and TRNG designs. a Bit stream with a specific probability generating product output through an AND gate. **b** Random number generator with controlled probability of 0 and 1 through specific pulse voltage and duration settings. **c** System schematic for stable noise capturing[117]. **d** Noise-enhanced forming for more detectable noise[118]. **e** Schematic of TRNG based on RTN of memristors (top) and the captured amplified RTN noise (bottom)[124]. **f** Platform constructed for TRNG with flicker noise on memristors[125]. Panel **c** is reproduced with the permission from ref. 117, copyright (IEEE, 2020). Panel **d** is reproduced with permission from ref. 118, copyright (IEEE, 2022). Panel **e** is reproduced with permission from ref. 124, copyright (Wiley, 2023). Panel **f** is reproduced with permission from ref. 125, copyright (IEEE, 2021).

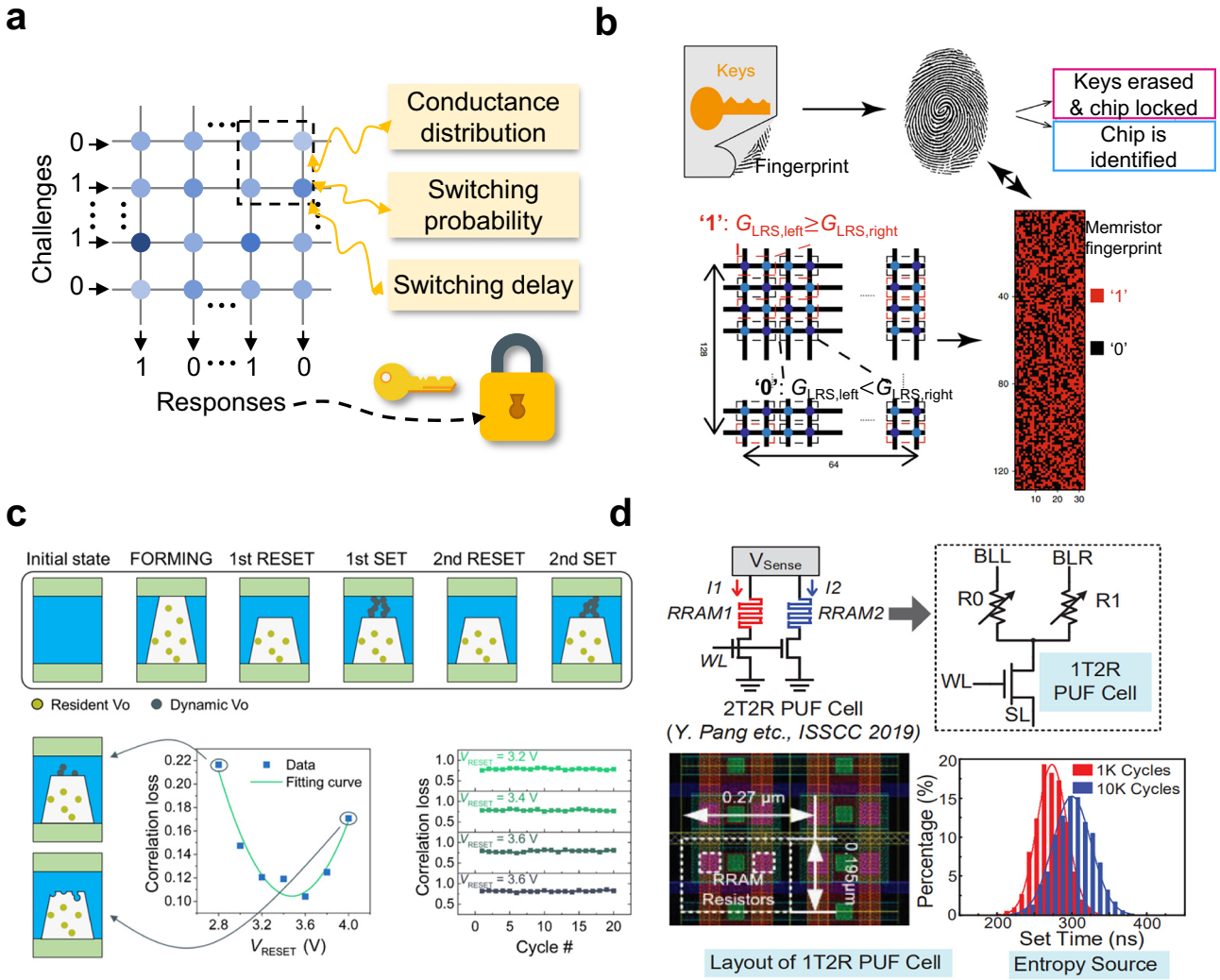

Fig. 8 | Memristor-based PUFs. a PUF workflow and crossbar structure for generation. Three main factors in crossbar are used for PUF generation. **b** Provable key destruction based on memristor array[148]. **c** Correlated PUF generation with filamentary switching[134]. **d** Compact and reconfigurable PUF designed with 1T2R cells[140]. Panel **b** is reproduced with permission from ref. 148, copyright (Springer Nature, 2018). Panel **c** is reproduced with permission from ref. 134, copyright (AAAS, 2022). Panel **d** is reproduced with permission from ref. 140, copyright (IEEE, 2024).

When deployed on memristor arrays, these two kinds of networks also show different behaviors due to inherent memristor noises. In the case of memristor-based deterministic neural networks, the high-precision weights of a pre-trained model are transferred to memristor conductance. The stored weights in the memristors may deviate from the intended values due to uncertainties like process variation and temperature variation, which inevitably impact the accuracy of inference and introduce computational instability. By contrast, in the case of memristor-based BNNs[157], representing parameters as probability distributions is valuable in the calculation of confidence intervals and probabilistic predictions, which leads to the fact that the inference accuracy degradation is minimized to the greatest extent possible. These behaviors are illustrated in Fig. 9c.

Recent research has focused on implementing Bayesian inference using memristor crossbar arrays to address robustness challenges in neural network accelerators. Gao et al.[158] introduced a unified Bayesian inference framework that integrates memristor variation models into algorithmic training, optimizing robustness through BNN training. This approach effectively mitigates accuracy degradation by approximately 50–60%, although it lacks verification with actual circuits and devices. Lin et al.[156] presented a method leveraging the intrinsic random noise of analog memristors. An experimental implementation of BNN on a 160 K memristor array demonstrated 97% accuracy in MNIST image classification and exhibited robustness against adversarial attacks, which is a very important application example for memristor-based BNNs, as shown in Fig. 9d. Another study[159] introduced a novel architecture for a Bayesian machine using real memristors, addressing energy efficiency and computational challenges in Bayesian inference. The prototype circuit, integrating 2048 memristors and 30,080 transistors, demonstrated significant energy efficiency improvements in gesture recognition compared to standard implementations on microcontrollers. Recent work[160] developed a stochastic CIM system efficiently performing in-situ random number generation and computation. Applied to a risk-sensitive reinforcement learning task, it achieved 10 times higher speed and 150 times higher energy efficiency in uncertainty decomposition compared to conventional digital computers. Complementing this, research[161] focused on implementing memristor-based BNNs to enhance uncertainty quantification in safety-critical applications, demonstrating high accuracy in arrhythmia classification and improved inference energy efficiency. In addition, Dalgaty, et al. implemented in-memory Markov Chain Monte Carlo sampling from set operations on memristor arrays and built BNNs for both supervised learning and reinforcement learning with excellent robustness[162]. Recent research has also highlighted the promising potential of BNNs leveraging memristors' inherent read noise characteristics for active learning

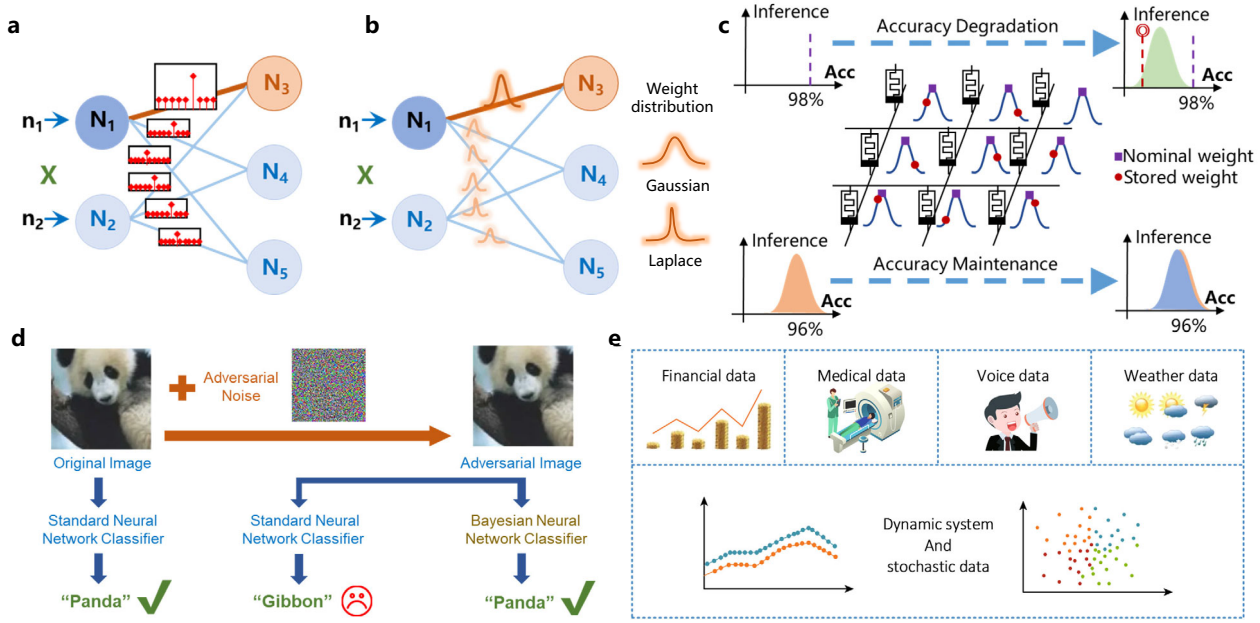

Fig. 9 | Generation of random distributions using memristors for BNNs. **a** Traditional deterministic neural network with fixed-value network weights. **b** BNNs with probability distributions being network weights[156]. **c** Weight deployments of convolution neural networks with accuracy degradation and BNNs with accuracy maintenance on memristor arrays[158]. **d** An example of using memristor-based BNN for adversarial robustness[156]. **e** Applications of memristor-based BNNs in various types of data processing[160]. Panels **b**, **d** are reproduced with permission from ref. 156, copyright (IEEE, 2019). Panel **c** is reproduced with permission from ref. 158, copyright (IEEE, 2021). Panel **e** is reproduced with permission from ref. 160, copyright (Springer Nature, 2023).

applications, offering an elegant approach to potentially reduce network training costs in resource-constrained scenarios[163].

These advancements demonstrate the potential of memristor-based BNNs, especially in edge computing and safety-critical applications. The unique noise properties of memristors enable efficient probabilistic computations, reducing computational burden while enhancing prediction reliability. Specifically, firstly, memristors provide a compact and energy-efficient way of storing and processing data with CIM capabilities. This is critical for edge applications where resources are limited. Secondly, by harnessing the inherent imperfections, the memristor-based BNNs increase the overall hardware resilience of the system. Future research may focus on scaling implementations and exploring new strategies and applications with memristor BNNs. The innovative BNN algorithms, on the one hand, are expected to be deeply developed to support a wider range of real-world dynamic applications (Fig. 9e), such as self-driving scenes[164,165], automated financial trading[164,165], medical diagnoses[166–168], and processing of voice[169,170] or weather data[171,172], etc. On the other hand, the development of efficient programming and readout techniques is crucial for the practical implementation of these networks, so BNN-oriented write-read circuits are worth studying.

Applications of other memristive devices working as generators

Besides oxide-based resistive memory, other types of memristors can also serve as effective generators.

Spintronic devices have frequently been used in recent years to generate probabilistic bits, forming a comparison with the functionality of quantum circuits. Borders et al. enhanced interfacial magnetic anisotropy by changing the free-layer thickness and junction diameters of MTJs[173] (Fig. 10a), enabling them to spontaneously produce random fluctuations between parallel and anti-parallel states with appropriate retention times, thereby achieving efficient integer factorization. Subsequently, Daniel et al. designed magnetic tunnel junctions with low energy barrier in its free layer[174] (Fig. 10b) and achieved tunable randomness for probabilistic bit generation through voltage control. Additionally, Piccinini et al. utilized the variability of switching voltages of self-heating PCM and demonstrated the TRNG[175], however this approach needs to address challenges related to potential system biases and device endurance limitations under repeated random number generation cycles. Shao et al. programmed the FeFETs to intermediate polarization states and generated random numbers with fast gate perturbations (Fig. 10c), which implemented high-quality TRNG and first demonstrated high-throughput homomorphic encryption on FeFET arrays[176]. Gong et al. utilized shot noise in Fe-diodes to implement low-cost TRNG while using 3D integration to create a Bayesian machine that achieved good classification results on MNIST[177].

Adel et al. (Fig. 10d) designed a circuit featuring a grid-like structure with multiple arrays of MTJs and implemented a PUF by comparing the resistance values of cells along different paths[178]. This design increases the CRP space which can demonstrate resistance against deep learning modeling attacks, but this work is limited to simulation and may suffer challenges in scalability due to the increased architectural complexity.

Multifunctional, reconfigurable designs that incorporate PUFs have also become a trend in works using other memristive devices. Chiu et al. designed a spintronic non-volatile CIM macro for edge AI with secure access control[179] utilizing the differential switching latency of the twin-cell structure with STT-MRAM and implemented a highly reliable PUF (Fig. 10e). Li et al. introduced a PUF with cycle-to-cycle variation by partially switching the FE layer polarization[180] (Fig. 10f). Notably, compared to workflows that use device-to-device variations, this work enables CRP updates through reprogramming, effectively expanding the application range in IoT scenarios, though it may also raise concerns about device lifetime.

Table 3 presents an overview of generators based on different memristive devices. PCM-based implementations, while promising with their excellent integration density, appear relatively less frequently in recent literature, potentially due to challenges associated with energy consumption and delay of phase change process[7,181]. In these applications, studies with oxide-based resistive memory have reported good throughput (especially in TRNG and SC)[117,118], while spintronic and ferroelectric devices have demonstrated excellent energy efficiency and integration density[173,177,179,180]. Overall, memristor-based generator chips

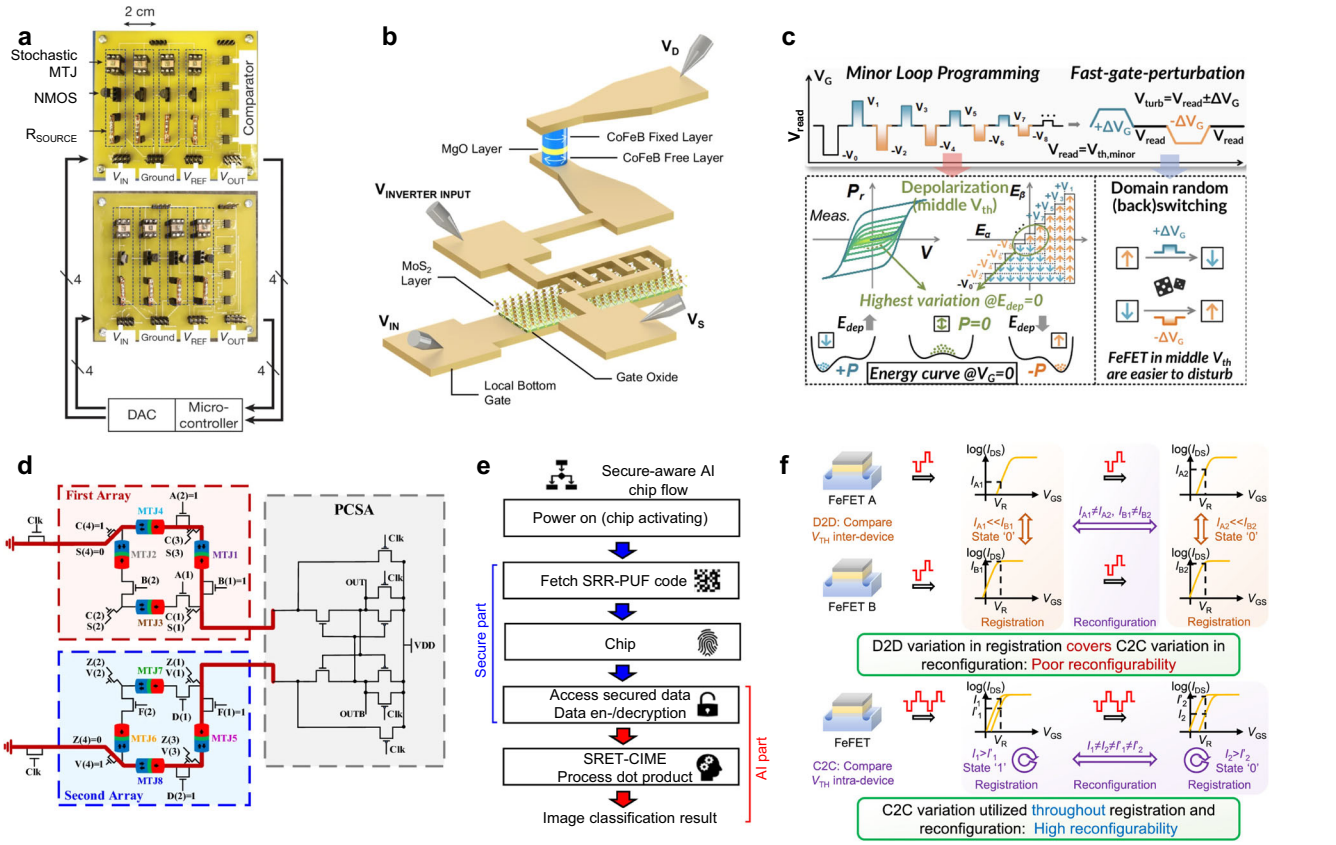

Fig. 10 | Demonstrations with other memristive devices as generators. **a** Probabilistic-bit generation on magnetic tunnel junctions for integer factorization[173]. **b** Compact implementation of on-chip probabilistic-bit generation with MTJ[174]. **c** Programing FeFETs to the intermediate polarization state for high throughput TRNG[176]. **d** A grid-like circuit designed for spintronic PUF with increases the challenge-response pair (CRP) space[178]. **e** A reconfigurable PUF with STT-MRAM designed for safe AI[179]. **f** A FeFET-based highly reconfigurable PUF with cycle-to-cycle variation[180]. Panel **a** is reproduced with permission from ref. 173, copyright (Springer Nature, 2019). Panel **b** is reproduced with permission from ref. 174, copyright (Springer Nature, 2024). Panel **c** is reproduced with permission from ref. 176, copyright (IEEE, 2024). Panel **d** is reproduced with permission from ref. 178, copyright (Springer Nature, 2024). Panel **e** is reproduced with permission from ref. 179, (Springer Nature, 2023). Panel **f** is reproduced with permission from ref. 180, copyright (Springer Nature, 2025).

are evolving toward multifunctionality[118,173,177,179], with improved integration density (3D integration) and integration scale emerging as common development themes across different device technologies.

Discussion and outlook

In summary, the reported techniques aim to harness the non-ideal characteristics of memristors for advanced computational functionalities, while mitigating their potential negative impacts. The integration of memristors as perturbators and generators in computational systems offers advantages in both energy efficiency and hardware overhead. However, several challenges remain in effectively utilizing these non-ideal characteristics for practical applications.

Current approaches to exploiting non-ideal device characteristics are predominantly founded on statistical principles. Consequently, harnessing less dominant or difficult-to-monitor non-ideal characteristics (such as thermal noise in oxide-based resistive memory) presents greater challenges compared to more prominent non-idealities (such as device-to-device switching voltage/delay variations, RTN in oxide-based resistive memory, interfacial magnetic anisotropy in spintronic devices, etc.). In certain research efforts, investigators leverage device operating mechanisms or specialized designs to amplify specific non-ideal characteristics under controlled conditions[125,174,176]. While these approaches facilitate easier capture of non-idealities, they simultaneously compromise numerical storage precision or even eliminate storage capabilities altogether, potentially limiting the multifunctional applications of these devices. Moreover, the accurate capture and utilization of these non-ideal characteristics often necessitates additional peripheral circuitry, creating significant challenges for designs requiring high throughput, computational density, and energy efficiency.

Furthermore, challenges also manifest differently depending on the specific application of non-ideal characteristics. For perturbators based on non-idealities, although these characteristics enhance the dynamic properties of the system and facilitate solving processes for specific computations (such as combinatorial optimization and reservoir computing), they inevitably compromise solution stability and robustness. This forces researchers to make critical trade-offs between computational speed and result accuracy. In the context of generators, frequent switching operations and the pursuit of high throughput pose challenges to the lifetime and operating environment of memristors (such as oxide-based resistive memory and HP memristors). Moreover, inherent device characteristics substantially impact the utilization of non-ideal properties. For instance, PCM's relatively high programming power consumption and extended programming times create potential limitations when employed as generators. Similarly, the limited integration scale of emerging memristors (such as FeFET and HP memristors) restricts their potential application domains.

To fully realize the potential of memristor noises in real-world computing tasks, several specific approaches need to be developed. A fundamental requirement is advancing our understanding of memristor noise mechanisms[39], as deeper insights would enhance the reliability of noise-enhanced techniques. For effective implementation, different application scenarios demand tailored strategies. With memristor-based perturbators, implementing precise regulation systems alongside streamlined peripheral circuits or advanced post-processing techniques can enhance operational

stability[22,76]. Memristor-based generators require streamlined and efficient peripheral circuits to collect high-quality noise-driven randomness while ensuring high throughput[140,160]. For noise-sensitive modules, redundant mapping and hybrid architectures can be employed to reduce noise interference on computational accuracy[6,18,182]. Complementing these approaches, the development of efficient programming and readout techniques represents a critical engineering advancement necessary for practical system implementation and optimal performance in complex computing environments[183,184].

Future research directions may include: (1) a deeper understanding of memristor physics and behavior to better model and control their non-ideal characteristics, (2) exploration of more advanced device operating mechanisms and integration approaches to enhance device performance and integration density, enabling chips to handle increasingly complex tasks, (3) development of compatible computational architectures and streamlined calibration techniques that leverage non-ideal characteristics for specific functions without sacrificing the advantages of non-volatility, and (4) investigation of hybrid systems combining memristors with other emerging technologies to create more robust and versatile computational platforms.

To conclude, leveraging memristor noises, which are generally viewed as non-idealities, for computational functionalities represents a promising direction for developing energy-efficient and resilient computing methods. This approach positions us to create algorithms and hardware that are not only computationally efficient but also capable of providing reliable and explainable decisions. As research progresses, the integration of memristor-based systems with appropriate noise utilization in real-world applications may lead to significant advancements in edge computing, artificial intelligence, and beyond.

Received: 15 October 2024; Accepted: 1 July 2025;
Published online: 14 July 2025

References

1. Mii, Y. J. Semiconductor innovations, from device to system. in *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. 276-281.
2. Horowitz, M. 1.1 Computing's energy problem (and what we can do about it). in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. 10-14.
3. Chua, L. Memristor-the missing circuit element. *IEEE Trans. Circuit Theory* **18**, 507–519 (1971).
4. Strukov, D. B., Snider, G. S., Stewart, D. R. & Williams, R. S. The missing memristor found. *Nature* **453**, 80–83 (2008). **This paper reports an example of memristors.**
5. Shi, Y. et al. Electronic synapses made of layered two-dimensional materials. *Nat. Electron.* **1**, 458–465 (2018).
6. Zidan, M. A., Strachan, J. P. & Lu, W. D. The future of electronics based on memristive systems. *Nat. Electron.* **1**, 22–29 (2018).
7. Lanza, M. et al. Memristive technologies for data storage, computation, encryption, and radio-frequency communication. *Science* **376**, eabj9979 (2022).
8. Yao, P. et al. Fully hardware-implemented memristor convolutional neural network. *Nature* **577**, 641–646 (2020).
9. Ielmini, D. Modeling the universal set/reset characteristics of bipolar RRAM bY Field- And Temperature-driven Filament. *Growth IEEE Trans. Electron Dev.* **58**, 4309–4317 (2011).
10. Zhang, W. et al. Neuro-inspired computing chips. *Nat. Electron.* **3**, 371–382 (2020).
11. Ielmini, D. & Wong, H. S. P. In-memory computing with resistive switching devices. *Nat. Electron.* **1**, 333–343 (2018).
12. Niu, D., Chen, Y., Xu, C. & Xie, Y. Impact of process variations on emerging memristor. in *Design Automation Conference*. 877-882.
13. Yu, S. *Resistive random access memory (RRAM)*. (Morgan & Claypool Publishers, 2016).
14. Zahoor, F., Azni Zulkifli, T. Z. & Khanday, F. A. Resistive Random Access Memory (RRAM): an overview of materials, switching mechanism, performance, multilevel cell (mlc) storage, modeling, and applications. *Nanoscale Res. Lett.* **15**, 90 (2020).
15. Zhao, H. et al. Energy-efficient high-fidelity image reconstruction with memristor arrays for medical diagnosis. *Nat. Commun.* **14**, 2276 (2023).
16. Song, W. et al. Programming memristor arrays with arbitrarily high precision for analog computing. *Science* **383**, 903–910 (2024).
17. Liu, B. et al. in *Proceedings of the 52nd Annual Design Automation Conference* Article 15 (Association for Computing Machinery, San Francisco, California, 2015).
18. Le Gallo, M. et al. Mixed-precision in-memory computing. *Nat. Electron.* **1**, 246–253 (2018).
19. Li, J. et al. Sparse matrix multiplication in a record-low power self-rectifying memristor array for scientific computing. *Sci. Adv.* **9**, eadf7474 (2023).
20. Nili, H. et al. Hardware-intrinsic security primitives enabled by analogue state and nonlinear conductance variations in integrated memristors. *Nat. Electron.* **1**, 197–202 (2018). **This work reports an early demonstration of memristive PUF with vertically integrated arrays.**
21. Du, Y. et al. Synaptic 1/f noise injection for overfitting suppression in hardware neural networks. *Neuromorphic Comput. Eng.* **2**, 034006 (2022).
22. Cai, F. X. et al. Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks. *Nat. Electron.* **3**, 409–418 (2020). **This paper reports an early demonstration of combinatorial optimization with memristor array.**
23. Moon, J. et al. Temporal data classification and forecasting using a memristor-based reservoir computing system. *Nat. Electron.* **2**, 480–487 (2019).
24. Mohseni, N., McMahon, P. L. & Byrnes, T. Ising machines as hardware solvers of combinatorial optimization problems. *Nat. Rev. Phys.* **4**, 363–379 (2022).
25. Liang, X. et al. Physical reservoir computing with emerging electronics. *Nat. Electron.* **7**, 193–206 (2024).
26. Lin, N. et al. In-memory and in-sensor reservoir computing with memristive devices. *APL Mach. Learn.* **2**, 010901 (2024).
27. Pang, Y., Gao, B., Lin, B., Qian, H. & Wu, H. Memristors for hardware security applications. *Adv. Electron Mater.* **5**, 1800872 (2019).
28. Carboni, R. & Ielmini, D. Stochastic memory devices for security and computing. *Adv. Electron Mater.* **5**, 1900198 (2019).
29. Nikolay, F. in *Memristors* (ed Chang Yao-Feng) Ch. 9 (IntechOpen, 2023).
30. Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R. & Eleftheriou, E. Memory devices and applications for in-memory computing. *Nat. Nanotechnol.* **15**, 529–544 (2020).
31. Nardone, M., Kozub, V., Karpov, I. & Karpov, V. Possible mechanisms for 1/f noise in chalcogenide glasses: a theoretical description. *Phys. Rev. B—Condens. Matter Mater. Phys.* **79**, 165206 (2009).
32. Islam, S., Shamim, S. & Ghosh, A. Benchmarking noise and dephasing in emerging electrical materials for quantum technologies. *Adv. Mater.* **35**, 2109671 (2023).
33. Yu, S., Jeyasingh, R., Yi, W. & Wong, H. S. P. Understanding the conduction and switching mechanism of metal oxide RRAM through low frequency noise and AC conductance measurement and analysis. in *2011 International Electron Devices Meeting*. 12.11.11-12.11.14.
34. Dutta, P. & Horn, P. Low-frequency fluctuations in solids: 1 f noise. *Rev. Mod. Phys.* **53**, 497 (1981).
35. Blanter, Y. M. & Büttiker, M. Shot noise in mesoscopic conductors. *Phys. Rep.* **336**, 1–166 (2000).

36. Kobayashi, K. & Hashisaka, M. Shot noise in mesoscopic systems: from single particles to quantum liquids. *J. Phys. Soc. Jpn.* **90**, 102001 (2021).
37. Nyquist, H. Thermal agitation of electric charge in conductors. *Phys. Rev.* **32**, 110 (1928).
38. Balatti, S. et al. Voltage-dependent random telegraph noise (RTN) in HfOx resistive RAM. in *2014 IEEE International Reliability Physics Symposium*. MY.4.1-MY.4.6.
39. Puglisi, F. M., Larcher, L., Padovani, A. & Pavan, P. A complete statistical investigation of RTN in HfO2-based RRAM in high resistive state. *IEEE Trans. Electron Dev.* **62**, 2606–2613 (2015).
40. Veksler, D. et al. Random telegraph noise (RTN) in scaled RRAM devices. in *2013 IEEE International Reliability Physics Symposium (IRPS)*. MY.10.11-MY.10.14.
41. Ambrogio, S. et al. Statistical fluctuations in HfOx resistive-switching. *Mem. Part II—Random Telegr. Noise. IEEE Trans. Electron Dev.* **61**, 2920–2927 (2014).
42. Yi, W. et al. Quantized conductance coincides with state instability and excess noise in tantalum oxide memristors. *Nat. Commun.* **7**, 11142 (2016).
43. Ackley, D. H., Hinton, G. E. & Sejnowski, T. J. A learning algorithm for boltzmann machines. *Cogn. Sci.* **9**, 147–169 (1985).
44. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **79**, 2554–2558 (1982).
45. Korte, B. H., Vygen, J., Korte, B. & Vygen, J. *Combinatorial Optimization*. Vol. 1 (Springer, 2011).
46. Blum, C. & Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv. ((CSUR))* **35**, 268–308 (2003).
47. Naseri, G. & Koffas, M. A. Application of combinatorial optimization strategies in synthetic biology. *Nat. Commun.* **11**, 2446 (2020).
48. Ulungu, E. L. & Teghem, J. Multi-objective combinatorial optimization problems: a survey. *J. Multi-Criteria Decis. Anal.* **3**, 83–104 (1994).
49. Srikakulapu, R. & Vinatha, U. Optimized design of collector topology for offshore wind farm based on ant colony optimization with multiple travelling salesman problem. *J. Mod. Power Syst. Clean. Energy* **6**, 1181–1192 (2018).
50. Ahmad, M., Mittal, N., Garg, P. & Khan, M. M. Efficient cryptographic substitution box design using travelling salesman problem and chaos. *Perspect. Sci.* **8**, 465–468 (2016).
51. Alam, M., Saki, A. A. & Ghosh, S. An efficient circuit compilation flow for quantum approximate optimization algorithm. in *2020 57th ACM/IEEE Design Automation Conference (DAC)*. 1-6.
52. Khan, S., Li, K. F., Manning, E. G. & Akbar, M. M. Solving the knapsack problem for adaptive multimedia systems. *Stud. Inform. Univ.* **2**, 157–178 (2002).
53. Qu, G. & Potkonjak, M. Analysis of watermarking techniques for graph coloring problem. in *Proceedings of the 1998 IEEE/ACM international conference on Computer-aided design*. 190-193.
54. Sharafi, P., Teh, L. H. & Hadi, M. N. Conceptual design optimization of rectilinear building frames: a knapsack problem approach. *Eng. Optim.* **47**, 1303–1323 (2015).
55. Lahami, M., Krichen, M., Bouchakwa, M. & Jmaiel, M. Using knapsack problem model to design a resource aware test architecture for adaptable and distributed systems. in *Testing Software and Systems: 24th IFIP WG 6.1 International Conference, ICTSS 2012, Aalborg, Denmark, November 19-21, 2012. Proceedings 24*. 103-118 (Springer).
56. Yang, K. et al. Transiently chaotic simulated annealing based on intrinsic nonlinearity of memristors for efficient solution of optimization problems. *Sci. Adv.* **6**, eaba9901 (2020).
57. Jiang, M. R., Shan, K. Y., He, C. P. & Li, C. Efficient combinatorial optimization by quantum-inspired parallel annealing in analogue memristor crossbar. *Nat. Commun.* **14**, 5927 (2023).
58. Taoka, K., Misawa, N., Koshino, S., Matsui, C. & Takeuchi, K. Simulated Annealing Algorithm & ReRAM Device Co-optimization for Computation-in-Memory. *2021 IEEE International Memory Workshop (Imw)*, 119-122 (2021).
59. Misawa, N., Taoka, K., Matsui, C. & Takeuchi, K. Domain specific ReRAM computation-in-memory design considering bit precision and memory errors for simulated annealing. *IEEE Int. Symp. Circ. S*, 3289-3293 (2022).
60. Bojnordi, M. N. & Ipek, E. Memristive boltzmann machine: a hardware accelerator for combinatorial optimization and deep learning. *Int. S High Perf Comp*. 1-13 (2016).
61. Matsumoto, S., Gyoten, H., Hiromoto, M. & Sato, T. RRAM/CMOS-hybrid architecture of annealing processor for fully connected ising model. *2018 IEEE 10th International Memory Workshop (Imw)*, 70-73 (2018).
62. Yue, W. et al. A scalable universal Ising machine based on interaction-centric storage and compute-in-memory. *Nat. Electron.* **7**, 904–913 (2024).
63. Mahmoodi, M. R., Prezioso, M. & Strukov, D. B. Versatile stochastic dot product circuits based on nonvolatile memories for high performance neurocomputing and neurooptimization. *Nat. Commun.* **10**, 5113 (2019).
64. Yi, S. I., Kumar, S. & Williams, R. S. Combinatorial optimization in hopfield networks with noise and diagonal perturbations. in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. 321-325.
65. Chiang, H. W., Nien, C. F., Cheng, H. Y. & Huang, K. P. ReAIM: A ReRAM-based adaptive ising machine for solving combinatorial optimization problems. in *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*. 58-72.
66. Medsker, L. R. & Jain, L. Recurrent neural networks. *Des. Appl.* **5**, 2 (2001).
67. Hochreiter, S. Long short-term memory. *Neural Computation MIT-Press* (1997).
68. Kates-Harbeck, J., Svyatkovskiy, A. & Tang, W. Predicting disruptive instabilities in controlled fusion plasmas through deep learning. *Nature* **568**, 526–531 (2019).
69. Lappalainen, J. K. et al. Connectome-constrained networks predict neural activity across the fly visual system. *Nature* **634**, 1132–1140 (2024).
70. Tanaka, G. et al. Recent advances in physical reservoir computing: a review. *Neural Netw.* **115**, 100–123 (2019).
71. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149 (2009).
72. Gao, C. et al. Toward grouped-reservoir computing: organic neuromorphic vertical transistor with distributed reservoir states for efficient recognition and prediction. *Nat. Commun.* **15**, 740 (2024).
73. Yaremkevich, D. D. et al. On-chip phonon-magnon reservoir for neuromorphic computing. *Nat. Commun.* **14**, 8296 (2023).
74. Gao, C., Gaur, P., Almutairi, D., Rubin, S. & Fainman, Y. Optofluidic memory and self-induced nonlinear optical phase change for reservoir computing in silicon photonics. *Nat. Commun.* **14**, 4421 (2023).
75. Chen, R. et al. Thin-film transistor for temporal self-adaptive reservoir computing with closed-loop architecture. *Sci. Adv.* **10**, eadl1299.
76. Zhong, Y. N. et al. A memristor-based analogue reservoir computing system for real-time and power-efficient signal processing. *Nat. Electron.* **5**, 672–681 (2022). **This paper reports a fully analog reservoir computing system with both dynamic and non-volatile memristors.**

77. Wang, S. et al. Echo state graph neural networks with analogue random resistive memory arrays. *Nat. Mach. Intell.* **5**, 104–113 (2023).
78. Du, C. et al. Reservoir computing using dynamic memristors for temporal information processing. *Nat. Commun.* **8**, 2204 (2017). **This work reports an early demonstration of memristive reservoir computing.**
79. Marinella, M. J. & Agarwal, S. Efficient reservoir computing with memristors. *Nat. Electron.* **2**, 437–438 (2019).
80. Sun, L. F. et al. In-sensor reservoir computing for language learning via two-dimensional memristors. *Sci. Adv*. **7**, eabg1455 (2021).
81. Zhu, X., Wang, Q. & Lu, W. D. Memristor networks for real-time neural activity analysis. *Nat. Commun.* **11**, 2439 (2020).
82. Midya, R. et al. Reservoir computing using diffusive memristors. *Adv. Intell. Syst.-Ger.* **1**, 1900084 (2019).
83. Soriano, M. C. et al. Delay-based reservoir computing: noise effects in a combined analog and digital implementation. *IEEE Trans. Neural Netw. Learn. Syst.* **26**, 388–393 (2015).
84. Du, C., Ma, W., Chang, T., Sheridan, P. & Lu, W. D. Biorealistic implementation of synaptic functions with oxide memristors through internal ionic dynamics. *Adv. Funct. Mater.* **25**, 4290–4299 (2015).
85. Ohno, T. et al. Short-term plasticity and long-term potentiation mimicked in single inorganic synapses. *Nat. Mater.* **10**, 591–595 (2011).
86. Zhong, Y. et al. Dynamic memristor-based reservoir computing for high-efficiency temporal signal processing. *Nat. Commun.* **12**, 408 (2021).
87. Sun, W. et al. High area efficiency (6 TOPS/mm2) multimodal neuromorphic computing system implemented by 3D multifunctional RRAM array. in *2023 International Electron Devices Meeting (IEDM)*. 1-4.
88. Jaeger, H. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *Bonn. Ger. Ger. Natl. Res. Cent. Inf. Technol. GMD Tech. Rep.* **148**, 13 (2001).
89. Zhongrui Wang, M. X. et al. Efficient modelling of ionic and electronic interactions by resistive memory-based reservoir graph neural network. *PREPRINT (Version 1) available at Research Square* (2024).
90. Wang, X. & Li, H. Reservoir computing with a random memristor crossbar array. *Nanotechnology* **35**, 415205 (2024).
91. Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531–2560 (2002).
92. Payvand, M. et al. Self-organization of an inhomogeneous memristive hardware for sequence learning. *Nat. Commun.* **13**, 5793 (2022).
93. Soures, N., Hays, L. & Kudithipudi, D. Robustness of a memristor based liquid state machine. in *2017 International Joint Conference on Neural Networks (IJCNN)*. 2414-2420.
94. Sayyaparaju, S., Shawkat, M. S. A., Adnan, M. M. & Rose, G. S. Circuit techniques for efficient implementation of memristor based reservoir computing. in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. 1-5.
95. Lin, N. et al. Resistive memory-based zero-shot liquid state machine for multimodal event data learning. *Nat. Comput. Sci.* **5**, 37–47 (2025).
96. Yoo, S. et al. Efficient data processing using tunable entropy-stabilized oxide memristors. *Nat. Electron.* **7**, 466–474 (2024).
97. John, R. A. et al. Reconfigurable halide perovskite nanocrystal memristors for neuromorphic computing. *Nat. Commun.* **13**, 2074 (2022).
98. Bae, J. et al. Tunable ion energy barrier modulation through aliovalent halide doping for reliable and dynamic memristive neuromorphic systems. *Sci. Adv.* **10**, eadm7221 (2024).
99. Choi, S. et al. 3D-integrated multilayered physical reservoir array for learning and forecasting time-series information. *Nat. Commun.* **15**, 2044 (2024).
100. Liang, X. et al. Rotating neurons for all-analog implementation of cyclic reservoir computing. *Nat. Commun.* **13**, 1549 (2022).
101. Si, J. et al. Energy-efficient superparamagnetic Ising machine and its application to traveling salesman problems. *Nat. Commun.* **15**, 3457 (2024).
102. Kang, Y. G. et al. Solving max-cut problem using spiking boltzmann machine based on neuromorphic hardware with phase change memory. *Adv. Sci.* **11**, 2406433 (2024).
103. Toprasertpong, K. et al. Reservoir computing on a silicon platform with a ferroelectric field-effect transistor. *Commun. Eng.* **1**, 21 (2022).
104. Nako, E., Toprasertpong, K., Nakane, R., Takenaka, M. & Takagi, S. Reservoir computing system with HZO/Si FeFETs in parallel configuration: experimental demonstration of speech classification. *IEEE Trans. Electron Dev.* **70**, 5657–5664 (2023).
105. Chen, Z. et al. All-ferroelectric implementation of reservoir computing. *Nat. Commun.* **14**, 3585 (2023).
106. Taniguchi, T., Ogihara, A., Utsumi, Y. & Tsunegi, S. Spintronic reservoir computing without driving current or magnetic field. *Sci. Rep.-Uk* **12**, 10627 (2022).
107. Soucarros, M., Clédière, J. & Dumas, C. & Elbaz-Vincent, P. Fault analysis and evaluation of a true random number generator embedded in a processor. *J. Electron. Test.* **29**, 367–381 (2013).
108. Nguyen, N., Kaddoum, G., Pareschi, F., Rovatti, R. & Setti, G. A fully CMOS true random number generator based on hidden attractor hyperchaotic system. *Nonlinear Dyn.* **102**, 2887–2904 (2020).
109. Gaines, B. R. in *Advances in Information Systems Science: Volume 2* (ed Julius T. T.) 37-172 (Springer US, 1969).
110. Jo, S. H., Kim, K.-H. & Lu, W. Programmable resistance switching in nanoscale two-terminal devices. *Nano Lett.* **9**, 496–500 (2009).
111. Knag, P., Lu, W. & Zhang, Z. A native stochastic computing architecture enabled by memristors. *IEEE Trans. Nanotechnol.* **13**, 283–293 (2014).
112. Zhao, Y. et al. A physics-based model of RRAM probabilistic switching for generating stable and accurate stochastic bit-streams. in *2019 IEEE International Electron Devices Meeting (IEDM)*. 32.34.31-32.34.31.
113. Gupta, S. et al. SCRIMP: a general stochastic computing architecture using ReRAM in-memory processing. in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 1598-1601.
114. Salehi, S. A. Low-cost stochastic number generators for stochastic computing. *IEEE Trans. VLSI Syst.* **28**, 992–1001 (2020).
115. Mathew, S. K. et al. 2.4 Gbps, 7 mW all-digital PVT-variation tolerant true random number generator for 45 nm CMOS high-performance microprocessors. *IEEE J. Solid-State Circuits* **47**, 2807–2821 (2012).
116. Yu, F. et al. A survey on true random number generators based on chaos. *Discret. Dyn. Nat. Soc.* **2019**, 2545123 (2019).
117. Lin, B. H. et al. A high-performance and calibration-free true random number generator based on the resistance perturbation in RRAM array. *2020 IEEE International Electron Devices Meeting (Iedm)* (2020).
118. Li, X. Q. et al. First demonstration of homomorphic encryption using multi-functional RRAM arrays with a novel noise-modulation scheme. *Int. Elect. Devices Meet* 33.35.31–33.35.34 (2022).
119. Equbal, M. S., Ketkar, T. & Sahay, S. Hybrid CMOS-RRAM true random number generator exploiting coupled entropy sources. *IEEE Trans. Electron Dev.* **70**, 1061–1066 (2023).
120. Wen, C. et al. Advanced data encryption using 2D materials. *Adv. Mater.* **33**, 2100185 (2021).
121. Soni, R. et al. Probing Cu doped Ge0.3Se0.7 based resistance switching memory devices with random telegraph noise. *J. Appl. Phys.* **107**, 024517 (2010).
122. Terai, M., Sakotsubo, Y., Saito, Y., Kotsuji, S. & Hada, H. Memory-state dependence of random telegraph noise of Ta_2O_5/TiO_2 Stack ReRAM. *IEEE Electr. Device L* **31**, 1302–1304 (2010).

123. Huang, C. Y., Shen, W. C., Tseng, Y. H., King, Y. C. & Lin, C. J. A contact-resistive random-access-memory-based true random number generator. *IEEE Electr. Device Lett.* **33**, 1108–1110 (2012). **This work reports an early implementation of memristive TRNG.**
124. Song, M. S. et al. Optimization of random telegraph noise characteristics in memristor for true random number generator. *Adv. Intell Syst-Ger* **5**, 2200358 (2023).
125. Gong, T. C. et al. A 128 kb Stochastic computing chip based on RRAM flicker noise with high noise density and nearly zero autocorrelation on 28-nm CMOS platform. *2021 IEEE International Electron Devices Meeting (IEDM)* (2021). **This work reports a large-scale memristive SC chip.**
126. Jiang, H. et al. A novel true random number generator based on a stochastic diffusive memristor. *Nat. Commun.* **8**, 882 (2017).
127. Woo, K. S. et al. A high-speed true random number generator based on a CuTe1—diffusive memristor. *Adv. Intell. Syst. Ger.* **3**, 2100062 (2021).
128. Qin, Y. B. et al. A high-speed true random number generator based on unified selector-RRAM. *IEEE Electr. Device Lett.* **44**, 1967–1970 (2023).
129. Song, S. Y., Huang, P., Shen, W. S., Liu, L. F. & Kang, J. F. A 3.3-Mbit/s true random number generator based on resistive random access memory. *Sci. China Inform. Sci.* **66**, 219402 (2023).
130. Li, X. R. et al. A 144-fJ/Bit reliable and compact TRNG based on the diffusive resistance of 3-D resistive random access. *Mem. IEEE Trans. Electron Dev.* **70**, 4139–4144 (2023).
131. John, R. A. et al. Halide perovskite memristors as flexible and reconfigurable physical unclonable functions. *Nat. Commun.* **12**, 3681 (2021).
132. Gao, B. et al. A unified PUF and TRNG design based on 40-nm RRAM with high entropy and robustness for IoT security. *IEEE Trans. Electron Dev.* **69**, 536–542 (2022).
133. Li, J., Cui, Y. J., Wang, C. H., Gu, C. Y. & Liu, W. Q. A fully configurable PUF using dynamic variations of resistive crossbar arrays. *IEEE Trans. Nanotechnol.* **21**, 737–746 (2022).
134. Gao, B. et al. Concealable physically unclonable function chip with a memristor array. *Sci. Adv.* **8**, eabn7753 (2022).
135. Woo, K. S. et al. Tunable stochastic memristors for energy-efficient encryption and computing. *Nat. Commun.* **15**, 3245 (2024).
136. Oh, J. et al. Memristor-based security primitives robust to malicious attacks for highly secure neuromorphic systems. *Adv. Intell. Syst.-Ger.* **4**, 2200177 (2022).
137. Rührmair, U. et al. Modeling attacks on physical unclonable functions. in *Proceedings of the 17th ACM Conference on Computer and Communications Security*. 237-249.
138. Adam, G. C., Khiat, A. & Prodromakis, T. Challenges hindering memristive neuromorphic hardware from going mainstream. *Nat. Commun.* **9**, 5267 (2018).
139. Pang, Y. et al. 25.2 A reconfigurable RRAM physically unclonable function utilizing post-process randomness source with $<6 \times 10-6$ native bit error rate. in *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*. 402-404.
140. Cao, Y., Yang, H., Yang, J., Liu, Q. & Liu, M. A 67F2 Reconfigurable PUF using 1T2R RRAM switching competition in 28 nm CMOS with 5e-9 bit error rate. in *2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. 1-2.
141. Yang, J. et al. A physically unclonable function with BER <0.35% for secure chip authentication using write speed variation of RRAM. in *2018 48th European Solid-State Device Research Conference (ESSDERC)*. 54-57.
142. Xue, X. et al. A 28 nm 512Kb adjacent 2T2R RRAM PUF with interleaved cell mirroring and self-adaptive splitting for extremely low bit error rate of cryptographic key. in *2019 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. 29-32.
143. Esatu, T. K. et al. Highly reliable and secure PUF using resistive memory integrated into a 28 nm CMOS. *Process. IEEE Trans. Electron Dev.* **70**, 2291–2296 (2023).
144. Chen, A. Reconfigurable physical unclonable function based on probabilistic switching of RRAM. *Electron. Lett.* **51**, 615–617 (2015).
145. Zhao, X., Zhao, Q., Liu, Y. & Zhang, F. An ultracompact switching-voltage-based fully reconfigurable RRAM PUF with low native instability. *IEEE Trans. Electron Dev.* **67**, 3010–3013 (2020).
146. Tseng, P. H. et al. ReRAM-based pseudo-true random number generator with high throughput and unpredictability characteristics. *IEEE Trans. Electron Dev.* **68**, 1593–1597 (2021).
147. Lin, B. et al. A unified memory and hardware security module based on the adjustable switching window of resistive memory. *IEEE J. Electron Devices Soc.* **8**, 1257–1265 (2020).
148. Jiang, H. et al. A provable key destruction scheme based on memristive crossbar arrays. *Nat. Electron.* **1**, 548–554 (2018).
149. Li, J., Cui, Y., Wang, C., Liu, W. & Kvatinsky, S. A Concealable RRAM physical unclonable function compatible with in-memory computing. in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 1-6.
150. Lee, H. Y. et al. Evidence and solution of over-RESET problem for HfOX based resistive memory with sub-ns switching speed and high endurance. in *2010 International Electron Devices Meeting*. 19.17.11-19.17.14.
151. Salje, E. K. H., Ding, X. & Zhao, Z. Noise and finite size effects in multiferroics with strong elastic interactions. *Appl. Phys. Lett.* **102**, 152909 (2013).
152. Hu, X. W. et al. A high area-efficiency RRAM-based strong PUF with multi-entropy source and configurable double-read process. *IEEE Int. Symp. Circ. S*, 2438-2442 (2022).
153. Park, J. et al. Highly reliable physical unclonable functions using memristor crossbar with tunneling conduction. in *2022 International Electron Devices Meeting (IEDM)*. 18.13.11-18.13.14.
154. Li, J., Cui, Y., Wang, C., Gu, C. & Liu, W. A fully configurable PUF using dynamic variations of resistive crossbar arrays. *IEEE Trans. Nanotechnol.* **21**, 737–746 (2022).
155. Neal, R. M. *Bayesian learning for neural networks*. Vol. 118 (Springer Science & Business Media, 2012).
156. Lin, Y. et al. Bayesian neural network realization by exploiting inherent stochastic characteristics of analog RRAM. in *2019 IEEE International Electron Devices Meeting (IEDM)*. 14.16. 11-14.16. 14 (IEEE).
157. Li, L. et al. Naive Bayes classifier based on memristor nonlinear conductance. *Microelectron J.* **129**, 105574 (2022).
158. Gao, D. et al. Bayesian inference based robust computing on memristor crossbar. In *Des. Aut. Con.*, 121–126 (2021).
159. Harabi, K. E. et al. A memristor-based Bayesian machine. *Nat. Electron.* **6**, 52–63 (2023).
160. Lin, Y. et al. Uncertainty quantification via a memristor Bayesian deep neural network for risk-sensitive reinforcement learning. *Nat. Mach. Intell.* **5**, 714–723 (2023). **This paper reports a large-scale BNN on a memristor system.**
161. Bonnet, D. et al. Bringing uncertainty quantification to the extreme-edge with memristor-based Bayesian neural networks. *Nat. Commun.* **14**, 7530 (2023).
162. Dalgaty, T. et al. In situ learning using intrinsic memristor variability via Markov chain Monte Carlo sampling. *Nat. Electron.* **4**, 151–161 (2021). **This paper reports a memristive Bayesian machine with Markov chain Monte Carlo sampling.**
163. Lin, Y. et al. Deep Bayesian active learning using in-memory computing hardware. *Nat. Comput. Sci.* **5**, 27–36 (2025).
164. Ticknor, J. L. A Bayesian regularized artificial neural network for stock market forecasting. *Expert Syst. Appl.* **40**, 5501–5506 (2013).

165. Jang, H. & Lee, J. Generative Bayesian neural network model for risk-neutral pricing of American index options. *Quant. Financ* **19**, 587–603 (2019).
166. Begoli, E., Bhattacharya, T. & Kusnezov, D. The need for uncertainty quantification in machine-assisted medical decision making. *Nat. Mach. Intell.* **1**, 20–23 (2019).
167. Topol, E. J. High-performance medicine: the convergence of human and artificial intelligence. *Nat. Med* **25**, 44–56 (2019).
168. McLachlan, S., Dube, K., Hitman, G. A., Fenton, N. E. & Kyrimi, E. Bayesian networks in healthcare: distribution by medical condition. *Artif. Intell. Med.* **107** (2020).
169. Xie, X. R., Liu, X. Y., Lee, T. & Wang, L. Bayesian learning for deep neural network adaptation. *IEEE-Acm Trans. Audio Speech Lang. Process.* **29**, 2096–2110 (2021).
170. Gläser, C., Heckmann, M., Joublin, F. & Goerick, C. Combining auditory preprocessing and Bayesian estimation for robust formant tracking. *IEEE Trans Audio Speech* **18**, 224–236 (2010).
171. Du, Y. et al. Predicting weather-related failure risk in distribution systems using Bayesian neural. *Netw. IEEE Trans. Smart Grid* **12**, 350–360 (2021).
172. Abistado, K. G., Arellano, C. N. & Maravillas, E. A. Weather forecasting using artificial neural network and Bayesian network. *J. Adv. Comput Intell.* **18**, 812–817 (2014).
173. Borders, W. A. et al. Integer factorization using stochastic magnetic tunnel junctions. *Nature* **573**, 390–393 (2019). **This work demonstrates probabilistic computing using spintronics technology.**
174. Daniel, J. et al. Experimental demonstration of an on-chip p-bit core based on stochastic magnetic tunnel junctions and 2D MoS2 transistors. *Nat. Commun.* **15**, 4098 (2024).
175. Piccinini, E., Brunetti, R. & Rudan, M. Self-heating phase-change memory-array demonstrator for true random number generation. *IEEE Trans. Electron Dev.* **64**, 2185–2192 (2017).
176. Shao, H. et al. First demonstration of high throughput and reliable homomorphic encryption using FeFET arrays for resource-limited IoT clients. in *2024 IEEE International Electron Devices Meeting (IEDM)*. 1-4.
177. Gong, T. et al. First demonstration of a Bayesian machine based on unified memory and random source achieved by 16-layer stacking 3D Fe-Diode with high noise density and high area efficiency. in *2023 International Electron Devices Meeting (IEDM)*. 1-4.
178. Adel, M. J., Rezayati, M. H., Moaiyeri, M. H., Amirany, A. & Jafari, K. A robust deep learning attack immune MRAM-based physical unclonable function. *Sci. Rep.-Uk* **14**, 20649 (2024).
179. Chiu, Y.-C. et al. A CMOS-integrated spintronic compute-in-memory macro for secure AI edge devices. *Nat. Electron.* **6**, 534–543 (2023). **This paper reports a 6.6 Mb STT-MRAM CIM macro with integrated PUFs.**
180. Li, T. et al. Demonstration of high-reconfigurability and low-power strong physical unclonable function empowered by FeFET cycle-to-cycle variation and charge-domain computing. *Nat. Commun.* **16**, 189 (2025).
181. Xiong, F. et al. Towards ultimate scaling limits of phase-change memory. in *2016 IEEE International Electron Devices Meeting (IEDM)*. 4.1.1-4.1.4.
182. Cai, F. et al. A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations. *Nat. Electron.* **2**, 290–299 (2019).
183. Zhang, W. et al. Edge learning using a fully integrated neuro-inspired memristor chip. *Science* **381**, 1205–1211 (2023).
184. Jiang, Z. et al. COPS: An efficient and reliability-enhanced programming scheme for analog RRAM and on-chip implementation of denoising diffusion probabilistic model. in *2023 International Electron Devices Meeting (IEDM)*. 1-4.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China Project 62404187 and in part by the Theme-based Research Scheme (TRS) project T45-701/22-R of the Research Grants Council (RGC), Hong Kong SAR.

Author contributions

C.D., Z.L. and N.W. conceived the idea. C.D., Y.R., and Z.L. wrote the manuscript. All authors contributed to discussion and revisions of the manuscript at all stages.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43246-025-00876-2.

Correspondence and requests for materials should be addressed to Zhengwu Liu.

Peer review information *Communications Materials* thanks the anonymous reviewers for their contribution to the peer review of this work. Primary Handling Editors: Aldo Isidori. A peer review file is available.