2024 Design, Automation & Test in Europe Conference (DATE 2024) – Multi-Partner Project

NeuSpin: Design of a Reliable Edge Neuromorphic System Based on Spintronics for Green AI

Soyed Tuhin Ahmed[‡‡†], Kamal Danouchi[‡], Guillaume Prenat[‡], Lorena Anghel[‡], Mehdi B. Tahoori[†]
[†]Karlsruhe Institute of Technology, Karlsruhe, Germany, [‡‡]corresponding author, email: soyed.ahmed@kit.edu
[‡]Univ. Grenoble Alpes, CEA, CNRS, Grenoble INP, and IRIG-Spintec, Grenoble, France

Abstract—**Internet of Things (IoT) and smart wearable devices for personalized healthcare will require storing and computing ever-increasing amounts of data. The key requirements for these devices are ultra-low-power, high-processing capabilities, autonomy at low cost, as well as reliability and accuracy to enable Green AI at the edge. Artificial Intelligence (AI) models, especially Bayesian Neural Networks (BayNNs) are resource-intensive and face challenges with traditional computing architectures due to the memory wall problem. Computing-in-Memory (CIM) with emerging resistive memories offers a solution by combining memory blocks and computing units for higher efficiency and lower power consumption. However, implementing BayNNs on CIM hardware, particularly with spintronic technologies, presents technical challenges due to variability and manufacturing defects. The NeuSPIN project aims to address these challenges through full-stack hardware and software co-design, developing novel algorithmic and circuit design approaches to enhance the performance, energy-efficiency and robustness of BayNNs on sprintronic-based CIM platforms.**

Index Terms—**Bayesian NNs, Spintronics, Computation-In-Memory (CIM)**

I. Introduction

Artificial Intelligence's utilization of cloud computing presently accounts for 76% of the overall traffic in data centers [1]. Projections indicate a fivefold surge of this percentage from 2016 to 2021. By 2025, it is anticipated to constitute nearly 20% of the world's electricity consumption [2]. The emergence of new applications in personalized healthcare through the Internet of Things (IoT) and smart wearable devices will demand the storage and processing of progressively larger data volumes within extremely low-power systems.

Neural networks (NNs), an AI algorithm, have revolutionized various fields, offering powerful tools for pattern recognition, decision-making, and predictive analytics. However, their predictions are often overconfident, even when they are wrong or receive out-of-distribution (OOD) data. Bayesian Neural Networks (BayNNs), a subclass of NNs, overcome the crucial drawback of NNs by offering the ability to provide uncertainty estimates in predictions. The predictive uncertainty of an NN model is particularly important for critical decision-making scenarios, such as autonomous driving, medical image analysis, and industrial predictive maintenance. However, the deployment of BayNN faces significant challenges due to its inherent hardware requirements. Specifically, the 'memory wall' problem is a concern when implemented in traditional Von Neumann architectures, due to separate computing units and memory blocks. This bottleneck significantly limits the efficiency and scalability of BayNNs in practical high-demand applications.

Neuromorphic computing, emulating the brain's functions through direct neural network and synaptic array implementation, presents superior processing capabilities in AI tasks when compared to classical microprocessors. Integrating neuromorphic hardware into Green AI inherently resolves data transfer challenges by reducing reliance on cloud data transfer. This integration heralds a new paradigm poised to permeate various IT sectors and facilitate the creation of genuinely autonomous devices.

Computing-in-Memory (CIM) architectures, as a hardware realization of neuromorphic computing, have emerged as a promising solution to overcome some of these limitations. In CIM architectures, the multiply and accumulate (MAC) operation of NN can be carried out where data already exists, eliminating the memory wall problem. In particular, the integration of spintronics-based technologies into CIM offers extremely low power consumption, high endurance, and high computational efficiency [3], while showing compatibility with existing semiconductor processes. However, such emerging advanced technologies are not without their challenges. Non-idealities like manufacturing variations and defects, as weel as stochastic behavior of spintronic memories add layers of difficulties to the implementation of BayNNs on CIM architectures.

The NeuSPIN project (**currently in the final stage**) sets out to confront the challenge of migrating cloud-based AI algorithms to On-Chip processing, emphasizing high accuracy and minimal energy consumption, primarily leveraging spintronic technology as the chosen Non-Volatile Memory (NVM) approach. Through innovative neuron and synapse designs coupled with Bayesian Neural Networks as the core AI model, the project aims to pioneer a dependable neuromorphic framework for energy-efficient AI processing at the edge. This initiative involves optimizing material stacks and device parameters specifically for AI applications, integrating spintronics technology into the fabric to minimize power consumption, and developing technology-aware statistical training algorithms. Moreover, the collaboration between the esteemed German - Karlsruhe Institute of Technology (KIT) - and the French - University of Grenoble Alpes (UGA) - universities aims to solidify a leading partnership in Edge-AI research, positioning Europe as a front-runner in AI development at the edge.

The project is presented in the next sections. The following section II describes the background on spintronic memories, uncertainty estimation, BayNNs, and discusses related papers. Afterwards, in Section II-D, challenges associated with BayNN integration into CIM are discussed. Then, in Section III several proposed approaches are discussed. In Section IV, key takeaways of NeuSpin project are discussed. Lastly, Section V concludes the paper.

II. Background

A. Spintronics Tenchnology

The fundamental component of magnetic memory (MRAM) lies in the Magnetic Tunnel Junction (MTJ), composed of two ferromagnetic layers: the Free Layer (FL) and the Reference Layer (RL), divided by a thin insulating layer. The device's overall resistance state is determined by the relative magnetization of these layers, whether Parallel (P) or Anti-Parallel (AP).

979-8-3503-4859-0/DATE24/© 2024 EDAA

In MRAM technology, two distinct switching mechanisms exist: Spin Transfer Torque (STT), prevalent in STT-MRAM, and Spin-Orbit Torque (SOT), employed in SOT-MRAM. Unlike STT-MRAM's two-terminal design, SOT-MRAM functions as a three-terminal device. While both offer high endurance and swift switching, SOT-MRAM stands out by segregating read and write paths within its structure. This segregation enables SOT-MRAM to provide tunable resistances, adjustable to several MΩ. Such a distinctive capability holds significant promise, especially in Matrix-Vector Multiplication operations within crossbar arrays.

SOT-MRAM, comprising an MTJ mounted on a heavy metal substrate, allows also for the integration of multiple MTJs on the same layer, simulating a multi-value cell. Moreover, both STT and SOT devices exhibit stochastic switching behavior, meaning that given specific current amplitudes and durations passing through or beneath the MTJ, the switching process occurs with a certain probability. Consequently, these devices serve a dual role, functioning not just as memory storage but also potentially serving as sources of randomness.

B. Uncertainty Estimation

In NNs, uncertainty estimation is a critical process in which the network provides a measure of confidence of the prediction (predictive uncertainty) along with the prediction itself. The predictive uncertainty reflects the model's belief in its output based on the given input data and its learned parameters [4]. The ability to estimate uncertainty in its prediction allows NNs to acknowledge the limits of their knowledge.

Uncertainty in deep learning is primarily divided into two types: aleatoric and epistemic. Aleatoric uncertainty, also called data uncertainty, is irreducible even if the model revives more data. This is because aleatoric uncertainty arises due to the inherent noise (randomness or variability) present in the data generation process. In contrast, epistemic uncertainty, also called model uncertainty, arises from incomplete knowledge within the model or data. This type of uncertainty can be reduced with additional data or with a more comprehensive understanding of the sources of uncertainty. The ability to estimate epistemic uncertainty empowers NNs to recognize the boundaries of their knowledge.

C. Bayesian Neural Networks

BayeNNs represent a significant shift from traditional NN paradigms during training and inference, as they embrace a probabilistic approach. Specifically, conventional NNs are described by a deterministic function $\mathbf{f}:\mathbb{R}^D\times\mathbb{R}^P\rightarrow\mathbb{R}^C$, where D, P, and C represent the dimensions of inputs $\mathbf{x}$, learnable parameters $\boldsymbol{\theta}$, and outputs $\mathbf{f}(\mathbf{x},\boldsymbol{\theta})$, respectively. In BayNN, the parameters $\boldsymbol{\theta}$ are not single point estimates but rather random variables, following a distribution $\boldsymbol{\theta}\sim p(\boldsymbol{\theta})$. The learning objective comprises the Bayesian framework for computing the posterior distribution $p(\boldsymbol{\theta}\mid\mathcal{D})$ given the observed data $\mathcal{D}$. Unfortunately, the exact computation of this posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ is intractable, especially when dealing with complex and complex high-dimensional models such as NNs. This is due to the high-dimensional integral calculation (over all possible values of $\boldsymbol{\theta}$) in the denominator of Bayes' theorem:

$$p(\mathcal{D})=\int p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})\,d\boldsymbol{\theta}.$$

As a result, direct computation of the integral in closed form becomes practically impossible, and we resort to approximation techniques such as variational inference (VI) and Monte Carlo dropout (MC-dropout). VI introduces a variational distribution $q_{\boldsymbol{\omega}}(\boldsymbol{\theta})\approx p(\boldsymbol{\theta}\mid\mathcal{D})$, often a Gaussian with diagonal covariance, to approximate the true posterior. The parameters ($\boldsymbol{\omega}$) of this distribution are optimized to minimize the Kullback-Leibler divergence with the true posterior, leading to an efficient approximation method.

On the other hand, MC-Dropout utilizes dropout layers not only during training but, more importantly, also during inference. Thus, MC-Dropout effectively simulates samples from an approximate posterior distribution.

D. Limitations and Challenges of BayNNs for CIM Implementation

Conventional BayNNs typically have either 32-bit floating-point precision [5] or quantized to high-bit precision, e.g., 8- to 16-bit precision [6], for their implications. However, since spintronic memories have only two stable states, regardless of which approximation is used for BayNNs, low-precision quantization of their parameters or learned statistical distributions is required. As a result, the quantization error is a concern for uncertainty estimates and inference accuracy.

In terms of VI, the main challenge lies in a) implementing the posterior distribution in the CIM, b) efficiently sampling from it for the inference step, as described in [7], and c) memory consumption. Specifically, on-the-fly sampling can lead to high computational costs at run time. It can be described as random sampling $\boldsymbol{\theta}\sim q_{\boldsymbol{\omega}}(\boldsymbol{\theta})$ followed by forward passing of the given $\mathbf{x}^*$ through the NN with $\boldsymbol{\theta}$ as its parameters.

On the other hand, for MC-Dropout, the main challenges lie in a) the realization of the dropout module in CIM, b) the sampling latency, and c) the shear number of dropout modules required. Specifically, conventional MC-Dropout applies dropout to each neuron of a layer, and another approach (MC-DropConnect) applies to each weight. Since the number of neurons and weights in an NN can be millions, the number of Dropout modules in the hardware can be huge and the overall sampling latency can be long.

The NeuSpin project, as a result, is focused on full-stack optimization, from the algorithm all the way to the circuit level, to solve the above-mentioned challenges.

E. Related Works

Previous studies have investigated hardware solutions for Bayesian Neural Networks, which are usually based on traditional CMOS technology. In the paper [8], a novel FPGA implementation is proposed that utilizes quadratic nonlinear activation functions. Work in [9] implemented an FPGA-based design aimed at accelerating Bayesian neural networks by employing an intermediate-layer caching technique. The approach described in [10] involves a CIM implementation in which the crossbar array stores the variance parameter and stochastic resistive (RRAM) devices are used to sample the probability distribution at the input of the array. In [11], the authors leverage the imperfections present in RRAM devices to implement Bayesian learning techniques In [12], the authors introduced a stochastic selector to facilitate weight dropout for neural sampling. The research in [13] showed the application of an ensemble of resistive crossbar arrays to store probabilistic weights for implementing BayNN. In Yang et al. [14], crossbar arrays were used to construct Bayesian neural networks with the help of low-barrier MTJs. The paper in [15] presented an alternative implementation with MRAM-based crossbar arrays that can represent mean and variance.

Our project aims at minimizing the reliance on Random Number Generators for Bayesian neural networks while proposing In-Memory computation to reduce excessive power consumption caused by frequent data transfers between memory and the core unit [16]. To achieve this, we adopt a hardware-software co-design approach, proposing architectural solutions involving fewer stochastic units and CIM architecture. Moreover, we leverage all spintronics functionalities in our hardware implementation to achieve a low-power design. Overall, the proposed approaches mitigate the inherent stochasticity of the spintronic devices by considering them as a feature rather than a foe.

III. Scalable Spintronic-Based Bayesian NNs

As mentioned, BayNNs are the principal methods for estimating predictive uncertainty in a machine learning application. Despite this, the computational cost and power consumption make the use of BayNNs on embedded hardware (with limited resources) is unattractive. Conventional BayNNs suffer from the *resource scalability* issue. As the size of the models and datasets increases, the computational and memory requirements also increase. In the case of BayNN implementations, this also means that the number of Random Number Generators (RNGs), to be realized by stochastic spintronic devices, and memory consumption to store model parameters also increases with model size. For example, in MC-Dropout [5] and MC-DropConnect [17], the number of Dropout modules equals the total number of weights and activations in the model. Similarly, the memory consumption of certain VI and ensemble implementations can be $2-10\times$ higher. This means that achieving the desired performance with minimal energy consumption, memory usage, and latency is challenging for BayNNs. Efficient resource scalability is particularly important in resource-constrained environments with edge AI, such as mobile devices or embedded systems. Scalable approaches for BayNN realization based on spintronic technology, as a part of the NeuSpin project, are described below.

A. Dropout Based BayNN

MC-Dropout typically has a lower memory consumption compared to other approximation methods, as it has the same number of point estimation parameters as a conventional NN. However, they have their own challenges, as mentioned earlier. We present the dropout-based BayNN approaches considered in the NeuSpin project in the next sections.

1) SpinDrop: In this project, we introduce for **the first time the binary Bayesian Neural Network (BinBayNN)** [18], [19] utilizing the inherent stochastic properties of spintronic devices for the implementation of the dropout module and the deterministic properties of BayNN weight storage for in-memory Bayesian inference. To implement dropout-based BayNN, each neuron in the array was equipped with a dedicated dropout module, which allowed the potential dropout of a neuron based on a specified probability. To enable control over the current and, consequently, the probability of the MTJ, CMOS transistors were integrated with the MTJ. The process involved generating a bitstream by alternating SET and RESET operations. Following a ”SET” write operation, the MTJ’s state was read using a sense amplifier to verify the occurrence of the switch, effectively indicating the dropout signal. Post-read operation, the MTJ was ”RESET” to the P-state in preparation for the next round of dropout signal generation [18], [19]. In classic BNNs, the standard matrix-vector multiplications are replaced with XNOR operations. Thus, it is necessary to design the spintronic-based bit cell that allows for this particular operation. The proposed architecture is organized around the MTJ crossbar array, in which each trained weight is stored in a unit represented by two 1T-1MTJ cells. In addition, a word-line decoder is used with the capability to enable multiple consecutive addresses. A SpinDrop module, described in the previous section, is connected to each pair of word-line to implement the dropout concept.

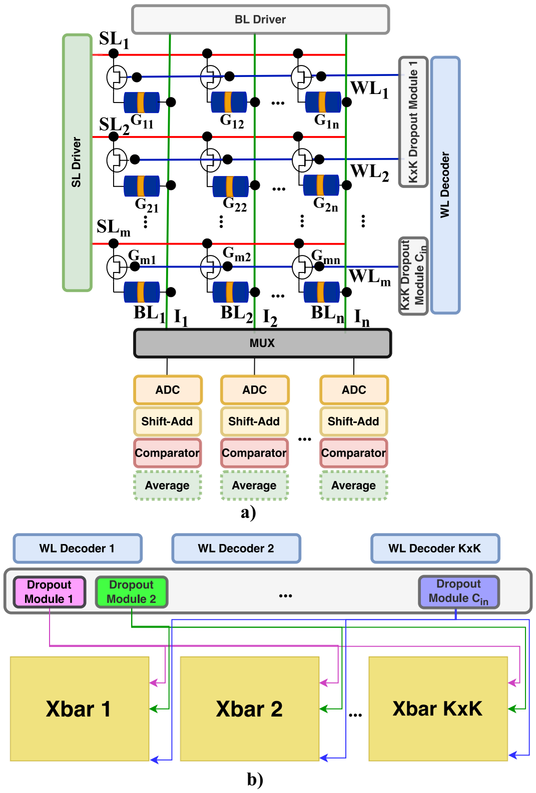

Fig. 1. Crossbar design for the MC-SpatialDropout based on mapping strategy (a) ① and (b) strategy②. In (b), only the Dropout module and WL decoder are shown. Everything else is abstracted.

A specifically designed learning objective is proposed. Evaluation of the proposed approach shows up to 100% detection of out-of-distribution data, an improvement in accuracy of $\sim 2\%$, and up to 15% for corrupted data.

2) Spatial-SpinDrop: However, one drawback of the SpinDrop structure is that it requires a considerable number of dropout modules per layer, typically one for each neuron. Although the dropout module can be reused, the number of dropout modules still increases with network size. Also, such an implementation is not adapted for convolutional layers.

Expanding on the SpinDrop concept, we developed the MC-SpatialDropout based [20] BayNN to reduce the number of dropout modules and allow easier implementation of the dropout module for convolutional layers. In MC-SpatialDropout, instead of the conventional dropout module that randomly disables neurons, spatial dropout is used. Spatial dropout drops entire feature maps, making it more suitable for CNNs where spatial correlations are vital. Furthermore, this method reduces the complexity of circuit design for implementing dropout in CNNs. The overall effort to co-design the algorithm and hardware leads to a reduction in the number of dropout modules per network by a factor of $9\times$ and energy consumption by $94.11\times$, while maintaining comparable predictive performance and uncertainty estimates, the architecture is depicted in Fig.1.

In addition, two prevalent mapping strategies are explored for mapping the convolutional layer with SpatialSpinDrop. In the first strategy (①), every kernel, shaped as $K \times K \times C_{in}$,

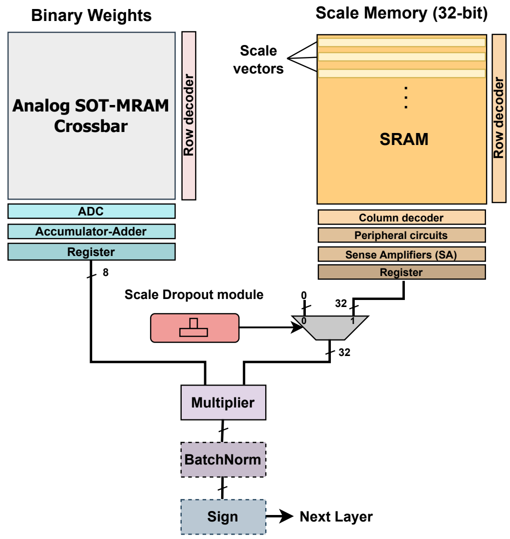

Fig. 2. Proposed inference architecture for Scale-Dropout.

is unfolded into a crossbar column, as outlined in [21].On the contrary, the second strategy (②) involves mapping each kernel onto smaller crossbars of $K \times K$ dimensions, with a shape of $C_{in} \times C_{out}$, as detailed in [22]. As a result, the Dropout module needs to be generalizable to different mapping strategies, which is not the case with the SpinDrop concept. Thus, the MC-SpatialDropout approach is $2.94\times$ more energy efficient than the SpinDrop concept presented earlier.

3) SpinScaleDrop: To further improve the efficiency of BayNN, we introduced a novel dropout approach called SpinScaleDrop [23]. In the scale dropout approach, a *scalar dropout mask* is applied to the scale vector by scale modulation rather than information zeroing for each layer. Thus, **only a single dropout module is per layer**. The scale vector is defined as a learnable parameter through gradient descent, and its shape is specifically designed to be suitable for the CIM architecture and to have low memory overhead. SpinScale-Dropout introduces randomness in the scale vector and, thus, the activation of a layer, potentially reducing co-adaptation between the scale vector and binary weights.

Furthermore, a *layer-dependent adaptive scale dropout* approach is proposed that selects the dropout probability based on the parameter size of the layer. Thus, extensive design-space exploration is mitigated to find the optimal dropout probability and location of the dropout layer for each model. The proposed BayNN with Scale Dropout can be trained using stochastic gradient descent. The training process involves sampling a scaled network for each forward pass, and the gradients for each parameter are averaged over the training instances of each mini-batch. In terms of using scale-dropout as a Bayesian approximation, the learning objective involves a novel regularization function for the scale vector to encourage it to be positive and centered around one. This regularization approach aligns well with the nature of binary NN, where the weights are binarized to ± 1.

Model uncertainty can be obtained by performing multiple forward passes with an enabled scale dropout during Bayesian inference. Each forward pass generates a stochastic scale vector, allowing Monte Carlo estimates of the model's predictive mean and uncertainty.

In terms of hardware implementation, spintronics-based scale dropout is modeled differently due to the manufacturing and in-field variation of the MTJs in the spintronic-based Scale-Drop module. Specifically, the dropout probability is defined as a stochastic variable, and the dropout probability is fitted to a Gaussian distribution. Such a module allows for the stochastic activation of the Scale vector that is stored in a neighboring memory. Furthermore, a novel CIM architecture is proposed that incorporates a SOT-MRAM crossbar array, an SRAM memory for scale vector storage, and a spintronics-based scale dropout module for stochastic activation, see Fig.2.

In terms of results, we have demonstrated up to 1% improvement in predictive performance. Furthermore, our approach showed more than $100\times$ energy savings compared to existing methods.

4) Inverted Normalization with Affine Dropout: The approach introduces a novel normalization and dropout method to enhance the robustness and inference accuracy of BayNNs, **a self-healing BayNN**, particularly in CIM architectures [24].

In traditional normalization techniques, such as batch or layer normalization, normalization is first performed, and then an optional affine transformation is performed. Unlike the traditional approach, the inverted normalization layer treats the affine parameters (β and γ) as similar to the weights and biases of the NN, that is, their primary objective is to minimize loss via gradient descent. Furthermore, the affine transformation is *necessary* and performed before normalization, ensuring that the learning process remains stable.

Furthermore, the Affine Dropout approach is introduced, which adds stochasticity to the weighted sum by randomly dropping the weight and bias of the inverted normalization layer with probability p. The implementation involves sampling two binary dropout masks, one for weight and the other for bias of Affine Dropout, from a Bernoulli distribution. Dropout masks are kept at a scalar value (vector-wise dropout) instead of a vector (element-wise dropout) to reduce the number of RNGs in the model. The sampled masks are then multiplied by the weights and biases of the inverted normalization layer, setting the dropped weights and biases to ones and zeros, respectively.

In terms of Bayesian inference and uncertainty estimation, the proposed affine dropout acts as an alternative to the conventional dropout in a Bayesian setting. Thus, during Bayesian inference, multiple forward passes through the network with independently sampled weights and bias masks for each layer result in a stochastic output distribution. The output distribution is treated as an approximation of the Gaussian process, following the theoretical foundation of existing work [5].

The evaluation of the approach shows an improvement in inference accuracy by up to 55.62%. In addition, in specific models such as LSTM-based time series prediction, the root mean square error (RMSE) score is reduced by up to 46.7%. Furthermore, the proposed method is capable of detecting up to 55.03% and 78.95% of OOD instances for uniform noise and random rotation experiments, respectively. As a result, an improvement of 14.61% is achieved compared to previous work.

B. Variational Inference (VI) Based

In terms of inference accuracy and quality of uncertainty, Variational Inference (VI) typically offers a more precise inference and a more accurate representation of the uncertainty. However, the challenges lie in their implementation in the CIM architecture as a result of their distributional parameters and on-the-fly sampling requirements. In this section, we briefly discuss the full-stack optimization approaches for VI-based BayNN from NeuSpin.

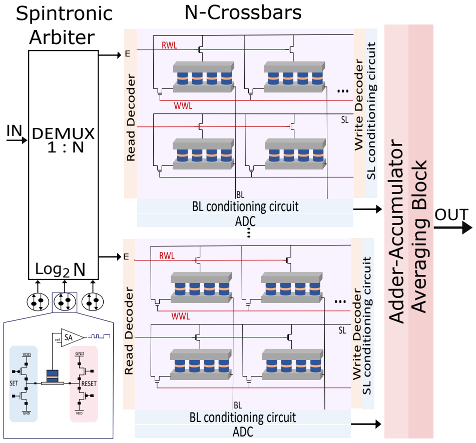

Fig. 3. Proposed layer architecture

1) Bayesian Sub-Set Parameter Inference: In conventional VI-based BayNNs, the variational distributions are typically applied to weights or neuron activation. Consequently, memory consumption and computational complexity are high, as weights represent a large proportion (more than 90%) of memory utilization. Furthermore, implementation challenges in CIM-based hardware due to limited stable states are a factor, as VI-based BayNNs are not suitable for binary weights.

In this project, we introduce *Bayesian subset parameter inference*, an efficient and scalable Bayesian NN framework with both deterministic and stochastic parameters [25]. Larger parameter groups (e.g., weights) are kept deterministic, while Bayesian treatment is only applied to the small parameter group, e.g., scale vector. Specifically, Bayesian treatment is applied to learn the distribution of the scale vector, while other parameters are learned via maximum likelihood. Consequently, our approach is the **first binary VI-based BayNNs framework with spintronic-based CIM implementation**.

We proposed a novel CIM architecture with two separate crossbars per layer: one for deterministic weights and the other for the Bayesian scale. Furthermore, a multi-level device composed of multiple MTJs is implemented to quantitatively represent Bayesian parameters. Additionally, the stochastic behavior of the SOT devices is utilized as a random number generator for sampling.

Experimental results show that the proposed approach has comparable accuracy to full-precision models while estimating uncertainty efficiently. The effectiveness in handling out-of-distribution data is demonstrated by the increase in negative log-likelihood (NLL) under dataset shifts. Furthermore, the proposed approach requires up to 70× lower power consumption and 158.7× lower storage memory requirements compared to traditional methods.

2) SpinBayes: In this project [7], we introduce *Bayesian in-memory approximation* for efficiently mapping the posterior distribution of BayNN to spintronic-based CIM architecture. The main idea behind the approach is to create a memory-friendly distribution that can be efficiently mapped and sampled by CIM hardware. Furthermore, to enable efficient sampling from the mapped posterior distribution on CIM hardware, a novel network topology named ”SpinBayes” is introduced.

The SpinBayes topology presented in Fig. 3 is tailored for CIM architectures leveraging spintronics and incorporates an Arbiter for each layer. This spintronic structure is built upon a multi-value SOT bit-cell crossbar array. Within a layer, the spintronic stochastic Arbiter is implemented at the periphery of crossbars, selecting specific crossbars for Bayesian inference in each forward pass. The Arbiter generates a random binary one-hot vector to determine the selection.

Furthermore, the proposed approximation approach introduces CIM- and spintronics-aware post-training quantization. In terms of the memory cells in the crossbar, we proposed novel MTJ-based multi-value cells for quantized weight storage. In our proposed design, a certain number of conductance levels can be programmed. We perform design-time exploration to optimize bit-precision, followed by MTJ-based multi-value design.

The proposed method is evaluated in classification tasks with up to 100 classes and semantic segmentation tasks on two safety-critical tasks: medical image diagnosis and automotive scene understanding tasks. Compared to SOTA methods, improvements in classification accuracy of up to 1.14% and uncertainty estimation of up to 20.16% are shown. Furthermore, the proposed approach can detect up to 100% samples from several out-of-distribution datasets.

IV. Key Takeaways and Future Works

The key takeaways from the NeuSpin project can be summarized as 1) **Effective Detection of Out-of-Distribution Data**: In multiple studies, we find that BayNNs have shown the ability to detect up to 100% out-of-distribution data, highlighting their robustness in handling unexpected or noisy input. 2) **Improvement in Inference Accuracy for Corrupted Data**: We find that the implementation of Bayesian methods has a significant improvement in inference accuracy, especially in scenarios involving corrupted data. This improvement is critical in applications where data integrity cannot always be guaranteed. 3) **Incorporation of Spintronic Technologies**: The use of spintronic-based memories, particularly in CIM architectures, is advantageous in terms of lower energy consumption and switching speed, which are essential for efficient hardware implementations of BNNs. Also, by using their stochastic regime, RNG implementation is possible using the same technology. 4) **Modeling defects in Devices**: Understanding and accounting for modeling defaults in devices is crucial. These defaults might include inherent variations, imperfections, or manufacturing irregularities in the spintronic-based devices used for BayNN implementations. Incorporating these defaults into the modeling process helps in creating more accurate representations of the hardware and leads to better algorithm-hardware co-design strategies. 5) **Algorithm-Hardware Co-Design**: The trend towards algorithm-hardware co-design, where Bayesian algorithms are specifically tailored to leverage the features of modern hardware architectures like spintronics and CIM, enables more efficient and practical implementations of BayNNs. 6) **Quantized BayNNs**: Quantization or binarization of parameters and activations in BayNNs is a key strategy not only to reduce memory overhead but also to overcome limited states of spintronics memories. Thus, some of the inherent costs of BayNNs can be reduced, making them more feasible for deployment in resource-constrained safety-critical applications. 7) **Resource-aware Methods for Bayesian Approximation**: Novel dropout techniques, such as Scale Dropout and Affine Dropout, and memory-centric approximations allow not only the reduction of the number of RNGs but also the efficient implementation of BayNN. 8) **Inherent Robustness and Self--healing**: BayNNs are inherently robust to variations and faults compared to conventional NNs. Their self-healing property can be further enhanced with specifically designed techniques such as inverted normalization with Affine Dropout.

TABLE I
COMPARISON OF PROPOSED METHODS

	Method	Inference accuracy (%) for VGG	OOD	Energy
Dropout Based	SpinDrop	91.95%	–	2.00 μJ/Image
	Spatial-SpinDrop	90.34%	99.99 %	0.68 μJ/Image
	SpinScaleDropout	90.45%	95.36 %	0.18 μJ/Image
Variational Inference Based	Bayesian Sub-Set Parameter	90.62%	–	0.30 μJ/Image
	SpinBayes	–	99.86%	0.26 μJ/Image

V. CONCLUSION

Bayesian NNs are inherently capable of estimating uncertainty in predictions, overcoming the limitations of conventional NNs. However, they are inherently resource-hungry in terms of power, memory consumption, latency, and the number of RNGs per model. Some of their costs can be reduced by implementing them in a spintronics-based CIM architecture. However, implementing BayNN in a CIM architecture is not straightforward and has several challenges. The objective of NeuSpin project, as a French-German collaborative project, is full stack optimization, from algorithm, BayNN topology, to circuit design, to optimize BayNN training, inference, and CIM implementation using spintronic non-volatile technology. This can greatly reduce hardware footprint while providing energy efficiency and uncertainty estimation for critical edge AI application. Key achievements include the ability to detect up to 100% of out-of-distribution data, consistently improve inference accuracy, especially in the presence of corrupted data, and enhance robustness against non-ideal properties of spintronics. Additionally, a notable breakthrough is achieved with the integration of algorithmic innovation with advanced hardware technologies such as spintronics and CIM architectures, leading to substantial reductions in energy consumption (up to 100×) and memory overhead (158.7×). Consequently, our novel approaches have paved the way for efficient, reliable, and scalable BayNNs in various real-world and safety-critical applications. The collaboration between the two partners has been a success so far, thanks to the complementary competences ans skills of the two teams, but also thanks to the possibility to use realistic data and parameters of spontronic devices fabricated and characterized in the SPINTEC facility.

REFERENCES

[1] U. Gupta *et al.*, "Chasing carbon: The elusive environmental footprint of computing," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2021, pp. 854–867.

[2] C. Magazzino *et al.*, "Investigating the link among ict, electricity consumption, air pollution, and economic growth in eu countries," *Energy Sources, Part B: Economics, Planning, and Policy*, vol. 16, no. 11-12, pp. 976–998, 2021.

[3] T. Lee *et al.*, "World-most energy-efficient mram technology for non-volatile ram applications," in *2022 IEDM*. IEEE, 2022, pp. 10–7.

[4] A. Nguyen *et al.*, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[5] Y. Gal *et al.*, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[6] R. Cai *et al.*, "Vibnn: Hardware acceleration of bayesian neural networks," *ACM SIGPLAN Notices*, vol. 53, no. 2, pp. 476–488, 2018.

[7] S. T. Ahmed *et al.*, "SpinBayes: Algorithm-Hardware Co-Design for Uncertainty Estimation Using Bayesian In-Memory Approximation on Spintronic-Based Architectures," *ACM Transactions on Embedded Computing Systems*, vol. 22, no. 5s, pp. 131:1–131:25, Sep. 2023. [Online]. Available: https://doi.org/10.1145/3609116

[8] H. Awano *et al.*, "Bynqnet: Bayesian neural network with quadratic activations for sampling-free uncertainty estimation on fpga," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 1402–1407.

[9] H. Fan *et al.*, "Fpga-based acceleration for bayesian convolutional neural networks," *IEEE TCAD*, vol. 41, no. 12, pp. 5343–5356, 2022.

[10] A. Malhotra *et al.*, "Exploiting oxide based resistive ram variability for bayesian neural network hardware design," *IEEE Transactions on Nanotechnology*, vol. 19, pp. 328–331, 2020.

[11] T. Dalgaty *et al.*, "In situ learning using intrinsic memristor variability via markov chain monte carlo sampling," *Nature Electronics*, vol. 4, no. 2, pp. 151–161, 2021.

[12] S. Dutta *et al.*, "Neural sampling machine with stochastic synapse allows brain-like learning and inference," *Nature Communications*, vol. 13, no. 1, p. 2571, Dec. 2022. [Online]. Available: https://www.nature.com/articles/s41467-022-30305-8

[13] D. Bonnet *et al.*, "Bringing uncertainty quantification to the extreme-edge with memristor-based bayesian neural networks," *Researchsquare*, 2023.

[14] K. Yang *et al.*, "All-spin bayesian neural networks," *IEEE Transactions on Electron Devices*, vol. 67, no. 3, pp. 1340–1347, 2020.

[15] A. Lu *et al.*, "An Algorithm-Hardware Co-Design for Bayesian Neural Network Utilizing SOT-MRAM's Inherent Stochasticity," *IEEE-JXCDC*, 2022.

[16] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 10–14.

[17] A. Mobiny *et al.*, "Dropconnect is effective in modeling uncertainty of bayesian deep networks," *Scientific reports*, 2021.

[18] S. T. Ahmed *et al.*, "Binary bayesian neural networks for efficient uncertainty estimation leveraging inherent stochasticity of spintronic devices," in *NANOARCH'22: 17th ACM International Symposium on Nanoscale Architectures*. ACM, 2022, pp. 1–6.

[19] ——, "Spindrop: Dropout-based bayesian binary neural networks with spintronic implementation," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 13, no. 1, pp. 150–164, 2023.

[20] ——, "Spatial-spindrop: Spatial dropout-based binary bayesian neural network with spintronics implementation," *arXiv preprint arXiv:2306.10185*, 2023.

[21] T. Gokmen *et al.*, "Training deep convolutional neural networks with resistive cross-point devices," *Frontiers in neuroscience*, vol. 11, p. 538, 2017.

[22] X. Peng *et al.*, "Optimizing weight mapping and data flow for convolutional neural networks on rram based processing-in-memory architecture," in *IEEE ISCAS*. IEEE, 2019, pp. 1–5.

[23] S. T. Ahmed *et al.*, "Scale-dropout: Estimating uncertainty in deep neural networks using stochastic scale," *arXiv preprint arXiv:2311.15816*, 2023.

[24] ——, "Enhancing reliability of neural networks at the edge: Inverted normalization with stochastic affine transformations," in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2024, pp. 1–6.

[25] ——, "Scalable spintronics-based bayesian neural network for uncertainty estimation," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2023, pp. 1–6.