# COSIME: FeFET based Associative Memory for In-Memory Cosine Similarity Search

Che-Kai Liu[1,2], Haobang Chen[1], Mohsen Imani[3], Kai Ni[4], Arman Kazemi[2], Ann Franchesca Laguna[5], Michael Niemier[2], Xiaobo Sharon Hu[2], Liang Zhao[1], Cheng Zhuo[1,*], and Xunzhao Yin[1,6,*]
[1]College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China
[2]Department of Computer Science and Engineering, University of Notre Dame, IN, USA
[3]Department of Computer Science, University of California, Irvine, CA, USA
[4]Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, NY, USA
[5]Department of Computer Technology, De La Salle University, Manilla, Philippines
[6]Zhejiang Lab, China *Corresponding authors, email: {czhuo, xzyin1}@zju.edu.cn

## ABSTRACT

In a number of machine learning models, an input query is searched across the trained class vectors to find the closest feature class vector in cosine similarity metric. However, performing the cosine similarities between the vectors in Von-Neumann machines involves a large number of multiplications, Euclidean normalizations and division operations, thus incurring heavy hardware energy and latency overheads. Moreover, due to the memory wall problem that presents in the conventional architecture, frequent cosine similarity-based searches (CSSs) over the class vectors requires a lot of data movements, limiting the throughput and efficiency of the system. To overcome the aforementioned challenges, this paper introduces COSIME, a general in-memory associative memory (AM) engine based on the ferroelectric FET (FeFET) device for efficient CSS. By leveraging the one-transistor AND gate function of FeFET devices, current-based translinear analog circuit and winner-take-all (WTA) circuitry, COSIME can realize parallel in-memory CSS across all the entries in a memory block, and output the closest word to the input query in cosine similarity metric. Evaluation results at the array level suggest that the proposed COSIME design achieves 333× and 90.5× latency and energy improvements, respectively, and realizes better classification accuracy when compared with an AM design implementing approximated CSS. The proposed in-memory computing fabric is evaluated for an HDC problem, showcasing that COSIME can achieve on average 47.1× and 98.5× speedup and energy efficiency improvements compared with an GPU implementation.

## 1 INTRODUCTION

Cosine similarity measures the similarity between two vectors in an inner product space. It is widely used in a number of machine learning models such as hyperdimensional computing (HDC) and deep neural networks (DNNs). During the inference phase of these machine learning applications, a large number of cosine similarity-based searches (CSSs) are often needed. While CSS has been extensively studied in many algorithm-level approaches [1], and can be executed in digital machines, it requires a large number of multiplications, as well as $L_2$ normalizations and divisions. Moreover, given the extensive search operations required by many machine learning algorithms such as HDC classifications, CSS also causes massive data movements between the memory and the processing units, i.e. the memory wall problem, limiting its performance and efficiency. These challenges posed by CSS becomes even more significant when deployed in power and resource constrained scenarios [2], calling for innovative and efficient hardware design for CSS.

Compute-in-memory (CiM) is a promising architectural paradigm that enables operations across entire memory blocks by integrating some basic processing capabilities inside the memory to overcome the memory wall problem. For example, content addressable memories (CAMs) [3, 4], which support parallel search across the stored vectors in memory against the input query, have been proposed as associative memories (AMs) to accelerate inference for machine learning applications e.g., few-shot learning, transformer, etc., [5, 6]. Moreover, in conjunction with non-volatile memory (NVM) technologies such as resistive RAM (ReRAM), ferroelectric FET (FeFET), etc., and customized sense amplifier (SA), CAM designs have demonstrated great potential as highly energy efficient AMs for nearest neighbor (NN) search using the Hamming distance metric [6–9].

However, it can be seen in Fig. 1 that the CAM designs supporting Hamming distance based search achieves energy efficiency at the expense of non-negligible accuracy loss for classification tasks. The AM in [10] supports a specific approximated CSS by approximating the denominator of cosine calculation and exploiting the the quasi-orthogonal property of hyperdimensional vector. That said, such approximation still causes slight accuracy loss, and is limited only to the HDC application. Therefore, a more general CiM based CSS that can not only offer energy efficiency and performance improvements but also maintain comparable accuracy to the full precision CSS implemented in software is highly desirable.

In this paper, we propose COSIME, an energy efficient, FeFET based in-memory search engine that implements parallel CSS across the memory to identify the word closest to the input query. COSIME incorporates several novel circuit designs summarized below. FeFETs are exploited as non-volatile storage to store the pre-trained data words (e.g., pre-trained class vectors in machine learning applications) in two FeFET arrays, where one array enables a row-wise dot product calculation between the input query and all stored words and the other array implements bit counting of the stored vectors. Current-mode analog circuits are employed to efficiently


*ICCAD '22, October 30-November 3, 2022, San Diego, CA, USA*

ACM ISBN 978-1-4503-9217-4/22/10…$15.00
https://doi.org/10.1145/3508352.3549412

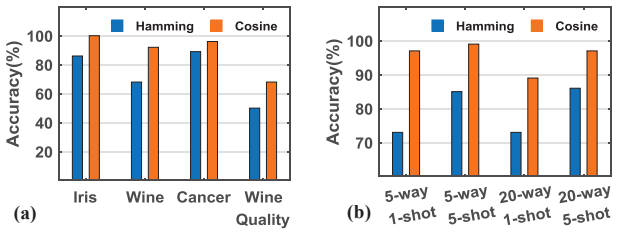


**Figure 1: Accuracy of (a) nearest neighbor classification, (b) few-shot learning tasks with Hamming distance based search and CSS [7], respectively.**

realize the cosine similarity calculation as well as the NN search. Since the analog circuits are independent of the memory array, the proposed COSIME design is not limited to FeFET technology, but can also be applied to other NVMs with access transistors.

To validate the functionality and evaluate the scalability and robustness of COSIME, NN searches based on cosine similarity are performed and analyzed. Evaluation results suggest that COSIME achieves 90.5× energy saving and 333× latency reduction compared to the AM design that implements approximated CSS [10]. To benchmark COSIME at the application level, we use COSIME for HDC classification where COSIME is implemented as an inference-accelerating AM. In this setup, COSIME demonstrates 47.1× and 98.5× speedup and energy efficiency improvement while maintaining the same classification accuracy compared with a GPU. To the best of our knowledge, COSIME is the first CiM design that supports NN search based on accurate cosine similarity.

## 2 BACKGROUND

In this section, we review FeFET basics and justify why FeFET is a favorable choice for implementing CiM based CSS. We then summarize recent efforts on designing AMs for similarity search.

### 2.1 FeFET Basics

FeFET based on recently discovered ferroelectric $HfO_2$ is a competitive candidate for high-speed, high-density, and low-power embedded NVM due to its intrinsic transistor structure, CMOS compatibility, excellent scalability, and superior energy efficiency [11]. As shown in Fig. 2(a), applying a positive (negative) gate voltage pulse sets the FeFET to low-$V_{TH}$ (high-$V_{TH}$) state (Fig. 2(b)). Unlike others NVMs whose memory write is driven by current and consumes significant power, FeFET exhibits superior write energy efficiency since the polarization switching is driven by an electric field, rather than large conduction currents [6].

It is demonstrated in [12, 13] that by connecting a series resistor with proper resistance value on the FeFET source/drain, the ON state current will be only limited by the series resistance, as shown in Fig. 2(c). As a result, the ON state current variation is significantly reduced with such 1FeFET1R structure and made independent from the FeFET $V_{TH}$ variation. This suggests possible tuning to the FeFET ON current for both the low-$V_{TH}$ and high-$V_{TH}$ states. [14] experimentally demonstrated a back-end-of-line (BEOL) 1FeFET1R structure, validating the aforementioned ON current tuning scheme with smaller cell area than other devices. A resistor with less than 8% variability is demonstrated. Given the small 1R variability and relatively large resistance of R, the ON state

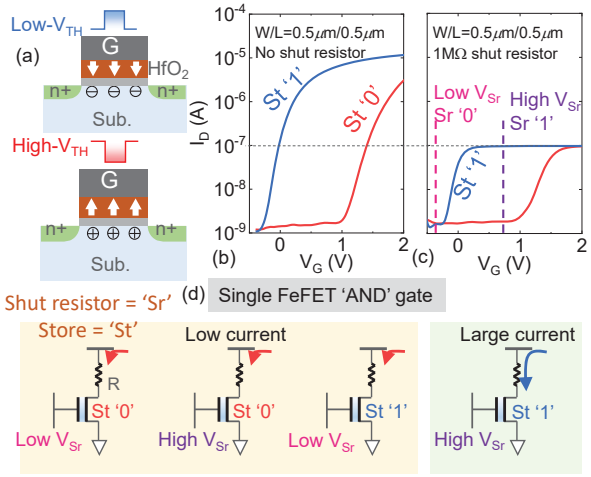


**Figure 2: (a) FeFET operation principles. $I_D$-$V_G$ characteristics of the high-$V_{TH}$ and low-$V_{TH}$ states for (b) a single FeFET and (c) a FeFET with a series resistor on the drain. (d) A single FeFET can compactly realize the AND gate.**

current of 1FeFET1R cell is approximately proportional to $\frac{1}{R}$ due to the large $M\Omega$ resistance of 1R, and the ON state current variation $\Delta I$, i.e., the derivation of ON state current, is proportional to $-\frac{\Delta R}{R \times R}$, where $\frac{\Delta R}{R}$ refers to the 1R variability. Therefore, the impact of the 1R variability on the ON state current is negligible [12].

In this work, the 1FeFET1R structure is adopted, and proposed to realize a compact AND gate (i.e., dot product for binary vectors) by storing one operand as the FeFET $V_{TH}$ state and applying the other operand as the gate voltage, as shown in Fig. 2(d) [15]. Such cell is leveraged in this work to calculate the cosine similarity in-memory.

### 2.2 Existing Associative Memory with Similarity Search

With advances in NVMs, binary/ternary/multi-bit CAM design have been proposed for energy efficient and ultra-dense associative search in various applications, e.g., IP routers, look-up table, reconfigurable computing and machine learning models, etc. [16–20]. Typically, CAM works in the exact match mode, in which only the stored vectors that exactly match the query are identified. However, NN search is also highly desirable as it identifies the vector closest to the query, a core computation in many machine learning models. With the exact matching mode, to identify the NN, multi-step searches with queries of increasing distance to the target query are applied, incurring overheads in energy and latency.

This can be overcome by leveraging the approximate matching mode of CAM, which directly computes the Hamming distance on the match line (ML) of CAM. Recent work [6–9] implemented NN search based on Hamming distance for few-shot learning tasks. However, they suffer from significant accuracy loss compared with CSS, as shown in Fig. 1. Recently, an AM supporting approximate CSS was proposed in [10]. This design specifically targets to HDC classification problems and approximates the denominator of cosine calculation by exploiting the quasi-orthogonal property of hyperdimensional vectors, thus limiting its application to other

machine learning models. In this work, we propose a more general AM design that supports NN search based on cosine similarity.

## 3 COSIME: IN-MEMORY COSINE SIMILARITY SEARCH ENGINE

COSIME implements in-memory cosine similarity search for the NN, a critical operation in the inference phase of BNN and HDC models as well as other machine learning models (e.g., for few-shot learning). Fig. 3 shows the architecture of COSIME. It consists of two FeFET memory arrays, a current-mode translinear circuit block for each row in the memory arrays, and an analog winner-take-all (WTA) circuit. Below we elaborate the cosine similarity metric derivation, as well as the detailed design of the COSIME components.

### 3.1 Cosine Similarity

Cosine similarity (or cosine distance) has been used as a distance metric for measuring the difference between an input feature query vector and the stored vectors. Specifically,

$$\cos\langle \vec{a}, \vec{b} \rangle = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \times \|\vec{b}\|} \tag{1}$$

Without loss of generality, in this work we assume that $\vec{a}$ is the binary input vector whose bits are either 0 or 1, and $\vec{b}$ as the class binary vector stored in the memory block. Eq. (1) equals to 1 means that $\vec{a}$ and $\vec{b}$ are exactly the same, while Eq. (1) equals to 0 means that $\vec{a}$ and $\vec{b}$ are orthogonal to each other. The numerator of Eq. (1) is simply the dot product of two vectors. The denominator is the product of the $L_2$ norm of the two vectors, while $L_2$ norm of a binary vector is the square root of the number of '1's in the vector. Directly computing $L_2$ norm requires a complex circuit, hindering the denominator from efficient hardware implementation. Therefore, prior works typically simplify the cosine equation by removing the denominator, or approximating the denominator to a constant value. Doing so may introduce significant errors.

COSIME aims to obtain the closest vector (i.e., NN) to the input query in terms of cosine similarity. From Eq. (1), cosine similarity can be equivalently expressed in a more circuit friendly variant without affecting the search output. Specifically, for computing cosine similarity, COSIME removes the need for square root operation without any accuracy loss by squaring both the numerator and denominator as shown in Eq. (2).

$$\cos^2\langle \vec{a}, \vec{b} \rangle = \frac{(\vec{a} \cdot \vec{b})^2}{(\|\vec{a}\| \times \|\vec{b}\|)^2} \tag{2}$$

Note that in Eq. (2), the denominator consists of the squared norm of the stored vector $\vec{b}$ which is the number of '1's within $\vec{b}$, and squared norm of input query vector $\vec{a}$ which is shared by all the cosine similarity metrics, and thus can be removed during the CSS. In this sense, the cosine similarity metric can be equivalently expressed as the $X^2/Y$ operator, where $X$ denotes the dot product $(\vec{a} \cdot \vec{b})^2$, and $Y$ denotes $\|\vec{b}\|^2$, i.e., the number of '1's within $\vec{b}$. Based on the above formulation, we illustrate the circuits implementing the computation of $X$, $Y$ and $X^2/Y$.

### 3.2 FeFET Memory arrays

We employ two identical FeFET-based non-volatile memory arrays to store the class vectors. As shown in Fig. 3(a), the gates of the FeFETs within a column are connected to the bitlines (BLs), while the drains of the FeFETs within a word share a wordline (WL). As discussed in Sec. 2.1, FeFETs can be used as a single transistor AND gate, enabling the memory array implementing in-memory binary bitwise dot product naturally.

During search, for the FeFET memory array on the left side of Fig. 3(a), high/low voltages are applied to the bitlines $BL$ according to the respective bit values in the input query. Only when the FeFET of a cell stores '1' (corresponding to low $V_{TH}$ state), and its gate voltage is at high level indicating an input bit '1', the cell is turned on, conducting $I_{ON}$ current from the wordline WL to ground. The resulting output current ($I_x$) flowing through a WL therefore represents the dot product of this word and the input query, i.e., $X$. To implement the squared norm of the stored vector, i.e., $Y$, the FeFET memory array on the right side of Fig. 3(a) is used, and stores the identical class vectors as the left memory array. All the bitlines of this array, however, are applied with high gate voltage, turning on the FeFETs storing '1'. It is easy to see that the magnitude of the output current $I_y$ of a word in this array represents the number of '1's within the stored vector, i.e., $Y$. Note that the magnitude of the output currents $I_x$, $I_y$ can be adjusted by tuning the resistor within the 1FeFET1R structure as discussed in Sec. 2.1, ensuring that the input currents are in the working range of the following translinear circuit model.

### 3.3 Translinear Circuits

To implement the key operation $X^2/Y$ for CSS, we propose to employ the translinear circuit from [21] and feed the output currents of FeFET memory arrays $I_x$ and $I_y$ into this analog arithmetic circuit. Fig. 3(b) shows the schematic of the translinear circuit implementing efficient current-mode squaring and division. This translinear circuit mainly consists of a translinear loop (indicated by the blue arrow) including clockwise (CW) transistors M1, M4 and counter-clockwise (CCW) transistors M2, M5. The transistors along the loop are operating in the subthreshold (weak inversion) region, and their drain-source currents can be characterized by the following expression [22]:

$$I_{DS} \approx I_0 \frac{W}{L} e^{\frac{V_{GS}}{\eta V_T}} \tag{3}$$

where $I_0$ denotes the drain current $I_D$ when $V_{GS} = V_T$, $V_T$ denotes the thermal voltage, $\eta$ the subthreshold slope factor.

The relation between the $V_{GS}$'s of the transistors along the translinear loop follows Kirchoff's Law, i.e.,

$$\sum_{CW} V_{GS} = \sum_{CCW} V_{GS} \tag{4}$$

from Eq. (3), we obtain:

$$V_{GS} = V_T \eta ln(\frac{I_{DS}}{I_0}) \tag{5}$$

By substituting Eq. (5) into Eq. (4) while keeping the loop transistors in the subthreshold region, the translinear circuit generates the analog output current $I_z$ as below:

$$I_z = \frac{I_x^2}{I_y} \tag{6}$$

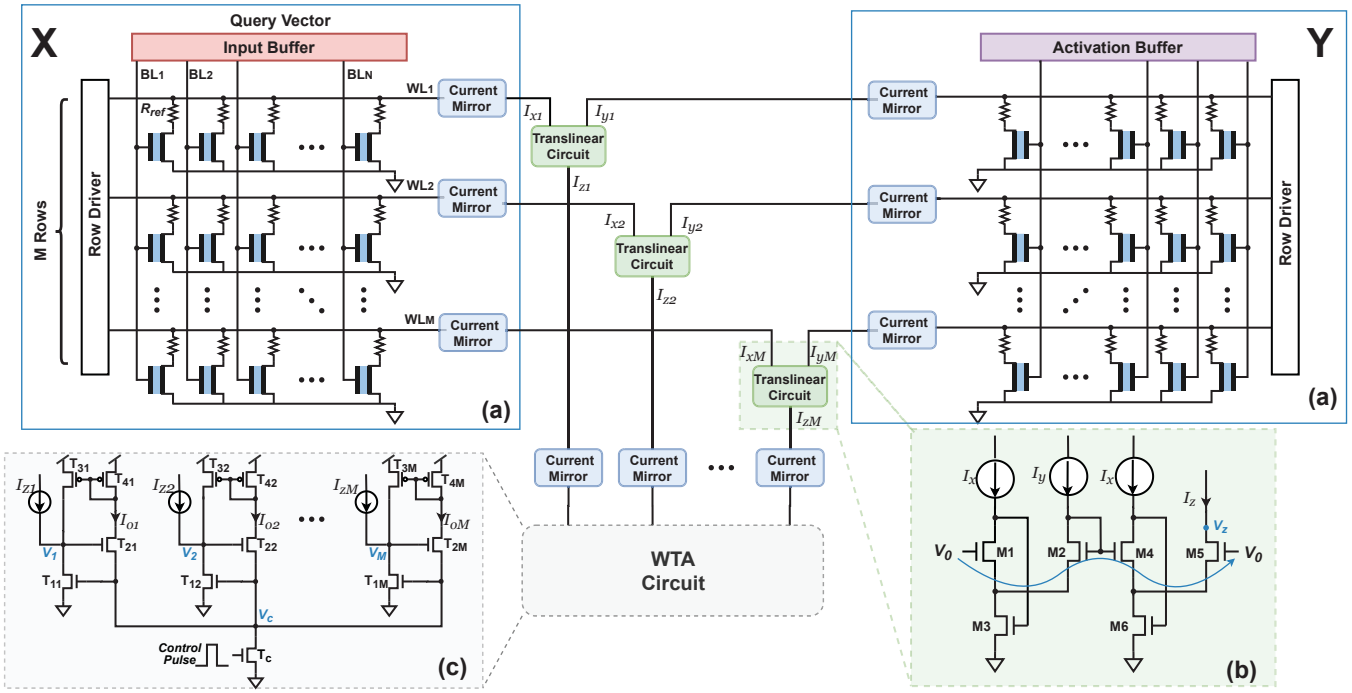


**Figure 3: COSIME overview. (a) 1FeFET1R memory array. (b) Translinear circuit. (c) Winner-Take-All (WTA) circuit.**

The operating voltage $V_0$ in Fig. 3(b) is set to 0.6V to keep the translinear circuit in the subthreshold region, and $I_y$ around 600nA corresponding to the average squared $L_2$ norm of the stored vectors. Fig. 4(a) shows the operating region with respect to the input current $I_x$, where the simulated transfer characteristic aligns with the theoretical result. It can be seen that the input current $I_x$ from the FeFET memory array should be within the operating range to guarantee the functionality of the translinear circuit. In order to maintain the correct functioning of the translinear circuit and guarantee the scalability of COSIME, we propose to adjust the resistor in every 1FeFET1R to satisfy the required input current range of the translinear circuit. For example, when the memory arrays scale to $N$ times row-wise, the input current per row from the memory arrays can remain constant by tuning the 1FeFET1R structure as presented in [12], thus reducing the 1FeFET1R cell ON current to $\frac{1}{N}$ times.

$$I_z = \frac{(\frac{I_x}{N} \times N)^2}{\frac{I_y}{N} \times N} = \frac{I_x^2}{I_y} \tag{7}$$

Moreover, the input current range can also be guaranteed by adjusting the size ratio $W/L$ of the current mirror associated with the translinear circuits.

## 3.4 Winner-Take-All Circuit

By employing the FeFET memory arrays and translinear circuits discussed above, the squared cosine distances between the stored vectors and the input query are effectively represented by the output currents of the translinear circuits. The last stage of the NN search is to find the maximum current as the ultimate search result. The conventional maximum current selection implementation is

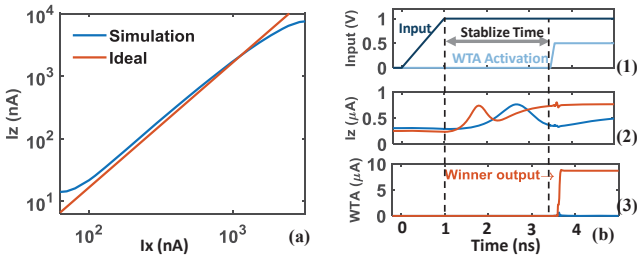


**Figure 4: (a) The transfer characteristic of the translinear circuit, where the center linear region indicates the operating region for the input. (b) Waveforms of (1) input and WTA activation for NN search, (2) translinear circuit output, and (3) WTA output.**

a current comparator-based tree structure which requires a huge number of transistors and increases the latency as the number of stored class vectors increases [9]. Here, we propose to utilize a current-mode O(N) WTA circuit presented in [23], which can offer efficient maximum current detection operation.

Fig. 3(c) shows the schematic of the WTA circuit. It consists of a gated transistor $T_C$ as the current source, a coupled transistor pair (i.e., the sourcing transistor $T_{1i}$ and the output transistor $T_{2i}$) and an output feedback current mirror (i.e., $T_{3i}$ and $T_{4i}$) for each input and output branch. The WTA circuit generally operates by inhibiting the transistor pairs sourcing smaller input currents, and amplifying the transistor pair sourcing the maximum input current. When one of the input currents is larger than others, the gate voltage $V_c$ of the corresponding sourcing transistor is driven to a higher level, while the drain voltages of other sourcing transistors are driven to

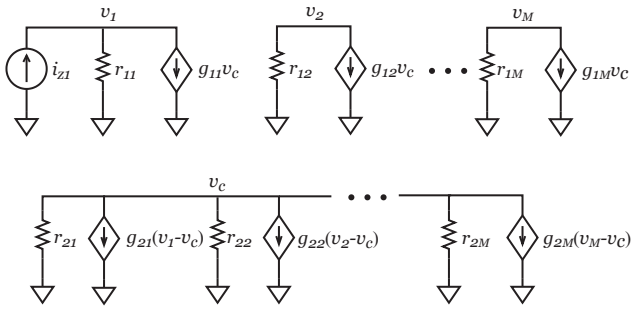


**Figure 5: Small-signal model of the M-rail-input WTA circuit.**

a lower level to maintain the smaller input currents. As a result, the reduced drain voltages will drive less output currents through the output transistors, and the output transistor corresponding to the maximum input drives a larger output current. The feedback current mirror on the output path adds the output current back to the input path, thus further exacerbating the current differences among the inputs. Such a WTA circuit therefore can distinguish input currents with even 1% difference. To this end, the WTA realizes NN search in cosine similarity metric. Fig. 4(b) shows the transient input and output waveforms of the WTA circuit. Note that to ensure correct WTA functionality, the WTA is activated after the input currents, i.e. the translinar circuit outputs, become stable.

## 3.5 Scalability of WTA Circuit

The 2-rail-input WTA circuit demonstrated in [9] may cost a large number of two-rail-input WTA components to construct a comparison tree. Instead, we hereby deploy an M-rail-input WTA circuit in [24] to our COSIME design. In [24], the derivation 2-rail-input WTA's output w.r.t. the input changes is given. To understand M-rail-input WTA's counterpart, below we elaborate why the transfer characteristics between the inputs and the outputs of WTA are weakly correlated with the number of input rail in COSIME, thus validating the scalability of COSIME by using the M-rail-input WTA circuit.

Fig. 5 shows the small-signal circuit model of the M-rail-input WTA circuit excluding the feedback current mirrors. The small signal of $V_1$, $V_2$, and $V_c$ are denoted as $v_1$, $v_2$, and $v_c$, respectively. For a particular operating point $[I_{z1}, \ldots, I_{zM}, I_{o1}, \ldots, I_{oM}]$, without loss of generality, we assume a small change in $I_{z1}$, denoted as $i_{z1}$. the corresponding small-signal parameters of the sourcing and output transistors in the subthreshold region are $g_{11} = \frac{I_{z1}}{V_T} \ldots g_{1M} = \frac{I_{zM}}{V_T}$, $g_{21} = \frac{I_{o1}}{V_T} \ldots g_{2M} = \frac{I_{oM}}{V_T}$, $r_{11} = \frac{V_A}{I_{z1}} \ldots r_{1M} = \frac{V_A}{I_{zM}}$, and $r_{21} = \frac{V_A}{I_{o1}} \ldots r_{2M} = \frac{V_A}{I_{oM}}$, where $V_A$ is the Early voltage and $V_T$ is the thermal voltage. Applying Kirchhoff's current law to the small signal circuits in Fig. 5 yields:

$$\begin{cases} i_{z1} = v_1 \dfrac{I_{z1}}{V_A} + v_c \dfrac{I_{z1}}{V_T} \\ v_j \dfrac{I_{zj}}{V_A} = -v_c \dfrac{I_{zj}}{V_T}, \quad \forall j \in [2, M] \\ \displaystyle\sum_{i=1}^{M} [\frac{I_{oi}}{V_T}(v_i - v_c) + v_c \frac{I_{oi}}{V_A}] \end{cases} \tag{8}$$

Given that Early voltage $V_A >> V_T = kT/q$, solving Eq. 8 yields:

$$\begin{aligned} \frac{dV_1}{dI_{z1}} &= \frac{v_1}{i_{z1}} = \frac{1}{I_{z1}}(V_T + V_A(1 - \frac{I_{o1}}{I_c})) \\ \frac{dV_j}{dI_{z1}} &= \frac{v_j}{i_{z1}} = \frac{-1}{I_{z1}} V_A(\frac{I_{o1}}{I_c}) \end{aligned} \tag{9}$$

where $I_c = \sum_{j=1}^{M} I_{oj}$. The $j$th current $I_{oj}$ (see Fig. 3(c)) of the output transistor operating in the subthreshold region can be expressed as:

$$I_{oj} = I_o exp((V_j - V_c)/V_T) \tag{10}$$

then the term in Eq. 9

$$\frac{I_{o1}}{I_c} = \frac{1}{1 + \sum_{j=2}^{M} exp((V_j - V_1)/V_T)} \tag{11}$$

Substituting Eq. 11 into Eq. 9 obtains:

$$\begin{aligned} \frac{dV_1}{dI_{z1}} &= \frac{1}{I_{z1}}(V_T + V_A(1 - \frac{1}{1 + \sum_{j=2}^{M} exp((V_j - V_1)/V_T)})) \\ \frac{dV_j}{dI_{z1}} &= -V_A \frac{1}{I_{z1}}(\frac{1}{1 + \sum_{j=2}^{M} exp((V_j - V_1)/V_T)})) \end{aligned} \tag{12}$$

Given that the initial input currents $I_{z1} = I_{z2} = \cdots = I_{zM} = I_m$, the initial gate voltage of $T_{21}, \ldots, T_{2M}$ should be identical, i.e., $V_m$. Eq. 12 can be simplified as below:

$$\begin{aligned} &\frac{dV_1}{dI_{z1}} = \frac{1}{I_{z1}}(V_T + V_A) \text{ and } \frac{dV_j}{dI_{z1}} = 0, \quad \text{when } V_j - V_1 \gg V_T \\ &\frac{dV_1}{dI_{z1}} = \frac{V_T}{I_{z1}} \text{ and } \frac{dV_j}{dI_{z1}} = -\frac{V_A}{I_{z1}}, \quad \text{when } V_j - V_1 \ll V_T \end{aligned} \tag{13}$$

It can be seen that with $V_j - V_1 \gg V_T$ and $V_j - V_1 \ll V_T$, the dynamics of the output transistors with the input current are independent of the number of input rails M. When $V_j \approx V_1 \approx V_m$, Eq. 12 simplifies to:

$$\begin{aligned} \frac{dV_1}{dI_{z1}} &= \frac{M-1}{M}\frac{V_A}{I_{z1}} \\ \frac{dV_2}{dI_{z1}} &= \frac{-1}{M}\frac{V_A}{I_{z1}} \end{aligned} \tag{14}$$

In 2-rail-input WTA, $V_1$ is a linear function of $I_{z1}$ with a slope of $\frac{V_A}{2I_{z1}}$ [25], and in an M-rail-input WTA, $V_1$ is a linear function of $I_{z1}$ with a slope of $\frac{M-1}{M}\frac{V_A}{I_{z1}}$ as shown in Eq. 14. As the number of input rails scales up, the winner's behavior w.r.t. the dynamics of the output transistor only differs by a constant factor. While for losers $j \in [2, M]$, as the input rail number increases, the behavior w.r.t. the dynamics differ by $\frac{1}{M}$. To this end, the impact of the number of input rails on the winner's output is negligible, which can be seen from Fig. 6(a), where the latency of COSIME changes little with increasing number of input rails, i.e., number of class vectors.

# 4 EVALUATION

In this section, we first evaluate COSIME in terms of energy and latency at the array level. We then investigate the scalability and robustness of COSIME upon device variations. We finally benchmark COSIME for binary HDC inference and compare it with a GPU implementation. We have simulated all the COSIME components at the circuit-level with Cadence Spectre. The write voltage for the 1FeFET1R CAM is ±4$V$. The 45nm PTM high-performance model is adopted for CMOS transistors [26], and the Preisach model [27]

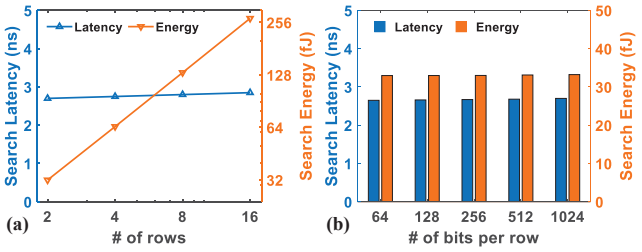


**Figure 6: Search energy and delay of COSIME with (a) varying number of rows/vectors (1024 bits per row), and (b) varying number of dimensions, respectively.**

is used for FeFET. The array wordlength is 1024 bits. The search delay is measured from the beginning of the search operation when the FeFET memory arrays are activated until the WTA circuit generates the output. Besides, the search delay is measured under the worst case, where two non-identical stored vectors are closest to each other, i.e., they only differ by 1 bit at the denominator, and the resulted squared cosine similarities are $cos^2\theta = 1/4$ and $1/5$, respectively.

## 4.1 Array-Level Evaluation

Fig. 6(a) shows the latency and energy trends of COSIME in terms of the number of words in the memory array. It can be seen that the increasing class vectors participating the NN search of COSIME have negligible impacts on the latency, aligning with the discussion in Sec. 3.5. The search energy of COSIME mainly consists of two parts: the WTA circuit along with its amplification current mirrors consuming up to 56% of the total energy, and the squared cosine translinear circuits along with their associated current mirrors taking around 43%. As the number of classes increases, i.e., the number of current paths of WTA circuit increases, the total search energy grows linearly. This is due to the fact that the increasing number of the WTA circuit branches introduces more input and output currents provided by the supply rails.

In addition to varying the number of rows (i.e., classes) within the FeFET memory arrays, we also investigate the scalability of COSIME by varying the number of bits in a word (i.e., dimensions) in terms of energy and latency metrics. As pointed out in Sec. 3.3, we maintain the correct functionality of translinear circuit and thus COSIME by tuning the resistor within the 1FeFET1R structure in the FeFET memory arrays. Fig. 6(b) reports the search energy and latency of COSIME with varying number of bits per word in the memory arrays. As can be seen, the latency and search energy of COSIME has negligible change when increasing wordlength from 64 to 1024, as the total current provided by the supply rails is kept the same based on the tuning method discussed in Sec. 3.3.

Here we also validate the robustness of COSIME design upon device variability. The device-to-device variability of the FeFETs is extracted from [12], i.e., $\sigma_{LVT} = 54mV$ for low-$V_{TH}$ state and $\sigma_{HVT} = 82mV$ for high-$V_{TH}$ state. The variability of the resistor in the 1FeFET1R cell is extracted from [14], i.e., 8%. The MOSFET device is assumed with 10% size and 10% $V_{TH}$ variations, and the supply voltage is assumed with 10% variation. Fig. 7(a) shows the output waveforms of COSIME for 100 Monte Carlo simulations. The array-level results indicate a 90% search accuracy of COSIME with

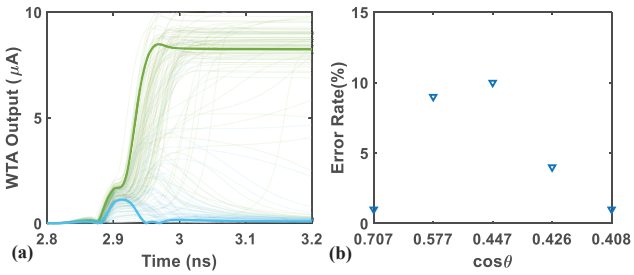


**Figure 7: Monte Carlo simulations considering all device-to-device variations: (a) The output waveforms of COSIME in the worst-case, achieving 90% accuracy. (b) The error rates of different cosine similarity outputs with an output $cos\theta = 0.5$.**

similarity threshold $\cos\theta = 0.5$ even in the worst case[1]. Fig. 7(b) shows the array-level search function error rates of COSIME generating different cosine similarity values when one entry of the array generates an output corresponding to $cos\theta = 0.5$. It can be seen from Fig. 7(b) that as the cosine similarities of two stored words with the input query get closer, the error rate of COSIME increases, due to the closer current inputs to the WTA circuit. Yet the maximum error rate is $\approx 10\%$, which would have minimal impact on the application-level accuracy for many machine learning and neuromorphic applications, such as HDC [9, 10, 29].

Fig. 8 demonstrates and compares different types of AMs implementing various distance metric calculations between the input query and stored vectors for neural network or HDC models. Following the feature extractor and additional function layer as shown in Fig. 8(a), employing conventional memory (Fig. 8(b)) such as DRAM to support NN search incurs significant data movement overhead as all the memory entries need to be sequentially transferred from the memory unit to the processing unit to calculate the cosine similarity between the input query and stored vectors. FeFET based AM in [6] (Fig. 8(c)) deploys a 2FeFET TCAM array to implement approximate search in terms of Hamming distance between the input query and stored vectors. Fig. 8(d) from [7] exploits the multi-bit characteristic of FeFET to build an AM implementing NN search in a novel distance metric (i.e., MCAM distance). Compared with prior AM designs, COSIME (Fig. 8(e)) achieves superior search energy, performance and area overhead, while still maintaining high accuracy as an AM implementing NN search in cosine similarity distance metric.

TABLE 1 summarizes the distance metric, search energy per bit and latency of different AM designs. The results show that COSIME offers 90.5× more energy efficiency and 333× less latency than the approximated CSS design from [10]. In comparision with the existing AMs with Hamming distance [9] or recently proposed Euclidean distance [30], COSIME is also superior in terms of search energy and latency. The significant improvements of COSIME over the counterpart approximated CSS design mainly benefit from the following aspects: (1) the advantages of FeFET in read/write energy [6]; (2) the 1FeFET1R structure limiting the conducting current within COSIME, which improves the energy efficiency and functionality;

[1]The worst case refers to that the cosine values corresponding to the WTA outputs are $cos\theta = 1/2$ and $1/\sqrt{5}$, respectively. In this case, the two stored vectors differ by 1 bit, which is the harshest situation for WTA circuit to distinguish the corresponding array currents. Such assumption is pessimistic.

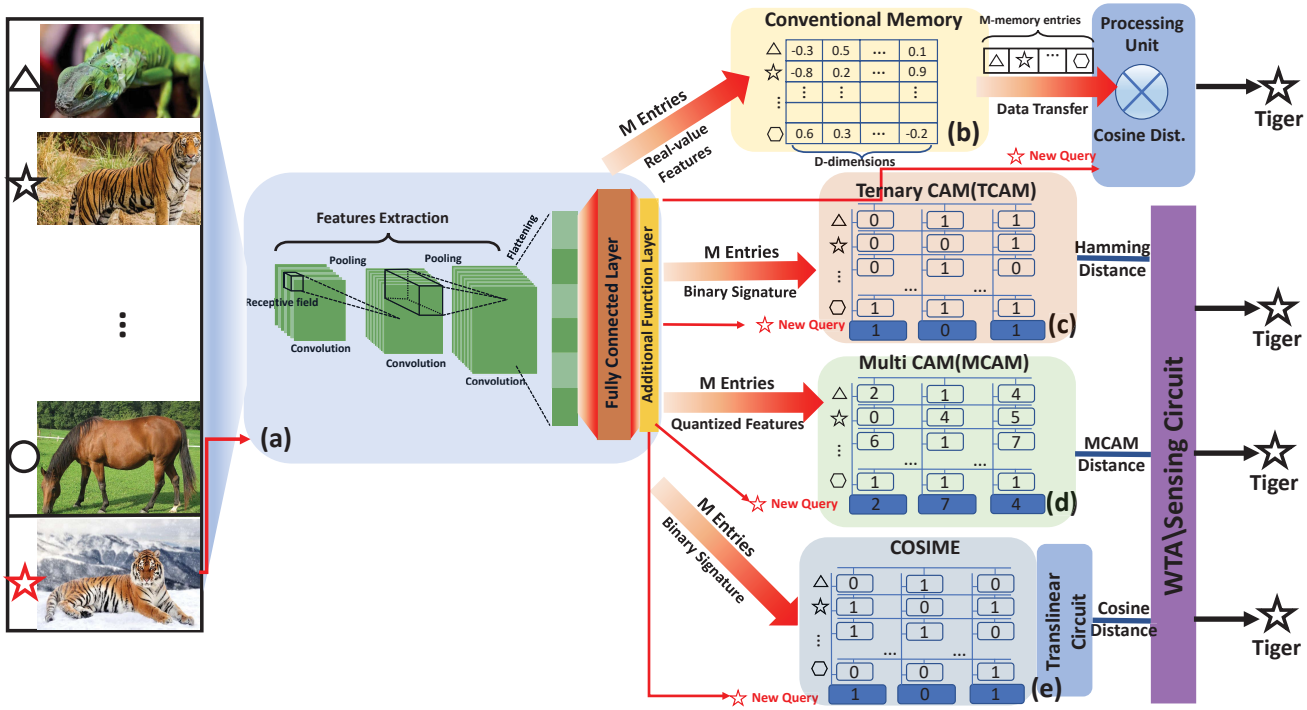


**Figure 8: Different AM based implementation along with (a) the feature extractor such as CNN [6] and HDC [28]. The additional function layer (AFL) following the feature extraction performs local sensitive hashing (LSH) [6] or quantization [7]. (b) Conventional memory such as DRAM. (c) FeFET based AM, e.g. [6], implementing approximate search in Hamming distance. (d) MCAM in [7], performing NN search in MCAM distance. (e) COSIME, performing CSS.**

and (3) relatively simple analog circuits in COSIME compared with the capacitor and analog-to-digital converter (ADC) in [10]. Moreover, TABLE 1 demonstrates the area overheads of different AMs. Both the A-HAM in [9] and the $E^2$-MCAM in [30] consume high area overhead since a tree-based loser-take-all (LTA) circuitry and sufficiently large flash cells supporting the 3-bit storage are used, respectively. The approximated CSS design in [10] consumes 1.31× area overhead than COSIME since it adopts ADC for its RRAM readout. On the contrary, COSIME exhibits ultra-low area overhead since: (1) the analog peripherals of COSIME arrays consume much less area than the peripherals of other designs; (2) ultra-compact 1FeFET1R structure has been successfully demonstrated without consuming extra area overhead [14].

## 4.2 Case Study: Hyperdimensional Classification

To validate the effectiveness of COSIME as an AM at application level, we benchmark our proposed COSIME array in the context of HDC models for classification as a case study. HDC is based on the understanding that the brain computes with patterns of neural activity that are not readily associated with numbers, and has been proven as effective for many cognitive tasks, such as object tracking [32], speech recognition [33], image classification [34, 35], etc. Due to the size of the brain's circuits, neural patterns can be modeled with hypervectors [36]. HDC builds upon a well-defined set of operations with random hypervectors, is extremely robust upon failures, and offers a computational paradigm that is easily applied to learning problems [37]. For HDC classification, the first step is to encode data into high-dimensional space. Then, HDC performs a learning task over encoded data by performing a single-pass training. The training generates a hypervector representing each class. During the inference phase, as an input query comes which contains the sample data to be classified, it is searched across the stored class hypervectors for the closest one in terms of cosine similarity. A class with the highest similarity to the query is selected as the inference prediction. HDC uses cosine similarity as an ideal distance metric, while prior work approximated with Hamming distance for easier hardware implementation. However, this approximation often results in accuracy loss.

Here, we evaluate HDC classification accuracy and efficiency over three large-scale data sets given in Table 2. Fig. 9(a) shows the HDC classification accuracy when the dimensionality of hypervectors varies from $D = 256$ to $D = 1k$. The results are reported using our proposed COSIME (cosine similarity) and Hamming distance [38] as the similarity metrics. The results show that HDC achieves maximum accuracy with dimensionality $D = 1k$. Reducing this dimensionality to 512 and 256 results in 1.7% and 12.2% accuracy loss. Our evaluation also indicates that by using cosine similarity for distance metric HDC achieves significantly higher accuracy (on average 7%) as compared to Hamming distance metric. Such observation is consistent with Fig. 1, and demonstrates the

**Table 1: Comparison of Existing AMs with Different Distance Metrics**

| Memory | Technology | Metric | Search Energy per bit($fJ$) | Latency($ns$) | Area*($mm^2$) | Process (nm) |
|---|---|---|---|---|---|---|
| **A-HAM** [9] | RRAM | Hamming | 0.20(×0.7) | 8.92(×2.9) | 0.524(×26.5) | 45 |
| **FeFET TCAM** [6] | FeFET | Hamming | 0.40(×1.4) | 0.36(×0.12) | 0.010§(×0.51) | 45 |
| **E$^2$-MCAM⋆ (1.5 V)** [30] | Flash | Euclidean$^2$ | 0.56(×1.95) | 5.85(×1.95) | 0.192(×9.7) | 55 |
| **Approx. Cosine** [10] | RRAM | Approx. Cosine | 25.9(×90.5) | 1000(×333) | 0.026¶(×1.31) | 90/65† |
| **COSIME (this work)** | FeFET | **Cosine** | **0.286**(×1) | **3**(×1) | **0.0198**(×1) | **45** |

*: Assuming 256 × 256 array size. §: Area associated with sensing is not included. ¶: Area is estimated via Neurosim3.0 [31] and scaled to 45nm technology for fair comparison. ⋆: E$^2$-MCAM stores 3 bits per cell for search, and the sensing circuitry energy is not included. †: NVM is based on 90nm CMOS while digital peripherals are based on 65m.

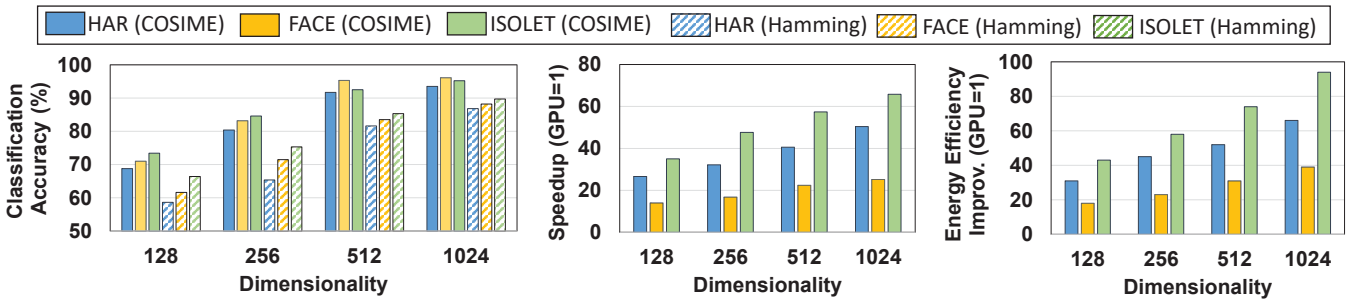


**Figure 9: (a) Classification accuracy of HDC using the proposed COSIME and Hamming distance as similarity metric. (b) Computation speedup and (c) energy efficiency improvement of COSIME compared to GPU.**

benefits of COSIME, which implements CSS for machine learning and HDC tasks. Regarding the errors induced by COSIME circuits, [9, 10, 29] have shown that the HDC classification is able to achieve negligible accuracy loss compared with the original accuracy with up to 20% error rate in the AM. Therefore, COSIME, as an AM for HDC, is robust to the device variation even considering the worst case for the HDC classification, as the maximum error rate ≈ 10%, which is below the HDC error tolerance.

In HDC, the associative search dominates both the training and inference phases (e.g., taking over 90% of training time [39]). Fig. 9(b), (c) show the energy efficiency improvement and execution time speedup of associative search running on COSIME over an NVIDIA 1080 GPU. Our results indicate that COSIME provides higher speedup and energy efficiency in higher dimensionality. For example, COSIME achieves 47.1× faster and 98.5× higher energy efficiency on average than GPU with $D = 1k$ dimensions. COSIME provides higher benefits for applications with more classes. For example, ISOLET, which has the highest number of classes (see Table 2), receives the highest speedup and energy efficiency compared to the GPU implementation. This efficiency comes from (1) the capability of COSIME based AM to enable fast and parallel search operations and (2) addressing data movement issues by eliminating data access to off-chip memory.

## 5 CONCLUSION

Hardware acceleration for CSS is important for edge intelligence and AI models. In this paper, we propose, for the first time, COSIME, a FeFET based AM that performs CSS in-memory. COSIME consists of compact FeFET memory arrays for dot product and squared $L_2$ norm operations, translinear circuits for squaring and division, and the WTA circuit for NN search. The functionality, scalability and robustness of COSIME have been validated. The energy and latency results of COSIME at the array level indicate 90.5× and 333× improvements over the state-of-the-art approximated CSS design, respectively. HDC application benchmarking suggests that COSIME achieves 47.1× speedup and 98.5× energy efficiency improvement over an GPU implementation. Note that the proposed COSIME design is not limited to FeFET technology, but is rather general and can be applied for other NVMs with access transistors. This is because the peripheral circuitry of COSIME is largely independent of the NVM array as long as the array output currents are within the sensing range. Therefore, COSIME paves a promising way towards efficient CiM designs for CSS in data-intensive applications.

**Table 2: Datasets ($n$: feature size, $K$: number of classes)**

| | $n$ | $K$ | Train Size | Test Size | Description |
|---|---|---|---|---|---|
| **UCIHAR** | 561 | 12 | 6,213 | 1,554 | Activity Recognition[40] |
| **FACE** | 608 | 2 | 522,441 | 2,494 | Face Recognition[41] |
| **ISOLET** | 617 | 26 | 6,238 | 1,559 | Voice Recognition [42] |

## ACKNOWLEDGEMENTS

This work was partially supported by the National Key R&D Program of China (2018YFE0126300), Zhejiang Provincial Key R&D program (2022C01232), NSF (LQ21F040006, LD21F040003), NSFC (62104213, 92164203), and Zhejiang Lab (2021MD0AB02). Niemier was supported in part by ASCENT, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. Imani was supported in part by National Science Foundation (NSF) #2127780 and Semiconductor Research Corporation (SRC) #2988.001.

## REFERENCES


[1] Z. Gong, P. Zhong, Y. Yu, and W. Hu, "Diversity-promoting deep structural metric learning for remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, vol. 56, no. 1, pp. 371–390, 2017.

[2] S. Salahuddin, K. Ni, and S. Datta, "The era of hyper-scaling in electronics," *Nature Electronics*, vol. 1, no. 8, pp. 442–450, 2018.

[3] X. S. Hu, M. Niemier, A. Kazemi, A. F. Laguna, K. Ni, R. Rajaei, M. M. Sharifi, and X. Yin, "In-memory computing with associative memories: A cross-layer perspective," in *2021 IEEE International Electron Devices Meeting (IEDM)*, pp. 25–2, IEEE, 2021.

[4] H. Amrouch, D. Gao, X. S. Hu, A. Kazemi, A. F. Laguna, K. Ni, M. Niemier, M. M. Sharifi, S. Thomann, X. Yin, *et al.*, "Iccad tutorial session paper ferroelectric fet technology and applications: From devices to systems," in *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1–8, IEEE, 2021.

[5] R. Karam, R. Puri, S. Ghosh, and S. Bhunia, "Emerging trends in design and applications of memory-based computing and content-addressable memories," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1311–1330, 2015.

[6] K. Ni, X. Yin, A. F. Laguna, S. Joshi, S. Dünkel, M. Trentzsch, J. Müller, S. Beyer, M. Niemier, X. S. Hu, *et al.*, "Ferroelectric ternary content-addressable memory for one-shot learning," *Nature Electronics*, vol. 2, no. 11, pp. 521–529, 2019.

[7] A. Kazemi, M. M. Sharifi, A. F. B. Laguna, F. Muller, X. Yin, T. Kampfe, M. Niemier, and X. S. Hu, "Fefet multi-bit content-addressable memories for in-memory nearest neighbor search," *IEEE Transactions on Computers (TC)*, 2021.

[8] A. Kazemi, M. M. Sharifi, A. F. Laguna, F. Müller, R. Rajaei, R. Olivo, T. Kämpfe, M. Niemier, and X. S. Hu, "In-memory nearest neighbor search with fefet multi-bit content-addressable memories," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1084–1089, IEEE, 2021.

[9] M. Imani, A. Rahimi, D. Kong, T. Rosing, and J. M. Rabaey, "Exploring hyperdimensional associative memory," in *IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 445–456, IEEE, 2017.

[10] G. Karunaratne, M. Schmuck, M. Le Gallo, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi, "Robust high-dimensional memory-augmented neural networks," *Nature communications*, vol. 12, no. 1, pp. 1–12, 2021.

[11] T. Böscke, J. Müller, D. Bräuhaus, U. Schröder, and U. Böttger, "Ferroelectricity in hafnium oxide: Cmos compatible ferroelectric field effect transistors," in *IEEE International Electron Devices Meeting (IEDM)*, 2011.

[12] T. Soliman, F. Müller, T. Kirchner, T. Hoffmann, H. Ganem, E. Karimov, T. Ali, M. Lederer, C. Sudarshan, T. Kämpfe, *et al.*, "Ultra-low power flexible precision fefet based analog in-memory computing," in *IEEE International Electron Devices Meeting (IEDM)*, 2020.

[13] X. Yin, F. Müller, Q. Huang, C. Li, M. Imani, Z. Yang, J. Cai, M. Lederer, R. Olivo, N. Laleni, *et al.*, "An ultra-compact single fefet binary and multi-bit associative search engine," *arXiv preprint arXiv:2203.07948*, 2022.

[14] D. Saito, T. Kobayashi, H. Koga, N. Ronchi, K. Banerjee, Y. Shuto, J. Okuno, K. Konishi, L. D. Piazza, A. Mallik, J. V. Houdt, M. Tsukamoto, K. Ohkuri, T. Umebayashi, and T. Ezaki, "Analog in-memory computing in fefet-based 1t1r array for edge ai applications," in *Symposium on VLSI Technology*, 2021.

[15] X. Yin, K. Ni, D. Reis, S. Datta, M. Niemier, and X. S. Hu, "An ultra-dense 2fefet tcam design based on a multi-domain fefet model," *IEEE Transactions on Circuits and Systems II: Express Briefs (TCAS-II)*, vol. 66, no. 9, pp. 1577–1581, 2018.

[16] T. Kohonen, *Associative memory: A system-theoretical approach*, vol. 17. Springer Science & Business Media, 2012.

[17] X. Yin, C. Li, Q. Huang, L. Zhang, M. Niemier, X. S. Hu, C. Zhuo, and K. Ni, "Fecam: A universal compact digital and analog content addressable memory using ferroelectric," *IEEE Transactions on Electron Devices (TED)*, vol. 67, no. 7, pp. 2785–2792, 2020.

[18] C. Li, F. Müller, T. Ali, R. Olivo, M. Imani, S. Deng, C. Zhuo, T. Kämpfe, X. Yin, and K. Ni, "A scalable design of multi-bit ferroelectric content addressable memory for data-centric computing," in *2020 IEEE International Electron Devices Meeting (IEDM)*, pp. 29–3, IEEE, 2020.

[19] R. Rajaei, M. M. Sharifi, A. Kazemi, M. Niemier, and X. S. Hu, "Compact single-phase-search multistate content-addressable memory design using one fefet/cell," *IEEE Transactions on Electron Devices (TED)*, vol. 68, no. 1, pp. 109–117, 2020.

[20] A. F. Laguna, H. Gamaarachchi, X. Yin, M. Niemier, S. Parameswaran, and X. S. Hu, "Seed-and-vote based in-memory accelerator for dna read mapping," in *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1–9, IEEE, 2020.

[21] B. A. Minch, "Translinear circuits," tech. rep., 2009.

[22] A. G. Andreou and K. A. Boahen, "Translinear circuits in subthreshold mos," *Analog Integrated Circuits and Signal Processing*, vol. 9, no. 2, pp. 141–166, 1996.

[23] J. A. Starzyk and X. Fang, "Cmos current mode winner-take-all circuit with both excitatory and inhibitory feedback," *ELECTRONICS LETTERS*, 1993.

[24] J. Lazzaro, S. Ryckebusch, M. A. Mahowald, and C. A. Mead, "Winner-take-all networks of o (n) complexity," *Advances in neural information processing systems (NIPS)*, vol. 1, 1988.

[25] J. Lazzaro, S. Ryckebush, M. Mahowald, and C. A. Mead, "Winner-take-all networks of o(n) complexity," tech. rep., Cal-tech, 1988.

[26] Nanoscale Integration and Modeling (NIMO) Group, "Predictive technology model." http://ptm.asu.edu/, 2007.

[27] K. Ni, M. Jerry, J. A. Smith, and S. Datta, "A circuit compatible accurate compact model for ferroelectric-fets," in *Symposium on VLSI Technology*, 2018.

[28] Y. Ni, Y. Kim, T. Rosing, and M. Imani, "Algorithm-hardware co-design for efficient brain-inspired hyperdimensional learning on edge," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 294–299, IEEE, 2022.

[29] G. Karunaratne, M. Le Gallo, G. Cherubini, L. Benini, A. Rahimi, and A. Sebastian, "In-memory hyperdimensional computing," *Nature Electronics*, vol. 3, no. 6, pp. 327–337, 2020.

[30] A. Kazemi, S. Sahay, A. Saxena, M. M. Sharifi, M. Niemier, and X. S. Hu, "A flash-based multi-bit content-addressable memory with euclidean squared distance," in *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 1–6, IEEE, 2021.

[31] P.-Y. Chen, X. Peng, and S. Yu, "Neurosim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures," in *2017 IEEE International Electron Devices Meeting (IEDM)*, pp. 6–1, IEEE, 2017.

[32] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp. 4705–4713, 2015.

[33] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning (ICML)*, pp. 173–182, PMLR, 2016.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems (NIPS)*, vol. 25, 2012.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 770–778, 2016.

[36] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive computation*, 2009.

[37] A. Hernandez-Cane, N. Matsumoto, E. Ping, and M. Imani, "Onlinehd: Robust, efficient, and single-pass online learning using hyperdimensional system," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021.

[38] M. Imani, J. Morris, J. Messerly, H. Shu, Y. Deng, and T. Rosing, "Bric: Locality-based encoding for energy-efficient brain-inspired hyperdimensional computing," in *Proceedings of the 56th Annual Design Automation Conference (DAC)*, pp. 1–6, 2019.

[39] M. Imani, Z. Zou, S. Bosch, S. A. Rao, S. Salamat, V. Kumar, Y. Kim, and T. Rosing, "Revisiting hyperdimensional learning for fpga and low-power architectures," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 221–234, IEEE, 2021.

[40] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living*, Springer, 2012.

[41] A. Angelova, Y. Abu-Mostafam, and P. Perona, "Pruning training sets for learning of object categories," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[42] "Uci machine learning repository." http://archive.ics.uci.edu/ml/datasets/ISOLET.