

# Power and area efficient stochastic artificial neural networks using spin-orbit torque-based true random number generator

Cite as: Appl. Phys. Lett. **118**, 052401 (2021); <https://doi.org/10.1063/5.0035857>

Submitted: 31 October 2020 • Accepted: 17 January 2021 • Published Online: 01 February 2021

Min Song, Wei Duan, Shuai Zhang, et al.

## COLLECTIONS

Paper published as part of the special topic on [Spin-Orbit Torque \(SOT\): Materials, Physics, and Devices](#)



View Online



Export Citation



CrossMark

## ARTICLES YOU MAY BE INTERESTED IN

[Spin-orbit torques: Materials, physics, and devices](#)

Applied Physics Letters **118**, 120502 (2021); <https://doi.org/10.1063/5.0039147>

[Field-free and sub-ns magnetization switching of magnetic tunnel junctions by combining spin-transfer torque and spin-orbit torque](#)

Applied Physics Letters **118**, 092406 (2021); <https://doi.org/10.1063/5.0039061>

[Integrator based on current-controlled magnetic domain wall](#)

Applied Physics Letters **118**, 052402 (2021); <https://doi.org/10.1063/5.0041362>

Time to get excited.  
Lock-in Amplifiers – from DC to 8.5 GHz

Find out more

Zurich Instruments

# Power and area efficient stochastic artificial neural networks using spin-orbit torque-based true random number generator

Cite as: Appl. Phys. Lett. **118**, 052401 (2021); doi: [10.1063/5.0035857](https://doi.org/10.1063/5.0035857)

Submitted: 31 October 2020 · Accepted: 17 January 2021 ·

Published Online: 1 February 2021



View Online



Export Citation



CrossMark

Min Song,<sup>1</sup> Wei Duan,<sup>1</sup> Shuai Zhang,<sup>2</sup> Zhenjiang Chen,<sup>2</sup> and Long You<sup>2,a)</sup> 

## AFFILIATIONS

<sup>1</sup>Hubei Key Laboratory of Ferro and Piezoelectric Materials and Devices, Faculty of Physics and Electronic Science, Hubei University, Wuhan 430062, China

<sup>2</sup>School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan 430074, China

**Note:** This paper is part of the Special Topic on Spin-Orbit Torque (SOT): Materials, Physics and Devices.

<sup>a)</sup>Author to whom Correspondence should be addressed: [lyou@hust.edu.cn](mailto:lyou@hust.edu.cn)

## ABSTRACT

Hardware implementations of Artificial Neural Networks (ANNs) using conventional binary arithmetic units are computationally expensive and energy-intensive together with large area footprints. Stochastic computing (SC) is an unconventional computing paradigm that operates on stochastic bit streams. It can offer low-power and area-efficient hardware implementations and has shown promising results when applied to ANN hardware circuits. SC relies on stochastic number generators (SNGs) to map input binary numbers to stochastic bit streams. The SNGs are conventionally implemented using random number generators (RNGs) and comparators. Linear feedback shifted registers (LFSRs) are typically used as the RNGs, which need far more area and power than the SC core, counteracting the latter's main advantages. To mitigate this problem, in this Letter, RNGs employing Spin-Orbit Torque (SOT)-induced stochastic switching of perpendicularly magnetized Ta/CoFeB/MgO nanodevices have been proposed. Furthermore, the SOT true random number generator (TRNG) is integrated with the simple CMOS stochastic computing circuits to perform a stochastic artificial neural network. To further optimize power and area efficiency, a fully parallel architecture and TRNG-sharing scheme are presented. The proposed stochastic ANN using the SOT-based TRNG obtains a negligible inference accuracy loss, compared with the binary version, and achieves  $9\times$  and  $25\times$  improvement in terms of area and power, respectively, compared with the ANN using LFSRs.

Published under license by AIP Publishing. <https://doi.org/10.1063/5.0035857>

Artificial Neural Networks (ANNs) are currently the main driving force behind the development in the field of neuroscience, artificial intelligence, and machine learning. Decades of research in ANNs, despite our limited understanding of biological Neural Networks (NNs), have shown promising results in applications such as pattern recognition, image classification, and speech recognition.<sup>1,2</sup> The core computations in ANNs can be summarized as matrix multiplication and accumulate operations on the multiple dimensional arrays. Therefore, their hardware implementation using CMOS technologies suffers from huge computational resource requirements, which cause a lot of area cost and computation power.<sup>3,4</sup> This has resulted in the exploration of several post-CMOS technologies such as spintronics,<sup>5-7</sup> oxide-based resistive memory,<sup>8,9</sup> and phase change materials<sup>10,11</sup> that can provide orders of magnitude energy improvement in comparison with CMOS implementations. Meanwhile, most of these

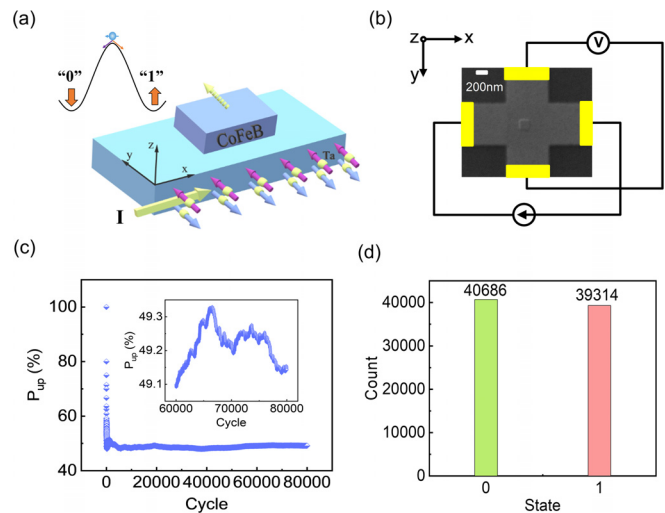
“neuro-mimetic” algorithms are based on deterministic computational units driven by the fact that the underlying CMOS hardware used to implement such algorithms is deterministic in nature. In contrast to a conventional deterministic computational circuit design, a stochastic computing (SC) circuit requires a low hardware complexity with a high fault tolerance.<sup>12-14</sup> Furthermore, SC's drawback of low precision is well-tolerated in ANNs, since many classification tasks of ANNs do not require high accuracy computation.<sup>15,16</sup> In SC, data, which are interpreted as probabilities and called Stochastic Numbers (SNs), are represented in the form of bit streams of “0”s and “1”s and generated by circuits called Stochastic Number Generators (SNGs). Traditionally, SNGs are composed of pseudorandom number generators [such as Linear Feedback Shift Registers (LFSR)] and comparators. When CMOS-based implementation of LFSRs is used to generate a vast number of pseudorandom bit streams, those bit streams are both periodic

and cross-correlated. Such correlations are usually undesirable from a computational perspective. Pseudorandom bit stream generation can also incur high energy and area consuming.<sup>17,18</sup> A preferable solution would be to rely on “true” random number generators (TRNGs) that generate random bits based on physical phenomena that are intrinsically random. The inherent stochastic switching in nonvolatile memory (NVM) devices makes bit streams generated by their circuits aperiodic and truly random.<sup>19–21</sup> In particular, spin-orbit torque (SOT)-Magnetoresistive Random Access Memory (MRAM) offers advantages over other types of NVM technologies in neural network applications,<sup>22,23</sup> due to its significant endurance, high energy efficiency, and CMOS compatibility.<sup>24,25</sup> Recently, spin-orbit torque,<sup>26,27</sup> generated by the spin Hall effect (SHE) and/or Rashba effect in heavy metal (HM)/ferromagnet (FM) bilayers, has attracted great attention because of its ability to switch the magnetization in an FM layer that can be directly controlled through the input.<sup>28</sup> Deterministic current-induced SOT full magnetization switching without any magnetic field was reported.<sup>29,30</sup> While a lot of research has focused on reducing its critical switching current density for deterministic switching,<sup>31–33</sup> attempts have been made to exploit the stochastic switching characteristics of nanomagnets by SOT to use them as TRNGs or SNGs,<sup>34</sup> which could produce bit streams representing any fraction between 0 and 1. Specifically, the magnetization of the FM layer is initialized along its hard axis direction by the SOT. Once the current is removed, the magnetization will go back to one of the two stable states along its easy axis with an equal probability. Therefore, SOT-based SNGs can be simulated to produce parallel stochastic bit streams for SC.<sup>35–37</sup> The parallel stochastic architecture achieves comparable accuracy to its sequential counterparts; however, duplicate SNGs cause the area and energy requirement of their SC applications much larger than those of conventional binary implementation.

In this work, we first experimentally demonstrate a spin-orbit torque-based TRNG, which relies on the process-induced SOT switching current variations, to construct SNGs. Sign-magnitude (SM) representation is introduced in arithmetic operations using SOT-based SNGs. Then, a fully parallel stochastic ANN is implemented through SOT-based SNGs integrated with CMOS stochastic computing circuits. To avoid duplicate hardware cost, a TRNG sharing scheme is finally proposed, which significantly reduces the area and power of the fully parallel network while maintaining the overall performance in terms of accuracy.

The essential idea of the TRNG is depicted in Fig. 1(a). When an in-plane charge current flows through the Ta layer, the spin current is generated due to the spin Hall effect in the Ta layer and is transmitted across the Ta/CoFeB interface. This results in the application of a torque on the CoFeB layer. The torque will align the magnetization to the in-plane direction and induce stochastic switching,<sup>38</sup> when there is no assistance of in-plane magnetic fields,<sup>27</sup> exchange coupling,<sup>39</sup> or breaking of geometrical symmetry.<sup>32,40</sup>

Our film stack consists of Ta (10 nm)/CoFeB (1 nm)/MgO (1 nm)/Ta (2 nm) (from the bottom), which is sputtered on a thermally oxidized Si substrate at room temperature. A scanning electron microscope image of a typical Hall bar structure with a magnetic device ( $200 \times 200 \text{ nm}^2$ ) at the cross section along with the setup of the measurement is shown in Fig. 1(b). To investigate the stochastic switching induced by SOT, we studied the upward switching probability of the TRNG as a function of switching cycles [Fig. 1(c)]. In each



**FIG. 1.** True random number generator implemented by SOT-induced stochastic switching of the nanomagnet. (a) Schematic of the device structure and principle of the TRNG. (b) Scanning electron microscope image of a nanomagnet along with the setup of the measurement. (c) Upward switching probability as a function of switching cycles. The inset shows the detail in the range of cycle 60 000–80 000. In each cycle, a current of  $I_w = 1.2 \text{ mA}$  under  $H_x = 35 \text{ Oe}$  is applied to the TRNG followed by a read current of  $50 \mu\text{A}$  to detect the orientation of the magnetization. (d) Counts of upward and downward states over the 80 000 switching cycles.

cycle, a current of  $I_w = 1.2 \text{ mA}$  under  $H_x = 35 \text{ Oe}$  is applied to the TRNG followed by a read current of  $50 \mu\text{A}$  to detect the anomalous Hall voltage that is proportional to the z-component of the magnetization, which is generated according to the anomalous Hall effect (AHE), arising from the spin-orbit interaction of the spin-polarized current carriers in conductive ferromagnetic materials. Here, we define the upward state as bit 1 while the downward state as bit 0. The proportion of these two states over 80 000 is 40 686/39 314 (0/1), as shown in Fig. 1(d). The ratio is very close to 1, indicating that the TRNG has good performance to generate high-quality random bits. The requirement of the small field may be associated with a superfluous bias field in the structure, which was probably introduced via our non-optimized fabrication process.<sup>20</sup> The random codes generated by the SOT-TRNG have passed the NIST SP800–22 test,<sup>41</sup> showing high quality of randomness (for details, please see Table S1 of the supplementary material).

The SNG, which consists of a random number generator (RNG) and a comparator, converts a given binary number to stochastic bit stream form, as shown in Fig. 2(a). SC uses bit streams to represent numbers encoded as the probability of 1's in the total number of bits.<sup>14,42</sup> In our case, a fractional number within (0, 1) is encoded from the n-bit binary number, through comparing with an n-bit true random number generated by the SOT-TRNG. If the binary number is larger than the true random number, a 1 is generated; otherwise, the output is a 0. The schematic circuit of the SOT-based TRNG is shown in the red dotted box, and the gray square area denotes the SOT device region. The output (a “0–1” code sequence) of the RNG is distributed randomly due to the stochastic switching of the magnetization (see Sec. 2 of the supplementary material for description of circuit implementation in detail).

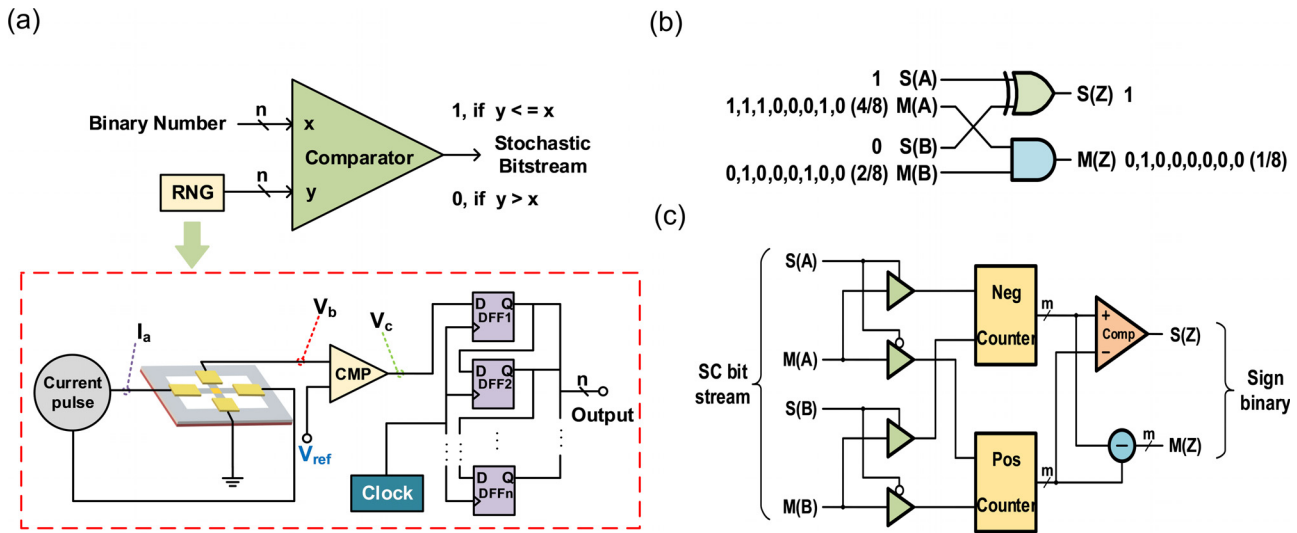


FIG. 2. The SOT-induced stochastic switching of the TRNG without external field. (a) The schematic circuits of the SNG. (b) A stochastic multiplier for the sign-magnitude representation. (c) A stochastic adder/subtractor for the sign-magnitude representation.

In SC, utilizing stochastic bit streams, complex arithmetic operations can be implemented by simple gates. For example, multiplication can be implemented by an AND gate in the unipolar format or an XNOR gate in the bipolar format. Unipolar and bipolar representations are most commonly used two encoding schemes.<sup>16</sup> However, the unipolar format cannot directly handle signed values, which are usually needed for ANNs. While, in the bipolar format, the maximum relative error occurs when both operands are near zero, which will result in a higher error rate. Here, sign-magnitude<sup>43</sup> representation, which handles the sign and magnitude of an SC number separately, is used to expand the unipolar representation range and takes advantage of unipolar encoding accuracy in SC. Figure 2(b) illustrates the SC multiplier taking two sign-magnitude SC numbers. The sign-magnitude stochastic multiplier is implemented by an XOR gate for computing the sign bit and a unipolar multiplier for computing the magnitude bits. For the sign-magnitude addition, we use two counters: one for the group of negative stochastic inputs and the other for that of positive ones, as shown in Fig. 2(c). In every cycle, the counters count the number of ones in the input SC bit streams to generate negative and positive sums. The comparator is used to compare the value of the two counters and decides the sign bit of the final result. The larger count value is subtracted from the smaller one to obtain the addition calculation result in binary form. While addition operation may be complicated in SM-SC than that in the conventional SC, SM-SC has peculiar operation and resource efficiency in ANNs using SOT-based TRNG applications.

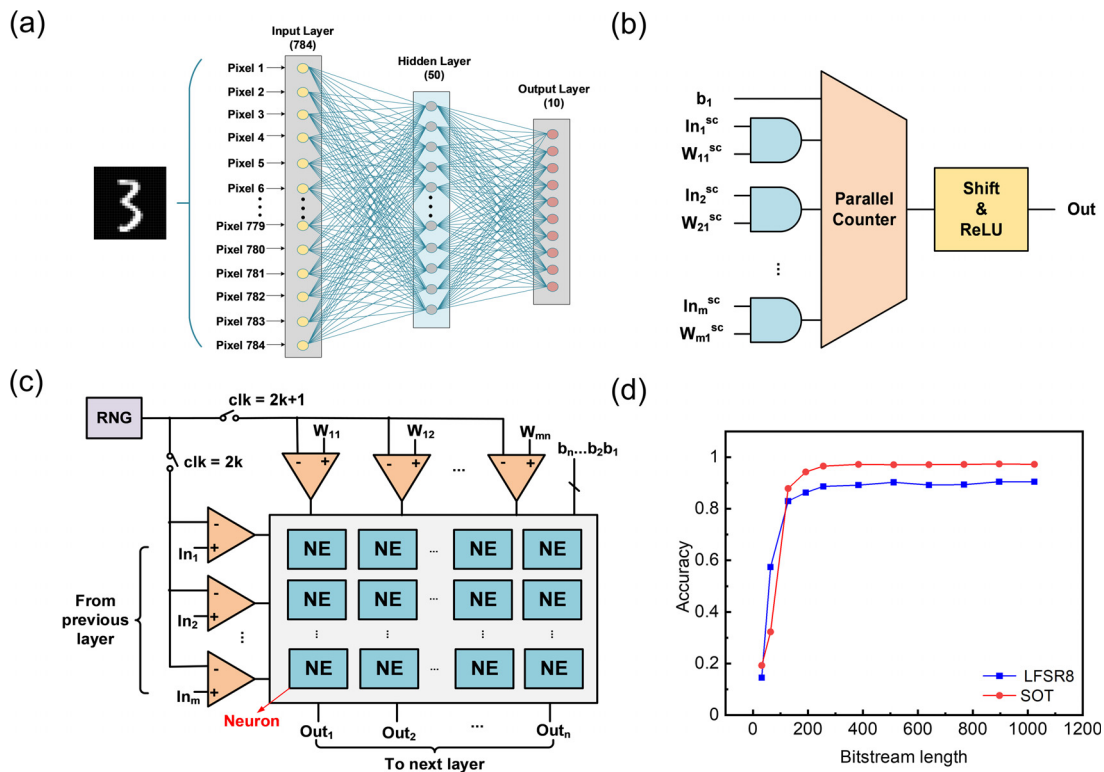
Then, a fully connected (FC) stochastic ANN is implemented by stochastic computing with a configuration of 784–50–10, meaning 784 visible input nodes, 50 hidden-layer nodes, and 10 output nodes, as shown in Fig. 3(a). The SC hardware structure of the proposed neuron is shown in Fig. 3(b); input bit streams are first multiplied with weight bit streams, then parallel counters accumulate multiplied products with bias, after that, the shifters scale the result in proper interval, and finally the neuron's output is calculated using the ReLU function.

Parallel counters will count the number of 1's and convert stochastic bit streams to binary numbers. The ReLU function is very simple to implement; if sign bit indicates positive, the output of ReLU function equals the input; otherwise, the output equals 0. To improve the overall throughput, generally fully parallel architecture is used to shorten the latency for the SC design.

When LFSRs are used in stochastic computing, duplicate SNGs are necessary for a parallel architecture to avoid correlation, which cause substantial hardware overhead. Thanks to the true random property of the SOT-based TRNG, we propose an RNG sharing method on the premise of better accuracy. Only one RNG is equipped to generate the necessary uncorrelated randomness for all input pixels and weights in the entire stochastic neural network. The structure for each layer is shown in Fig. 3(c), and the random numbers generated by one RNG compare with inputs and weights, respectively, using the time division multiplexing method. For each layer, input  $In$ , weight  $W$ , and bias  $b$  are binary numbers, to do the binary-to-stochastic conversion, and the inputs  $In_1 \sim In_m$  compare with random bitstreams generated by the RNG at  $clk = 2k$ , while weights  $W_{11} \sim W_{mn}$  compare with random bitstreams generated by the RNG at  $clk = 2k + 1$ . Then,  $In_1 \sim In_m$  and their corresponding weights in the stochastic format will be calculated with the bias in neurons in parallel. Note that we can use only one RNG to complete the operation in one layer; furthermore, if this RNG is connected to each layer with the same sharing scheme, no more RNGs are needed in the whole ANN. First, we train the ANN in python with the cross-entropy error loss function using the error backpropagation algorithm and check its accuracy on the MNIST dataset. The MNIST handwritten digit image dataset consists of 60 000 training data and 10 000 testing data, with  $28 \times 28$  grayscale image and 10 classes.<sup>44</sup>

Then, we obtain the inference accuracy of the ANN, in which the SOT-TRNG and LFSR RNG are used as hardware components, adopting our one-RNG sharing scheme. Experimentally measured raw data of the SOT-TRNG are used in the SNG to generate the stochastic





**FIG. 3.** Architecture and simulation results of the stochastic ANN. (a) The fully connected neural network structure consists of 784 input neurons, 50 hidden neurons, and 10 output neurons. (b) Structure of the proposed neuron. (c) Parallel architecture and RNG sharing schemes for fully connected layers. (d) Inference accuracy of the ANN with the SOT-TRNG and LFSR RNG using the same sharing scheme.

bitstreams. The red line in Fig. 3(d) represents SC ANN inference accuracy with our one-SOT-TRNG sharing scheme, while the blue line represents inference accuracy with an eight-bit LFSR using the same sharing scheme. It can be seen that with the increase in the bit stream length, the inference accuracy of the ANN with the SOT-based TRNG gets much higher than that of the ANN with the LFSR, which may be in large part by true randomness. When a LFSR is used to generate a vast number of pseudorandom bit streams, those bit streams are both periodic and cross-correlated, while bit streams generated by the SOT-TRNG behave aperiodic and truly random. These phenomena can be clearly seen from the two-dimensional grayscale patterns as shown in Fig. S1 of the [supplementary material](#).

The stochastic ANN with the proposed sharing SOT-TRNG, the stochastic ANN with LFSR RNGs, and ANN of conventional binary (32-bit floating-point) are realized with Verilog Hardware Description Language at the register-transfer level. We use a Synopsys Design Compiler to synthesize the ANN with the 45 nm Nangate Open Cell Library.<sup>45</sup> The stochastic ANN based on the SOT- and LFSR-RNG have the same fully parallel hardware architecture. To obtain high inference accuracy, the ANN using LFSRs without the RNG sharing scheme is implemented. In the fully parallel structure, the ANN using LFSRs needs 39984 LFSRs to avoid correlation, while the ANN using the SOT-TRNG needs only one RNG due to its true random feature. The enormous chip area (estimated to be 120 mm<sup>2</sup>) and power (estimated to be 17 W) make binary ASIC unpractical for implementation

using the fully parallel structure; therefore, we employ a semi-parallel approach in the binary design. The inference accuracy and synthesis results in terms of area, power, energy and latency for testing each input image with different configurations are summarized in [Table I](#).

The inference accuracy of test data in the MNIST dataset of ANNs plays a crucial role in the performance of the system. The inference accuracy of the proposed approach is higher than the corresponding bit stream length of the SC network with LFSRs. The test accuracy of our proposed ANN with 1024-bit stream is 97.23%, while the test accuracy of the floating-point network is 97.51%. There is only a 0.28% point difference between the two networks, showing a negligible performance loss of the proposed ANN compared with the binary version.

The random numbers used in the SC-ANN are produced by the AHE read scheme in SOT nanomagnets composed of Ta/CoFeB/MgO heterostructures. However, for practical applications, these heterostructures can be implemented as the free-layer stack in Magnetic Tunnel Junctions (MTJs), in which a larger Tunnel Magnetoresistance (TMR) will be obtained. Therefore, we use the SOT-MTJ for the performance estimation of the SC ANN. The area and power of the SOT-TRNG have been estimated by the hardware simulation results of the SOT MTJ-based TRNG circuit (for details, see Secs. 4–6 of the [supplementary material](#)). The area consumption of the SOT-TRNG is mainly contributed by transistors, due to the small dimension and back-end integration ability of spintronic devices. When one SOT-TRNG is

**TABLE I.** Synthesis and simulation results for a 784-50-10 network at 50 MHz and 1 V in 45-nm CMOS technology.

	SC using a SOT-TRNG (this work)			SC using LFSRs			Binary
	256	512	1024	256	512	1024	
Stream length (bit)	256	512	1024	256	512	1024	...
Accuracy (%)	96.55	97.09	97.23	95.68	96.67	96.65	97.51
Power (mW)	8.25	8.31	8.43	210.32	210.41	210.50	216.65
Area (mm <sup>2</sup> )	0.24	0.24	0.24	2.62	2.62	2.62	2.46
Energy ( $\mu$ J)	0.09	0.18	0.36	2.40	4.55	8.87	0.32
Latency ( $\mu$ s)	11.4	21.64	42.12	11.4	21.64	42.12	1.46

shared, the area of the proposed ANN is decreased to 9.16% and 9.76%, compared with the ANN with LFSRs and conventional binary implementation, respectively. At the same time, the ANN with SOT-based TRNG has up to 25.49 $\times$  and 26.26 $\times$  power efficiency, compared with the ANN counterparts. Energy consumption of SC networks rises with the increase in bit stream length, due to SC's long latency. In general, SC's clearest drawback may be its long latency needed by stochastic-binary and binary-stochastic number conversion. There is a trade-off between the area, power consumption, accuracy, and latency in the proposed architectures. The results show that our work has significantly lower hardware cost and power efficiency than the ANN with LFSRs and conventional binary implementation.

In summary, an efficient stochastic implementation of the ANN is proposed using the SOT-based TRNG. Experimental results show that our proposed design obtains similar inference accuracy compared to conventional binary implementation while achieving lower error rates compared to the stochastic counterpart with LFSRs. In terms of hardware cost, compared to the conventional CMOS-based floating-point binary implementation, our design reduces the area occupation up to 91.3% and consumes 96.2% less power. Compared to the stochastic implementation using LFSRs, our design saves up to 91.8% area and 96.1% energy consumption. Our proposed hybrid stochastic ANN takes advantage of both spintronic devices and simple stochastic computing to enable a high-performance and low-cost computing paradigm.

See the [supplementary material](#) for the details about the proposed ANN using the SOT-based TRNG.

## AUTHORS' CONTRIBUTIONS

M.S. and W.D. contributed equally to this work.

The authors acknowledge financial support from the National Natural Science Foundation of China (NSFC Grant Nos. 61904051, 61821003, and 61674062); the Research Project of Wuhan Science and Technology Bureau (No. 2019010701011394); and the Fundamental Research Funds for the Central Universities (No. HUST: 2018KFYXKJC019).

The authors declare that they have no competing interest.

## DATA AVAILABILITY

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

## REFERENCES

- 1P. Bashivan, K. Kar, and J. J. DiCarlo, *Science* **364**(6439), eaav9436 (2019).
- 2A. Prieto, B. Prieto, E. Ortigosa, E. Ros, F. Pelayo, J. Ortega, and I. Rojas, *Neurocomputing* **214**, 242 (2016).
- 3A. K. Jain, M. Jianchang, and K. M. Mohiuddin, *Computer* **29**(3), 31 (1996).
- 4G. Zhang, B. E. Patuwo, and M. Y. Hu, *Int. J. Forecast.* **14**(1), 35 (1998).
- 5T. Jungwirth, J. Sinova, A. Manchon, X. Marti, J. Wunderlich, and C. Felser, *Nat. Phys.* **14**(3), 200 (2018).
- 6F. Pulizzi, *Nat. Mater.* **11**(5), 367 (2012).
- 7S. Bhatti, R. Sbiaa, A. Hirohata, H. Ohno, S. Fukami, and S. N. Piramanayagam, *Mater. Today* **20**(9), 530 (2017).
- 8Y. Yang, P. Sheridan, and W. Lu, *Appl. Phys. Lett.* **100**(20), 203112 (2012).
- 9X. Liu, S. M. Sadaf, S. Park, S. Kim, E. Cha, D. Lee, G. Jung, and H. Hwang, *IEEE Electron Device Lett.* **34**(2), 235 (2013).
- 10L. Wang, L. Tu, and J. Wen, *Sci. Technol. Adv. Mater.* **18**(1), 406 (2017).
- 11L. Wang, S.-R. Lu, and J. Wen, *Nanoscale Res. Lett.* **12**(1), 347 (2017).
- 12A. Alaghi and J. P. Hayes, *ACM Trans. Embedded Comput. Syst.* **12**(2s), 1 (2013).
- 13B. R. Gaines, in *Advances in Information Systems Science: Volume 2*, edited by J. T. Tou (Springer US, Boston, MA, 1969), p. 37.
- 14B. D. Brown and H. C. Card, *IEEE Trans. Comput.* **50**(9), 891 (2001).
- 15J. Yu, K. Kim, J. Lee, and K. Choi, "Accurate and efficient stochastic computing hardware for convolutional neural networks," in *2017 IEEE International Conference on Computer Design (ICCD)* (IEEE, 2017), p.105.
- 16A. Alaghi, W. Qian, and J. P. Hayes, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **37**(8), 1515 (2018).
- 17M. Riedel, in *Stochastic Computing: Techniques and Applications*, edited by W. J. Gross and V. C. Gaudet (Springer International Publishing, Cham, 2019), p. 121.
- 18A. Ardakani, F. Leduc-Primeau, N. Onizawa, T. Hanyu, and W. J. Gross, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **25**(10), 2688 (2017).
- 19S. Sahay and M. Suri, *Semicond. Sci. Technol.* **32**(12), 123001 (2017).
- 20H. Chen, S. Zhang, N. Xu, M. Song, X. Li, R. Li, Y. Zeng, J. Hong, and L. You, in *2018 IEEE International Electron Devices Meeting (IEDM)* (2018), p. 36.5.1.
- 21A. Fukushima, T. Seki, K. Yakushiji, H. Kubota, H. Imamura, S. Yuasa, and K. Ando, *Appl. Phys. Express* **7**, 083001 (2014).
- 22W. A. Borders, H. Akima, S. Fukami, S. Moriya, S. Kurihara, Y. Horio, S. Sato, and H. Ohno, *Appl. Phys. Express* **10**(1), 013007 (2017).
- 23Y. Cao, A. Rushforth, Y. Sheng, H. Zheng, and K. Wang, *Adv. Funct. Mater.* **29**(25), 1970175 (2019).
- 24K. L. Wang, J. G. Alzate, and P. Khalili Amiri, *J. Phys. D* **46**(7), 074003 (2013).
- 25S. A. Wolf, J. Lu, M. R. Stan, E. Chen, and D. M. Treger, *Proc. IEEE* **98**(12), 2155 (2010).
- 26I. M. Miron, K. Garello, G. Gaudin, P.-J. Zermatten, M. V. Costache, S. Auffret, S. Bandiera, B. Rodmacq, A. Schuhl, and P. Gambardella, *Nature* **476**(7359), 189 (2011).
- 27L. Liu, C.-F. Pai, Y. Li, H. W. Tseng, D. C. Ralph, and R. A. Buhrman, *Science* **336**(6081), 555 (2012).
- 28J. Ma, J. Hu, Z. Li, and C.-W. Nan, *Adv. Mater.* **23**(9), 1062 (2011).
- 29Y. Cao, Y. Sheng, K. W. Edmonds, Y. Ji, H. Zheng, and K. Wang, *Adv. Mater.* **32**(16), 1907929 (2020).

- <sup>30</sup>Q. Ma, Y. Li, D. B. Gopman, Y. P. Kabanov, R. D. Shull, and C. L. Chien, *Phys. Rev. Lett.* **120**(11), 117703 (2018).
- <sup>31</sup>S. Fukami, T. Anekawa, C. Zhang, and H. Ohno, *Nat. Nanotechnol.* **11**, 621 (2016).
- <sup>32</sup>L. You, O. Lee, D. Bhowmik, D. Labanowski, J. Hong, J. Bokor, and S. Salahuddin, *Proc. Natl. Acad. Sci. U. S. A.* **112**(33), 10310 (2015).
- <sup>33</sup>M. Kazemi, G. E. Rowlands, E. Ipek, R. A. Buhrman, and E. G. Friedman, *IEEE Trans. Electron Devices* **63**(2), 848 (2016).
- <sup>34</sup>Y. Kim, X. Fong, and K. Roy, *IEEE Magn. Lett.* **6**, 1 (2015).
- <sup>35</sup>J. Hu, B. Li, C. Ma, D. Lilja, and S. J. Koester, *IEEE Trans. Electron Devices* **66**(8), 3620 (2019).
- <sup>36</sup>G. Finocchio, T. Moriyama, R. De Rose, G. Siracusano, M. Lanuzza, V. Puliafito, S. Chiappini, F. Crupi, Z. Zeng, T. Ono, and M. Carpentieri, *J. Appl. Phys.* **128**(3), 033904 (2020).
- <sup>37</sup>P. Debashis, R. Faria, K. Y. Camsari, and Z. Chen, *IEEE Magn. Lett.* **9**, 1 (2018).
- <sup>38</sup>K. Cai, M. Yang, H. Ju, S. Wang, Y. Ji, B. Li, K. W. Edmonds, Y. Sheng, B. Zhang, N. Zhang, S. Liu, H. Zheng, and K. Wang, *Nat. Mater.* **16**(7), 712 (2017).
- <sup>39</sup>Y.-C. Lau, D. Betto, K. Rode, J. M. D. Coey, and P. Stamenov, *Nat. Nanotechnol.* **11**(9), 758 (2016).
- <sup>40</sup>G. Yu, P. Upadhyaya, Y. Fan, J. G. Alzate, W. Jiang, K. L. Wong, S. Takei, S. A. Bender, L.-T. Chang, Y. Jiang, M. Lang, J. Tang, Y. Wang, Y. Tserkovnyak, P. K. Amiri, and K. L. Wang, *Nat. Nanotechnol.* **9**(7), 548 (2014).
- <sup>41</sup>See <https://github.com/arcetri/sts> for “National Institute of Standards and Technology, NIST Statistical Test Suite, 2017.”
- <sup>42</sup>J. Hayes, in Proceedings-Design Automation Conference (2015).
- <sup>43</sup>A. Zhakatayev, S. Lee, H. Sim, and J. Lee, in 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC) (2018), p. 1.
- <sup>44</sup>Y. LeCun, C. Cortes, and C. J. C. Burges, see <http://yann.lecun.com/exdb/mnist/> for “The MNIST Database of Handwritten Digits, 1998.”
- <sup>45</sup>See <https://si2.org/open-cell-library/> for “Nangate Inc., Nangate 45 nm Open Cell Library, 2009.”