SANDIA REPORT
SAND2022-13142
Printed September,2022

Probabilistic Nanomagnetic Memories for Uncertain and Robust Machine Learning

Christopher H Bennett, T. Patrick Xiao, Samuel Liu, Leonard Humphrey, Jean Anne Incorvia, Bert J. Debusscherre, Daniel Ries, and Sapan Agarwal

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@osti.gov
Online ordering:	http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.gov
Online order:	https://classic.ntis.gov/help/order-methods

ABSTRACT

This project evaluated the use of emerging spintronic memory devices for robust and efficient variational inference schemes. Variational inference (VI) schemes, which constrain the distribution for each weight to be a Gaussian distribution with a mean and standard deviation, are a tractable method for calculating posterior distributions of weights in a Bayesian neural network such that this neural network can also be trained using the powerful backpropagation algorithm. Our project focuses on domain-wall magnetic tunnel junctions (DW-MTJs), a powerful multi-functional spintronic synapse design that can achieve low power switching while also opening the pathway towards repeatable, analog operation using fabricated notches .

Our initial efforts to employ DW-MTJs as an all-in-one stochastic synapse with both a mean and standard deviation didn't end up meeting the quality metrics for hardware-friendly VI. In the future, new device stacks and methods for expressive anisotropy modification may make this idea still possible. However , as a fall back that immediately satisfies our requirements, we invented and detailed how the combination of a DW-MTJ synapse encoding the mean and a probabilistic Bayes-MTJ device , programmed via a ferroelectric or ionically modifiable layer, can robustly and expressively implement VI. This design includes a physics-informed small circuit model, that was scaled up to perform and demonstrate rigorous uncertainty quantification applications, up to and including small convolutional networks on a grayscale image classification task, and larger (Residual) networks implementing multi-channel image classification.

Lastly, as these results and ideas all depend upon the idea of an inference application where weights (spintronic memory states) remain non-volatile, the retention of these synapses for the notched case was further interrogated. These investigations revealed and emphasized the importance of both notch geometry and anisotropy modification in order to further enhance the endurance of written spintronic states. In the near future, these results will be mapped to effective predictions for room temperature and elevated operation DW-MTJ memory retention, and experimentally verified when devices become available.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Table 0-1.

Abbreviation	Definition
DNN	Deep Neural Network
BNN	Bayesian Neural Network
VI	Variational Inference
MVM	Matrix Vector Multiply
TRNG	True Random Number Generators
MTJ	Magnetic Tunnel Junction
STT	Spin Transfer Torque
SOT	Spin Orbit Torque
DW-MTJ	Domain-Wall Magnetic Tunnel Junction
VCMA	Voltage-Controlled Magnetic Anisotropy
TMR	Tunneling Magnetoresistance
ECE	Expected Calibration Error
CIs	Confidence Intervals
ADC	Analog to Digital Conversion
MAC	Multiply and Accumulate
RT	Room Temperature

1. INTRODUCTION

The powerful ability of deep neural networks (DNNs) to generalize has driven their wide proliferation in the last decade to many applications. However, particularly in applications where the cost of a wrong prediction is high, there is a strong desire for algorithms that can reliably quantify the confidence in their predictions [2]. Bayesian neural networks (BNNs) can provide the generalizability of DNNs, while also enabling rigorous uncertainty estimates by encoding their parameters as probability distributions learned through Bayes' theorem such that predictions sample trained distributions [3]. Probabilistic weights can also be viewed as an efficient form of model ensembling, reducing overfitting [4]. In spite of this, the probabilistic nature of BNNs makes them slower and more power-intensive to deploy in conventional hardware, due to the large number of random number generation operations required [5]. Some proposals to increase the energy efficiency of digital BNNs via pipelining have been made [6], but ultimately these approaches hit an efficiency wall due to the serial nature of random number generation. In contrast, emerging memory devices pose an attractive set of possible options for true random number generators (TRNGs) at a less than 1 pJ /bit energy footprint [7].

In recent years, in-memory computing has also emerged to enable orders-of-magnitude more efficient processing of data-intensive DNN algorithms. These systems alleviate the memory wall problem in conventional architectures, while also leveraging the efficiency and parallelism of analog computation [8, 9]. A variety of computational memory devices have been proposed as artificial synapses for DNNs: resistive random access memories (ReRAM) [10, 11], phase change memories [12, 13], electrochemical memories [14–17], designer ionic/electronic thin films [18], magnetic memories [19], and others. However, these synaptic devices cannot directly implement BNN weights, which are not static but are sampled from trained probability distributions.

Spintronic devices possess properties that make them promising for data storage, in-memory computing for DNNs, and probabilistic computing. Spintronic devices typically use the magnetic tunnel junction (MTJ) as the building block [20] and have demonstrated high energy efficiency, scalability, and endurance [21–23]. Magnetic spin textures such as domain walls [24–27] and skyrmions [28, 29] can implement complex, tunable behaviors that can realize higher-order neurons and synapses. Spintronic devices also have unique intrinsic stochastic properties [30–32]. Recently, stochasticity in MTJs has been experimentally demonstrated to produce conductance noise due to thermal fluctuations in magnetization experienced by the free ferromagnetic layer. Importantly, the distribution of conductance noise is dictated by the magnetic energy landscape, which can be manipulated using a variety of methods including magnetic field [33], spin transfer torque [34], spin orbit torque [35], and voltage-controlled magnetic anisotropy (VCMA) [36, 37]. As a result, the tunable random bitstream readout of stochastic MTJs can be used to implement Boltzmann machines for probabilistic computing [38]. While proposals for spin-based BNNs have been made [39, 40], they relied upon either streaming generated RNGs from the periphery into each array or using digital circuitry to fully compose the weight used in the sampling step. These decisions majorly reduce the efficiency of a hardware spintronic BNN design by increasing the energy cost of the basic sampling operation. Lastly, ReRAM devices have also been used to implement probabilistic weights [41–43], but required many devices per weight since the weight's mean and standard deviation cannot be independently encoded at the device level.

In this work, we introduce a novel array design for efficient probabilistic matrix-vector multiplication (MVM) sample steps with the inference operation fully supported by *in-situ* analog spintronic device electrical operation . We target BNNs that are trained using the variational inference method to represent each weight as a normal distribution with a trained mean (μ) and standard deviation (σ). The BNNs are deployed on a spintronic system where each weight is encoded by a domain-wall memory with multi-bit precision in μ, and a stochastic spintronic memory that independently encodes σ with multi-bit precision. The devices are directly integrated in the same array, and are used together in a probabilistic MVM. The accuracy and quality of uncertainty predictions from the proposed hardware are evaluated using realistic in-memory computing simulations, based on stochastic device properties obtained from micromagnetic simulations. We show that the proposed spintronic implementations of BNNs give accurate, well-calibrated uncertainty estimates for complex classification and regression problems that match software BNN implementations, and are superior to comparable DNNs. These BNN predictions require 10-100× less energy than conventional hardware by efficiently combining the RNG and MVM operations in the analog domain.

1.0.1. Bayesian Neural Networks

A Bayesian neural network uses probabilistic weights to make predictions with a quantified uncertainty. Though there are other ways to quantify uncertainty, a BNN produces well-calibrated uncertainties by learning the weight probability distributions using Bayes' theorem [3]:

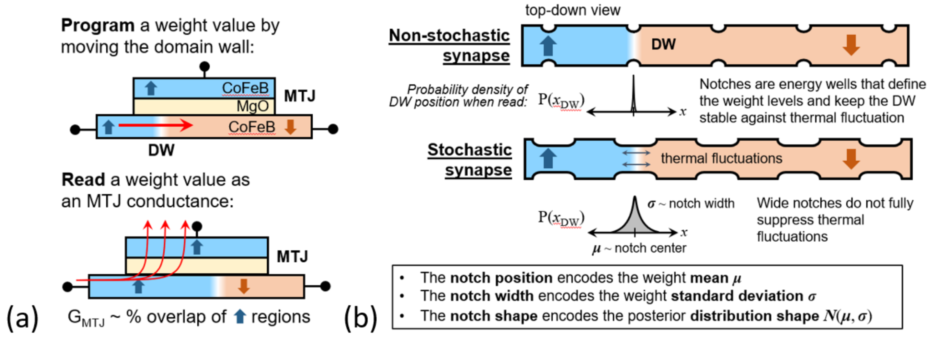
$$P(\Theta|\mathcal{D}) = \frac{P(\mathcal{D}|\Theta)P(\Theta)}{P(\mathcal{D})} \tag{1.1}$$

$P(\Theta|\mathcal{D})$ is known as the posterior distribution of the model's weights Θ after it has been exposed to the training data $\mathcal{D}$. After training, the distributions are fixed. When evaluated on new data, multiple predictions are made using different samples of the posterior weight distribution, and the statistics of these predictions is used to quantify the uncertainty. In this work, BNNs are trained in software, then their effectiveness on unseen data is evaluated using simulations of the proposed spintronic hardware.

During the training phase, computing the right-hand side of this equation is computationally expensive. Furthermore, the posterior distribution for a weight can be an arbitrarily complex distribution that is difficult to implement in analog hardware. For these reasons, we approximate Bayes' theorem using variational inference (VI) [44]. VI is used to constrain the distribution for each weight to be a Gaussian distribution $\mathcal{N}(\mu, \sigma)$, parameterized by a mean μ and a standard deviation σ, as shown in Fig. 3-3A. These parameters can be efficiently trained using the backpropagation algorithm [45]. We use the Tensorflow Probability framework and the Flipout method [46] to implement BNN training with VI. Our proposed hardware is compatible with Gaussian-distributed weights trained using any method. As a baseline, the trained BNNs are compared to iso-topology deep neural networks (DNNs) with deterministic weights. DNNs were trained using Tensorflow Keras. Details on the specific trained networks are given in Sections 3.0.4 and 3.0.5.

2. NOTCH-ENCODABLE NOISE SOURCES

Based on our desire for an all-in-one stochastic synapse, our initial project plan led us to explore the limits for expressing both a mean and standard deviation within an individual DW-MTJ device. Effectively, as shown in Fig 1, the mean is represented by the notch that the device is programmed to, and the mean is represented by the natural jitter of the weight within the given notch. Therefore, for the application of bayesian inference, the width of the notch is extended as compared to the standard neural network case (where minimizing the sigma or movement of the notch is advantageous to limit extraneous inference noise).



Program a weight value by moving the domain wall:
CoFeB
MgO
CoFeB
MTJ
DW
Read a weight value as an MTJ conductance:
MTJ
(a)
G_{MTJ} ~ % overlap of ⬆ regions
(b)
top-down view
Non-stochastic synapse
DW
Probability density of DW position when read:
$P(x_{DW})$
x
Notches are energy wells that define the weight levels and keep the DW stable against thermal fluctuation
Stochastic synapse
thermal fluctuations
$P(x_{DW})$
σ ~ notch width
x
μ ~ notch center
Wide notches do not fully suppress thermal fluctuations
• The notch position encodes the weight mean μ
• The notch width encodes the weight standard deviation σ
• The notch shape encodes the posterior distribution shape $N(\mu, \sigma)$

Figure 2-1. (A) Lateral view of our chosen spintronic nanosynapse where the top visualizing shows how programming (notch selection) can be makde using a large current (voltage) across the track; then subsequently the device can be read out non-destructively via the large top electrode. (B) Basic conceptual design for a notched all-in-one stochastic synapse is given and distinguished relative to non-stochastic sister device. As visualized, the larger (wider) notches allow for natural encoding of the σ value. In addition, the notch shape can also represent a designer distribution shape if desired.

2.0.1. Mechanisms for intrinsic stochasticity modifications

As the inherent stochasticity of the domain due to thermal effects is an open question in nanophysics, the domain wall-magnetic tunnel junction device was analyzed through the context of micromagnetic simulations that emulate the elementary motion of the chiral domain-wall (DW) structure based on fundamental properties of the thin films as well as applied current. Early on, we realized that besides temperature itself (not a feasible tunable approach), an electrically feasible way to alter the magnitude of the random domain wall motion would be the voltage-controlled magnetic anisotropy (VCMA) effect. VCMA modification of DW jitter can be accomplished either by a) applying strain via a piezoelectric substrate , b) or by applying voltage to an oxide layer on top of the free layer with the domain wall. Some possible contacting schemes utilizing a piezoelectric are presented in Fig 2(a). However, this scheme suffers from a complicated contacting scheme. In order to maintain an array-usable three terminal structure, voltage-based VCMA is preferable . However, the effect must be also be made non-volatile (that is, different voltages applied to every synapse during inference would be an unacceptable energy cost. For that reason, a second option was considered: magneto-ionic modification of the active layer. Magneto-ionic modification induces an anisotropy or stiffness change in the material through the introduction of oxygen ions (anionic motion) into the free layer [47]. Specifically, the migration of oxygen into the free layer reduces magnetic order and reduces anisotropy or stiffness. One example set of material stacks that would realize this effect are shown in Fig2(b). Analogously to the normal DW-MTJ programming mode, the anisotropy effects could be realized during a separated ‘anisotropy programming mode’ ; then, during inference time, the modified synapses would continue to realize their appropriate mean and jitter values.

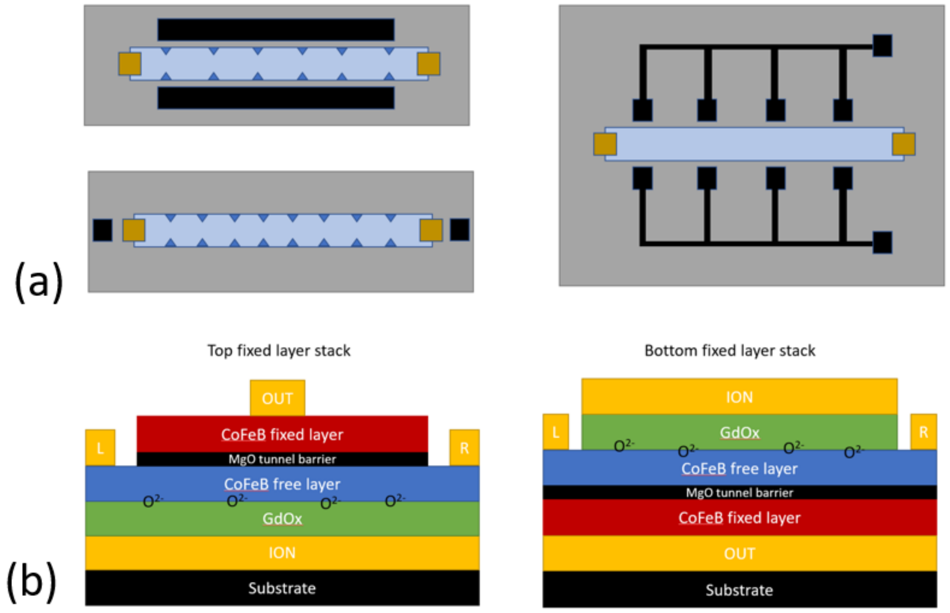


Figure 2-2. (A) Birds-eye view of several iterations of a contacting scheme, where black wires help to apply piezoelectric effect via applied current to enable non-volatile modification of anisotropy. In the two left schemes notches are still fabricated as before, whereas in the right figure the application points constitute notches. (B) Alternative magneto-ionic modification approach. Two iterations of this stack are given, where the ionic reservoir is either above or below the free layer (where domain wall propagates). The top stack is more analogous to the current DW-MTJ approach with the injection of oxygen ions across the green active layer leading to a broad range anisotropy values.

The micromagnetics solver mumax3 is also used to measure the variation in domain wall position over a period of time, done for various levels of anisotropy that can be modulated using magneto-ionics. From the collected data, we discovered there is an approximately 3.5:1 ratio in maximum to minimum amount of noise in the system. This effect also works for multiple notch positions. While this ratio is somewhat sufficient to implement toy uncertainty quantification problems such as the one shown in Fig. 2-4, which is only a 1 dimensional mapping, we discovered using our CrossSim software that at least 5x+ is necessary to effectively encode the gaussian standard deviation spreads on more complicated tasks . This result is echoed in the next section as well (Fig. 3-5).

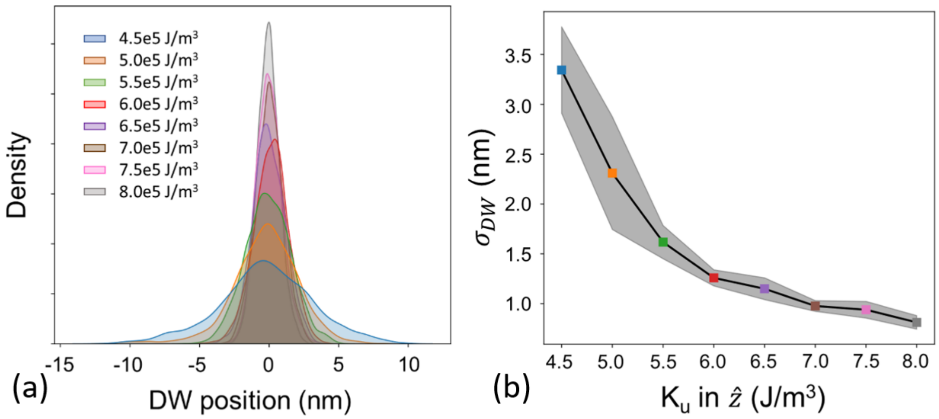


Figure 2-3. (A) MuMax simulations demonstrating the effect of the anisotropy modification upon the spread of domain-wall sampled over time within a single notc, where the plus and minus distances are computed with respect to the center of the notch. (B) Sigma values for the computed histogram from (A) as a function of the Anisotropy. Colors for each of the points correspond to the color in the left pane.

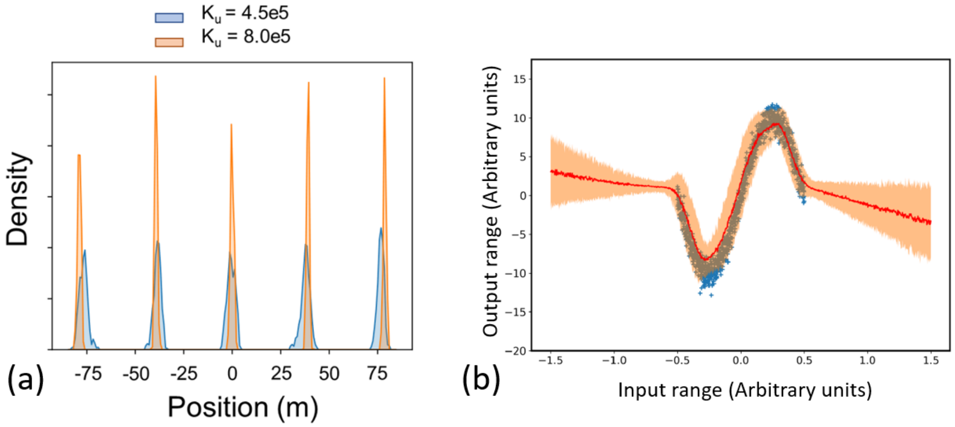


Figure 2-4. (A)MuMax simulations showing verification of modifiable sigma values within several weights/notches along a single track, with the width of the two curves color-corresponding to the respective anisotropy value pictured above the pane. As visible, the lower anisotropy (less stiff) track allows the DW to wander more, increasing the width of the quasi-Gaussian spread (B) Demonstration of how anisotropically modified DW jitter over this range can be used on a simple neural network trained to predict the sine function over a small range. As visible, at this ratio/range, the out-of-domain region (out of$-0.5 < x < 0.5$) does demonstrate some increase in variance after sampling the small neural network.

3. COMBINED SPINTRONIC BIT CELL FOR ROBUST UNCERTAINTY QUANTIFICATION

To encode a BNN's weight probability distributions, our fully spintronic Bayesian artificial synapse compactly integrates a tunable noise source with a programmable artificial synapse that encodes the mean component of the weight. The tuning range of the conductance noise should ideally cover a large range in order to encode both wide (highly noisy) and narrow (nearly deterministic) weight probability distributions. The proposed Bayes-MTJ utilizes the physical stochasticity and voltage controllability of magnetic materials to realize this functionality, and further uses magneto-ionics to ensure that the encoded noise properties are non-volatile.

The Bayes-MTJ structure is shown in Fig. 3-1A, and is based on a cylindrical in-plane MTJ. Both of the in-plane axes (i.e. the *x*-*y* plane) are easy axes for the free layer's magnetization, and thus thermal fluctuations can readily cause random changes in the free layer's in-plane magnetization. These fluctuations generate noise in the conductance across the MTJ, and this noise fully spans the range between the maximum conductance state (free and reference layers parallel) and the minimum conductance state (free and reference layers anti-parallel). Experiments validating this effect in cylindrical in-plane magnetic systems have been shown previously [48]. Since the noise always spans the full conductance range of the device, the magnitude of conductance noise can be controlled by modulating the MTJ's tunnel magnetoresistance (TMR) ratio via the voltage-controlled magnetic anisotropy (VCMA) effect. Modulation of the TMR ratio using an applied voltage across the oxide layer has has been demonstrated previously, both experimentally and theoretically [49–52].

An externally applied voltage is not an efficient implementation of tunable noise because each device encodes a unique probability distribution and thus would require an independent VCMA voltage during an inference operation. However, there are at least two ways that non-volatile encoding of the noise magnitude can be accomplished. Firstly, a ferroelectric or multiferroic layer can be introduced to the stack to induce a polarization field at the interface, implementing an effective electric field that can be modulated to an appropriate state using applied voltage [53–55]. Another option is to introduce an ion-conductive layer to reversibly modulate the oxidation state of the free layer. Ion migration is induced using an electric field, resulting in non-volatile changes in magnetic properties such as the magnetic anisotropy [56–59] and magnetoresistance [60–62]. Oxidation of the free layer has been shown to reduce the TMR of MTJ stacks[63]. In this paper, these effects will be approximated using an effective built-in voltage V_{bi} across the MgO tunnel barrier that is set during programming.

The Bayes-MTJ can be represented by a macrospin Landau-Lifshitz-Gilbert (LLG) model described as follows [49]:

$$\frac{m}{t} = -\gamma\mu_0 m \times H_{eff} + \alpha m \times \frac{m}{t} - \beta P J_{STT} m \times (m \times m_r) \qquad (3.1)$$

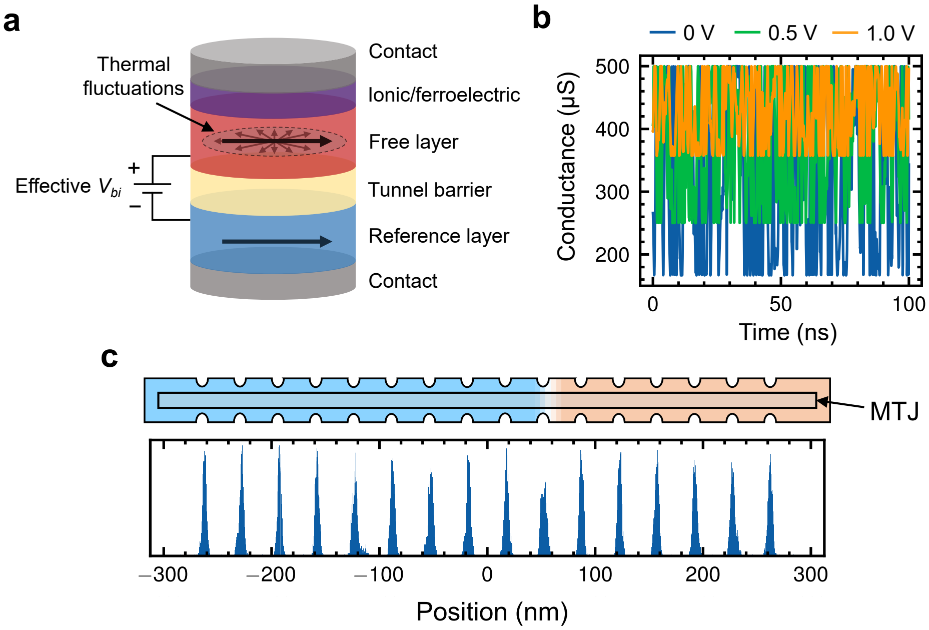


Figure 3-1. (A) Structure of the Bayes-MTJ. Thermal fluctuations cause random changes in the in-plane magnetization of the free layer that manifest as noise in the tunnel magnetoresistance, (B) Simulated noise in the Bayes-MTJ conductance. The applied voltage modulates the TMR and the magnitude of the noise via the VCMA effect. (C) Structure of a notched DW-MTJ synapse. Bottom shows the distribution of DW position over 25 ns after being initialized in each of 16 notches.

where m and m_p are the magnetization unit vector of the free and reference layers respectively, γ is the Gilbert gyromagnetic ratio, α is the damping parameter, P is the spin polarization, and J_{STT} is applied spin transfer torque current density. $\beta = \gamma\hbar/2et_F M_s$, where $\hbar$ is the reduced Planck constant, e is electron charge, t_F is the thickness of the free layer, and M_s is saturation magnetization. Additionally, a random vector representing thermal fluctuations at finite temperature is added to each time step into the effective field term, similar to the implementation in MuMax3 [64]:

$$H_{therm} = \eta\sqrt{\frac{2\mu_0\alpha k_B T}{M_s\gamma V\Delta t}} \tag{3.2}$$

where η is a random vector from a standard normal distribution updated every time step, μ_0 is vacuum permeability, k_B is the Boltzmann constant, T is the absolute temperature, V is the cell volume, and Δt is the simulation time step. Relevant simulation values for an in-plane anisotropy CoFeB/MgO/CoFeB system are presented in Table 1.

The VCMA effect modulates the anisotropy field as well as the resistance when a voltage is applied. The anisotropy field is modeled with the following:

$$\hat{z}H_k = \frac{2K_i}{t_{free}M_s\mu_0} - \frac{2\kappa_s V_{bi}}{\mu_0 M_s t_{ox} t_{free}} \tag{3.3}$$

where K_i is the anisotropy energy, t_{free} and t_{ox} are the thickness of the free layer and oxide layer respectively, κ_s is the VCMA coefficient, and V_{bi} is the built-in voltage. The resistance of the MTJ can be expressed as:

$$R = R_p \frac{1 + (\frac{V_{bi}}{V_h})^2 + TMR}{1 + (\frac{V_{bi}}{V_h})^2 + \frac{1}{2} TMR(1 + \sin\theta \cos\phi)} \tag{3.4}$$

where R_p is resistance when the magnetizations of free and reference layers are parallel, V_h is the voltage at which the TMR ratio is halved, and θ and ϕ are the polar coordinates for the unit vector magnetization of the free layer.

In Fig. 3-1B, the conductance of the Bayes-MTJ device is sampled for 100 ns at a VCMA voltage of 0V, 0.5V, and 1V. In each case, the conductance varies randomly and continuously between the fully parallel and fully anti-parallel states of the MTJ. Increasing the VCMA voltage decreases the TMR ratio, which narrows the range of allowed output conductance and thus reduces the magnitude of conductance noise. This device acts as a tunable noise source that is used by the cell design in Section 3.0.2 to encode the standard deviation of a probability distribution.

3.0.1. Domain Wall Static Weight Encoder

To encode the static or mean value of a weight probability distribution, we use a domain wall-magnetic tunnel junction (DW-MTJ) artificial synapse [26, 32]. This three-terminal device has previously been shown to have extremely low read and write noise, an important feature for the precise encoding of static weights. The DW-MTJ device contains a ferromagnetic rectangular wire that produces a magnetic domain wall (DW). The wire lies underneath a tunnel barrier and a reference magnetic layer to form an MTJ. The DW-MTJ can encode multiple conductance states based on the DW position, which controls the proportion of the free layer that is parallel or anti-parallel to the reference layer. Notches are also lithographically defined along the edges of the wire to provide linearly spaced, repeatable states and reduce drift of the DW due to thermal fluctuations. A write operation is performed by passing current in the direction of the desired DW motion, in-plane to the stack, while a read operation is performed by measuring resistance perpendicular to the stack (through the tunnel barrier). DW motion is mediated by spin transfer torque (STT) and an additional spin orbit torque (SOT) component provided by the heavy metal layer underneath the free layer. A top-down schematic of the device is shown in Fig. 3-1C.

To model the more complicated physical dynamics of the DW in the free layer, the MuMax3 micromagnetics solver is used [64]. The finite temperature LLG equation described previously is solved for each timestep for a multi-spin system. The constants used for the perpendicular magnetic anisotropy CoFeB/MgO/CoFeB system in this simulation are shown in Table 2. To characterize the intrinsic noise of a DW-MTJ, a DW is created at a notch within the track and the position of the DW is sampled over 25 ns at 300 K. This is repeated for all 16 levels to characterize the variation in DW position, shown in Fig. 3-1C. On average, the DW-MTJ's conductance noise is approximately 0.335% of the full conductance range dictated by its TMR.

3.0.2. Probabilistic In-Memory Matrix-Vector Multiplication

We propose a novel cell design shown in Fig. 3-2A to combine the Bayes-MTJ tunable noise source with a DW-MTJ static weight, collectively encoding the trained weight probability distributions in BNNs. The cell uses the difference in conductance of two DW-MTJs to represent both positive and negative weight means. All three devices are connected on one end to the same metal column so that their output currents add. The fabrication challenges of simultaneously integrating both types of devices are important to note. Since the proposed cell centers around the use of the in-plane magnetization Bayes-MTJ, one solution is to use in-plane magnetization DW-MTJ devices [65] to enable monolithic integration of both devices on the same material stack. However, when scaling and energy efficiency is a concern, out-of-plane magnetic systems are typically desired for the DW-MTJ device. This is because in-plane domain walls are generally wider and more sensitive to track roughness [66], limiting scaling in contrast to out-of-plane systems. In this case, heterogenous integration of two different magnetic material stacks is necessary. One solution is for different stacks to be grown in different areas of the wafer for integration during the growth phase [67]. Another possibility is to use flip chip integration, allowing devices to be fabricated on two different magnetic substrates before being bonded together for final integration [68].

To realize independent control of the weight means by the DW-MTJs and the weight standard deviations by the Bayes-MTJ, the time-averaged conductance of the Bayes-MTJ must be canceled out so that the device contributes only zero-centered random noise. To accomplish this, a bipolar volage pulse is applied to the Bayes-MTJ device consisting of two pulses of equal duration and amplitude but opposite polarity. The resulting bipolar current is integrated over the full duration on a capacitor at the bottom of a column, using a current conveyor (CC) circuit. The CC acts as a current buffer with large output resistance while maintaining a virtual ground on the column [69]. The time-averaged conductance of the Bayes MTJ contributes equal but opposite currents during the two halves of the pulse, and gets canceled out in the final capacitor charge so that only the noise contribution remains. An important advantage of this approach is that the cancellation does not depend on the value of the time-averaged conductance, so that device-to-device MTJ variations can be tolerated.

Figure 3-2B shows the accumulated charge from the output of a Bayes-MTJ alone during a read pulse with length t_{read} = 2 ns, for five independent pulses. The dashed black line depicts the output of a deterministic resistor with R_p = 2 kΩ. Each run with a Bayes-MTJ is an independent sample from the encoded weight probability distribution. There is a clear difference in the noise distribution at different applied voltage, where the final accumulated charge has a much tighter distribution around 0C when 2V is applied due to the reduced TMR. Fig. 3-2C shows the distribution of the charge noise after 2 ns for two effective V_{bi}, with 20,000 samples each. The distribution is not Gaussian, but can effectively approximate BNNs trained with normally distributed weights, as shown in the next section.

The integrated charge Q from a Bayes-MTJ can then be converted to an effective conductance noise via $\delta G_{BMTJ} = Q/V_{read}\,t_{read}$, where V_{read} is the read voltage (note that Q scales linearly with V_{read} so δG_{BMTJ} is independent of V_{read}). The dependence of the conductance noise standard deviation on built-in voltage is shown in Fig. 3-2D, for t_{read} = 2 ns. The range of modulation between maximum and minimum noise standard deviation is 38.9:1. Fig. 3-2E shows how the noise standard deviation depends on the pulse length at 0V built-in voltage. A 2 ns sample time is chosen to maximize the cycle-to-cycle fluctuations in capacitor charge. A longer integration time averages out the effective conductance noise.

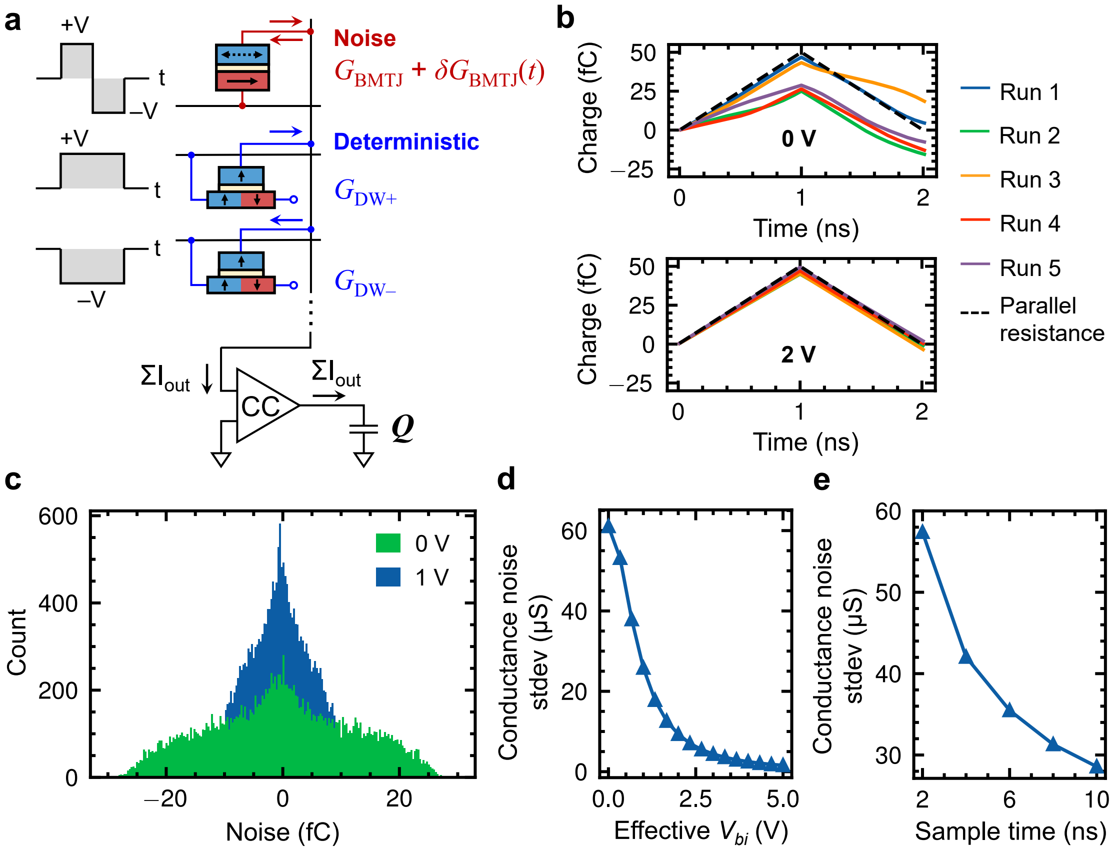


Figure 3-2. (A) Probabilistic cell with one Bayes-MTJ and two DW-MTJ devices to encode a programmable Gaussian distribution. The third DW-MTJ terminal is used only during programming. Currents on a column are integrated on a capacitor. (B) Charge on the capacitor Q vs. time due to the Bayes-MTJ current at two values of V_{bi}. Each run is an independent read using a 2ns bipolar volage pulse. (C) Distribution of the final capacitor charge $Q(T = 2ns)$induced by noise in the Bayes-MTJ, at two values of V_{bi} with 20,000 samples each. (D) Standard deviation of $Q(T = 2ns)$ vs. V_{bi} on the Bayes-MTJ, (E) $Q(T)$ vs. the total pulse length T at $V_{bi} = 0$V.

The two DW-MTJs are driven by unipolar pulses of the same amplitude and total duration as the bipolar pulse: one positive and one negative, so that their currents are subtracted. Currents from multiple cells of this type can be summed on the same column, and the same read pulses can be broadcast to a row of cells. This implements a fully analog, in-memory MVM where every matrix element is sampled simultaneously from an independent probability distribution. The amplitude of the three pulses applied to each row is proportional to the corresponding element of the input vector. The integrated charge can be read out as a capacitor voltage that represents the final probabilistic matrix-vector product.

3.0.3. *Mapping BNNs to Bayes-MTJ Arrays*

For each probabilistic weight in the trained BNNs, the mean μ is mapped to the difference in conductance $(G_{DW+} - G_{DW-})$ of a DW-MTJ device pair. The standard deviation σ is encoded in the effective conductance noise of the Bayes-MTJ tunable noise source, defined in Section 3.0.2. The simulated Bayes-MTJ noise distribution in Fig. 3-2C does not exactly follow a Gaussian distribution. The Bayes-MTJ noise distribution is zero-symmetric, strictly bounded, and has the same shape regardless of V_{bi}, which controls the width of the distribution. To compactly model this distribution for large arrays, the following analytical distribution is used, up to a normalization constant:

$$P(x) = \frac{\pi}{2} A \sin\left(\frac{\pi}{2}(x+1)\right) + \frac{1-A}{B\sqrt{2\pi}} \exp\left(-\left(\frac{x}{B}\right)^2\right) \tag{3.5}$$

where $A = 0.9298$ and $B = 0.0367$ are fitting parameters, and x is a random variable in the range $(-1, +1)$. For a desired value of σ, a random value x is sampled from this distribution and is converted to a conductance fluctuation by:

$$\delta G_{BMTJ}(x) = 61.06 S \times \left(\frac{\sigma}{\mu_{max}}\right) \times 2.379x \tag{3.6}$$

where 61.06 μS is the maximum Bayes-MTJ effective conductance noise at $V_{bi} = 0\text{V}$ (using $V_{read} = 0.1\text{V}$, $t_{read} = 2$ ns, and $R_p = 2$ kΩ). The constant 2.379 accounts for the difference in the standard deviation between $P(x)$ and the standard normal $\mathcal{N}(0, 1)$. The σ value is normalized by μ_{max}, the largest absolute value of μ for the layer, which is mapped to the parallel resistance of the DW-MTJ in Table II. The value of $R_{p,DW}$ was tuned to fit the BNN's σ values inside the available conductance noise range.

Fig. 3-3B shows the simulated Bayes-MTJ noise distribution at a voltage of 0.5V alongside its analytical distribution (blue) and a Gaussian distribution with the same standard deviation (red). Fig. 3-3C shows the distribution of σ, expressed in terms of the target Bayes-MTJ conductance standard deviation, for a five-layer Fashion MNIST BNN to be described in Section **??**. The range between the green dashed lines represents the σ values that can be encoded by the Bayes-MTJ having V_{bi} between 0V and 5V, which will be the range used through the rest of the paper unless otherwise stated. Excluding the first layer, the vast majority (99.5%) of the σ values in the BNN can be encoded by the Bayes-MTJ, with outliers clipped to the nearest value inside the range. The first layer's σ values are almost entirely zero, so it is implemented by a standard array where no read pulses are delivered to the Bayes-MTJ rows.

For the spintronic hardware simulations of BNNs in the following sections, we extend the CrossSim modeling framework [70] for analog accelerators to model in-memory computations with tunable stochastic elements. The Bayes-MTJ is modeled using the analytical distribution above. The μ values were linearly quantized to be compatible with four bits of precision in each DW-MTJ conductance (16 notches), and the σ values were nonlinearly quantized to support four bits of precision in the VCMA voltage.

3.0.4. Quantifying Classification Uncertainty

For classification problems, a DNN typically has a softmax output layer, which can be interpreted as a vector of probabilities p for every class. The information entropy of this vector measures the amount of uncertainty in a given prediction: $H(p) = -\sum_i p_i \log p_i$, where i indexes the class.

The uncertainty of a BNN is based on sampling N predictions, each yielding a probability vector p. The overall prediction and confidence are based on the expectation value of the probability vector formed from the N samples: $E[p]$. Multiple sampling of the probabilistic weights also allows the predicted uncertainty for a given input to be decomposed into an aleatoric and epistemic uncertainty [71]:

$$H_{total} = H_{aleatoric} + H_{epistemic} \tag{3.7}$$

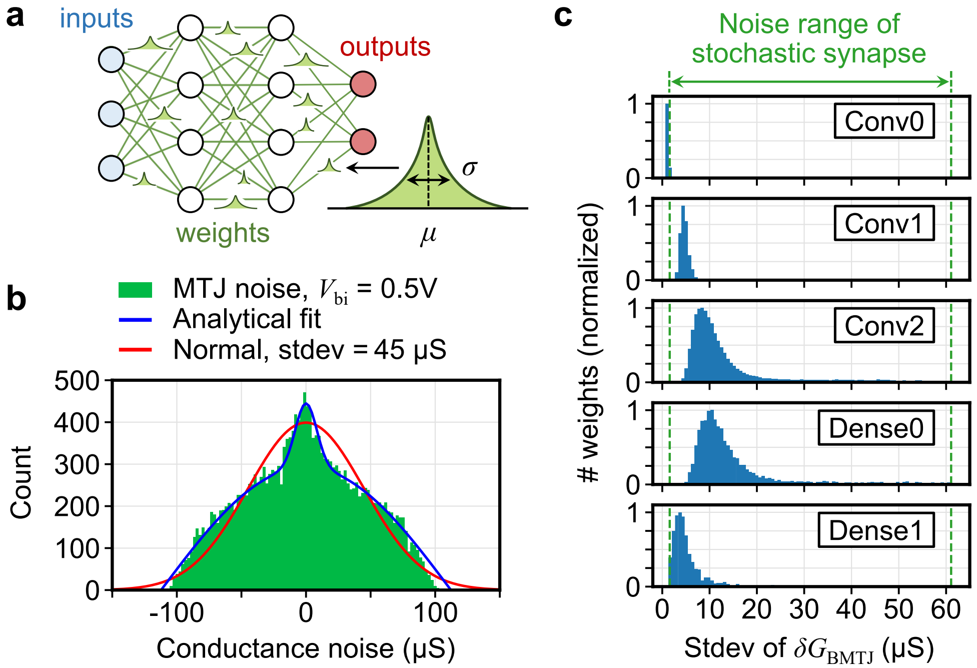


Figure 3-3. (A) Schematic of a Bayesian neural network where each weight follows a Gaussian distribution $\mathcal{N}(\mu, \sigma)$, (B) Analytical fit to the BayesMTJ noise distribution from LLG simulations, and Gaussian distribution with the same standard deviation, (C) Distribution of σ values for each layer of a Fashion MNIST BNN, mapped to Bayes-MTJ conductance noise. The first layer's weights are implemented without activating the Bayes-MTJ.

where

$$H_{total} = H(E[p]), \quad H_{aleatoric} = E[H(p)] \tag{3.8}$$

Aleatoric uncertainty $H_{aleatoric}$ originates from randomness or ambiguity inherent in the data, and the epistemic uncertainty $H_{epistemic}$ originates from the model's lack of knowledge [72]. Aleatoric uncertainty tends to be high when the input data is noisy, while epistemic uncertainty tends to be high if the input is out of distribution, i.e. has properties that are distinct from the training data. Epistemic uncertainty is particularly useful in enabling the neural network to make safe extrapolations to out-of-distribution data [73]. Thus, the BNN offers two potential advantages over the DNN baseline: (1) better calibrated uncertainty estimates, and (2) meaningful decomposition of uncertainty.

The loss function used for variational inference is a sum of the prediction's categorical cross entropy and the Kullback-Leibler (KL) divergence of each posterior distribution with the prior, aggregated over all the weights. The KL divergence term is responsible for approximating Bayes' theorem [45], while for the DNN baseline only the categorical cross entropy loss is used.

A DNN and a BNN were trained on the Fashion MNIST dataset [74] with ten classes, both using the LeNet-5 architecture [75], but with sigmoids replaced by Rectified Linear Unit (ReLU) activations and average pooling replaced by max pooling. The DNN has 61.7K parameters and the BNN has 123.2K parameters, since each weight has two parameters (μ and σ). The bias weights in the BNN are left deterministic so that they can be implemented digitally within the accelerator. The same optimizer

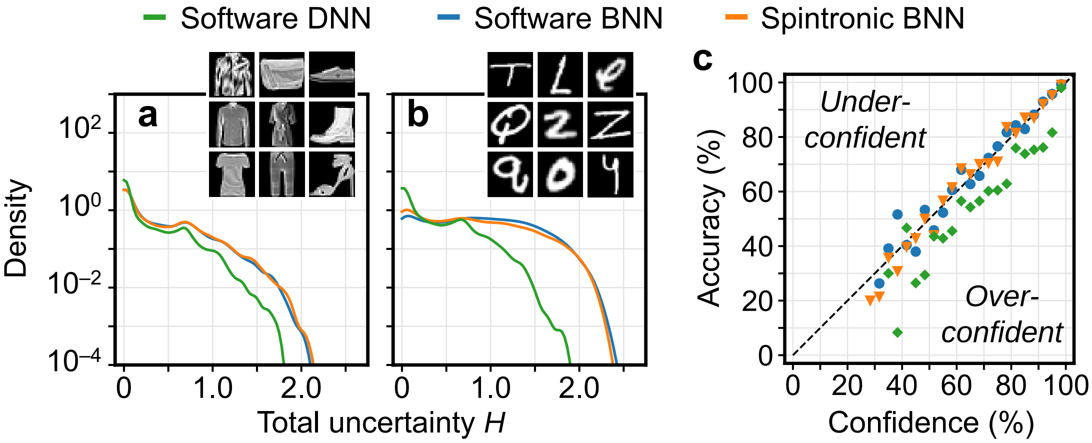


Figure 3-4. Distribution of predicted uncertainty (kernel density estimation) by the software DNN, software BNN, and BNN simulated on spintronic hardware, on (A) the Fashion MNIST test set, and (B) the EMNIST-Letters test set. Both the DNN and BNN were trained on Fashion MNIST. Uncertainties are in units of information entropy. (C) Uncertainty calibration curve of the three cases on the Fashion MNIST test set. Note that most of the predictions lie in the highest confidence bands.

(Adam), number of epochs (20), and learning rate (10^{-3}) are used for both models. Fig. 3-3C shows the distribution of trained σ values in the BNN for each layer.

First, both networks were evaluated on the Fashion MNIST test set (10,000 images) and the EMNIST-Letters test set (10,000 images) of handwritten letters [76], representing out-of-distribution data where the network should predict high uncertainty. The BNN was evaluated both in software and simulated on the spintronic hardware, and was sampled 100 times unless otherwise specified. Fig. 3-4A and B show that all cases predict low uncertainty on Fashion MNIST and higher uncertainty on EMNIST-Letters. However, the DNN still has a prominent peak at low uncertainty for letters, whereas the BNN has a much higher uncertainty overall, as expected.

To more quantitatively assess the quality of these uncertainty estimates, a calibration curve [77] is used, shown in Fig. 3-4C. For each network, the Fashion MNIST test set is split into bins based on the confidence of the prediction. If the uncertainty is well calibrated, the confidence should match the accuracy of the images in the bin: e.g. for images where the network has 50% confidence, it should ideally be correct 50% of the time. Fig. 3-4C shows that the BNN is better calibrated than the DNN, which is over-confident, and that the spintronic BNN closely implements the software BNN despite the limited noise range, limited noise precision, and the difference in distribution shape. An overall metric for the quality of the uncertain estimate is the expected calibration error [77]:

$$ECE = \sum_{m}^{M} \frac{N_m}{N_{test}} |acc(x_m) - conf(x_m)| \tag{3.9}$$

where x_m is the set of images in the m^{th} confidence bin, N_m is the number of images in this bin, and $N_{test} = 10,000$ is the size of the test set. The accuracy and ECE for the three cases are shown in Table 3.

We further probe the differences between the BNN and DNN by experimenting with images that are linear superpositions of Fashion MNIST clothing items and EMNIST letters. This is parameterized by the letter fraction, where 0% is a Fashion MNIST image and 100% is a letter image, as shown in Fig. 3-5A. Fig. 3-5B shows the ECE vs letter fraction, where 1000 random clothing-letter pairs were generated for each letter fraction from 0% to 90%, separated by 10% intervals. The label for each image is the original Fashion MNIST label. The ECE is less meaningful at very high letter fractions where the image is very weakly related to its label. The BNNs, including the spintronic implementation, have lower ECE at all values of the letter fraction, indicating better calibrated uncertainties. Fig. 3-5C shows how the spintronic hardware's ECE changes as the noise On/Off ratio of the Bayes-MTJ is decreased below what can be achieved with V_{bi} = 5V (see Fig. 3-2D). A ratio larger than 10 can accurately capture the small σ values in the network.

Finally, Fig. 3-5D and E compare the decomposed uncertainty components of the DNN and spintronic BNN, respectively. The DNN baseline is deterministic, so it cannot predict a non-zero epistemic uncertainty. For both models, the aleatoric uncertainty peaks at an intermediate letter fraction, though this is more evident in the BNN. This is hypothesized to be due to the fact that images with near-equal mixtures of letters and clothing items have the greatest number of overlapping spatial features and thus appear more noisy. Meanwhile, the BNN's predicted epistemic uncertainty increases nearly monotonically with letter fraction, which matches the fact that a higher letter fraction means that the image is farther away from the training distribution. The epistemic uncertainty is important for increasing the BNN's uncertainty for images with large letter fraction where the original Fashion MNIST label is harder to predict.

3.0.4.1. CIFAR-100 Experiments

To demonstrate the feasibility of the spintronic BNN accelerator on a more complex problem and a larger-scale algorithm, deep residual networks (ResNets) [78] were trained on the CIFAR-100 image classification dataset with 100 classes [79]. The ResNet topology in Fig. 3-6A was used to train both a DNN and a BNN having 1.25M and 2.50M parameters, respectively. To improve accuracy, both networks were trained with data augmentation (random horizontal flips, random horizontal shifts ≤10%, and random vertical shifts ≤10%) applied to the training images. Both networks were trained for 100 epochs with the same optimizer (Adam) and learning rates. Fig. 3-6B shows the distribution of σ values in the BNN for each layer. To facilitate mapping to the Bayes-MTJ, a maximum value constraint was imposed on σ during training.

The spintronic hardware implementation of the BNN used the same assumptions as for Fashion MNIST, except that we represent μ values with eight bits of precision using bit slicing [9]: each μ value uses two pairs of DW-MTJ devices with 16 notches per device. One pair encodes the higher four bits and is integrated with the Bayes-MTJ that encodes the 4-bit σ value. The other pair encodes the lower four bits in a separate array where the Bayes-MTJ rows are left unused. The Bayes-MTJ is not used for the first convolution layer where most of the σ values are near zero. To improve energy efficiency, the batch normalization operation is folded into the convolution μ and σ values [80].

The ECEs of the trained ResNets on the CIFAR-100 test set (see Table 3) are larger than for Fashion MNIST due to the greater complexity of the task: correct predictions with high confidence were less dominant in CIFAR-100. The BNN reduces the ECE by 7×, at a cost of just 0.41% in top-1 accuracy. Fig. 3-6C shows the calibration curves. The spintronic implementation of the BNN tends to be more confident than the software BNN. We hypothesize that this is because the analog accelerator resamples the Bayes-MTJ noise on every probabilistic MVM, and thus each instance of weight re-use in a convolution layer independently resamples the posterior weight distributions. Averaged across the ResNet, a given weight is

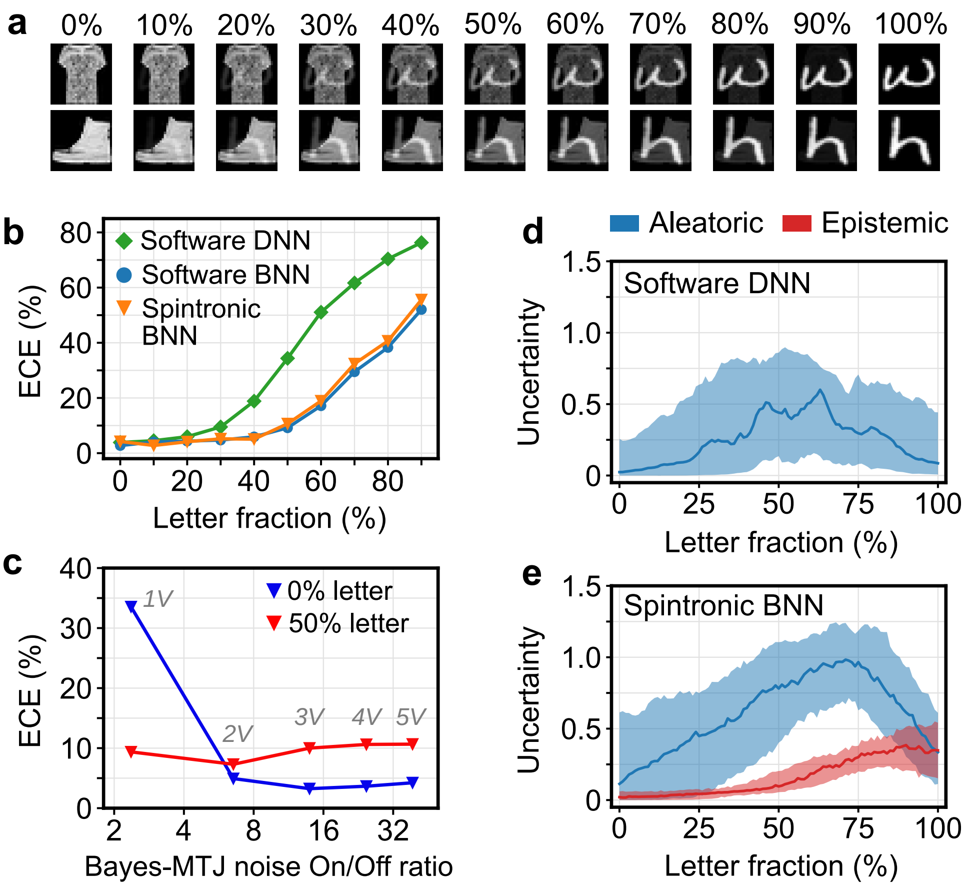


Figure 3-5. (A) Continuous transformation from Fashion MNIST to EMNIST-Letters images by varying the letter fraction, (B) ECE vs. letter fraction for the software DNN, software BNN, and spintronic implementation of the BNN, (C) Dependence of the ECE on the Bayes-MTJ noise On/Off ratio ($\partial G_{BMTJ,max}/\partial G_{BMTJ,min}$). The maximum value of V_{bi} needed to achieve the On/Off ratio is labeled. (D) Uncertainty vs. letter fraction predicted by the DNN, (E) Uncertainty vs. letter fraction predicted by the BNN, decomposed into aleatoric and epistemic uncertainty components. Uncertainties are in units of information entropy and the shaded regions contain the middle 50% of 100 FMNIST-to-Letters transformations tested.

re-sampled 51× per image in the analog accelerator. By contrast, the software (TensorFlow Probability) implementation only resamples weights once per batch of 32 images to reduce RNG overheads. The much more frequent resampling allows for greater cancellation of the noise in the subsequent layer, reducing the overall variance in the network's predictions and leading to greater confidence.

Fig. 3-6D shows that by varying the weighting factor on the KL divergence loss term relative to categorical cross entropy, BNNs can be trained at different points along the trade-off between accuracy and ECE. The ECE does not directly track this hyperparameter but rather has a minimum; the BNN is over-confident to the left of the minimum and under-confident to the right. The ECE minimum lies further to the right for the spintronic implementation. This is because the analog hardware is slightly more confident, so it tends to be well-calibrated where the software BNN is slightly under-confident.

As with Fashion MNIST, uncertainties far away from the training set were evaluated by continuously blending CIFAR-100 images with a different dataset: the Street View House Numbers (SVHN) dataset [81], which uses 32 × 32 RGB images similar to CIFAR-100. The ECE vs SVHN fraction is shown in Fig. 3-6E. The ResNet BNN and its spintronic hardware implementation produce significantly better-calibrated uncertainties on out-of-distribution data than a conventional classification ResNet.

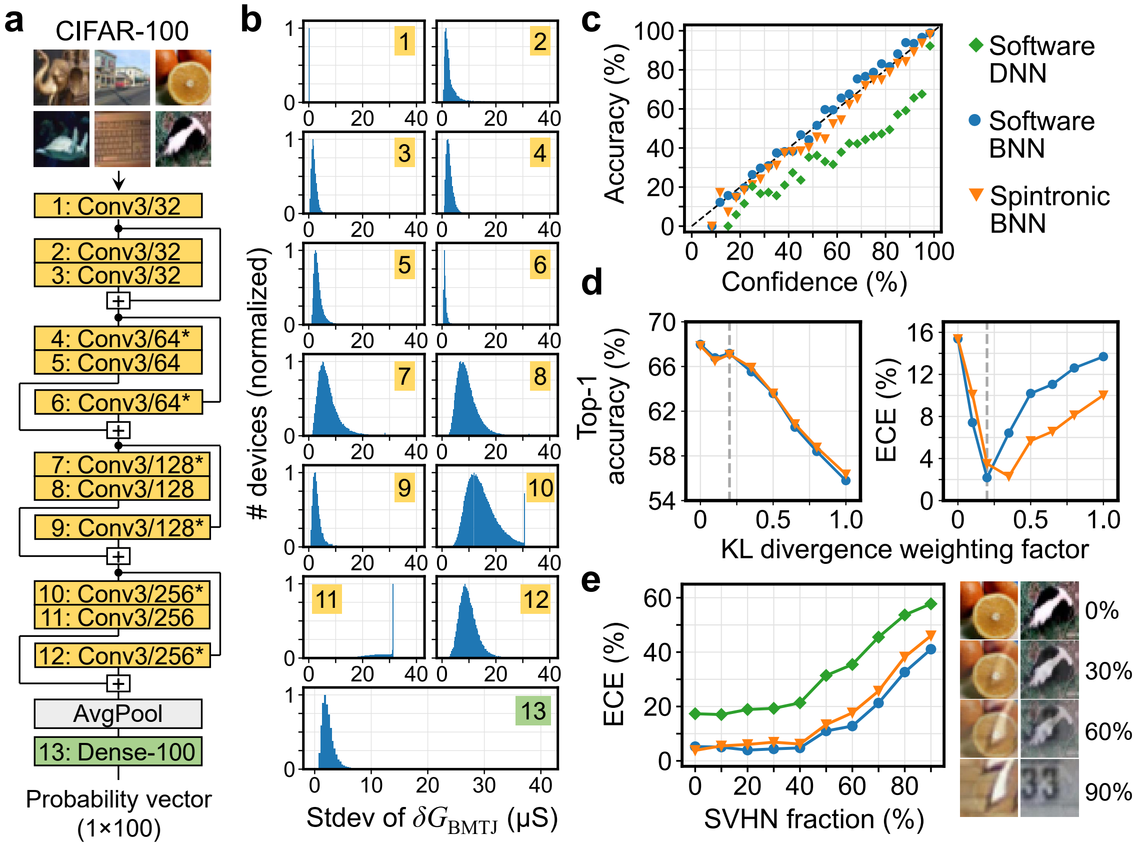


Figure 3-6. (A) ResNet topology for CIFAR-100 image classification, used for both the DNN and BNN. An asterisk denotes a stride of two. Convolutions other than layers 6, 9, and 12 are followed by batch normalization. (B) Distribution of trained σ values for each layer of the ResNet, mapped to Bayes-MTJ conductance noise. (C) Comparison of uncertainty calibration curves for the ResNet DNN, BNN, and BNN simulated on spintronic hardware, for the CIFAR-100 test set. (D) Accuracy and ECE on the CIFAR-100 test set for various BNNs trained with different weighting factors on the KL divergence term of the loss function. Each network is evaluated both in software and simulated on the spintronic accelerator. (E) ECE vs SVHN fraction for CIFAR-100 images continuously mixed with SVHN images. Results in (C) and (E) used a BNN with a KL divergence weighting factor of 0.2 and 100 samples per prediction. Results in (D) are based on 25 samples per prediction.

3.0.5. Quantifying Regression Uncertainty

The proposed spintronic BNN accelerator can also be used to efficiently quantify uncertainty with regression models, where a continuous quantity is predicted rather than a discrete class. We use the Auto MPG dataset [82], where the task is to predict an automobile's fuel efficiency given eight other attributes of the car which can be continuous (e.g. horsepower, weight) or discrete (e.g. model year, number of cylinders). The dataset of 398 cars is divided into 255 training, 64 validation, and 78 test examples. A simple BNN is trained for 500 epochs using VI with three dense layers that have 128, 32, and 1 output, respectively. Unlike the classification case, a negative log-likelihood loss function is used that assumes a normal distribution for the fuel efficiency y:

$$\mathcal{L}(y_{pred}, y_{true}, \sigma_0) = -\log\left[\frac{1}{\sigma_0\sqrt{2\pi}}\exp\left(-\frac{\left(y_{true}-y_{pred}\right)^2}{\sigma_0^2}\right)\right] \tag{3.10}$$

where y_{pred} is the predicted fuel efficiency, y_{true} is the true efficiency, and σ_0 is a hyperparameter that is used to calibrate the estimated uncertainty of the model. For this network topology, which produces point predictions, the corresponding DNN does not provide any uncertainty estimate because the output has no probabilistic interpretation.

The model's predictive uncertainty is obtained by defining confidence intervals (CIs) that contain some percentage of the 1000 BNN point predictions for each input. Fig. 3-7A shows the mean prediction and 90% CIs for the examples in the test set, where blue indicates that the true fuel efficiency lies within the 90% CI. For a model that produces well-calibrated uncertainties, a CI containing α% of the predictions should contain the true output for α% of the test inputs. Fig. 3-7B shows that the BNN gives well-calibrated uncertainties across the full range of CIs (values of α), and the spintronic hardware closely matches the ideal software results.

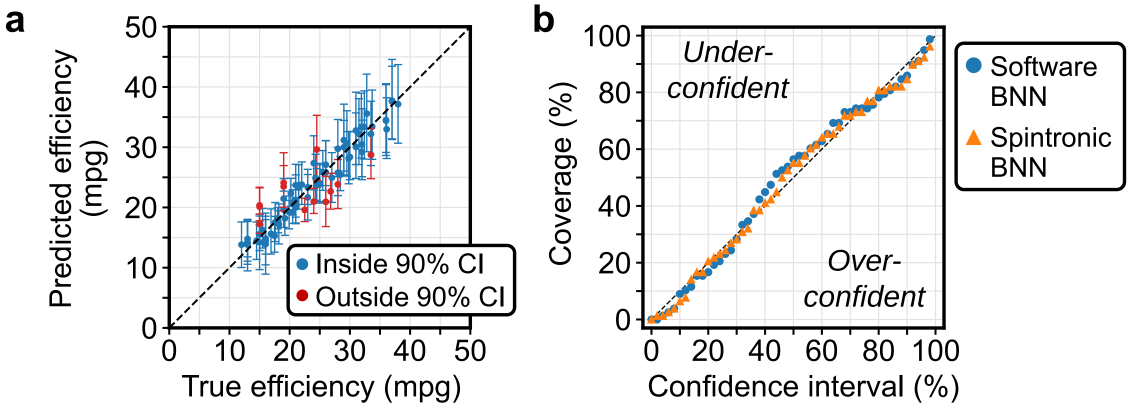


Figure 3-7. (A) Spintronic BNN regression results on the Auto MPG test set, comparing the predicted to true efficiency. Error bars show the 90% confidence interval obtained from sampling 100 BNN predictions. Blue indicates points where the true values lies inside the 90% confidence interval. (B) Calibration curve for the software and spintronic implementation of the regression BNN on the Auto MPG test set.

3.0.6. *Energy efficiency*

Compared to conventional digital implementations of BNNs, the proposed MTJ-based probabilistic MVM engine saves considerable energy by performing multi-bit RNG and multiply-accumulate (MAC) operations using low-voltage magnetic devices in the analog domain. Furthermore, the proposed hardware can be more efficient than previously proposed MTJ-based accelerators [40, 83] by integrating the two functions within the same array, without the need for intermediate digital processing to compute a probabilistic MVM.

Fig. 3-8A shows how the energy consumption per probabilistic MAC operation scales for the proposed spintronic accelerator. Circuit energies were computed based on a 40nm transistor process, assuming 8-bit precision for the analog-to-digital converter (ADC) and shared digital-to-analog converter (DAC). To reduce the current consumption of the CC, MTJs with higher resistance than listed in Tables 1 and 2 are assumed (Bayes-MTJ R_p = 10 kΩ, DW-MTJ R_p = 56 kΩ). We also consider the efficiency of a system that uses the highest MTJ resistances demonstrated in the literature [84] (Bayes-MTJ R_p = 1 MΩ, DW-MTJ R_p = 5.6 MΩ). Since the CC, ADC, or DAC dominate the energy, higher efficiency can be obtained in large arrays where these costs can be amortized over more MACs. Meanwhile, the cost of true RNG in state-of-the-art CMOS circuits is about 1.6 pJ/bit [1], or 6.4 pJ to generate a 4-bit random value that matches the assumed programming precision of the Bayes-MTJ. Multiplication of 4-bit values incurs an additional ∼0.05 pJ/MAC [85]. The spintronic accelerator can yield more than 100× energy improvement at large array sizes.

An energy cost associated with BNNs, whether implemented in digital software or a spintronic accelerator, is the cost of randomly sampling the prediction multiple times. Resampling the noisy weights is needed to produce well-calibrated uncertainties, and also improves accuracy by ensembling the predictions of multiple weight samples. Fig. 3-8B and C show how the accuracy and ECE on Fashion MNIST and CIFAR-100 depend on the number of samples for the spintronic BNN. The number of samples needed for convergence of accuracy and ECE depends on the task, and this number is the overhead factor of a BNN prediction over a DNN prediction on the same analog hardware.

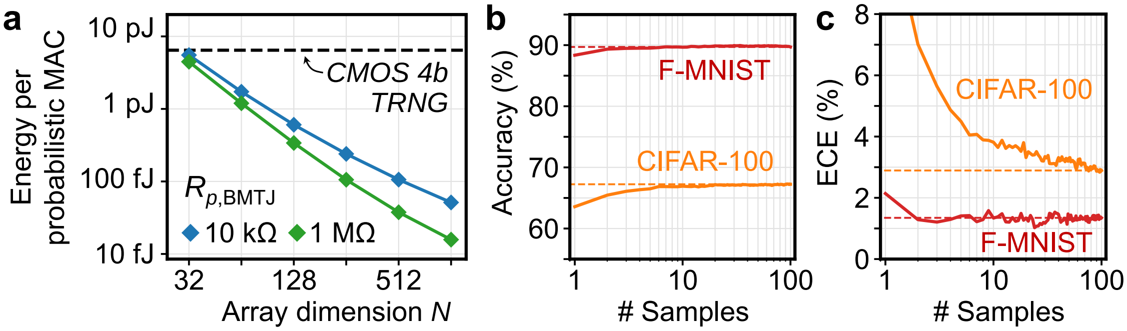


Figure 3-8. (A) Energy consumption per probabilistic MAC operation within an $N \times N$ probabilistic MVM executed by the spintronic in-memory computing accelerator (blue). Two values of the Bayes-MTJ parallel resistance are considered. The black dashed line shows the efficiency of performing the same probabilistic MACs using the CMOS True RNG from Bae *et al.* [1]. (B)-(C) Accuracy and ECE vs. number of sampled predictions from the spintronic BNN on the Fashion MNIST and CIFAR-100 datasets.

Symbol	Parameter	Value
α	Gilbert damping	0.01
M_s	Saturation magnetization	1×10^6 A/m
K_i	Anisotropy energy	0.08 J/m^2
κ_s	VCMA coefficient	75×10^{-15} J/m
P	Spin polarization	0.6
t_{MgO}	MgO thickness	1.5 nm
t_{free}	Free layer thickness	1.5 nm
d	MTJ diameter	50 nm
TMR	Tunnel magnetoresistance	200%
V_h	Voltage where TMR is halved	0.5 V
R_p	Parallel resistance	2 kΩ
T	Temperature	300 K

Table 3-1. Physical parameters used in the macrospin LLG simulations of the Bayes-MTJ.

Symbol	Parameter	Value
α	Gilbert damping	0.02
M_s	Saturation magnetization	8×10^5 A/m
K_u	Perpendicular anisotropy	5×10^5 J/m^3
A	Exchange constant	1.3×10^{-11} J/m
DMI	Dzyaloshinskii-Moriya interaction constant	-0.05 J/m^2
P	Spin polarization	0.7
t_{free}	Free layer thickness	1.5 nm
l	Free layer length	600 nm
w	Free layer width	40 nm
w_n	Notch width	10 nm
d_n	Notch depth	10 nm
s_n	Notch spacing	35 nm
R_p	Parallel resistance	11.1 kΩ
TMR	Tunnel magnetoresistance	200%
T	Temperature	300 K

Table 3-2. Physical parameters used in micromagnetics simulations of the DW-MTJ.

Metric	DNN (software)	BNN (software)	BNN (spintronic)
Accuracy (Fashion MNIST)	90.09%	89.98%	89.53%
ECE (Fashion MNIST)	3.28%	1.54%	0.83%
ECE (50% Letter fraction)	34.35%	12.27%	7.00%
Accuracy (CIFAR-100)	67.98%	63.65%	63.47%
ECE (CIFAR-100)	15.38%	7.59%	3.86%
ECE (50% SVHN fraction)	31.40%	13.62%	9.64%

Table 3-3. Accuracy and expected calibration error of trained networks.

4. RETENTION OF DW SYNAPSES

4.0.1. Methodology for elevated-temperature DW-MTJ simulations

The above probabilistic computing schemes that rely on spintronic synapses, as well as more conventional schemes implementing neural networks with purely scalar weights [26, 32], are only valid in the inference regime assuming that weights programmed remain programmed over time. DW-MTJ synaptic devices benefit from multiple strengths from relying on current-induced nucleation and reaction to external voltage forces, but the underlying physical equations for domain-wall propagation also notably expose them to an increased probability of state drift (DW movement) as temperatures increase. In this subsection of our project, we began exploring DW retention using archetypal domain wall track systems and simulating them at elevated temperatures using MuMax [64].

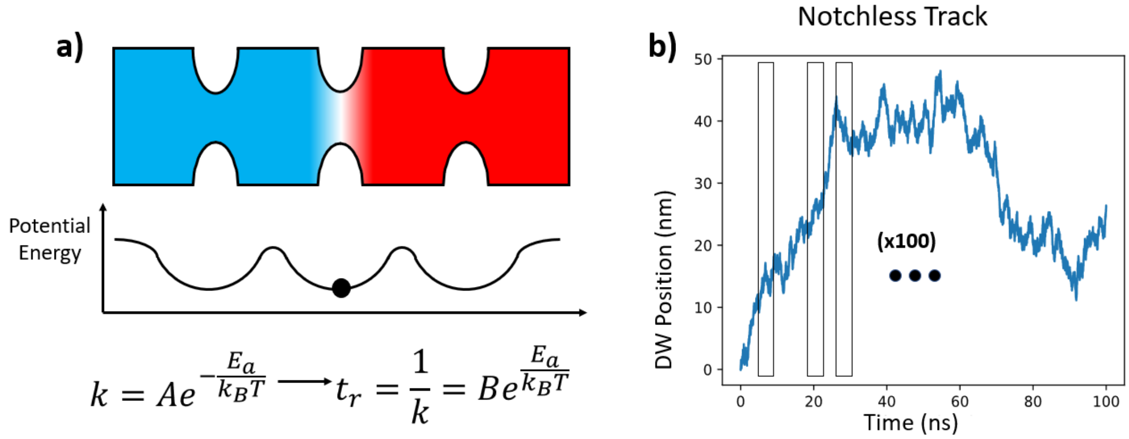


Figure 4-1. (A) Schematic for simulation methods; heavy metal track with structured notches evolves over time where notch represents energy wells and thermal energy can induce motion of the DW (white strip between blue and red regions in P and AP states, respectively). (B) Depiction of DW position in a notchless track demonstrating broader motion. Sampling strategy is indicated; the black rectangles overlaid on the image demonstrate the sampling window size method that is employed in following figures.

As shown in Fig. 4-1A, the overall research objective is to explore how the limits of energy barriers introduced into the DW racetrack (notches). Analytically, the rate of escape from a notch , k, is overall modeled as an Arrhenius-like process. Specifically, it can be modeled with respect to an exponentiated activation energy of the notch (E_a) divided by the Boltzmann constant and the ambient temperature T, where this term must be fit. In turn, the simplified retention time t_r is simply $1/k$. In order to provide a constant lens of analysis over different simulation time lengths, a sampling window approach was utilized where different sampling windows are nested over the total simulation time , as in 4-1B. These windows can be used to calculate a) standard deviation of distance traveled, or b) probabilities of escape from a notch (not relevant in the notchless track case where unlimited travel is allowed.

For the second case specifically, this methodology suggests that , once a DW propagates a certain thresholded distance Δd beyond the center of a given notch N , the detected jump at time Δt can be used to inform and fit predicted retention for that model system. One such example is shown in Fig. 4-2A; in this case, $\Delta t = 85ns$. Meanwhile, Fig. 4-2B demonstrates the relationship between an effective notch case and an unbounded notchless case. As a function of the sampling window , the standard deviation of DW propagation continues to increase in the notchless case, while it plateaus in the notched case (at $T = 300K$). However, in higher temperature scenarios, the notched case will not effectively plateau, leading to the need for additional mechanisms for enhanced retention. These scenarios are discussed in the following section.

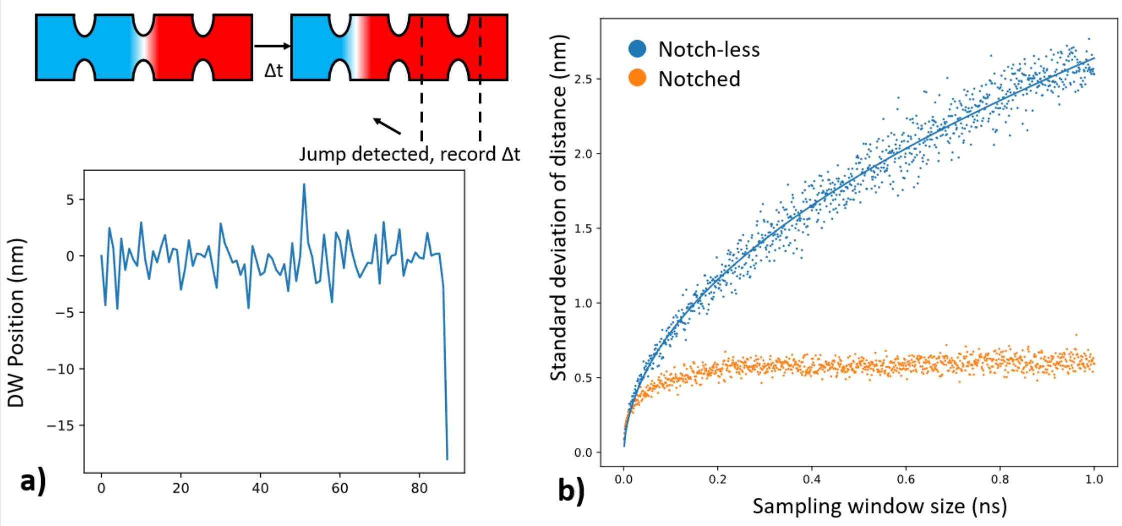


Figure 4-2. (A) Demonstration of a single retention simulation for a notched case where jump from originating notch is detected and recorded in order to inform retention studies. (B) Point cloud data for several hundred individual simulations usng a variety of sampling window widths. This data can help support quality claims for probability of retention at a given temperature. In this case, where $T = 300K$, and anisotropy $K_u = 1e6$, notch width $w_n = 10nm$, notched operation is reliable.

4.0.2. Anisotropy: a method for retention enhancement

At higher ambient temperatures, the odds of a domain wall escape continue to increase to an intolerable level, and the use of anisotropy (stiffness of the magnetic thin film the DW transits through) becomes critical. This dynamic is highlighted in the overall contrast between Fig. 4-3A,B. Specifically, at around room temperature ($T = 300k$) , some anisotropy values for the track such as those around $K_u = 1e6$ or a bit lower allow for adequate retention, and less stiff simulated variants behavior almost equivalent to the notchless cases. In the high temperature scenario, even the stiffest simulated tracks show inadequate retention.

In addition to anisotropy, further notch geometries w_n can be explored in order to further enhance geometry. The aggregate results of merit for retention (e..g predicted retention time t_r given) are shown in

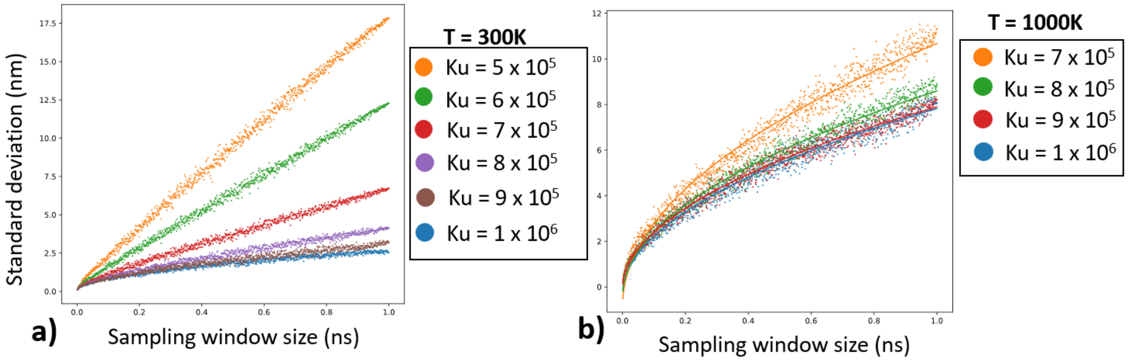


Figure 4-3. (A) Standard deviation computed over various sampling windows in the low temperature scenario with multiple scenarios/tracks simulated for each anisotropy configuration (B) Same results for the high temperature environment.

Fig. 4-4A,B, where (A) explores in depth the temperature dependence of the model system for various notch widths given a sufficiently thick track, and (B) explores the end retention time predicted for different K_u given a set width geometry. In addition, the predicted t_r then allows us to fit the earlier equations. For instance, at the $K_u = 1e6$ case with $w_n = 10nm$, the energy activation energy is $E_a = 2e-19J$, with a fitting paramter of $B = 8.1e-14s$. For the same track width and at $K_u = 5e5$, these values fall to $2.8e-11$, wth a fitting parameter of $B = 8.1e-14s$

The aggregate results suggest that a sufficiently stiff track , with the smallest notch realistically fabricable , could open up the doorway to hundreds of hours of retention or better at room temperature. Immediate next steps including getting actual t_r values at room temperature or slightly above $T = 300-400K$, which unlike these higher T simulations require much longer simulation steps to validate these predictions.

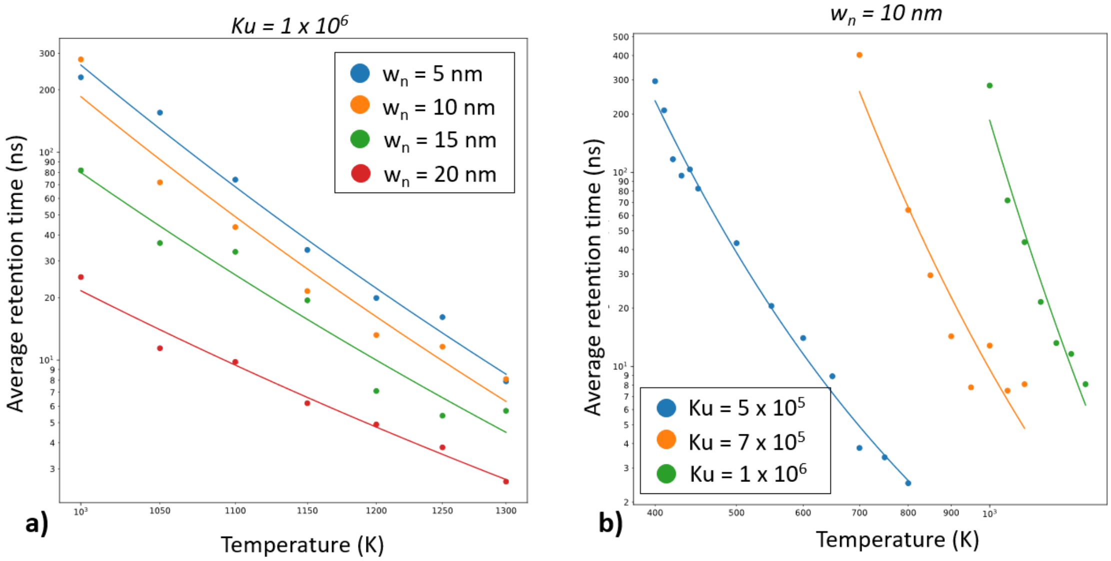


Figure 4-4. (A) Average retention time t_r, averaged over many individual simulation runs, as a function of the temperature simulated . In this pane, anisotropy is held constant and notch width is being varied. (B) The same simulation set-up as (a), but in this case notch width is held constant and notch anisotropy is being varied. As visible, anisotropy has a very significant effect, with larger values e.g. $K_u = 1e-6$ potentially paving the way towards days of retention at RT or slightly elevated above RT temperatures (full simulations for these lower temperatures were not completed due to computational tractability of the simulations).

5. SUMMARY AND PERSPECTIVE

Our results confirm that a Bayes-MTJ noise encoder (programmable standard deviation σ) and a pair of DW-MTJ devices constructing a spintronic synapse (programmable mean μ) can collectively encode expressive probability distributions with sufficient quality for real BNN operations. The two types of devices can be co-integrated within a compact nanofabric, paving the way to one-shot probabilistic matrix-vector multiplications in the analog domain. The proposed hardware can be $10-100\times$ more efficient than performing the same computation using conventional RNGs, and can be made even more so with more resistive MTJ devices. We simulated classification and regression Bayesian neural networks whose trained probabilistic weights are encoded using the novel spintronic technology. Despite device non-idealities (non-Gaussian noise distribution, limited range and precision in representing σ and μ), the spintronic BNN implementation produces well-calibrated and decomposable uncertainty estimates on CIFAR-100, Fashion MNIST, and perturbed versions of these datasets. The spintronic hardware yields high-fidelity accuracy and ECE metrics that are nearly identical or superior to those produced by a software BNN. To demonstrate feasibility on more complex tasks and to relax device programming precision and range requirements, future work will investigate closer co-design of the algorithm and device by integrating device properties into the VI training of the BNN.

Meanwhile, our intial exploration of domain-wall memory retention has us cautiously optimistic about the feasibiliity of synapses that last weeks or longer, even in slightly elevated temperature environments. However, this exploration made it clear that a simple device design that does not intelligently account for the energy barrier of the trapped domain wall will not be able to deliver these metrics. In the next research cycle, further experimental and theoretical explorations of mechanisms for effective and long-lasting anisotropy modification, including VCMA-based circuit modulation ideas , will be critical. Even more exotically, ideas such as magneto-ionics could present a win-win by not only inducing more expressive single synapse behavior, but at the same time solidfying retention into a range that isn't achievable in current DW-MTJ device designs.

BIBLIOGRAPHY

[1] Sang-Geun Bae, Yongtae Kim, Yunsoo Park, and Chulwoo Kim. 3-Gb/s high-speed true random number generator using common-mode operating comparator and sampling uncertainty of D flip-flop. *IEEE Journal of Solid-State Circuits*, 52(2):605–610, 2017. doi: 10.1109/JSSC.2016.2625341.

[2] Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/7180cffd6a8e829dacfc2a31b3f72ece-Paper.pdf.

[3] David J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472, may 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.3.448. URL https://doi.org/10.1162/neco.1992.4.3.448.

[4] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. Hands-on Bayesian neural networks—a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2):29–48, 2022. doi: 10.1109/MCI.2022.3155327.

[5] Ruizhe Cai, Ao Ren, Ning Liu, Caiwen Ding, Luhao Wang, Xuehai Qian, Massoud Pedram, and Yanzhi Wang. Vibnn: Hardware acceleration of bayesian neural networks. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, ASPLOS '18, page 476–488, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450349116. doi: 10.1145/3173162.3173212. URL https://doi.org/10.1145/3173162.3173212.

[6] Ruizhe Cai, Ao Ren, Ning Liu, Caiwen Ding, Luhao Wang, Xuehai Qian, Massoud Pedram, and Yanzhi Wang. Vibnn: Hardware acceleration of bayesian neural networks. *ACM SIGPLAN Notices*, 53(2):476–488, 2018.

[7] Roberto Carboni and Daniele Ielmini. Stochastic memory devices for security and computing. *Advanced Electronic Materials*, 5(9):1900198, 2019.

[8] Abu Sebastian, Manuel Le Gallo, Riduan Khaddam-Aljameh, and Evangelos Eleftheriou. Memory devices and applications for in-memory computing. *Nature Nanotechnology*, 15:529–544, 2020. ISSN 17483395. doi: 10.1038/s41565-020-0655-z. URL http://dx.doi.org/10.1038/s41565-020-0655-z.

[9] T. Patrick Xiao, Christopher H. Bennett, Ben Feinberg, Sapan Agarwal, and Matthew J. Marinella. Analog architectures for neural network acceleration based on non-volatile memory. *Applied Physics Reviews*, 7(3):031301, 2020. doi: 10.1063/1.5143815. URL https://doi.org/10.1063/1.5143815.

[10] Can Li, Miao Hu, Yunning Li, Hao Jiang, Ning Ge, Eric Montgomery, Jiaming Zhang, Wenhao Song, Noraica Dávila, Catherine E Graves, , Zhiyong Li, John Paul Strachan, Peng Lin, Zhongrui Wang, Mark Barnell, Qing Wu, R. Stanley Williams, J. Joshua Yang, and Qiangfei Xia. Analogue signal and image processing with large memristor crossbars. *Nature electronics*, 1(1):52–59, 2018.

[11] Peng Yao, Huaqiang Wu, Bin Gao, Jianshi Tang, Qingtian Zhang, Wenqiang Zhang, J Joshua Yang, and He Qian. Fully hardware-implemented memristor convolutional neural network. *Nature*, 577 (7792):641–646, 2020.

[12] Selina La Barbera, Denys R. B. Ly, Gabriele Navarro, Niccolò Castellani, Olga Cueto, Guillaume Bourgeois, Barbara De Salvo, Etienne Nowak, Damien Querlioz, and Elisa Vianello. Narrow heater bottom electrode-based phase change memory as a bidirectional artificial synapse. *Advanced Electronic Materials*, 4:1800223, 9 2018. ISSN 2199160X. doi: 10.1002/aelm.201800223.

[13] V. Joshi, M. Le Gallo, Simon Haefeli, I. Boybat, S. Nandakumar, C. Piveteau, M. Dazzi, B. Rajendran, A. Sebastian, and E. Eleftheriou. Accurate deep neural network inference using computational phase-change memory. *Nature Communications*, 11, 2020.

[14] Paschalis Gkoupidenis, Nathan Schaefer, Benjamin Garlan, and George G. Malliaras. Neuromorphic functions in PEDOT:PSS organic electrochemical transistors. *Advanced Materials*, 27:7176–7180, 2015. ISSN 15214095. doi: 10.1002/adma.201503674.

[15] Yu-Pu Lin, Christopher H Bennett, Théo Cabaret, Damir Vodenicarevic, Djaafar Chabi, Damien Querlioz, Bruno Jousselme, Vincent Derycke, and Jacques-Olivier Klein. Physical realization of a supervised learning system built with organic memristive synapses. *Scientific reports*, 6(1):1–12, 2016.

[16] Yiyang Li, T. Patrick Xiao, Christopher H. Bennett, Erik Isele, Armantas Melianas, Hanbo Tao, Matthew J. Marinella, Alberto Salleo, Elliot J. Fuller, and A. Alec Talin. In situ parallel training of analog neural network using electrochemical random-access memory. *Frontiers in Neuroscience*, 15, 4 2021. ISSN 1662-453X. doi: 10.3389/fnins.2021.636127.

[17] Dmitry Kireev, Samuel Liu, Harrison Jin, T. Patrick Xiao, Christopher H. Bennett, Deji Akinwande, and Jean Anne Incorvia. Metaplastic and energy-efficient biocompatible graphene artificial synaptic transistors for enhanced accuracy neuromorphic computing. 2022. doi: 10.48550/ARXIV.2203.04389. URL `https://arxiv.org/abs/2203.04389`.

[18] Donald A Robinson, Michael E Foster, Christopher H Bennett, Austin Bhandarkar, Elizabeth R Webster, Aleyna Celebi, Nisa Celebi, Elliot J Fuller, Vitalie Stavila, Catalin D Spataru, et al. Tunable intervalence charge transfer in ruthenium prussian blue analogue enables stable and efficient biocompatible artificial synapses. *arXiv e-prints*, pages arXiv–2207, 2022.

[19] Seungchul Jung, Hyungwoo Lee, Sungmeen Myung, Hyunsoo Kim, Seung Keun Yoon, Soon-Wan Kwon, Yongmin Ju, Minje Kim, Wooseok Yi, Shinhee Han, Baeseong Kwon, Boyoung Seo, Kilho Lee, Gwan-Hyeob Koh, Kangho Lee, Yoonjong Song, Changkyu Choi, Donhee Ham, and Sang Joon Kim. A crossbar array of magnetoresistive memory devices for in-memory computing. *Nature*, 601: 211–216, 1 2022. ISSN 0028-0836. doi: 10.1038/s41586-021-04196-6.

[20] S. Ikeda, K. Miura, H. Yamamoto, K. Mizunuma, H. D. Gan, M. Endo, S. Kanai, J. Hayakawa, F. Matsukura, and H. Ohno. A perpendicular-anisotropy CoFeB–MgO magnetic tunnel junction. *Nature Materials*, 9:721–724, 9 2010. ISSN 1476-1122. doi: 10.1038/nmat2804.

[21] Lin Xue, Chi Ching, Alex Kontos, Jaesoo Ahn, Xiaodong Wang, Renu Whig, Hsin wei Tseng, James Howarth, Sajjad Hassan, Hao Chen, Mangesh Bangar, Shurong Liang, Rongjun Wang, and Mahendra Pakala. Process optimization of perpendicular magnetic tunnel junction arrays for last-level cache beyond 7 nm node. pages 117–118. IEEE, 6 2018. ISBN 978-1-5386-4218-4. doi: 10.1109/VLSIT.2018.8510642.

[22] J. Grollier, D. Querlioz, K. Y. Camsari, K. Everschor-Sitte, S. Fukami, and M. D. Stiles. Neuromorphic spintronics. *Nature Electronics*, 3:360–370, 2020. ISSN 25201131. doi: 10.1038/s41928-019-0360-9. URL http://dx.doi.org/10.1038/s41928-019-0360-9.

[23] E. Raymenants, O. Bultynck, D. Wan, T. Devolder, K. Garello, L. Souriau, A. Thiam, D. Tsvetanova, Y. Canvel, D. E. Nikonov, I. A. Young, M. Heyns, B. Soree, I. Asselberghs, I. Radu, S. Couet, and V. D. Nguyen. Nanoscale domain wall devices with magnetic tunnel junction read and write. *Nature Electronics*, 4:392–398, 6 2021. ISSN 2520-1131. doi: 10.1038/s41928-021-00593-x.

[24] Otitoaleke Akinola, Xuan Hu, Christopher H. Bennett, Matthew Marinella, Joseph S. Friedman, and Jean Anne C. Incorvia. Three-terminal magnetic tunnel junction synapse circuits showing spike-timing-dependent plasticity. *Journal of Physics D: Applied Physics*, 52, 2019. ISSN 13616463. doi: 10.1088/1361-6463/ab4157.

[25] Saima A. Siddiqui, Sumit Dutta, Astera Tang, Luqiao Liu, Caroline A. Ross, and Marc A. Baldo. Magnetic domain wall based synaptic and activation function generator for neuromorphic accelerators. *Nano Letters*, 20:1033–1040, 2020. ISSN 15306992. doi: 10.1021/acs.nanolett.9b04200.

[26] Thomas Leonard, Samuel Liu, Mahshid Alamdar, Can Cui, Otitoaleke G. Akinola, Lin Xue, T. Patrick Xiao, Joseph S. Friedman, Matthew J. Marinella, Christopher H. Bennett, and Jean Anne C. Incorvia. Shape-dependent multi-weight magnetic artificial synapses for neuromorphic computing. 2021. doi: 10.48550/ARXIV.2111.11516. URL https://arxiv.org/abs/2111.11516.

[27] Wesley H. Brigner, Naimul Hassan, Xuan Hu, Christopher H. Bennett, Felipe Garcia-Sanchez, Can Cui, Alvaro Velasquez, Matthew J. Marinella, Jean Anne C. Incorvia, and Joseph S. Friedman. Domain wall leaky integrate-and-fire neurons with shape-based configurable activation functions. *IEEE Transactions on Electron Devices*, 69:2353–2359, 5 2022. ISSN 0018-9383. doi: 10.1109/TED.2022.3159508.

[28] Priyamvada Jadaun, Can Cui, Sam Liu, and Jean Anne C. Incorvia. Adaptive cognition implemented with a context-aware and flexible neuron for next-generation artificial intelligence. 2020. doi: 10.48550/ARXIV.2010.15748. URL https://arxiv.org/abs/2010.15748.

[29] Kyung Mee Song, Jae Seung Jeong, Biao Pan, Xichao Zhang, Jing Xia, Sunkyung Cha, Tae Eon Park, Kwangsu Kim, Simone Finizio, Jörg Raabe, Joonyeon Chang, Yan Zhou, Weisheng Zhao, Wang Kang, Hyunsu Ju, and Seonghoon Woo. Skyrmion-based artificial synapses for neuromorphic computing. *Nature Electronics*, 3:148–155, 2020. ISSN 25201131. doi: 10.1038/s41928-020-0385-0. URL http://dx.doi.org/10.1038/s41928-020-0385-0.

[30] Abhronil Sengupta, Priyadarshini Panda, Parami Wijesinghe, Yusung Kim, and Kaushik Roy. Magnetic tunnel junction mimics stochastic cortical spiking neurons. *Scientific Reports*, 6:30039, 9 2016. ISSN 2045-2322. doi: 10.1038/srep30039.

[31] Gopalakrishnan Srinivasan, Abhronil Sengupta, and Kaushik Roy. Magnetic tunnel junction based long-term short-term stochastic synapse for a spiking neural network with on-chip STDP learning. *Scientific Reports*, 6:29545, 9 2016. ISSN 2045-2322. doi: 10.1038/srep29545.

[32] Samuel Liu, T. Patrick Xiao, Can Cui, Jean Anne C. Incorvia, Christopher H. Bennett, and Matthew J. Marinella. A domain wall-magnetic tunnel junction artificial synapse with notched geometry for accurate and efficient training of deep neural networks. *Applied Physics Letters*, 118: 202405, 5 2021. ISSN 0003-6951. doi: 10.1063/5.0046032. URL `https://aip.scitation.org/doi/10.1063/5.0046032`.

[33] K. Hayakawa, S. Kanai, T. Funatsu, J. Igarashi, B. Jinnai, W.A. Borders, H. Ohno, and S. Fukami. Nanosecond random telegraph noise in in-plane magnetic tunnel junctions. *Physical Review Letters*, 126:117202, 3 2021. ISSN 0031-9007. doi: 10.1103/PhysRevLett.126.117202.

[34] William A. Borders, Ahmed Z. Pervaiz, Shunsuke Fukami, Kerem Y. Camsari, Hideo Ohno, and Supriyo Datta. Integer factorization using stochastic magnetic tunnel junctions. *Nature*, 573:390–393, 9 2019. ISSN 0028-0836. doi: 10.1038/s41586-019-1557-9.

[35] Vaibhav Ostwal and Joerg Appenzeller. Spin–orbit torque-controlled magnetic tunnel junction with low thermal stability for tunable random number generation. *IEEE Magnetics Letters*, 10:1–5, 2019. ISSN 1949-307X. doi: 10.1109/LMAG.2019.2912971.

[36] Jialin Cai, Bin Fang, Like Zhang, Wenxing Lv, Baoshun Zhang, Tiejun Zhou, Giovanni Finocchio, and Zhongming Zeng. Voltage-controlled spintronic stochastic neuron based on a magnetic tunnel junction. *Physical Review Applied*, 11:034015, 3 2019. ISSN 2331-7019. doi: 10.1103/PhysRevApplied.11.034015.

[37] Christopher Safranski, Jan Kaiser, Philip Trouilloud, Pouya Hashemi, Guohan Hu, and Jonathan Z. Sun. Demonstration of nanosecond operation in stochastic magnetic tunnel junctions. *Nano Letters*, 21:2040–2045, 3 2021. ISSN 1530-6984. doi: 10.1021/acs.nanolett.0c04652.

[38] Jan Kaiser, William A. Borders, Kerem Y. Camsari, Shunsuke Fukami, Hideo Ohno, and Supriyo Datta. Hardware-aware in situ learning based on stochastic magnetic tunnel junctions. *Physical Review Applied*, 17:014016, 1 2022. ISSN 2331-7019. doi: 10.1103/PhysRevApplied.17.014016.

[39] Kezhou Yang, Akul Malhotra, Sen Lu, and Abhronil Sengupta. All-spin bayesian neural networks. *IEEE Transactions on Electron Devices*, 67(3):1340–1347, 2020.

[40] Anni Lu, Yandong Luo, and Shimeng Yu. An algorithm-hardware co-design for Bayesian neural network utilizing SOT-MRAM's inherent stochasticity. *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, 8(1):27–34, 2022. doi: 10.1109/JXCDC.2022.3177588.

[41] Yudeng Lin, Qingtian Zhang, Jianshi Tang, Bin Gao, Chongxuan Li, Peng Yao, Zhengwu Liu, Jun Zhu, Jiwu Lu, Xiaobo Sharon Hu, He Qian, and Huaqiang Wu. Bayesian neural network realization by exploiting inherent stochastic characteristics of analog RRAM. In *2019 IEEE International Electron Devices Meeting (IEDM)*, pages 14.6.1–14.6.4, 2019. doi: 10.1109/IEDM19573.2019.8993616.

[42] Thomas Dalgaty, Eduardo Esmanhotto, Niccolo Castellani, Damien Querlioz, and Elisa Vianello. Ex situ transfer of Bayesian neural networks to resistive memory-based inference hardware. *Advanced Intelligent Systems*, 3(8):2000103, 2021. doi: https://doi.org/10.1002/aisy.202000103. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202000103.

[43] Akul Malhotra, Sen Lu, Kezhou Yang, and Abhronil Sengupta. Exploiting oxide based resistive ram variability for bayesian neural network hardware design. *IEEE Transactions on Nanotechnology*, 19: 328–331, 2020.

[44] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. doi: 10.1080/01621459.2017.1285773. URL https://doi.org/10.1080/01621459.2017.1285773.

[45] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/blundell15.html.

[46] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*, 2018.

[47] L Baldrati, AJ Tan, M Mann, R Bertacco, and GSD Beach. Magneto-ionic effect in cofeb thin films with in-plane and perpendicular-to-plane magnetic anisotropy. *Applied Physics Letters*, 110(1):012404, 2017.

[48] Punyashloka Debashis, Rafatul Faria, Kerem Y. Camsari, Joerg Appenzeller, Supriyo Datta, and Zhihong Chen. Experimental demonstration of nanomagnet networks as hardware for Ising computing. pages 34.3.1–34.3.4. IEEE, 12 2016. ISBN 978-1-5090-3902-9. doi: 10.1109/IEDM.2016.7838539.

[49] Yoichi Shiota, Shinichi Murakami, Frédéric Bonell, Takayuki Nozaki, Teruya Shinjo, and Yoshishige Suzuki. Quantitative evaluation of voltage-induced magnetic anisotropy change by magnetoresistance measurement. *Applied Physics Express*, 4:043005, 3 2011. ISSN 1882-0778. doi: 10.1143/APEX.4.043005.

[50] Peisen Li, Aitian Chen, Dalai Li, Yonggang Zhao, Sen Zhang, Lifeng Yang, Yan Liu, Meihong Zhu, Huiyun Zhang, and Xiufeng Han. Electric field manipulation of magnetization rotation and tunneling magnetoresistance of magnetic tunnel junctions at room temperature. *Advanced Materials*, 26:4320–4325, 7 2014. ISSN 09359648. doi: 10.1002/adma.201400617.

[51] Kaili Zhang, Deming Zhang, Chengzhi Wang, Lang Zeng, You Wang, and Weisheng Zhao. Compact modeling and analysis of voltage-gated spin-orbit torque magnetic tunnel junction. *IEEE Access*, 8: 50792–50800, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2980073.

[52] Viola Krizakova, Eva Grimaldi, Kevin Garello, Giacomo Sala, Sebastien Couet, Gouri Sankar Kar, and Pietro Gambardella. Interplay of voltage control of magnetic anisotropy, spin-transfer torque, and heat in the spin-orbit-torque switching of three-terminal magnetic tunnel junctions. *Physical Review Applied*, 15:054055, 5 2021. ISSN 2331-7019. doi: 10.1103/PhysRevApplied.15.054055.

[53] Aitian Chen, Yan Wen, Bin Fang, Yuelei Zhao, Qiang Zhang, Yuansi Chang, Peisen Li, Hao Wu, Haoliang Huang, Yalin Lu, Zhongming Zeng, Jianwang Cai, Xiufeng Han, Tom Wu, Xi-Xiang Zhang, and Yonggang Zhao. Giant nonvolatile manipulation of magnetoresistance in magnetic tunnel junctions by electric fields via magnetoelectric coupling. *Nature Communications*, 10:243, 12 2019. ISSN 2041-1723. doi: 10.1038/s41467-018-08061-5.

[54] Mei Fang, Sangjian Zhang, Wenchao Zhang, Lu Jiang, Eric Vetter, Ho Nyung Lee, Xiaoshan Xu, Dali Sun, and Jian Shen. Nonvolatile multilevel states in multiferroic tunnel junctions. *Physical Review Applied*, 12:044049, 10 2019. ISSN 2331-7019. doi: 10.1103/PhysRevApplied.12.044049.

[55] Jiawei Wang, Aitian Chen, Peisen Li, and Sen Zhang. Magnetoelectric memory based on ferromagnetic/ferroelectric multiferroic heterostructure. *Materials*, 14:4623, 8 2021. ISSN 1996-1944. doi: 10.3390/ma14164623.

[56] Uwe Bauer, Lide Yao, Aik Jun Tan, Parnika Agrawal, Satoru Emori, Harry L. Tuller, Sebastiaan van Dijken, and Geoffrey S. D. Beach. Magneto-ionic control of interfacial magnetism. *Nature Materials*, 14:174–181, 2 2015. ISSN 1476-1122. doi: 10.1038/nmat4134.

[57] L. Baldrati, A. J. Tan, M. Mann, R. Bertacco, and G. S. D. Beach. Magneto-ionic effect in CoFeB thin films with in-plane and perpendicular-to-plane magnetic anisotropy. *Applied Physics Letters*, 110: 012404, 1 2017. ISSN 0003-6951. doi: 10.1063/1.4973475.

[58] Aik Jun Tan, Mantao Huang, Can Onur Avci, Felix Büttner, Maxwell Mann, Wen Hu, Claudio Mazzoli, Stuart Wilkins, Harry L. Tuller, and Geoffrey S. D. Beach. Magneto-ionic control of magnetism using a solid-state proton pump. *Nature Materials*, 18:35–41, 1 2019. ISSN 1476-1122. doi: 10.1038/s41563-018-0211-5.

[59] Fen Xue, Noriyuki Sato, Chong Bi, Jun Hu, Jinliang He, and Shan X. Wang. Large voltage control of magnetic anisotropy in CoFeB/MgO/OX structures at room temperature. *APL Materials*, 7: 101112, 10 2019. ISSN 2166-532X. doi: 10.1063/1.5101002.

[60] Yingfen Wei, Sylvia Matzen, Cynthia P. Quinteros, Thomas Maroutian, Guillaume Agnus, Philippe Lecoeur, and Beatriz Noheda. Magneto-ionic control of spin polarization in multiferroic tunnel junctions. *npj Quantum Materials*, 4:62, 12 2019. ISSN 2397-4648. doi: 10.1038/s41535-019-0201-0.

[61] Martin Nichterwitz, Shashank Honnali, Jonas Zehner, Sebastian Schneider, Darius Pohl, Sandra Schiemenz, Sebastian T. B. Goennenwein, Kornelius Nielsch, and Karin Leistner. Control of positive and negative magnetoresistance in iron oxide–iron nanocomposite thin films for tunable magnetoelectric nanodevices. *ACS Applied Electronic Materials*, 2:2543–2549, 8 2020. ISSN 2637-6113. doi: 10.1021/acsaelm.0c00448.

[62] Guofei Long, Qian Xue, Qiang Li, Yu Shi, Lin Li, Long Cheng, Peng Li, Junwei Zhang, Xixiang Zhang, Haizhong Guo, Jing Fu, Shandong Li, Jagadeesh S. Moodera, and Guo-Xing Miao. Interfacial control via reversible ionic motion in battery-like magnetic tunnel junctions. *Advanced Electronic Materials*, 7:2100512, 9 2021. ISSN 2199-160X. doi: 10.1002/aelm.202100512.

[63] Sungjung Joo, K. Y. Jung, B. C. Lee, Tae-Suk Kim, K. H. Shin, Myung-Hwa Jung, K.-J. Rho, J.-H. Park, Jinki Hong, and K. Rhie. Effect of oxidizing the ferromagnetic electrode in magnetic tunnel junctions on tunneling magnetoresistance. *Applied Physics Letters*, 100:172406, 4 2012. ISSN 0003-6951. doi: 10.1063/1.4704557.

[64] Arne Vansteenkiste, Jonathan Leliaert, Mykola Dvornik, Mathias Helsen, Felipe Garcia-Sanchez, and Bartel Van Waeyenberge. The design and verification of mumax3. *AIP Advances*, 4, 2014. ISSN 21583226. doi: 10.1063/1.4899186. URL http://dx.doi.org/10.1063/1.4899186.

[65] J. A. Currivan-Incorvia, S. Siddiqui, S. Dutta, E. R. Evarts, J. Zhang, D. Bono, C. A. Ross, and M. A. Baldo. Logic circuit prototypes for three-terminal magnetic tunnel junctions with mobile domain walls. *Nature Communications*, 7:3–9, 2016. ISSN 20411723. doi: 10.1038/ncomms10275.

[66] G. Catalan, J. Seidel, R. Ramesh, and J. F. Scott. Domain wall nanoelectronics. *Reviews of Modern Physics*, 84:119–156, 2 2012. ISSN 0034-6861. doi: 10.1103/RevModPhys.84.119.

[67] A. Chavent, V. Iurchuk, L. Tillie, Y. Bel, N. Lamard, L. Vila, U. Ebels, R.C. Sousa, B. Dieny, G. di Pendina, G. Prenat, J. Langer, J. Wrona, and I.L. Prejbeanu. A multifunctional standardized magnetic tunnel junction stack embedding sensor, memory and oscillator functionality. *Journal of Magnetism and Magnetic Materials*, 505:166647, 7 2020. ISSN 03048853. doi: 10.1016/j.jmmm.2020.166647.

[68] John H. Lau. Recent advances and new trends in flip chip technology. *Journal of Electronic Packaging*, 138, 9 2016. ISSN 1043-7398. doi: 10.1115/1.4034037.

[69] Matthew J. Marinella, Sapan Agarwal, Alexander Hsia, Isaac Richter, Robin Jacobs-Gedrim, John Niroula, Steven J. Plimpton, Engin Ipek, and Conrad D. James. Multiscale co-design analysis of energy, latency, area, and accuracy of a ReRAM analog neural training accelerator. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 8(1):86–101, 2018. doi: 10.1109/JETCAS.2018.2796379.

[70] T. Patrick Xiao, Christopher H. Bennett, Ben Feinberg, Matthew J. Marinella, and Sapan Agarwal. CrossSim: accuracy simulation of analog in-memory computing, online: https://github.com/sandialabs/cross-sim. 2022. URL https://github.com/sandialabs/cross-sim.

[71] Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 560–569. AUAI Press, 2018. URL http://auai.org/uai2018/proceedings/papers/207.pdf.

[72] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

[73] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS'17, page 5580–5590, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[74] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[75] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[76] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. EMNIST: Extending MNIST to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[77] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org, 2017.

[78] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[79] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[80] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2704–2713, June 2018.

[81] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL `http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf`.

[82] J. Ross Quinlan. Combining instance-based and model-based learning. In *Proceedings of the Tenth International Conference on International Conference on Machine Learning*, ICML'93, page 236–243, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc. ISBN 1558603077.

[83] Kezhou Yang, Akul Malhotra, Sen Lu, and Abhronil Sengupta. All-spin Bayesian neural networks. *IEEE Transactions on Electron Devices*, 67(3):1340–1347, 2020.

[84] J. Doevenspeck, K. Garello, B. Verhoef, R. Degraeve, S. Van Beek, D. Crotti, F. Yasin, S. Couet, G. Jayakumar, I. A Papistas, P. Debacker, R. Lauwereins, W. Dehaene, G. S. Kar, S. Cosemans, A. Mallik, and D. Verkest. SOT-MRAM based analog in-memory computing for DNN inference. In *2020 IEEE Symposium on VLSI Technology*, pages 1–2, 2020. doi: 10.1109/VLSITechnology18217.2020.9265099.

[85] Mark Horowitz. Computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 10–14, 2014. doi: 10.1109/ISSCC.2014.6757323.

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Technical Library	1911	sanddocs@sandia.gov

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop

Hardcopy—External

Number of Copies	Name(s)	Company Name and Company Mailing Address
1		

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.