CafeHD: A Charge-Domain FeFET-Based Compute-in-Memory Hyperdimensional Encoder with Hypervector Merging

Taixin Li[1*], Hongtao Zhong[1*], Juejian Wu[1], Thomas Kämpfe[2], Kai Ni[3], Vijaykrishnan Narayanan[4]
Huazhong Yang[1], Xueqing Li[1†]
[*]Equal Contributions; [1]BNRist, Electronic Engineering, Tsinghua University, China; [2]Fraunhofer IPMS, Germany
[3]University of Notre Dame, USA; [4]Pennsylvania State University, USA; [†]Email: xueqingli@tsinghua.edu.cn

***Abstract*—Hyperdimensional computing (HDC) is an emerging paradigm that employs hypervectors (HVs) to emulate cognitive tasks. In HDC, the most time-consuming and power-hungry process is encoding, the first step that maps raw data into HVs. There have been non-volatile memory (NVM) based computing-in-memory (CiM) HDC encoding designs, which exploit the intrinsic HDC characteristics of high parallelism, massive data, and robustness. These NVM-based CiMs have shown great potential in reducing encoding time and power consumption. Among them, the ferroelectric field-effect transistor (FeFET) based designs show ultra-high energy efficiency. However, existing FeFET-based HDC encoding designs face the challenges of energy-consuming current-mode addition, inefficient HV storage, limited endurance, and single encoding method support. These challenges limit the energy efficiency, lifetime, and versatility of the designs.**

This work proposes an energy-efficient charge-domain FeFET-based in-memory HDC encoder, i.e., CafeHD, with extended lifetime, good versatility, and comparable accuracy. Area-efficient charge-domain computing is proposed in HDC encoding for the first time, which enables CafeHD with ultra-low power and high scalability. An HV merging technique is explored to improve the performance. A low-cost partial MAJ interface is also proposed to reduce writes. Besides, CafeHD also supports two widely used encoding methods. Results show that CafeHD on average achieves 10.9×/12.7×/3.5× speedup and 103.3×/21.9×/6.3× energy efficiency with ∼84% write times reduction and similar accuracy compared with the state-of-the-art ReRAM/PCM/FeFET-based CiM design for HDC encoding, respectively.

***Index Terms*—hyperdimensional computing, computing-in-memory, charge-domain computing, ferroelectric transistors**

I. Introduction

Hyperdimensional computing (HDC) is an emerging computational framework that imitates the behavior of human brains by high-dimensional vectors, i.e., *hypervectors* (HVs) [1]. HVs are random and holographic vectors with thousands of independent and identically distributed (i.i.d.) elements. With only one or few shots and unidirectional dataflow in the training process, HDC is lightweight and energy-efficient, and has shown potential in many applications, such as language classification [2], speech recognition [3], etc.

Fig. 1(a) shows the widely used HDC framework. The first step is HDC encoding that maps the input raw data into HVs. It is a very compute-intensive and energy-consuming process, and can take about 99.9% and 94.6% of the training and inference time on average across four HDC-related datasets, respectively, as shown in Fig. 1(b). This is because the bit-wise operations in HDC encoding, such as binding (XOR), bundling (e.g., majority vote, MAJ), and permutation, are highly parallel and induce intensive memory access due to the high HV dimensionality [1]. Moreover, all elements are holographic and equally weighted in HVs, which enhances the robustness of HDC [4]. Therefore, non-volatile memory (NVM) based computing-in-memory (CiM) becomes a desirable candidate, which can perform *in-situ* operations in memory and significantly reduce data movement.

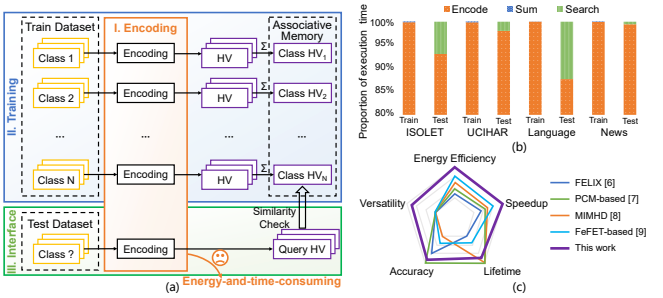


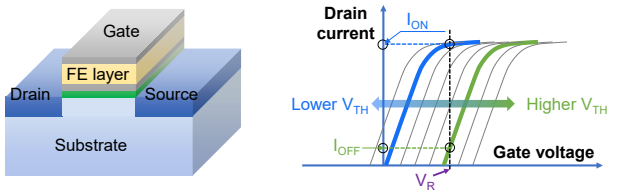
Fig. 1. Hyperdimensional computing (HDC): (a) Basic framework. (b) Time-consuming encoding. (c) Comparison with SOTA CiM-based encoders [6]–[9].

There have been several energy-efficient NVM-based CiMs for HDC encoding with impressive speedup based on various devices, such as resistive random access memories (ReRAMs) [5], [6], phase change memories (PCMs) [7], ferroelectric field-effect transistors (FeFETs) [8], [9], etc. Among these devices, FeFETs exploit lower power consumption and good CMOS compatibility thanks to the physical structure, voltage-driven write mechanism, and relatively high ON/OFF ratio. Therefore, the FeFET-based CiM has shown the promise of higher performance and energy efficiency compared with the current-driven NVM-based CiMs [8], [9].

However, the existing FeFET-based CiMs for HDC encoding [8], [9] perform bundling through current summation, heavily degrading energy efficiency. Besides, HVs are directly stored in memory for these designs, which occupies large memory footprints and limits performance. These designs also require costly interfaces (e.g., analog-to-digital converters [8]) or frequent writes [9] to handle massive intermediate data, causing large overheads or reliability issues [10]. Moreover, record-based encoding [11] and N-gram encoding [12] are two widely adopted encoding methods for different applications, but most existing designs support only one of them [5]–[9].

To cope with the above challenges, we propose CafeHD, a **ch**arge-domain **Fe**FET-based in-memory **HD**C encoder with HV merging to achieve high energy efficiency, extended lifetime, good versatility, and comparable accuracy. The detailed contributions are summarized as follows:

979-8-3503-4859-0/DATE24/© 2024 EDAA



(a) Planar FeFET structure (b) Typical FeFET $I_D - V_G$

Fig. 2. FeFET: (a) Planar physical structure. (b) I_D-V_G curve [13], [14].

- We propose an energy-efficient charge-domain FeFET-based hardware architecture with improved lifetime and versatility including three highlighted modules: (i) A NAND-type charge-domain XOR array that supports both record-based and N-gram encoding; (ii) A low-cost partial MAJ interface for write times reduction to achieve extended lifetime; (iii) A charge-domain MAJ array with local computing units (LCUs) to achieve both high energy efficiency and low area cost.
- We propose an HV merging technique instead of storing HVs directly to improve energy efficiency and performance with negligible accuracy loss.
- We perform extensive experiments and explore the optimal tradeoff between efficiency, accuracy, and reliability. Results show that CafeHD achieves 10.9×/12.7×/3.5× speedup and 103.3×/21.9×/6.3× energy efficiency with ∼84% write times reduction and similar accuracy compared with the state-of-the-art ReRAM/PCM/FeFET-based CiM for HDC encoding, respectively.

II. Background and Related Works

This section introduces the basics of FeFETs and HDC encoding, and analyzes the bottlenecks of the related works.

A. FeFET Basics

FeFET is a MOSFET with the insertion of a ferroelectric layer in the gate stack, as shown in Fig. 2(a) [13], [14]. The recent discovery of ferroelectricity in doped HfO_2 enables FeFETs with smaller sizes and lower operating voltage, which facilitates the integration of FeFETs into advanced technology platforms such as 28nm bulk [13] and 22nm FDSOI [14].

The FeFET memory states are controlled by the FE layer polarization. During the write operation, if a high positive write voltage is applied to the FeFET gate, the domains in the FE layer will switch to the positive state, which reduces the FeFET threshold voltage (V_{th}). Similarly, if a high negative write voltage is applied, the FeFET V_{th} will increase. The resulting hysteretic characteristic of the FeFET I_D-V_G curves are shown in Fig. 2(b). During the read operation, the read voltage (V_R) within the memory window can distinguish the I_D difference between two memory states.

Despite high energy efficiency and CMOS compatibility, HfO_2-based FeFETs still suffer from low endurance [10]. Although high-endurance FeFETs are being developed [15], frequent writes should be avoided in FeFET-based designs.

B. HDC Encoding

Encoding is the first step in HDC, which encodes the input raw data into high-dimensional space. Given an input feature vector $\vec{F} = \{f_1, f_2, \ldots, f_n\}$, HDC encoding maps it to a D-dimensional HV ($\vec{H} \in \{0,1\}^D$) that can reserve the information of feature values with their positions in $\vec{F}$. There are two widely used encoding methods, including record-based encoding and N-gram encoding. In general, N-gram encoding is more suitable for consecutive information such as texts, while record-based encoding can cover other tasks.

The record-based encoding [11] first generates two binary HV sets: (i) the position HV set $\{\vec{P}_1, \vec{P}_2, \ldots, \vec{P}_n\}$, where $\vec{P}_i \in \{0,1\}^D$, and (ii) the level HV set obtained by quantizing the feature values into Q levels $\{\vec{L}_1, \vec{L}_2, \ldots, \vec{L}_Q\}$, where $\vec{L}_i \in \{0,1\}^D$. Then, for each f_i in F, there is a corresponding $\vec{L}'_i$ based on the f_i value. Finally, XOR operations are performed between each $\vec{L}'_i$ and $\vec{P}_i$, and all XOR results are summed and then binarized by comparing each summed element value with $\frac{n}{2}$. The above process can be formulated as Eq. 1:

$$\vec{H} = \vec{P}_1 \oplus \vec{L}'_1 + \vec{P}_2 \oplus \vec{L}'_2 + \cdots + \vec{P}_n \oplus \vec{L}'_n \quad (1)$$

The N-gram encoding [12] uses permutation instead of position HVs to indicate feature positions. Every N adjacent level HVs, i.e., $\{\vec{L}'_i, \vec{L}'_{i+1}, \ldots, \vec{L}'_{i+N-1}\}$, are first encoded to $\vec{G}_i$ using Eq. 2. In this step, the k^{th} level HV $\vec{L}'_{i+k}$ is permutated (usually shifted, ρ) for $k-1$ times before being combined to $\vec{G}_i$ by XOR operations. N is a preset parameter whose values are typically 3, 4, or 5 (tri-gram, quad-gram, or penta-gram). Then all the obtained $n - N + 1$ N-gram HVs are summed and binarized, which can be formulated as Eq. 3.

$$\vec{G}_i = \vec{L}'_i \oplus \rho\vec{L}'_{i+1} \oplus \rho^2\vec{L}'_{i+2} \oplus \cdots \oplus \rho^{N-1}\vec{L}'_{i+N-1} \quad (2)$$

$$\vec{H} = \vec{G}_1 + \vec{G}_2 + \cdots + \vec{G}_{n-N+1} \quad (3)$$

C. Related Works

For both record-based and N-gram encoding, XOR and addition are two key operations. There have been several NVM-based CiM designs. SearcHD [5] and FELIX [6] exploit ReRAM crossbars to implement logic-in-memory (LiM) operations, achieving significant energy efficiency improvement compared with the digital solutions. However, the ReRAM current-driven write scheme makes the writes in these designs extremely energy-consuming. [7] presents a PCM-based CiM where XOR is approximated by AND operations performed within the PCM array. MIMHD [8] leverages multi-level FeFETs for XOR operations. However, addition and comparison in both [7] and [8] are implemented by the costly peripherals.

Recently, [9] proposes an energy-efficient FeFET-based CiM design, which performs in-memory XOR and utilizes MAJ as efficient addition and comparison implementations. However, the data transfer between the XOR and MAJ array involves frequent writes and may induce reliability issues. Meanwhile, [9] uses the summed current for addition. This computation mode is highly energy-consuming and unreliable when adding hundreds of currents considering the FeFET device variation. Moreover, the above designs mainly focus on one single encoding method and lack versatility due to the different applications of record-based and N-gram encoding.

III. Hardware Design

This section presents the proposed hardware architecture with three highlighted modules to significantly improve energy efficiency, lifetime, and versatility.

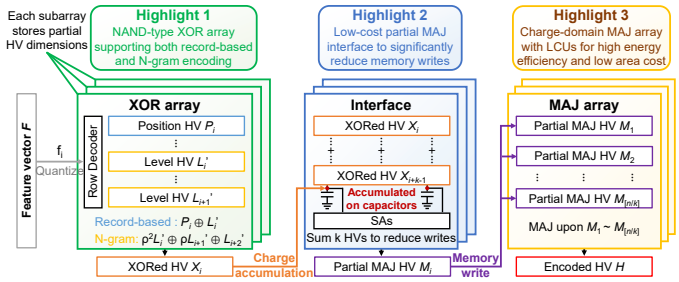


Fig. 3. Overview of the proposed hardware architecture in CafeHD.

A. Overall Architecture

Fig. 3 shows the proposed hardware architecture containing an XOR array, an interface, and an MAJ array. To support high-dimensional HV storage, HVs are partitioned into slices and mapped into multiple subarrays. Each feature dimension is quantized and input to the row decoder in the XOR array. The row decoder then selects the corresponding HVs and performs the XOR operations. A certain number of XOR results are summed on the capacitors in the interface and bundled to HVs by MAJ operations using sense amplifiers (SAs). After the partial MAJ HVs are written into the MAJ array, MAJ operations are carried out to output the final encoded HV.

The proposed design significantly improves performance thanks to three highlighted modules: (i) The proposed XOR array leverages a novel NAND-type structure to allow multiple input selections and thus support both record-based and N-gram encoding; (ii) The proposed partial MAJ interface can partially sum the HVs before the MAJ array, which avoids both costly peripherals and frequent writes in prior FeFET-based designs; (iii) The proposed MAJ array utilizes charge-domain computing for higher energy efficiency and exploits LCUs to lower the area cost, which provides better energy efficiency and scalability than existing current-domain MAJ CiMs.

B. NAND-Type In-Memory XOR Array

The in-memory XOR arrays in existing designs either select two rows as input or compute between a selected row and an external input. However, for the arrays where two rows are XORed, it is impractical to store all the shifted level HVs for N-gram encoding. Meanwhile, in record-based encoding, arrays with external inputs need to fetch position HVs from other memories, which severely degrades energy efficiency.

To address this issue, we propose the NAND-type charge-domain XOR array that supports both encoding methods by providing various input selections, as shown in Fig. 4. The HVs are pre-stored in the NAND-type FeFET array. Each cell contains two FeFETs controlled by the wordline (WL) to store complementary states. The top cells are connected to the mode selection units (MSUs) via the bitlines (BLs). The sourcelines (SLs) connect the bottom cells, the PMOS transistors for pre-charging, and the capacitors for storing the XOR results.

For record-based encoding, the two rows storing the position HV and level HV are selected as input. Both BLs are grounded through the turned-on $\mathrm{T_{ra}}$ and $\mathrm{T_{rb}}$ controlled by the control signal (Ctrl). To perform the XOR operation, SLs are first precharged to VDD and then floating. The two selected rows are activated by applying a read voltage (V_R) to the WLs.

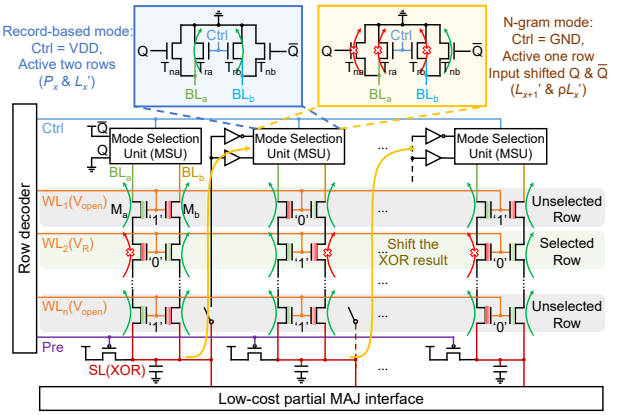


Fig. 4. CafeHD Highlight I: Proposed NAND-type charge-domain in-memory XOR array supporting both record-based and N-gram encoding.

TABLE I
CELL MODE FOR RECORD-BASED AND N-GRAM ENCODING

Method	Ctrl	Input 1 (level HV)			Input 2 (position/shifted HV)			Pull-down path	SL (XOR)
Record-based	VDD	Logic	M_a	M_b	Logic	M_a	M_b	/	/
		'0'	high-V_{th}	low-V_{th}	'0'	high-V_{th}	low-V_{th}	Right	'0'
					'1'	low-V_{th}	high-V_{th}	OFF	'1'
		'1'	low-V_{th}	high-V_{th}	'0'	high-V_{th}	low-V_{th}	OFF	'1'
					'1'	low-V_{th}	high-V_{th}	Left	'0'
N-gram	GND	Logic	M_a	M_b	Logic	T_{na} (Q)	T_{nb} ($\overline{Q}$)	/	/
		'0'	high-V_{th}	low-V_{th}	'0'	OFF (GND)	ON (VDD)	Right	'0'
					'1'	ON (VDD)	OFF (GND)	OFF	'1'
		'1'	low-V_{th}	high-V_{th}	'0'	OFF (GND)	ON (VDD)	OFF	'1'
					'1'	ON (VDD)	OFF (GND)	Left	'0'

Meanwhile, the unselected WLs are set to a positive voltage (V_{open}) to turn on all FeFETs. If the two cells store the same bit, the FeFETs on one side are all ON and form a pull-down path to discharge SL. Otherwise, each pull-down path has at least one OFF-state FeFET in series, and SL remains at VDD. Therefore, the XOR results are represented by the SL voltage.

N-gram encoding performs the XOR operation between the level HV and the shifted partial result from the MSU. Ctrl is grounded, and the pull-down path is controlled by both Q and $\overline{\mathrm{Q}}$. Therefore, Q can be XORed with the state of the selected cell by similar XOR operations. In the first cycle, SLs are initially grounded. With Q set to GND, the level HV can be read out and shifted to Q and $\overline{\mathrm{Q}}$ of the next column by buffers and inverters. The following $N-1$ cycles repeat the above process to generate the N-gram HV. Table I shows the detailed cell mode for the two encoding methods. The custom NAND structure combined with the novel MSU design achieves in-memory XOR support for both two encoding methods.

C. Partial MAJ Interface

The output HVs from the XOR array are transferred to the MAJ array for MAJ computation. However, frequent FeFET writes induce severe endurance degradation. Moreover, the write process is much slower and energy-consuming compared with XOR, which becomes the performance bottleneck of the encoding process. Therefore, we propose a low-cost partial MAJ interface that partially sums the HVs to reduce the memory writes and the speed mismatch.

Fig. 5(a) illustrates the workflow of the proposed interface. In the first k cycles, once the HVs are output from the XOR array, the capacitors are charged or discharged according to the XOR results. In this way, the accumulated charges and V_{sum}

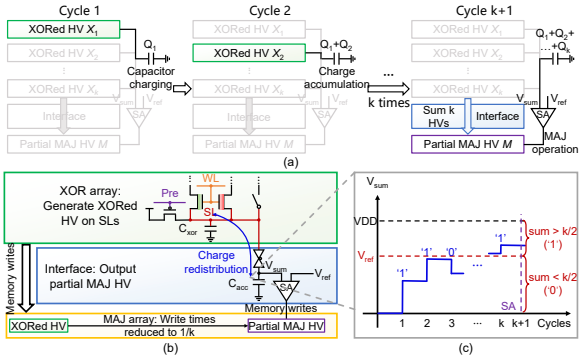


Fig. 5. CafeHD Highlight II: Proposed partial MAJ interface to reduce writes. (a) Workflow. (b) Circuit structure. (c) Transient waveform of V_{sum}.

represent the summed results. Afterward, V_{sum} is compared with a reference voltage (V_{ref}) to perform the MAJ operation with a batch size of k and output the partial MAJ HV.

Fig. 5(b) shows the interface circuit design. With the XORed HV indicated as the SL voltage, charge redistribution is performed between C_{xor} and C_{acc} once the transmission gate is turned on. As shown in Fig. 5(c), if the SL voltage is VDD, C_{acc} is charged, and V_{sum} increases. Otherwise, charges flow from C_{acc} to C_{xor}, causing V_{sum} to drop. After k iterations, V_{sum} higher than V_{ref} implies more 1's in the sum, so the SA outputs '1'. Similarly, the MAJ result is '0' if V_{sum} is lower than V_{ref}. By bundling every k HVs through the charge redistribution, memory writes are significantly reduced to $\frac{1}{k}$, which alleviates the speed mismatch between the write process and the XOR computation at low cost. Moreover, C_{acc} can be achieved by top-level metal layers above the array to save area.

We analyze the impact of k on the required capacitance of C_{acc}, the accuracy in applications, and write times reduction, and select the optimal k value in Section V-B and V-C.

D. Charge-Domain In-Memory MAJ Array

Conventional in-memory MAJ operations are implemented by activating multiple rows and sensing the summed currents [7], [9]. However, this process imposes high power consumption and poor accuracy. Here we propose a charge-domain MAJ array with LCUs to tackle these challenges.

Fig. 6(a) shows the proposed MAJ array with LCUs. The basic cell is the 1T AND-type FeFET, and each LCU contains 4 rows of FeFETs. To read out the data in the selected row, SL is first precharged to VDD. Then, WL, BL, and SL are set to V_R, GND, and floating, respectively. If the FeFET in the selected row is in a high-V_{th} state, SL will be pulled down. Otherwise, SL will stay at VDD. The inverter sets the voltage of node X consistent with the FeFET state, and the feedback loop accelerates the charging and discharging of the capacitor.

Fig. 6(b) shows the MAJ workflow. The output HVs from the interface are written into different LCUs sequentially. Afterward, all LCUs simultaneously read out the HVs to node X. The identical capacitors in the LCUs are connected so the readout results can be averaged by capacitor coupling and obtained on CBL. The CBL voltage is further compared with a reference voltage to perform MAJ upon r HVs. The intermediate MAJ HV is written back, and the encoded HV is produced by iterative MAJ upon all the intermediate HVs. The

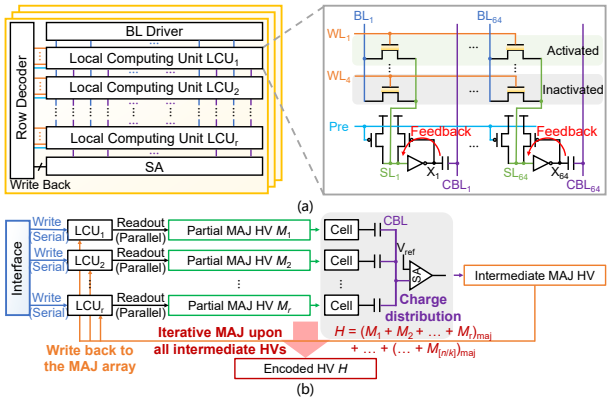


Fig. 6. CafeHD Highlight III: Proposed charge-domain MAJ array with improved energy efficiency at low area cost. (a) Overview. (b) Workflow.

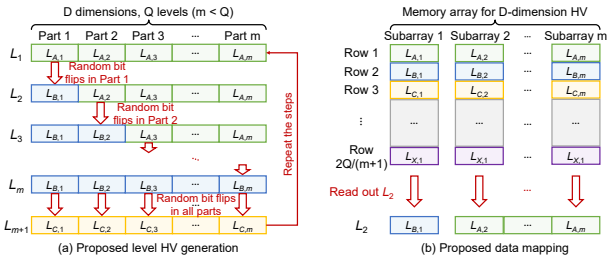


Fig. 7. Proposed HV merging technique: (a) The proposed level HV generation strategy. (b) The proposed data mapping of the level HVs.

charge-domain computing offers high energy efficiency and scalability, while LCUs reduce the capacitor area overheads.

The selection of r (the batch size for each MAJ operation) is further optimized to provide better tradeoff between accuracy and extra writebacks in Section V-B and V-C.

IV. Software Optimization

This section introduces the proposed HV merging technique, including the generation strategy and data mapping of level HVs, which is critical to performance improvement in CafeHD. This is because the level HVs are conventionally generated by repeatedly flipping a small number of random bits and directly pre-stored in memory, which induces high memory redundancy and further degrades performance.

Fig. 7(a) shows the proposed level HV generation strategy. $\vec{L}_1$ with D dimensions is initialized and partitioned into m slices. $\vec{L}_2$ is then generated by flipping random bits in the first slice. Based on $\vec{L}_2$, flipping random bits in the second slice can obtain $\vec{L}_3$. The similar process continues until $\vec{L}_m$. Considering the quasi-orthogonality required for character representation, $\vec{L}_{m+1}$ is generated by bit flips in all partitions. Thereafter, the above operations are repeated continuously.

Fig. 7(b) shows the proposed data mapping of the level HVs. Similar to HVs, the memory array is partitioned into m independent subarrays. Row 1 and Row 2 store $\vec{L}_1$ and $\vec{L}_m$, respectively, and so on. Therefore, only $\frac{2Q}{m+1}$ rows are required to store Q level HVs. The non-directly stored HVs are obtained by reading different rows. For example, to get $\vec{L}_2$, Subarray 1 reads Row 2, while other subarrays read Row 1. With the location constraint of bit flips, the HV merging technique significantly reduces the redundancy of storing level HVs and further improves the performance of CafeHD.

TABLE II
BASIC DATASET SETTINGS

Dataset	Feature size	Classes	Encoding method	Applications
ISOLET	617	26	record-based	Speech recognition [16]
UCIHAR	561	6	record-based	Activity recognition [17]
Language	152[1]	21	N-gram	Language recognition [18]
News[2]	718[1]	10	N-gram	News classification [19]

[1] Average feature size.
[2] Pre-processed from Reuters-21578 by removing classes with little data.

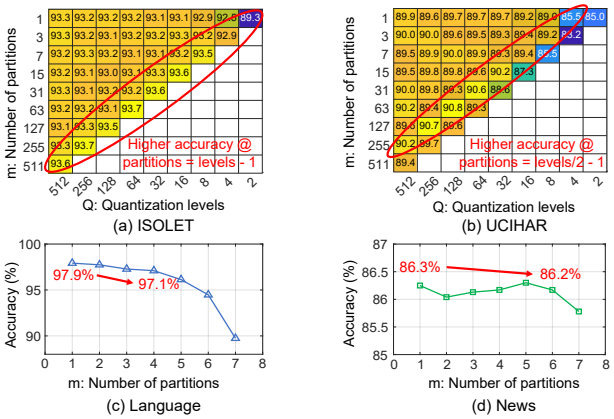


Fig. 8. Accuracy analysis of the proposed HV merging on four datasets.

The value of m affects the performance improvement and the accuracy in applications. Both effects are evaluated to select the optimal m in Section V-B and V-D.

V. EVALUATION AND DISCUSSION

A. Experimental Setup

We evaluate the proposed CafeHD at both hardware and software levels. For hardware-level evaluation, the array-level SPICE simulations are carried out. A widely used FeFET model in [20] and a 45nm CMOS process are utilized. The FeFET write pulse is $\pm$4V, and the pulse width is 10ns. V_R and V_{open} are set to 1V and 2.5V, respectively. The capacitors in the XOR and MAJ array are set to 1fF with 2.6% variation and 2fF with 1.4% variation, respectively [21]. A parasitic capacitance scaled by the array size is added for each line.

For software-level evaluation, CafeHD is benchmarked on four popular classification tasks, and Table II shows the dataset parameters. All HV dimensionality is 10000. Device variations and SA offsets are considered in the accuracy evaluation.

Four state-of-the-art NVM-based in-memory HDC encoding designs, including a ReRAM-based design (i.e., FELIX [6]), a PCM-based design [7], and two FeFET-based designs (i.e., MIMHD [8] and [9]), are compared as the baselines. FELIX [6] and MIMHD [8] are mainly designed for record-based encoding, while [7] and [9] are designed for N-gram encoding.

B. Accuracy Analysis

Here we evaluate the impact of the proposed level HV generation strategy for the HV merging technique. Fig. 8 shows the accuracy under different quantization levels (Q) and number of partitions (m). On the ISOLET and UCIHAR datasets, accuracy shows a falling trend as Q increases, while m has a small impact. Note that there is even a $\sim$0.5% accuracy improvement when $m = Q-1$ in the ISOLET dataset or $m = \frac{Q}{2}-1$ in the UCIHAR dataset. This is because $\vec{L}_1$ and $\vec{L}_Q$ are almost orthogonal when $m = Q-1$, thus achieving the best quantization. Besides, when $m = \frac{Q}{2}-1$, $\vec{L}_1 \sim \vec{L}_{Q/2}$ shows a symmetrical relationship with $\vec{L}_{Q/2+1} \sim \vec{L}_Q$, which exactly maps the data symmetry in the UCIHAR dataset. On the Language and News datasets, any two level HVs should be quasi-orthogonal to represent characters. Therefore, accuracy drops with more partitions due to the decrease in orthogonality.

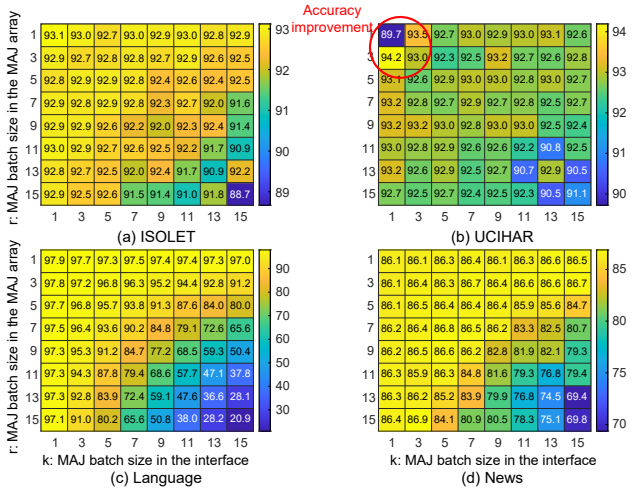


Fig. 9. Accuracy evaluation of the partial MAJ operations on four datasets.

Fig. 9 shows the accuracy for different MAJ batch sizes in the interface (k) and MAJ array (r). In all datasets, accuracy decreases when k and r increase. Since the data in the UCIHAR dataset are initially grouped, partial MAJ operations can provide >0.8% improved accuracy for this dataset.

C. Reliability Evaluation

To optimize the MAJ batch size k and r, we evaluate the tradeoff between reliability and efficiency. Fig. 10(a) shows the impact of k in the proposed partial MAJ interface. More writes can be reduced as k increases. However, when k exceeds 7, the write times reduction becomes insignificant, while the required capacitance of C_{acc} becomes intolerable. Hence we select $k = 7$ as the sweet spot with 85.7% write times reduction and $\sim 2\frac{C_{acc}}{C_{xor}}$. Fig. 10(b) shows the impact of r in the MAJ array. Smaller r induces more extra writebacks to the LCUs. Meanwhile, thanks to the charge-domain computing, the noise margin is still large when r reaches 15. Therefore, we consider to set r as large as possible while maintaining high accuracy. These optimizations in total reduce $\sim$84% writes on average.

D. Energy and Latency Evaluation

Fig. 11(a) shows the energy and latency of the proposed XOR array, which increases with more rows due to the parasitic capacitance. Writing and reading an LCU consume 40.7fJ/bit and 5.1fJ/bit, respectively. The charge distribution in the interface and MAJ array costs <1fJ energy per cell and <0.5ns latency. These low-power operations are attributed to the charge-domain computing throughout CafeHD.

Fig. 11(b) analyzes the impact of the HV merging on the performance of the XOR array. By reducing the required rows of the XOR array for HDC encoding, the HV merging technique brings 41%-170% energy and 39%-133% latency reduction on the four datasets. Therefore, the combination of

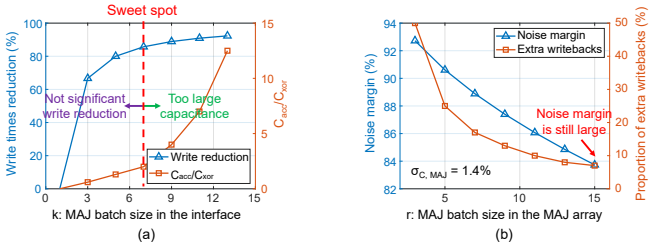


Fig. 10. Tradeoff between reliability and efficiency. (a) Impact of MAJ batch size in the interface. (b) Impact of MAJ batch size in the MAJ array.

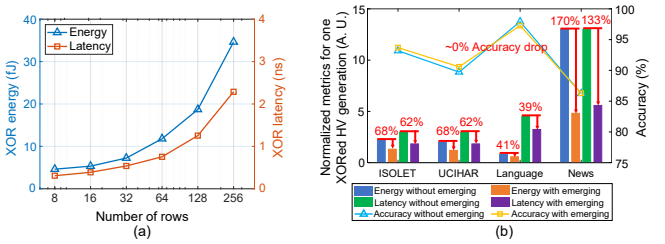


Fig. 11. (a) Energy and latency of the proposed XOR array. (b) Comparison of the XOR array performance and accuracy with and without HV merging.

charge-domain computing and HV merging enables CafeHD with ultra-high energy efficiency.

E. Overall Comparison with SOTA Designs

Table III shows the performance and accuracy comparison between the proposed CafeHD and existing NVM-based in-memory HDC encoding designs on the four datasets. On the ISOLET and UCIHAR datasets, CafeHD is compared with FELIX [6] and MIMHD [8]. The energy efficiency improvement is 101.1-105.5× and 7.4-7.9×, and the speedup is 10.3-11.4× and 1.7-1.9×, respectively. On the Language and News datasets, the energy efficiency is raised by 21.9× and 4.9×, and the latency is reduced by 12.7× and 5.1× on average compared with [7] and [9], respectively. Thanks to the charge-domain computing, the proposed interface, and the HV merging technique, CafeHD takes an all-around advantage of performance compared with all the existing designs.

For accuracy, it is seen that CafeHD shows the highest accuracy for all datasets except News thanks to reliable charge-domain computing and the proposed HV merging technique.

F. More Discussions and Future Work

CafeHD is mainly optimized for energy-efficient HDC encoding design. Although LCUs have been adopted in CafeHD to reduce the capacitor area overheads, lower parallelism and extra writebacks still may affect performance. More designs for a better balance between area and performance can be explored. Besides, the multi-level FeFET-based HDC encoders can be also designed to improve accuracy in the future.

VI. Conclusion

This paper proposes CafeHD, a charge-domain FeFET-based in-memory HDC encoder that achieves the highest reported energy efficiency. The proposed charge-domain in-memory design significantly improves energy efficiency at low area cost. Through the proposed XOR array with multiple input selections and partial MAJ interface, CafeHD reduces memory writes and supports two widely used encoding methods. We further propose an HV merging technique to bring extra improvement in energy efficiency and performance. Results

TABLE III
Energy per Input Vector, Latency per Input Vector, and Accuracy Compared With Existing In-Memory HDC Encoders

Dataset		FELIX [6]	MIMHD [8]	[7]	[9]	**CafeHD**
	NVM type	ReRAM	FeFET	PCM	FeFET	**FeFET**
ISOLET	Energy (nJ)	3900	274.8	/	/	**36.97**
	Latency (ns)	2200	360.0	/	/	**193.4**
	Accuracy	92.3%	91.8%	/	/	**93.3%**
UCIHAR	Energy (nJ)	3400	266.7	/	/	**33.62**
	Latency (ns)	2000	327.4	/	/	**193.4**
	Accuracy	91.2%	92.2%	/	/	**93.8%**
Language	Energy (nJ)	/	/	430.3	85.39	**15.05**
	Latency (ns)	/	/	1702	479.6	**114.2**
	Accuracy	/	/	92.8%	93.8%	**96.9%**
News	Energy (nJ)	/	/	1853	511.3	**121.7**
	Latency (ns)	/	/	10052	5863	**963.0**
	Accuracy	/	/	87.3%	/	**86.3%**

show that CafeHD achieves extraordinary energy efficiency, speedup, and write times reduction compared with the state-of-the-art NVM-based CiMs for HDC encoding.

Acknowledgment

This work is supported in part by NSFC (U21B2030, 92264204), in part by the National Key R&D Program of China (No. 2019YFA0706100), and in part by NSF (2008365, 2312866, 2235366).

References

[1] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive computation*, vol. 1, pp. 139–159, 2009.
[2] A. Rahimi *et al.*, "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing," in *ISLPED*, 2016, pp. 64–69.
[3] M. Imani, D. Kong, A. Rahimi *et al.*, "Voicehd: Hyperdimensional computing for efficient speech recognition," in *ICRC*, 2017, pp. 1–8.
[4] A. Rahimi *et al.*, "High-dimensional computing as a nanoscalable paradigm," *TCAS-I*, vol. 64, no. 9, pp. 2508–2521, 2017.
[5] M. Imani, X. Yin, J. Messerly *et al.*, "Searchd: A memory-centric hyperdimensional computing with stochastic training," *TCAD*, 2019.
[6] S. Gupta *et al.*, "Felix: Fast and energy-efficient logic in memory," in *International Conference on Computer-Aided Design*, 2018, pp. 1–7.
[7] G. Karunaratne, M. Le Gallo *et al.*, "In-memory hyperdimensional computing," *Nature Electronics*, vol. 3, no. 6, pp. 327–337, 2020.
[8] A. Kazemi *et al.*, "Mimhd: Accurate and efficient hyperdimensional inference using multi-bit in-memory computing," in *ISLPED*, 2021.
[9] Q. Huang, Z. Yang, K. Ni, M. Imani, C. Zhuo *et al.*, "FeFET Based In-Memory Hyperdimensional Encoding Design," *TCAD*, 2023.
[10] A. Keshavarzi, K. Ni, W. Van Den Hoek *et al.*, "Ferroelectronics for edge intelligence," *IEEE Micro*, vol. 40, no. 6, pp. 33–48, 2020.
[11] M. Imani, C. Huang *et al.*, "Hierarchical hyperdimensional computing for energy efficient classification," in *DAC*, 2018, pp. 1–6.
[12] L. Ge and K. K. Parhi, "Classification using hyperdimensional computing: A review," *IEEE Circuits and Systems Magazine*, 2020.
[13] M. Trentzsch *et al.*, "A 28nm HKMG super low power embedded NVM technology based on ferroelectric FETs," in *IEDM*, 2016, pp. 11–5.
[14] S. Dünkel *et al.*, "A FeFET based super-low-power ultra-fast embedded NVM technology for 22nm FDSOI and beyond," in *IEDM*, 2017.
[15] A. A. Sharma *et al.*, "High speed memory operation in channel-last, back-gated ferroelectric transistors," in *IEDM*, 2020, pp. 18–5.
[16] R. Cole *et al.*, "ISOLET," UCI Machine Learning Repository, 1994.
[17] J. Reyes-Ortiz *et al.*, "Human Activity Recognition Using Smartphones," UCI Machine Learning Repository, 2012.
[18] A. Rahimi *et al.*, "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing," in *ISLPED*, 2016, pp. 64–69.
[19] D. Lewis, "Reuters-21578 Text Categorization Collection," UCI Machine Learning Repository, 1997.
[20] S. Deng *et al.*, "A Comprehensive Model for Ferroelectric FET Capturing the Key Behaviors: Scalability, Variation, Stochasticity, and Accumulation," in *IEEE Symposium on VLSI Technology*, 2020, pp. 1–2.
[21] X. Ma *et al.*, "CapCAM: A Multilevel Capacitive Content Addressable Memory for High-Accuracy and High-Scalability Search and Compute Applications," *TVLSI*, vol. 30, no. 11, pp. 1770–1782, 2022.