FeBiM: Efficient and Compact Bayesian Inference Engine Empowered with Ferroelectric In-Memory Computing

Chao Li[1], Zhicheng Xu[2], Bo Wen[2], Ruibin Mao[2], Can Li[2], Thomas Kämpfe[3], Kai Ni[4] and Xunzhao Yin[1,5*]
[1]College of Information Science and Electronic Engineering, Zhejiang University
[2]Department of Electrical and Electronic Engineering, The University of Hong Kong; [3]Center Nanoelectric Technologies, Fraunhofer IPMS
[4]Electrical Engineering Department, University of Notre Dame; [5]Key Laboratory of CS&AUS of Zhejiang Province
*Corresponding author email: xzyin1@zju.edu.cn

ABSTRACT

In scenarios with limited training data or where explainability is crucial, conventional neural network-based machine learning models often face challenges. In contrast, Bayesian inference-based algorithms excel in providing interpretable predictions and reliable uncertainty estimation in these scenarios. While many state-of-the-art in-memory computing (IMC) architectures leverage emerging non-volatile memory (NVM) technologies to offer unparalleled computing capacity and energy efficiency for neural network workloads, their application in Bayesian inference is limited. This is because the core operations in Bayesian inference, i.e., cumulative multiplications of prior and likelihood probabilities, differ significantly from the multiplication-accumulation (MAC) operations common in neural networks, rendering them generally unsuitable for direct implementation in most existing IMC designs. In this paper, we propose FeBiM, an efficient and compact Bayesian inference engine powered by multi-bit ferroelectric field-effect transistor (FeFET)-based IMC. FeBiM effectively encodes the trained probabilities of a Bayesian inference model within a compact FeFET-based crossbar. It maps quantized logarithmic probabilities to discrete FeFET states. As a result, the accumulated outputs of the crossbar naturally represent the posterior probabilities, i.e., the Bayesian inference model's output given a set of observations. This approach enables efficient in-memory Bayesian inference without the need for additional calculation circuitry. As the first FeFET-based in-memory Bayesian inference engine, FeBiM achieves an impressive storage density of 26.32 Mb/mm^2 and a computing efficiency of 581.40 TOPS/W in a representative Bayesian classification task. These results demonstrate 10.7×/43.4× improvement in compactness/efficiency compared to the state-of-the-art hardware implementation of Bayesian inference.

1 INTRODUCTION

In-memory computing (IMC) has recently emerged as a promising solution to address the memory wall issues in conventional von Neumann hardware [1]. Leveraging the compactness and high energy efficiency of emerging non-volatile memory (NVM) technologies, many leading IMC accelerators have achieved impressive computing efficiency and throughput for data-intensive machine learning models, particularly neural networks (NNs) [2–5]. While conventional NN-based algorithms are widely used, they often struggle in situations where training data is insufficient or when interpretable results are needed [6, 7]. As a compelling alternative, Bayesian inference is particularly well-suited in low-data scenarios, providing explainable results with uncertainty estimation [8, 9].

The primary posterior calculation in Bayesian inference, which involves the cumulative product of prior and likelihood probabilities as per Bayes' theorem, however, differs from the multiply-and-accumulate (MAC) operations common in NN workloads. This difference renders Bayesian inference usually unsuitable for direct implementation with many existing IMC designs that typically focus on NN acceleration. In traditional complementary metal-oxide-semiconductor (CMOS)-based von Neumann implementations for Bayesian inference, such as CPU [10], GPU [11] and field-programmable gate array (FPGA) [12], accessing separate memory units for stored probabilities incurs significant area and energy overhead. Efforts to exploit the non-volatility and energy efficiency of emerging devices have led to the development of Bayesian inference prototypes utilizing random number generators (RNGs) built with magnetic tunnel junction (MTJ) [13], memtransistor [14] and magnetic random-access memory (MRAM) [15]. These implementations, however, are limited to Bayesian inference with binary evidence/events, and do not effectively address probability storage, rather generating required probabilities on demand, which is energy-consuming. A memristor-based Bayesian machine has been proposed [16], using near-memory stochastic computing to reduce memory access overhead. Yet, these implementations still require additional CMOS logic and multiple clock cycles for posterior calculations and complex sensing circuitry for final inference, thus compromising computing density and inference efficiency.

To address the aforementioned challenges in hardware implementation of Bayesian inference, we propose FeBiM, an efficient and compact in-memory Bayesian inference engine utilizing multi-level cell (MLC) ferroelectric field-effect transistors (FeFETs). The key contributions of this work are summarized as follows:

- We propose a compact crossbar array design using one FeFET per cell as probability storage unit and a compact and scalable winner-take-all (WTA) circuit for sensing. This multi-bit FeFET array enables efficient in-memory Bayesian inference in just one clock cycle, eliminating the need for extra calculation circuitry.
- We introduce a novel mapping scheme that associates quantized logarithmic probabilities with discrete FeFET states. This scheme enables the output currents of the crossbar to naturally represent the posterior probabilities, i.e., the cumulative product of priors and likelihoods given a set of observations.
- We thoroughly investigate the functionality, scalability and application level performance of FeBiM. In a representative Bayesian classification task, our proposed design shows a 10.7×/43.4× storage density/inference efficiency improvement compared to the state-of-the-art Bayesian machine.

DAC '24, June 23–27, 2024, San Francisco, CA, USA

ACM ISBN 979-8-4007-0601-1/24/06…$15.00
https://doi.org/10.1145/3649329.3656538

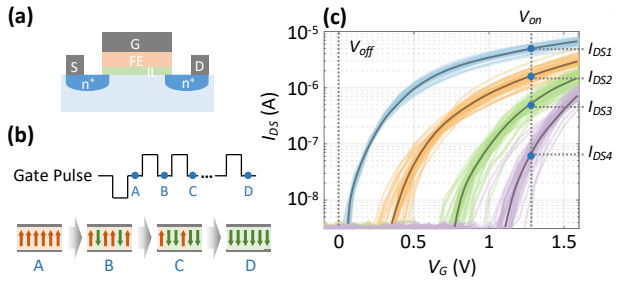

Figure 1: (a) Schematic of a FeFET device. (b) Partial polarization switching in an MFM capacitor induced by a write pulse train. (c) The multi-level I_D-V_G characteristics. Each dark curve represents a FeFET state.

The rest of the paper is organized as follows. Section 2 reviews the basics and relevant prior works. Section 3 introduces our FeFET-based IMC design for Bayesian inference. Section 4 presents the validation, scalability investigation and application benchmarking results of FeBiM. Finally, section 5 concludes the paper.

2 PRELIMINARIES AND RELATED WORKS

In this section, we briefly introduce the basics of FeFET device and its advantages in IMC, as well as Bayesian inference preliminaries and existing hardware implementations.

2.1 In-Memory Computing with FeFET

Recent years have seen significant research interests in FeFET. Unlike two-terminal resistive devices, the three-terminal FeFET functions as a standalone 1T non-volatile storage unit, featuring a high ON/OFF ratio, high energy efficiency, small footprint and CMOS compatibility [17–19]. As shown in Fig.1(a), a FeFET integrates HfO_2 as the ferroelectric dielectric layer within the gate stack of a MOSFET [20]. Write operations of FeFETs involve applying positive/negative voltage pulses to the gate. This switches the polarization in the ferroelectric layer towards the channel/gate metal, setting the FeFET into a low-V_{TH}/high-V_{TH} state. The polarization switching in FeFETs, driven by the electric field, results in superior write energy efficiency (~fJ/bit) compared to other NVM devices [21]. The binary states of FeFETs have been extensively studied and applied in existing IMC designs [22].

By varying the number of positive write pulses applied to the gate of FeFET after a full erase, as shown in Fig.1(b), partial polarization switching and multiple distinct V_{TH} states can be realized. For instance, in a 2-bit storage case shown in Fig.1 (c), 4 V_{TH} states with well-controlled device variation are achieved. When activated with V_{on} applied to the gate, a FeFET shows a current I_{DS} between the drain and source. When inhibited with V_{off}, the FeFET is in a cut-off state. The FeFET states are thus associated with corresponding I_{DS} values. Exploiting the multi-level characteristics of FeFET, recent IMC designs [23–27] have achieved improved density while maintaining computing robustness, compared to the single-level cell (SLC)-based and analog approaches, respectively.

2.2 Bayesian Inference

Bayesian inference is a key probabilistic framework that facilitates decision-making in scenarios with incomplete information. It incorporates evidence, assumptions, and prior knowledge to make informed decisions [28]. Unlike conventional NN-based machine learning approaches, Bayesian models excel at providing interpretable results and accurately estimating prediction certainty, even with limited data [8, 9]. Bayesian inference is widely applied in various fields, including medical diagnosis, where limited patient data requires the integration of expert knowledge [29], and in machine learning for decision-making under uncertainty [30].

At its core, Bayesian inference leverages Bayes' theorem to update the probability of an event based on observed evidence:

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)} \tag{1}$$

Here, $P(A)$ is the prior probability of event A, indicating the initial belief or knowledge, and $P(B)$ is the probability of observing evidence B. $P(B|A)$ is the likelihood probability of observing B given that event A has occurred. $P(A|B)$ represents the updated posterior probability of event A occurring, given that B is observed. Since the evidence probability $P(B)$ is constant for a given A, it is often omitted in calculations without affecting the order of magnitude of posteriors. Thus Eq. (1) can be simplified as:

$$P(A|B_1, B_2, ..., B_n) \propto P(A) \cdot P(B_1, B_2, ..., B_n|A) \tag{2}$$

For ease of modeling and calculation, the likelihood factors $P(B_i)$ can be considered conditionally independent in many real-life scenarios without compromising the prediction accuracy [31], and Eq. (2) can further be simplified as:

$$P(A|B) \propto P(A) \cdot \prod_{i=1}^{n} P(B_i|A) \tag{3}$$

In applications such as diagnosis/classification, the event with the highest posterior probability is selected as the final cause/category:

$$\hat{A} = \arg\max_{A} P(A|B) = \arg\max_{A} P(A) \cdot \prod_{i=1}^{n} P(B_i|A) \tag{4}$$

The cumulative product of prior and likelihoods in Eq. (3) is different from the commonly-used MAC operations in NN-based models. As a result, many existing IMC designs dedicated to MAC acceleration do not directly support Bayesian inference. Implementing Bayesian inference in hardware requires addressing the storage of model parameters (i.e., priors and likelihoods), the posterior calculations (Eq. (3)) and final decision-making (Eq. (4)). Conventionally, Bayesian inference models have been implemented on von Neumann platforms such as CPU [10], GPU [11], FPGA [12] and application-specific integrated circuit (ASIC) [32]. However, these platforms face bottlenecks in frequent data transfer between external memory and computing units, severely affecting the throughput and efficiency. To efficiently realize posterior calculations, some emerging NVM devices have been used to build RNGs in small-scale Bayesian inference prototypes, demonstrating compactness and better energy efficiency compared to traditional CMOS device [13–15]. However, these works do not address the storage and access of probabilities, a fundamental aspect of Bayesian inference implementation, and require dedicated circuits for on-demand probabilities generation. To mitigate excessive memory access, [16] builds digital NVM with memristors that performs near-memory posterior calculations, greatly reducing data movement overhead compared to conventional implementations. Yet, this approach requires additional CMOS logic and multiple clock cycles per posterior calculation and final result sensing, leading to lower computing density and efficiency.

3 FEBIM DESIGN AND WORKFLOW

To overcome the challenges in existing hardware implementation of Bayesian inference, we introduce FeBiM, a compact and efficient Bayesian inference engine empowered with multi-bit FeFET-based

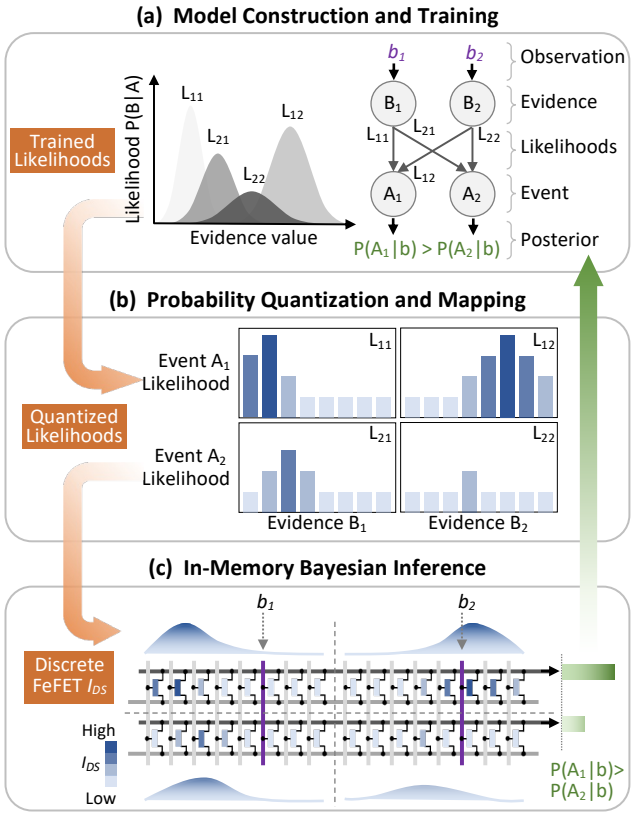

Figure 2: The overall workflow of FeBiM. Trained probabilities of the Bayesian model are quantized and mapped to discrete FeFET states. Given observed evidence values, the FeFET-based crossbar outputs maximum posterior.

IMC. In this section, we present the overall workflow of FeBiM, the core array design with just one multi-bit FeFET device per cell, and an effective scheme that maps the probabilities to the crossbar.

3.1 Overview of FeBiM

Fig. 2 illustrates the workflow of FeBiM. To begin with, a Bayesian inference model is constructed for the target problem. For example, Fig. 2(a) depicts a Bayesian network with two evidence nodes and two events, each having a uniform prior. The likelihood probabilities are first modeled on the training data. Then, the likelihoods are quantized to an adequate precision level without compromising the performance of the Bayesian model, as shown in Fig. 2(b). These quantized likelihoods are mapped to multiple FeFET states corresponding to specifically defined I_{DS}s, associated with specific writing configurations for programming the FeFETs. During the inference, discretized evidence values of the test samples activate corresponding crossbar columns. The stored likelihoods are accumulated along each row, as shown in Fig. 2(c). In this way, the posterior probabilities for each event are yielded as the crossbar outputs without extra calculation circuitry. These posteriors are sent to the sensing module to make the final prediction, completing one inference operation.

3.2 FeFET Crossbar Array Design and Operation

Fig. 3 shows the proposed crossbar design of FeBiM, which includes the core FeFET array, row driver, write/input buffer, sensing module and peripheral circuitry. In each array row, the drain nodes of FeFETs connect to a wordline (WL), and the source nodes are grounded to the sourceline (ScL). The FeFETs within the same column share a common bitline (BL) at their gates. We utilize a concise and scalable winner-take-all (WTA) circuit design [33] in the sensing module to detect the WL with the maximum current.

During write, the WL and ScL associated with the target row are grounded, and a 4V write voltage V_w with corresponding pulse number is applied to the gate of target FeFETs, programming the designated quantized prior or likelihood probabilities. To inhibit write disturbance, we apply a half bias $V_w/2$ scheme to the WLs and ScLs of unselected rows [34]. As indicated in Fig. 3, in orange/blue, the quantized prior/likelihood probabilities of a Bayesian model with n evidence nodes (each evidence value is quantized to m levels) and k events are programmed into the corresponding sections of the crossbar. During inference, the prior column is activated with V_{on}=0.5V on BL_1. One column of each likelihood block is activated according to the input evidence value on BLs, and other unselected columns are inhibited with V_{off}=-0.5V. The activated FeFET cells' I_{DS} accumulate along each WL as I_{WL}, representing the calculated posteriors (denoted in green). These I_{WL}s are then input to the WTA circuit (i.e., I_{CM}s) by the current mirrors. The WTA circuit identifies the WL with the maximum current, corresponding to the event with the highest posterior, and outputs a one-hot current result as the final inference decision (denoted with purple).

3.3 Probability Quantization and Mapping

To implement Eq. (3) in our crossbar design, we adopt logarithmic computing. In logarithmic domain, Eq. (3) can be rewritten as:

$$\log P(A|B) \propto \log P(A) + \sum_{i=1}^{n} \log P(B_i|A) \tag{5}$$

This formula naturally aligns with our crossbar's computational behavior, as described in Sec. 3.2. We initially convert the original probabilities into logarithmic values, then truncate very small probabilities to manage quantization precision efficiently. The quantization process involves two steps: discretizing evidence values to designated levels (corresponding with m BLs in each likelihood block, as described in Sec. 3.2), and quantizing the logarithmic likelihoods corresponding to the discretized evidence values with designated precision. After quantization, we apply column normalization to the likelihoods corresponding to the same evidence value (i.e., the likelihoods stored in the same column) and priors:

$$\begin{aligned} P'(A) &= \log P(A) + (1 - \max \log P(A)) \\ P'(B_i = b|A) &= \log P(B_i = b|A) + (1 - \max \log P(B_i = b|A)) \end{aligned} \tag{6}$$

where each column of the normalized probabilities is added with a constant, with their maximum values scaled to 1. This normalization enhances the differences among posteriors of multiple events without altering their order of magnitude, thus mitigating the accuracy degradation after quantization:

$$\hat{A} = \arg\max_A \log P(A|B) = \arg\max_A P'(A|B) \tag{7}$$

Finally, the normalized logarithmic probabilities P' are linearly mapped to discrete FeFET states with corresponding I_{DS} values and respective FeFET write configuration. As illustrated in Fig. 4(a), original probabilities P (denoted with blue) are truncated (i.e., P<0.1 replaced with 0.1), converted to logarithmic values, and normalized to P'. P' is uniformly quantized to 10 levels, which are then linearly mapped to FeFET I_{DS} values from 0.1 to 1.0 μA (denoted

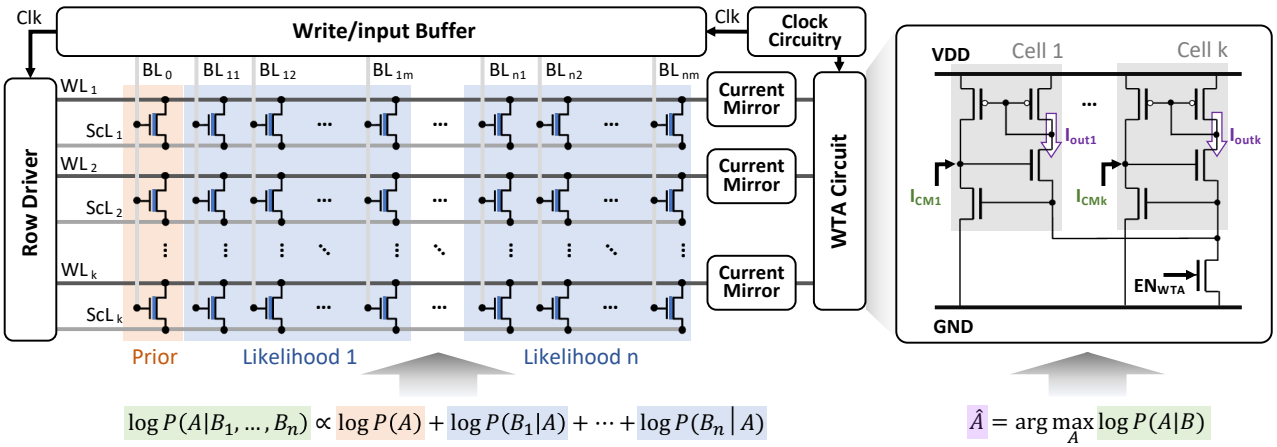

Figure 3: The proposed crossbar array effectively programs the priors and likelihoods in multi-bit FeFET cells. During inference, the posteriors are naturally calculated on each WL and fed into the WTA circuit, which detects the maximum posterior.

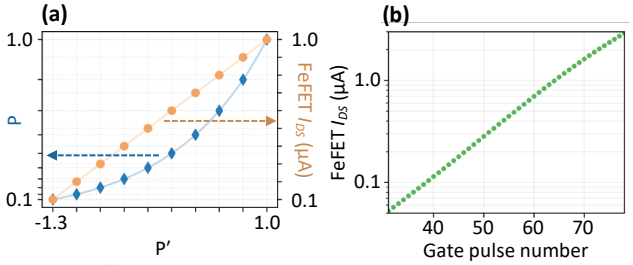

Figure 4: (a) Truncated probabilities P are first converted to the logarithmic values and normalized to P', which is then quantized and mapped to discrete FeFET states. (b) Relation between the gate pulse number and the FeFET state.

in orange). Fig. 4(b) shows the required gate pulse number to program FeFET into the designated state with required I_{DS}. For each discrete FeFET state associated with a quantized probability, the corresponding write configuration for programming the state into FeFET is determined thereof.

4 VALIDATION AND EVALUATION

In this section, we investigate the functionality and scalability of FeBiM, and benchmark its performance using a representative Bayesian classification task. The results demonstrate significant improvement in density and efficiency compared to the state-of-the-art designs.

4.1 Functionality and Performance

We first verify the correct implementation of Eq. (5) in the proposed crossbar design and the functionality of the WTA circuit. In our SPECTRE simulations, the experimentally calibrated Preisach model [35] is adopted for FeFETs, and the 45nm PTM model [36] is utilized for MOSFETs with TT process corner at 27°C and minimum sizes. To begin with, we validate the posterior calculation with two likelihood factors in the logarithmic domain. Two FeFET cells F_a and F_b in the same row WL_i are programmed with respective write configurations, storing corresponding P'_a and P'_b, as depicted in Fig. 4. By varying P'_a and P'_b, Fig. 5(a) shows the theoretical I_{WLi} values calculated from cell I_{DS} values, exactly matching the I_{WLi} values obtained from circuit calculations shown in Fig. 5(b). We then validate the functionality of the WTA circuit. Two separate WLs are connected to the WTA circuit, with I_{WL1} and I_{WL2} varying from

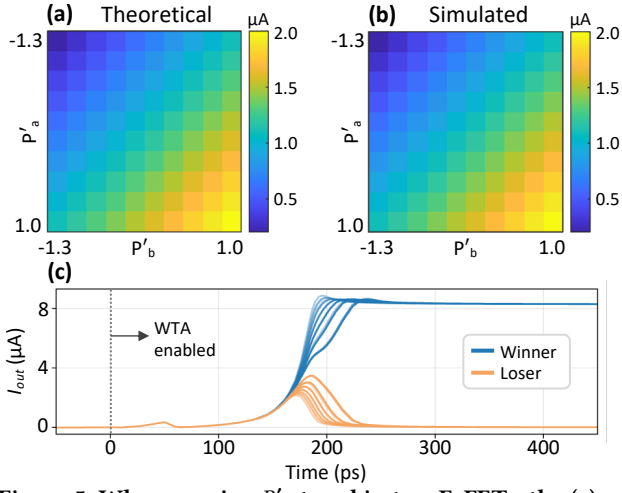

Figure 5: When varying P' stored in two FeFETs, the (a) calculated I_{WL} matches the (b) simulated I_{WL} precisely. (c) Transient simulation of the WTA circuit validates its function.

0.2μA and 2.0μA, respectively. Fig. 5(c) shows the WTA circuit's output, where the winner results (i.e., I_{WTA1} when $I_{WL1} > I_{WL2}$ and I_{WTA2} when $I_{WL2} > I_{WL1}$) are clearly distinguishable from the loser results in less than 300ps.

To evaluate the scalability and circuit-level performance of FeBiM, we test arrays with increasing numbers of rows and columns and measure the inference delay and energy consumption, as shown in Fig. 6. In these tests, all BLs are activated. The inference delay is measured as the time required to identify the winner output of the WTA circuit in the worst cases (i.e., minimum gap between adjacent I_{WL} values) after activating BLs. The inference energy includes the consumption of the crossbar array and the sensing module. The array part consists of the power dissipation in WL drivers and BL drivers. The sensing part includes the power consumption in current mirrors and the WTA circuit during inference. As array size increases, longer stabilization time leads to larger delay and higher power dissipation in both the crossbar and the sensing module.

4.2 Application Benchmarking

A Bayesian classifier, often used in tasks such as spam detection or sentiment analysis, applies Bayesian inference to assign a sample to

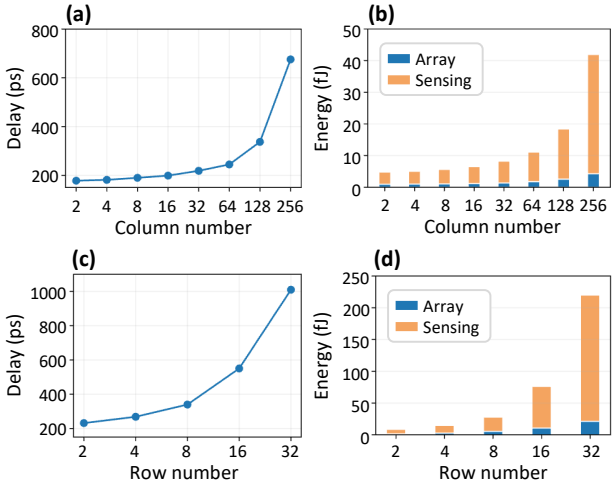

Figure 6: The (a) inference delay and (b) energy consumption of FeBiM with 2 rows and a growing number of columns. The (c) inference delay and (d) energy consumption of FeBiM with 32 columns and a growing number of rows.

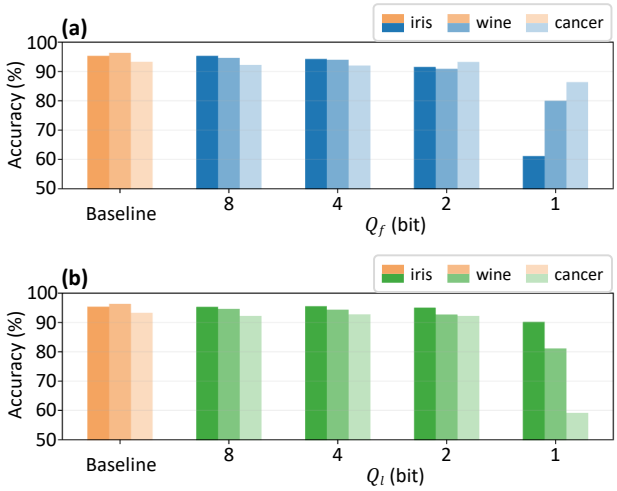

Figure 7: The inference accuracy of Gaussian Bayesian classifier on different datasets with respect to (a) varying feature quantization levels with 8-bit quantized likelihoods, and (b) varying likelihood quantization levels with 8-bit quantized features, when compared to the software baseline.

classes based on the highest posterior probability [37]. As described in Eq. (2), the posterior class probabilities ($P(A|B)$) are estimated based on the likelihoods of observed features (evidence values) given each class ($P(B|A)$) and the prior probabilities of the classes ($P(A)$). To simplify model construction and enhance likelihood computation, the Gaussian naive Bayesian classifier (GNBC), as a specific type of Bayesian classifier designed for continuous data, assumes conditional independence of features given the class label (Eq.3) and a Gaussian distribution for each feature: it estimates the mean and variance of each feature for each class and uses the Gaussian probability density function to compute the likelihood of a data point belonging to a class. Despite its simplified assumptions, GNBC typically performs well in practice [38].

Following the workflow of FeBiM in Fig. 2, we construct GNBC models using Python Scikit-learn [40] and train them on three datasets, *iris*, *wine* and *cancer*. The test/train ratio is 0.7, and the number of training-inference epochs is set to 100 for each dataset to determine the average inference accuracy. Fig. 7 shows the software-based inference accuracy using 64-bit floating point precision. Then, we apply our proposed mapping scheme to convert these probabilities to logarithmic values for quantization and normalization. Even with the feature quantization precision (Q_f, i.e., the levels of discretized evidence values as described in Sec. 3.3) or likelihood quantization precision (Q_l) reduced to as low as 2-bit, GNBCs display a negligible drop in inference accuracy on different datasets compared to the software baseline. These results indicate the effectiveness of our proposed mapping scheme in enabling high-performance Bayesian inference with limited precision.

We then implement the GNBC trained on *iris* on our proposed FeFET-based crossbar architecture. Fig. 8(a) shows the average inference accuracy of *iris*-GNBC with feature and likelihood quantization precisions varying from 1 to 8-bit. The quantization region with $\Delta_{acc} < 1\%$ is highlighted, where Δ_{acc} is the inference accuracy loss when compared to the software baseline. We choose Q_f=4bit and Q_l=2bit as the optimal quantization precision, achieving an inference accuracy of 94.64% with no significant gain observed at higher Q_f/Q_l precision levels. The quantized likelihoods are mapped to discrete FeFET I_{DS} and programmed into the crossbar. The I_{DS} states of FeFETs in the programmed crossbar storing the quantized likelihoods are shown in Fig. 8(b). The array consists of 3 WLs for 3 classes and 64 BLs for 4 4-bit features. Notably, due to the equal sample numbers in each class of *iris*, the prior is uniform and the prior block is omitted in Fig. 8(b) without affecting the posterior calculation and classification results.

To evaluate the inference accuracy of the hardware-implemented GNBC, we quantize the features of test samples as input for the crossbar. As described in Sec. 3.2, each quantized feature value activates a corresponding BL, and the I_{DS} of activated FeFETs are accumulated along each WL. The WL with maximum accumulated current is then detected by the WTA circuit as the predicted class of the test sample. The predictions of the in-memory GNBC are compared against the true labels of the test samples to calculate the classification accuracy. The number of the training-inference epochs of the in-memory GNBC is set as 100. The classification accuracy distribution of the circuit-implemented GNBC shows negligible degradation compared to the software baseline, as shown in Fig 8(c). We further investigate the robustness of FeBiM using Monte Carlo simulations with varying levels of FeFET V_{TH} variation. Compared to the baseline, the mean accuracy drop is just ~5% at $\sigma_{V_{TH}}$=45mV. Considering the experimental variation of FeFET device, such as [25] at 38mV, these results indicate the robustness and reliability of FeBiM in high-performance Bayesian classification.

In Table 1, we benchmark FeBiM's performance metrics in the context of *iris*-GNBC against other representative NVM-based Bayesian inference implementations. In light of a 2FeFET per cell IMC design at 45nm node [41], we lay out a 2×2 FeFET array and estimate a cell area of 0.076um^2. The array density reaches 26.32 Mb/mm^2 when storing 2 bits per cell. The average energy dissipation per inference is 17.20fJ, yielding a computing efficiency of 581.40 TOPS/W. As a result, FeBiM outperforms the state-of-the-art memristor-based Bayesian machine by 10.7× in storage density and 43.4× in efficiency. Compared to the RNG-based implementations, the computing density of FeBiM is improved by more than 3.0×.

Table 1: Comparison between FeBiM and other Bayesian inference implementations with emerging devices.

Reference	Technology	Device usage	Device configuration	Probability storage unit	Calculation circuitry	Sensing circuitry	Speed clk./inf.	Storage density Mb/mm^2	Computing density MO/mm^2	Efficiency TOPS/W
[13]	MTJ	RNG	SLC	*	RNG, logic gate, comparator, Muller C-element	PCSA	2000	*	0.23	0.013
[14]	Memtransistor	RNG	SLC	*	RNG, logic gate	Inverting amplifier	200	*	0.033	0.0025
[16]	Memristor	Memory	SLC	8×2T2R cells†	LFSR, comparator	PCSA	1~255‡	2.47$^\wedge$	0.034$^\wedge$	2.14~13.39‡
This work	FeFET	Memory	MLC	1F	\$^\diamond$	WTA circuit	1	26.32	0.69	581.40

$*$: Not mentioned. $\diamond$: Not required. $\dagger$: For 8-bit quantized likelihoods. $\ddagger$: Depending on the operation scheme. $\wedge$: Cell area data from [39], scaled to 45nm. Abbreviations. RNG: random number generator. PCSA: pre-charge sense amplifiers. LFSR: linear feedback shift register. clk./inf.:clock cycles per inference. MO: million operations. TOPS/W: trillion operations per second per watt.

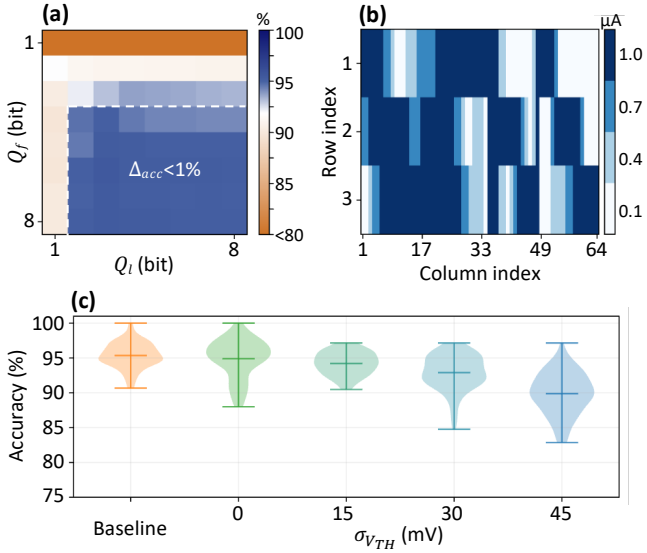

Figure 8: (a) Average inference accuracy of *iris*-GNBC with different quantization precision. (b) The states of FeFETs in a 3×64 crossbar implementing the *iris*-GNBC with Q_f=4bit and Q_l=2bit. (c) The inference accuracy distribution of circuit-implemented Bayesian classifier under different FeFET device variation levels as compared with the software baseline.

5 CONCLUSION

In this paper, we introduce FeBiM, a highly compact and efficient in-memory Bayesian inference engine utilizing multi-bit FeFETs. Our novel mapping scheme effectively encodes the trained probabilities of a Bayesian inference model within a compact FeFET-based crossbar, enabling natural logarithmic in-memory Bayesian inference and facilitating more efficient and reliable computations. When applied to a representative Bayesian classification task, FeBiM exhibited high inference accuracy and efficiency, maintaining robust performance despite variations and limited precision. As the first FeFET-based in-memory Bayesian inference engine, FeBiM achieved a remarkable storage density of 26.32 Mb/mm^2 and a computing efficiency of 581.40 TOPS/W. These results demonstrate significant advancements over the state-of-the-art Bayesian machine, showcasing FeBiM's potential in enhancing a broad range of Bayesian inference applications.

ACKNOWLEDGMENTS

This work was partially supported by National Key R&D Program of China (2022YFB4400300), NSFC (92164203, 62104213) and SGC Cooperation Project (Grant No. M-0612).

REFERENCES

[1] D. Ielmini *et al.*, "In-memory computing with resistive switching devices," *Nature electronics*, vol. 1, no. 6, pp. 333–343, 2018.
[2] A. Shafiee *et al.*, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *ACM SIGARCH*, vol. 44, no. 3, pp. 14–26, 2016.
[3] X. S. Hu *et al.*, "In-memory computing with associative memories: A cross-layer perspective," in *2021 IEEE IEDM*, pp. 25–2, 2021.
[4] Z. Yan *et al.*, "Improving realistic worst-case performance of nvcim dnn accelerators through training with right-censored gaussian noise," in *2023 IEEE/ACM ICCAD*, pp. 1–9, 2023.
[5] S. Jung *et al.*, "A crossbar array of magnetoresistive memory devices for in-memory computing," *Nature*, vol. 601, no. 7892, pp. 211–216, 2022.
[6] A. Qayyum *et al.*, "Secure and robust machine learning for healthcare: A survey," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 156–180, 2020.
[7] G. Yang *et al.*, "Unbox the black-box for the medical explainable ai via multi-modal and multi-centre data fusion," *Information Fusion*, vol. 77, pp. 29–52, 2022.
[8] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, 2015.
[9] N. Burkart *et al.*, "A survey on the explainability of supervised machine learning," *Journal of Artificial Intelligence Research*, vol. 70, pp. 245–317, 2021.
[10] R. J. Smith *et al.*, "Massively parallel bayesian inference for transient gravitational-wave astronomy," *MNRAS*, vol. 498, no. 3, pp. 4492–4502, 2020.
[11] C. Talbot *et al.*, "Parallelized inference for gravitational-wave astronomy," *Physical Review D*, vol. 100, no. 4, p. 043030, 2019.
[12] H. Awano *et al.*, "Bynqnet: Bayesian neural network with quadratic activations for sampling-free uncertainty estimation on fpga," in *2020 IEEE DATE*, pp. 1402–1407, IEEE, 2020.
[13] D. Vodenicarevic *et al.*, "Low-energy truly random number generation with superparamagnetic tunnel junctions for unconventional computing," *PRL*, vol. 8, no. 5, p. 054045, 2017.
[14] Y. Zheng *et al.*, "Hardware implementation of bayesian network based on two-dimensional memtransistors," *Nature communications*, vol. 13, no. 1, p. 5578, 2022.
[15] R. Faria *et al.*, "Implementing bayesian networks with embedded stochastic mram," *AIP Advances*, vol. 8, no. 4, 2018.
[16] K.-E. Harabi *et al.*, "A memristor-based bayesian machine," *Nature Electronics*, vol. 6, no. 1, pp. 52–63, 2023.
[17] J. Müller *et al.*, "Ferroelectricity in hfo 2 enables nonvolatile data storage in 28 nm hkmg," in *2012 IEEE Symposium on VLSI technology*, pp. 25–26, IEEE, 2012.
[18] A. I. Khan *et al.*, "The future of ferroelectric field-effect transistor technology," *Nature Electronics*, vol. 3, no. 10, pp. 588–597, 2020.
[19] X. Yin *et al.*, "Ferroelectric compute-in-memory annealer for combinatorial optimization problems," *Nature Communications*, vol. 15, no. 1, p. 2419, 2024.
[20] S. Dünkel *et al.*, "A fefet based super-low-power ultra-fast embedded nvm technology for 22nm fdsoi and beyond," in *2017 IEEE IEDM*, pp. 19–7, IEEE, 2017.
[21] K. Ni *et al.*, "Ferroelectric ternary content-addressable memory for one-shot learning," *Nature Electronics*, vol. 2, no. 11, pp. 521–529, 2019.
[22] X. Yin *et al.*, "Ferroelectric ternary content addressable memories for energy-efficient associative search," *IEEE TCAD*, vol. 42, no. 4, pp. 1099–1112, 2022.
[23] C. Li *et al.*, "A scalable design of multi-bit ferroelectric content addressable memory for data-centric computing," in *2020 IEEE IEDM*, pp. 29–3, IEEE, 2020.
[24] S. Shou *et al.*, "See-mcam: Scalable multi-bit fefet content addressable memories for energy efficient associative search," in *2023 IEEE/ACM ICCAD*, pp. 1–9, IEEE, 2023.
[25] T. Soliman *et al.*, "First demonstration of in-memory computing crossbar using multi-level cell FeFET," *Nature Communications*, vol. 14, no. 1, p. 6348, 2023.
[26] X. Yin *et al.*, "An ultracompact single-ferroelectric field-effect transistor binary and multibit associative search engine," *Advanced Intelligent Systems*, vol. 5, no. 7, p. 2200428, 2023.
[27] X. Yin *et al.*, "Deep random forest with ferroelectric analog content addressable memory," *arXiv preprint arXiv:2110.02495*, 2021.
[28] G. E. Box *et al.*, *Bayesian inference in statistical analysis*. John Wiley & Sons, 2011.
[29] D. Nikovski, "Constructing bayesian networks for medical diagnosis from incomplete and partially correct statistics," *IEEE TKDE*, vol. 12, no. 4, pp. 509–516, 2000.
[30] P. C. Trimmer *et al.*, "Decision-making under uncertainty: biases and bayesians," *Animal cognition*, vol. 14, pp. 465–476, 2011.
[31] D. Lowd *et al.*, "Naive bayes models for probability estimation," in *ICML*, pp. 529–536, 2005.
[32] G. G. Ko *et al.*, "A 3mm 2 programmable bayesian inference accelerator for unsupervised machine perception using parallel gibbs sampling in 16nm," in *2020 IEEE Symposium on VLSI Circuits*, IEEE, 2020.
[33] C.-K. Liu *et al.*, "Cosime: Fefet based associative memory for in-memory cosine similarity search," in *2022 IEEE/ACM ICCAD*, pp. 1–9, 2022.
[34] K. Ni *et al.*, "Write disturb in ferroelectric fets and its implication for 1t-fefet and memory arrays," *IEEE EDL*, vol. 39, no. 11, pp. 1656–1659, 2018.
[35] K. Ni *et al.*, "A circuit compatible accurate compact model for ferroelectric-fets," in *2018 IEEE Symposium on VLSI Technology*, pp. 131–132, IEEE, 2018.
[36] R. Vattikonda *et al.*, "Modeling and minimization of pmos nbti effect for robust nanometer design," in *2006 ACM/IEEE DAC*, pp. 1047–1052, 2006.
[37] A. H. Wang, "Don't follow me: Spam detection in twitter," in *2010 IEEE SECRYPT*, pp. 1–10, IEEE, 2010.
[38] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI*, vol. 3, pp. 41–46, 2001.
[39] Q. Liu *et al.*, "33.2 a fully integrated analog reram based 78.4 tops/w compute-in-memory chip with fully parallel mac computing," in *2020 IEEE ISSCC*, pp. 500–502, IEEE, 2020.
[40] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[41] X. Yin *et al.*, "Fecam: A universal compact digital and analog content addressable memory using ferroelectric," *IEEE TED*, vol. 67, no. 7, pp. 2785–2792, 2020.