Algorithm to Technology Co-Optimization for CiM-based Hyperdimensional Computing

Mahta Mayahinia*, Simon Thomann†, Paul R. Genssler‡,
Christopher Münch*, Hussam Amrouch†‡, and Mehdi B. Tahoori*
*Department of Computer Science and Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany
†Chair of AI Processor Design, TUM School of Computation, Information and Technology, Technical University of Munich;
Munich Institute of Robotics and Machine Intelligence, Munich, Germany
‡Semiconductor Test and Reliability, University of Stuttgart, Stuttgart, Germany
{mahta.mayahinia/christopher.muench/mehdi.tahoori}@kit.edu;genssler@iti.uni-stuttgart.de;{simon.thomann/amrouch}@tum.de

***Abstract*—Hyperdimensional computing (HDC) has been recognized as an efficient machine learning algorithm in recent years. Robustness against noise and simple computational operations, while being limited by the memory bandwidth, make it a perfect fit for the concept of computation in memory (CiM) with emerging nonvolatile memory (NVM) technologies. For an HDC accelerator based on NVM-CiM, there are different parameters from the algorithm all the way down to the technology that interact with each other and affect the overall inference accuracy as well as the energy efficiency of the accelerator. Therefore, in this paper, we propose, for the first time, a full-stack co-optimization method and use it to design an HDC accelerator based on NVM-based content addressable memory (CAM). By incorporating the device manufacturing variability and co-optimizing the algorithm and hardware design, HDC inference on our proposed NVM-based CiM accelerator can reduce the energy consumption by 3.27x, while compared to the purely software-based implementation, the inference accuracy loss is merely 0.125%.**

***Index Terms*—Hyperdimensional computing, Analog computing, Non-volatile memory, Content addressable memory (CAM)**

I. Introduction

Conventional compute-centric architectures face two serious challenges. First, the increasing demand for data and second, the limits of technology scaling. In compute-centric architectures, the data has to be transferred from the memory to the processing elements. Driven by machine learning, more and more data has to be processed, exposing this overhead of data transfers. This *memory wall* has emerged as the major bottleneck of compute-centric architectures. Traditionally, technology scaling enabled an increase in processing capabilities. However, the size of a transistor is approaching its fundamental physical limits.

To address both challenges at the same time, the Computation-in-Memory (CiM) paradigm is emerging as a promising solution. Instead of separating data and computing, the memory is enhanced to perform simple computations directly, eliminating the data transfer overhead. The concept of CiM can be realized with different memory technologies, from the conventional static RAM (SRAM) all the way to the emerging non-volatile memory (NVM). Besides the low static power consumption of the NVM technologies and improved density, *analog* CiM can be realized to further improve performance and energy efficiency at the cost of increased noise in the computations.

This work was partially supported by Advantest as part of the Graduate School "Intelligent Methods for Test and Reliability" at the University of Stuttgart, as well as by funding from the pilot program Core Informatics of the Helmholtz Association (HGF) at Karlsruhe Institute of Technology.

Hyperdimensional Computing (HDC) is an emerging brain-inspired algorithm that has gained considerable interest in the past years [1]. Not only for its capability of one-shot learning but also because of its robustness against noise in its computations. HDC is based on the concept of hypervectors (HVs), i.e., vectors with a very large dimension [2]. Using these large HVs as the basic processing elements provides sufficient redundancy and thus considerable robustness against noise. In addition, HDC's simple mathematical operations can be easily implemented in hardware. Hence, the combination of HDC and NVM-CiM enables data-intensive yet efficient machine learning.

A core operation of HDC is the computation of the similarity between two HVs. The different classes are represented by *prototype HVs*. During inference, the similarity of an unlabeled *query HV* to all the prototype HVs is computed and the most similar class is selected as the label. In hardware, this inference operation is performed by the Associative Memory (AM). In the literature, CiM-based AMs have been implemented [3]–[6]. At the circuit level, Content Addressable Memory (CAM)-based cells were employed to realize the AM. By employing NVM technologies in this CAM structure, prototype HVs are only written once, which can save energy. However, the length of the HVs challenges the NVM-CAM implementations. The noise in the computations overcomes even the robustness of HDC preventing a reliable operation [7].

The reliability of NVM-CiM is affected by two main factors. First, analog computing is inherently susceptible to noise. Second, the NVM technologies are still immature and hence more prone to manufacturing variability. Although hardware design techniques can partially improve the reliability of the NVM-CiM, there is a limit that the HDC inference algorithm can tolerate. Vastly incorrect similarity computations will impact the inference accuracy by selecting the incorrect class as the most similar to the query HV. Previous work did not precisely model the manufacturing variability of NVM at the technology level and, at the same time, consider its effect on the inference accuracy at the algorithm level. Further, the necessity of such an algorithm to technology co-optimization to achieve energy-efficient yet reliable HDC computations was not fully recognized in previous work. A holistic cross-level design and evaluation are essential to find the most efficient solutions.

In this paper, we introduce the framework for this algorithm to technology co-optimization of energy consumption and inference accuracy for an HDC accelerator based on various

979-8-3503-4859-0/DATE24/© 2024 EDAA

NVM-CiM technologies. Moreover, by optimizing analog and digital computing and developing a hierarchical similarity computation, we cover the entire technology stack to improve the energy efficiency of the hardware-based HDC inference with a negligible reduction in the inference accuracy.

Our novel contributions within this paper are as follows: (**i**). Taking into account four different NVM technologies, including Ferroelectric FET (FeFET), Phase Change Memory (PCM), Spin-Transfer Torque Magnetic RAM (STT-MRAM), and Redox-based RAM (ReRAM) to show the interaction of the device with the hardware and algorithm in the co-optimization of the entire HDC design space. (**ii**). Optimizing the analog and digital computing to implement similarity computation on large HVs directly in the hardware through a scalable hierarchical AM realized by NVM-CAM cells. (**iii**). Incorporating the proposed hardware design and its implication at the algorithmic level and investigating its impact on the HDC inference accuracy.

II. Background and related work

A. Background

1) CAM: For each CAM operation, there are two sets of data operands, *prototypes* which are fixed, and *query* which is compared with *all* the prototypes in parallel. Due to their high parallelism, CAM hardwares are used in high-performance applications.

Since prototypes are not changing dynamically, they need to be written only once in NVM, with almost no static power consumption. Thus, CAM cell based on NVM can be a good fit for this use case. Fig. 1(e) shows the CAM cell based on the NVM which has been proposed in [7]. In this CAM structure, the original and complementary bits of the prototype patterns are stored in the NVMs, while the original and negated bits of the query pattern are applied as binary voltage levels to the search-line and negated search-line, respectively. By decoding binary '1' ('0') as a high resistive state (HRS) (low resistive state (LRS)) in the prototype, and high (low) voltage level in the query patterns, if the search operation results in *match*, at the time of sampling, the voltage of the CAM match-line is greater than the case of *mismatch*. Due to the limited HRS-LRS ratio in different NVM technologies, the length of the match-line cannot be increased arbitrarily.

2) HDC: HDC is a rapidly growing alternative to classical machine learning algorithms [2]. The applicability of HDC has successfully been shown for language recognition [8], image and pattern classification [9], [10], circuit reliability [11], and more [12]–[14]. By using HVs with dimensions in the thousands, HDC and its operations are designed to create patterns and later recognize them. Moreover, the large dimension creates redundancy in the HVs, giving HDC its strong resilience to noise and errors. For the individual elements of the HVs, several different data types can be used, such as real or integer numbers, and for the hardware-based implementation, binary is the most compatible. To map real-world objects to the hyperspace, a small set of operations is used and their concrete implementations depend on the used data type. The utilized encoding scheme is application-dependent and the literature offers several, together with the analysis of the data type [1].

B. Related Work

Together with a PCM-based n-gram encoder, a PCM-based AM has been proposed and fabricated in [3]. The CiM-oriented multiply-accumulate (MAC) version of the XNOR has been performed in an analog manner to compute the similarity of the prototype and query HVs. Although this work has also utilized analog computing, the amount of digital computing and Analog-to-Digital Converter (ADC) activations are still considerable, resulting in high energy consumption [15].

In [4], besides a fully digital AM hardware, leveraging the resistive CAM structure has been taken into account as well. Moreover, a fully analog version of the AM hardware based on the resistive CAM has also been introduced. However, the incorporation of the technology-aware manufacturing variability in their simulation-based hardware model is not precisely performed, leading to less realistic results.

FeFET technology is also used in literature to construct AMs by using CAM-based arrays [6], [16]. While [16] ignores the impact of the final analog to digital step, [6] proposes a "synaptic comparator" utilizing intermediate states of the FeFET device. The problem with both [6], [16] is that the small CAM size ranging from 8 to $15\,\mathrm{bit}$, imposes high hardware overhead.

HDC for knowledge graphs accelerated by FeFET-based CiM has been discussed in [17]. Although [17] has benefited from chunk-wise analog computing, the compromise between multiple abstraction levels, namely, technology, circuit, architecture, and the algorithm has not been considered holistically.

Employing NVM-CAM-based design for accelerating the HDC inference implies strong interaction between the algorithm and the hardware, mainly due to the mixing of the digital and analog computations. Therefore, co-optimizing these interactions is crucial to reach an energy-efficient and highly accurate HDC inference. For such design space exploration, a co-optimization methodology and a full-stack framework are required. Such a framework needs to be flexible and realistic, i.e., sufficiently incorporated with the circuit-level and manufacturing variability details. The lack of the aforementioned methodology and framework is the shortcomings of the current literature.

III. Our Proposed Methodology

HDC can achieve high inference accuracy if the HV size is large enough as shown in Fig. 2. Reducing the size of the HV (< 100) results in an extremely low HDC inference accuracy of $\sim 40\%$. However, the scalability of the NVM-CAM is severely limited by the lower HRS-LRS ratio of NVM compared to SRAM, as well as the impact of manufacturing variability. In this section, for the sake of energy efficiency, we aim to employ the NVM-CAM structure for similarity computation. On top of that, we propose a full-stack solution to overcome the challenge of short NVM-CAM comparison length. Furthermore, we propose a methodology to co-optimize the algorithm and hardware design spaces to achieve a reliable and energy-efficient HDC inference accelerator.

A. Design Solutions at the Algorithmic Level

Generating the HVs (i.e., encoding) is performed at the algorithmic level. The data type and the length of these HVs are crucial features and need to be chosen based on specific

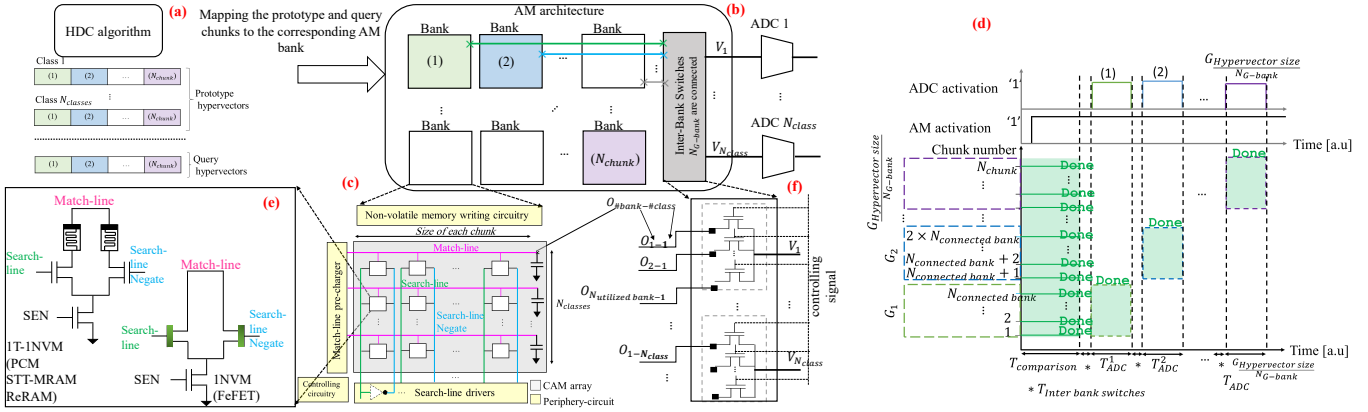


Fig. 1: Overview of the proposed method. (a) Split the HVs into chunks at the algorithmic level. (b) Multi-bank AM architecture to maintain the parallel search capability. (c) The array-level realization of the CAM structure. (d) The timing diagram shows the hierarchical calculation of the similarity measure. (e). The differential structure of the resistive CAM for 1T-1NVM (PCM, STT-MRAM, and ReRAM) and for 1T structure (FeFET). (f) Inter-Bank switches for scalability of the CAM-based accelerator for HDC.

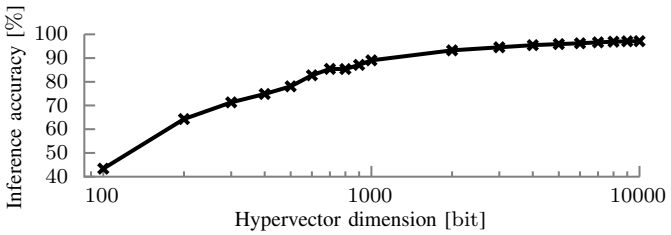


Fig. 2: Golden baseline inference accuracy of language recognition.

hardware characteristics. For the data type, binary is more compatible with the hardware. Also, the size of the HVs needs to be large to maintain high inference accuracy.

To co-optimize the inference algorithm based on the underlying hardware design and NVM technology, the large HVs need to be broken into *chunks* at the algorithm level. As we will discuss in Sec. III-B, efficient mapping of the HVs chunks to the NVM-CAM accelerator requires some extra information, which needs to be provided by the algorithm. This information determines if the chunk is from a prototype or a query and what is its position in the original HV. The role of the HDC inference accelerator hardware is to process the chunks such that the final output of the similarity computation will not be affected by chunking.

B. Design Solutions at the Hardware Level

The key feature of the AM is the capability of parallel search execution. To maintain the parallel search capability, the first requirement is to design a *multi-bank* AM architecture. Banks are memory sub-systems that can work independently, at the same time. As it is shown by Fig. 1(b), in a chunk-wise manner HDC inference, the number of chunks (N_{chunk}) is the size of the HV divided over the maximum length of the match-line. Each chunk of the prototypes and query HVs are mapped to the corresponding AM bank.

The array level design of each bank is shown in Fig. 1(c), the number of match-lines in each AM bank is equal to the number of the classes. So, in one AM bank, the corresponding query chunk is compared to the corresponding chunk of *all* the prototypes. As shown in Fig. 1(d), at each activation shot of the proposed AM architecture, at first, the similarity computation of the entire query (all the chunks in their corresponding banks) and prototype HVs can be performed. Nevertheless, assigning the query to the most similar class, equivalent to the highest inverse Hamming distance in the digital domain, requires the accumulation of similarity measures over all the chunks. To realize the cumulative similarity measure in the AM, we introduce the *inter-Bank switches* (Fig. 1(b)). Inter-Bank switches connect the group of match-lines (in total N_{G-bank}) corresponding to each class across the AM banks.

1) Hierarchical Similarity Measure Computation: According to Fig. 1(d), after parallel similarity computation in the multi-bank AM, the first group of banks (G_1) is connected via the inter-Bank switches, and the output voltage of the inter-Bank switches is the analog representation of the similarity measure of G_1, which is converted to digital bits with the help of an ADC. Please note that the higher the output voltage of the inter-Bank switch, the higher the output of the ADC, and the higher the similarity measure. However, the output of the ADC is not the exact inverse Hamming distance.

In the next time step, the configuration of the inter-Bank switches for the next group, and activation of the ADCs will be performed. This procedure is repeated for all the groups onward

$$(T^1_{\text{ADC}} : G_1 ... T^{\frac{\text{HV size}}{N_{G-bank}}}_{\text{ADC}} : G_{\frac{\text{HV size}}{N_{G-bank}}}).$$

For each prototype, the total similarity measure for the entire length of the HV can be obtained by summing up the digital representation of the similarity measure for each group, i.e., the output of the ADC. Eventually, the most similar class can be determined based on a winner-take-all mechanism, which simply compares the digital values corresponding to each class. As shown in Fig. 1(f), the inter-Bank switches are implemented by connecting each match-line output capacitance, to the source of an n-type transistor that acts as a switch. The drain of all the switch transistors corresponding to the same class are connected to a common node that is marked as V_x *(x: 1..*N_{class}*)* in Fig. 1(f), which is the input of the corresponding ADC.

C. Algorithm to Technology Co-Optimization

In each abstraction level of the design space, from the high-level inference algorithm all the way down to the technology

level, there are parameters that need to be co-optimized to reach an energy-efficient yet reliable hardware-based HDC accelerator with acceptable inference accuracy.

In the **algorithm level**, the size of the HV is a crucial parameter; the larger the HV, the higher the inherent HDC robustness. However, for larger HVs, the analog hardware design supporting that, imposes a considerable overhead in terms of the overall inference latency and energy, which is mostly due to ADC activations.

In the **hardware architecture level**, the number of connected banks through the inter-Bank switches (N_{G-bank}) is crucial. Increasing N_{G-bank} moves the output voltage levels of the inter-Bank switch (i.e., analog representation of the similarity measure) closer together. As these voltages are inputs to the ADC, it is important that the voltage corresponding to the most similar class can be differentiated from the other classes at least by one ADC quantization level.

The existing trade-off: On the other side, in the presence of manufacturing variability, the output voltage of the CAM match-line corresponding to the most similar class can be incorrectly sensed less than other classes and produces wrong similarity computation. While increasing the N_{G-bank} impairs the sensing reliabiliy due to a decrease in the sense margin, larger N_{G-bank} can decrease the standard deviation of the analog voltage distribution and makes the distribution narrower, which can improve the distinguishability. Despite this effect, however, for larger N_{G-bank}, the impact of shrinking the sense margin is more dominant and negatively affects the inference accuracy. In general, large N_{G-bank} is desirable since it decreases the number of ADC activations and hence, improves the latency and energy efficiency, however, at the cost of impaired sensing reliability. Therefore, the robustness of the HDC algorithm against its computation noise determines the maximum value for N_{G-bank}.

At last in the **technology** level, various NVM technologies can be utilized. The HRS-LRS ratio as well as the effect of the manufacturing variability are important technology-level aspects. The Maximum N_{G-bank}, which is an architectural parameter, is determined based on the employed NVM technology. Large HRS-LRS ratio (e.g. FeFET, PCM, and ReRAM technologies) is beneficial to increase the maximum N_{G-bank}. While a small HRS-LRS ratio (e.g., STT-MRAM technology) results in a smaller maximum N_{G-bank}.

The inference accuracy and energy consumption of the NVM-based HDC inference accelerator are determined by the aforementioned algorithm, hardware, and technology level parameters. The co-optimization methodology adjusts these parameters such that energy saving can be achieved at a small inference accuracy loss.

IV. Results and Discussion

A. Hardware-level Analysis

1) Front-End of the line Simulation: We perform a detailed SPICE-based electrical-level simulation of the CAM array (Fig. 1(c), Fig. 1(e)), as well as the inter-Bank switches (Fig. 1(f)). We consider the effect of the manufacturing variability for all of the explored NVM technologies, FeFET, PCM, STT-MRAM, and ReRAM. The HRS and LRS of these technologies are following a normal distribution (N $\sim$ (mean (μ), standard deviation (σ)).

TABLE I: Simulation setup tools and parameters.

Simulation tool	Cadence Virtuoso
Technology node for CMOS	Global Foundries 22FDX
Standard V_{DD} for CMOS	$0.8\,\mathrm{V}$
Temperature	$27\,^{\circ}\mathrm{C}$
Interconnect parameters	- Barrier: Ta-based, thickness: $3\,\mathrm{nm}$ - Horizontal/vertical dielectric = 2.55/3.9 - $RC_{\pi-model}$=$11.95\,\Omega$, $16.63\,\mathrm{aF}$ (per CAM cell)
FeFET model [18]	- Material = $Hf_{0.5}Zr_{0.5}O_2$ - Ferro layer thickness = $10\,\mathrm{nm}$ - HRS, LRS = $1\,\mathrm{T\Omega}$, $19.50\,\mathrm{k\Omega}$
PCM model [19]	- Fabrication node = $90\,\mathrm{nm}$ - PCM material thickness = $100\,\mathrm{nm}$ - HRS, LRS = $9\,\mathrm{M\Omega}$, $20\,\mathrm{k\Omega}$
MTJ model [20]	- RA = $7.5\,\Omega\,\mu\mathrm{m}^2$ - Nominal TMR = $150\,\%$ - HRS, LRS = $11.56\,\mathrm{k\Omega}$, $5.967\,\mathrm{k\Omega}$
ReRAM model: *JART* [21], [22]	- Filament radius = $45\,\mathrm{nm}$ - Disc region length = $0.6\,\mathrm{nm}$ - HRS, LRS = $105.51\,\mathrm{k\Omega}$, $1.975\,\mathrm{k\Omega}$

2) Back-End of the line Simulation: The resistive and capacitive (RC) parasitic of the metallic interconnect need to be considered for a realistic hardware model. We tune the electrical-level parameters of our proposed hardware design based on the trained data from HDC applications. Our front-end and back-end of the line parameters are outlined in Tab. I.

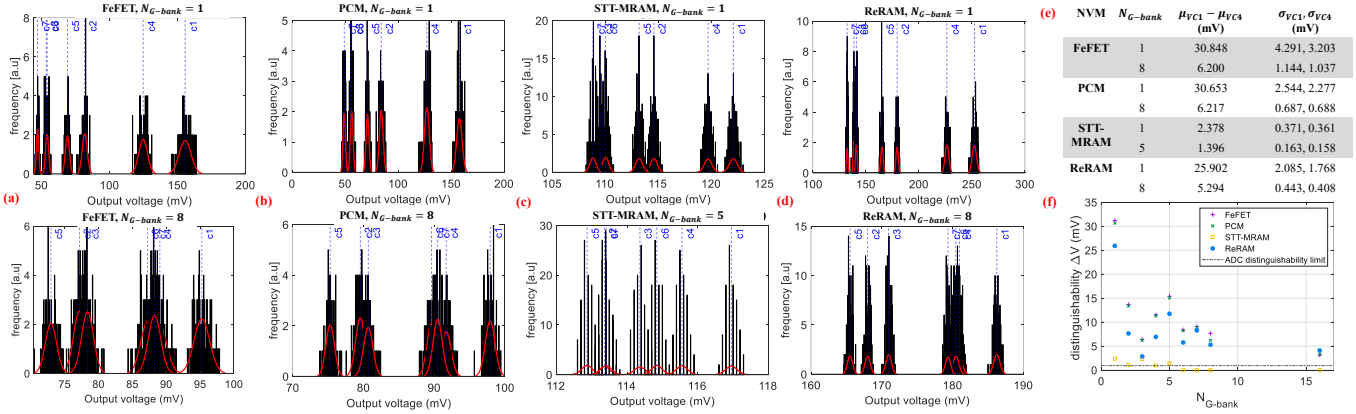
For the 10-bit ADC, we employ a voltage source of $950\,\mathrm{mV}$ [23]. Hence, the ADC quantization level is $0.928\,\mathrm{mV}$ and this value is the minimum voltage difference (ΔV) required for the most similar class to be distinguishable from the others.

Fig. 3(a-e) show the analog voltage distribution at the input of the ADC for two extreme cases of $N_{G-bank} = 1$, $N_{G-bank} = 8$ for FeFET, PCM, and ReRAM, and $N_{G-bank} = 5$ for STT-MRAM since it has the lowest HRS-LRS ratio. Due to the manufacturing variability and generally smaller HRS-LRS ratio compared to SRAM, in all the explored NVM technologies, the length of the CAM in each bank is limited to 64.

As discussed in Sec. III-B, a larger N_{G-bank} decreases the sense margin, which brings the distinct voltage levels closer together. At the same time, it also decreases the standard deviation (σ) making the distribution narrower (see Fig. 3(a-e)). The latter effect, although not dominant for larger N_{G-bank}, mitigates the error of the similarity computation to some extent. Fig. 3(f) shows the decreasing trend of the sense margin while increasing the N_{G-bank}. Due to the interconnect parasitics, the decreasing trend of sense margin depends on the to-be-compared prototype and query HVs, and hence, is not always monotonic. For STT-MRAM and $N_{G-bank} > 5$, the sense margin is below the ADC quantization level. For FeFET, PCM, and ReRAM, N_{G-bank} can be as high as 16. However, leveraging $N_{G-bank} = 16$ at the algorithmic level results in a poor inference accuracy and thus we limit N_{G-bank} to eight.

B. Algorithm-level Analysis

The first step toward algorithmic-level explorations is abstracting the hardware. In this regard, for different *block sizes* ($64 \times N_{G-bank}$), we model the nominal output voltage (mean



NVM	N_{G-bank}	$\mu_{VC1} - \mu_{VC4}$ (mV)	$\sigma_{VC1}, \sigma_{VC4}$ (mV)
FeFET	1	30.848	4.291, 3.203
	8	6.200	1.144, 1.037
PCM	1	30.653	2.544, 2.277
	8	6.217	0.687, 0.688
STT-MRAM	1	2.378	0.371, 0.361
	5	1.396	0.163, 0.158
ReRAM	1	25.902	2.085, 1.768
	8	5.294	0.443, 0.408

Fig. 3: (a-e) The expected ΔV with respect to the N_{G-bank} for different NVM technologies, (f) the distinguishability versus N_{G-bank}, (C_x, are the prototype classes, in the evaluated data set: C_1, C_4 are the most and the second most similar classes, respectively.

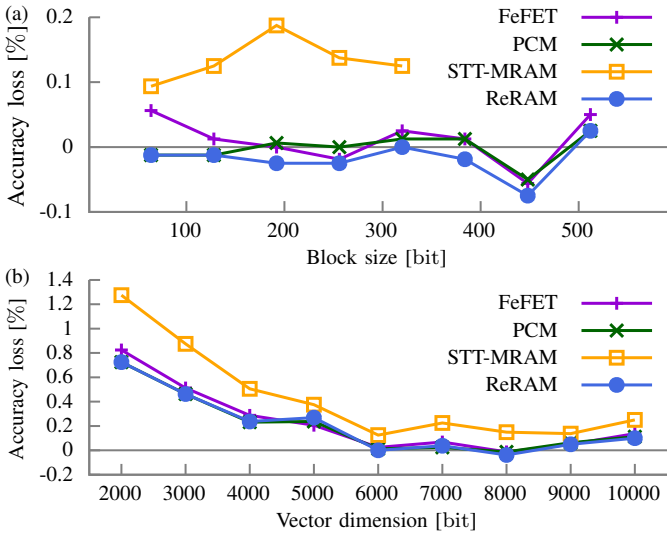


Fig. 4: Median inference accuracy loss of the tested technologies. (a) The relation of accuracy loss over block size at a hypervector (HV) dimension of 6000 bit. (b) The relation of accuracy loss over HV dimension at a block size of 320 bit.

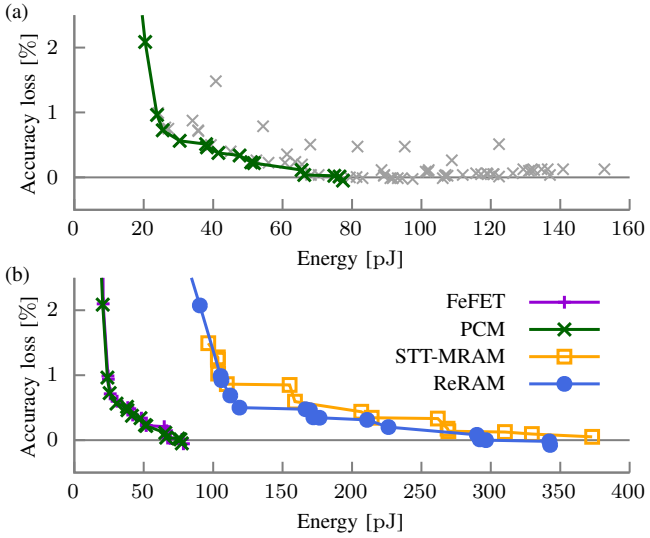


Fig. 5: Pareto optimal analysis of the different HV and block sizes with respect to accuracy loss and energy consumption. (a) The Pareto front of PCM at the connected dots and all non-optimal configurations in the background as a scatter plot. (b) Comparison of the different technologies.

(μ)) and its standard deviation (σ) as a function of inverse Hamming distance. Please note that the nominal output voltage and its standard deviation can be obtained through *single-run* and *Monte Carlo* simulations, respectively. By knowing the normal distribution of the output voltage, we can calculate the probability of the occurrence of falling into each ADC quantization level. Based on the block size, the output of the ADC in the digital domain corresponds to an inverse Hamming distance with a one-to-one mapping scheme.

To investigate the influence of the manufacturing variability on the different technologies, we evaluate HDC with several applications, including language recognition and gesture recognition with Electromyography (EMG). First, the inference accuracy is calculated in software as a golden baseline, then the abstracted hardware models are injected into the inference algorithm. We calculate the loss by subtracting hardware-based accuracy from the variation-free software-based baseline.

In Fig. 4(a), it can be seen the relation of accuracy loss versus the block size at the HV size of 6000 bit for the language dataset. As we can see in Fig. 4(a), the inference accuracy loss is not monotonically increasing with the block size. The competing effects of increasing the block size (or increasing the N_{G-bank}) on decreasing the sense margin and at the same time, decreasing the standard deviation (σ) of the voltage distributions is the reason behind the observations of the local extrema in Fig. 4(a). Besides, hardware implementation and manufacturing variability are not necessarily destructive for inference accuracy. In some cases, the models would classify incorrectly. Yet, the randomness of the non-ideal hardware corrects these errors and actually improves the inference accuracy. Hence, for some of the (technology, block size) pairs in Fig. 4(a), negative accuracy loss can also be observed.

Due to the low HRS-LRS ratio in STT-MRAM, only a block size up until 320 bit is feasible. Moreover, STT-MRAM technology exhibits a 0.1 % to 0.2 % higher accuracy loss than all the other technologies which are gravitating around 0 % loss. Relatively worse inference accuracy for the case of STT-MRAM is also explainable by its relatively smaller sense margin as shown in Fig. 3(c, e, f).

Fig. 4(b) shows an almost decreasing trend of inference accuracy loss with larger HVs. Moreover, as is depicted in Fig. 4(b),

TABLE II: Summary of the results and comparison with the related work, the size of the hypervector is 10 000.

Related work	Technology CMOS / NVM	Accuracy of hardware model	Energy per class [pJ]	Accuracy loss
[4]*	45 nm / ReRAM	+	2.2	0.5 %
[3]	65 nm / PCM	+++	1180.0	0.4 %
This work **	22 nm / FeFET	++	76.7	0.1 %
	22 nm / PCM		76.4	0.1 %
	22 nm / STT		135.9	0.2 %
	22 nm / ReRAM		144.3	0.1 %

* The fully analog implementation based on the resistive CAM
** The block size is 320 bit

by selecting higher HV dimensions, the negative impact of the NVM technology is vanishing. Typically, larger HVs benefit from more redundancy, leading to stronger robustness against noise. Therefore, the effect of the technology on the inference accuracy loss is reduced compared to smaller HVs.

By combining the inference accuracy numbers with the energy of the proposed NVM-CAM-based accelerator for the HDC inference, we evaluate the NVM technologies as Pareto fronts. Fig. 5(a) shows the optimal configurations in green and non-optimal ones in gray. The point outside the plotted area is the 1024-bit block with a high accuracy loss of 5.53 %. In Fig. 5(b), the different technologies are depicted together and their Pareto fronts follow a similar pattern. PCM and FeFET technologies are more energy efficient than ReRAM and STT-MRAM.

As discussed in Sec. III-C, the size of the HVs at the algorithm level and the block size (or N_{G-bank}) at the hardware architecture level are crucial parameters for our proposed HDC accelerator. The most energy-efficient configurations are small HVs (from the algorithm level) with large block sizes (at the hardware architecture level), which minimize ADC activation cycles at the cost of higher accuracy loss due to more analog computing. The EMG dataset confirms these conclusions.

According to our Pareto analysis, increasing the HV size improves the inference accuracy, at the cost of higher energy consumption. However, increasing the HV size beyond a certain value stagnates the inference accuracy, i.e., an increase in the energy does not have a considerable effect on the inference accuracy. Horizontal jumps on the Pareto fronts in Fig. 5(b) are showing the increase of the HV size.

As earlier discussed in Sec. IV-A, although the block size of 1024 bit ($N_{G-bank} = 16$) is possible, however, the accuracy loss is too high which cannot be compensated with the energy saving. The Pareto analysis also shows that the block size of 1024 bit, is not a good choice in the context of HDC.

Tab. II shows the summary of results in our proposed CAM-based HDC accelerator and its comparison with the related work. For all the NVM technologies, by co-optimizing the digital and analog computing, the block size of 64 bits can increase to 320 bits, which means ADC reduction for $\sim 5\times$. Compared to the non-optimized hardware design (with the block size of 64 bits, equal to the length of NVM-CAM), the reduction of the energy consumption for the explored NVM technologies are as follows: FeFET: $3.92\times$, PCM: $3.95\times$, STT-MRAM: $2.65\times$, and ReRAM: $2.54\times$. While the inference accuracy loss, average on various NVM technologies, is only 0.125 %.

V. Conclusion

In this paper, we have explored accelerating the HDC inference by leveraging the computation in non-volatile memory paradigm. We have introduced a hardware and algorithm co-optimization methodology and further used it to design a multi-bank associative memory based on NVM-CAM. The short length of the NVM-CAM is a challenge imposed by NVM technology while utilizing it for the similarity computation of the HVs. To overcome this challenge, we have proposed a chunk-wise HDC inference at the algorithm level, modified multi-bank associative memory, and the CiM architecture, and employed a hybrid analog and digital computing scheme for similarity computation at the circuit level. We have demonstrated that the co-optimization of the energy and inference accuracy results in considerably less energy consumption ($3.27\times$), with the inference accuracy loss of only 0.125 %.

References

[1] L. Ge *et al.*, "Classification Using Hyperdimensional Computing: A Review," *IEEE Circuits and Systems Magazine*, 2020.

[2] P. Kanerva, "Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors," *Cognitive Computation*, vol. 1, 2009.

[3] G. Karunaratne *et al.*, "In-memory hyperdimensional computing," *Nature Electronics*, vol. 3, 2020.

[4] M. Imani *et al.*, "Exploring Hyperdimensional Associative Memory," in *IEEE HPCA*, 2017.

[5] P. R. Genssler *et al.*, "DropHD: Technology/algorithm co-design for reliable energy-efficient nvm-based hyperdimensional computing under voltage scaling," in *IEEE DATE*, 2024.

[6] S. Thomann *et al.*, "All-in-memory brain-inspired computing using fefet synapses," *Front. Electron. 3: 833260. doi: 10.3389/felec*, 2022.

[7] J. Li *et al.*, "1 mb 0.41 μm² 2t-2r cell nonvolatile tcam with two-bit encoding and clocked self-referenced sensing," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 896–907, 2014.

[8] A. Rahimi *et al.*, "A Robust and Energy-Efficient Classifier Using Brain-Inspired Hyperdimensional Computing," in *IEEE ISLPED*, 2016.

[9] P. R. Genssler *et al.*, "Brain-inspired computing for wafer map defect pattern classification," in *IEEE International Test Conference*, 2021.

[10] D. Kleyko *et al.*, "Holographic graph neuron: A bioinspired architecture for pattern processing," *IEEE TNNLS*, 2017.

[11] P. R. Genssler *et al.*, "Brain-inspired computing for circuit reliability characterization," 2022.

[12] P. R. Genssler *et al.*, "HDCircuit: Brain-inspired hyperdimensional computing for circuit recognition," in *IEEE DATE*, 2024.

[13] A. Rahimi *et al.*, "Hyperdimensional biosignal processing: A case study for EMG-based hand gesture recognition," in *IEEE ICRC*, 2016.

[14] P. R. Genssler *et al.*, "Modeling and predicting transistor aging under workload dependency using machine learning," 2023.

[15] T. Chou *et al.*, "Cascade: Connecting rrams to extend analog dataflow in an end-to-end in-memory processing paradigm," in *MICRO*, 2019.

[16] K. Ni *et al.*, "Ferroelectric ternary content-addressable memory for one-shot learning," *Nature Electronics*, vol. 2, no. 11, pp. 521–529, 2019.

[17] H. E. Barkam *et al.*, "Reliable hyperdimensional reasoning on unreliable emerging technologies," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 2023, pp. 1–9.

[18] S. Kumar *et al.*, "Cross-layer reliability modeling of dual-port fefet: Device-algorithm interaction," *IEEE TCAS-I: Regular Papers*, 2023.

[19] M. Le Gallo *et al.*, "An overview of phase-change memory device physics," *Journal of Physics D: Applied Physics*, vol. 53, no. 21, p. 213002, 2020.

[20] F. Bernard-Granger *et al.*, "SPITT: A magnetic tunnel junction SPICE compact model for STT-MRAM," in *Proceedings of the MOS-AK Workshop of the Design, Automation & Test in Europe (DATE)*, 2015.

[21] C. Bengel *et al.*, "Variability-aware modeling of filamentary oxide-based bipolar resistive switching cells using spice level compact models," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2020.

[22] S. Wiefels *et al.*, "Hrs instability in oxide-based bipolar resistive switching cells," *IEEE Transactions on Electron Devices*, vol. 67, no. 10, 2020.

[23] M. Liu *et al.*, "A 10-bit 2.5-gs/s two-step adc with selective time-domain quantization in 28-nm cmos," *IEEE TCAS-I: Regular Papers*, 2022.