nature electronics

SUPPLEMENTARY INFORMATION

https://doi.org/10.1038/s41928-019-0321-3

In the format provided by the authors and unedited.

Ferroelectric ternary content-addressable memory for one-shot learning

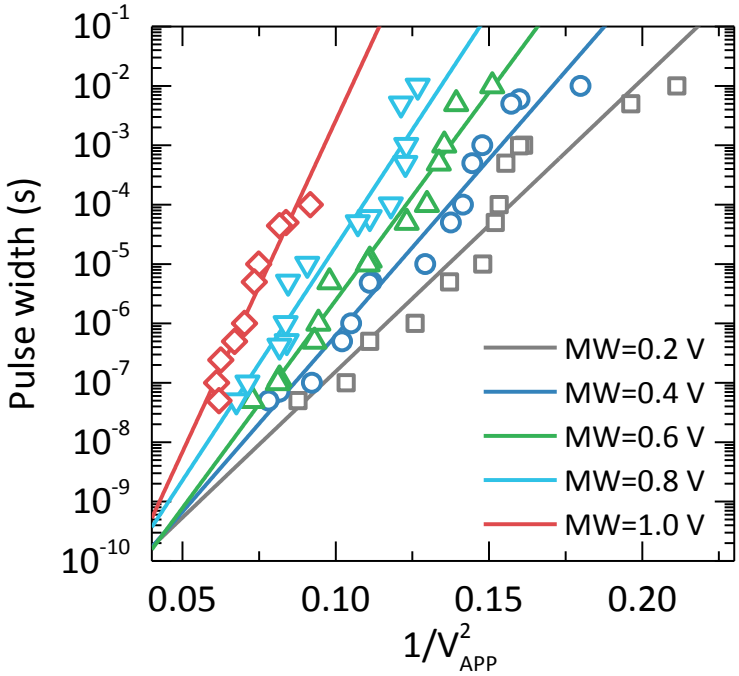
Kai Ni[1]*, Xunzhao Yin[1], Ann Franchesca Laguna[1], Siddharth Joshi[1], Stefan Dünkel[2], Martin Trentzsch[2], Johannes Müller[2], Sven Beyer[2], Michael Niemier[1], Xiaobo Sharon Hu[1] and Suman Datta[1]

[1]University of Notre Dame, Notre Dame, IN, USA. [2]GLOBALFOUNDRIES Fab1 LLC & Co. KG, Dresden, Germany. *e-mail: kni@nd.edu

Supplementary Information

Single FeFET Characteristics

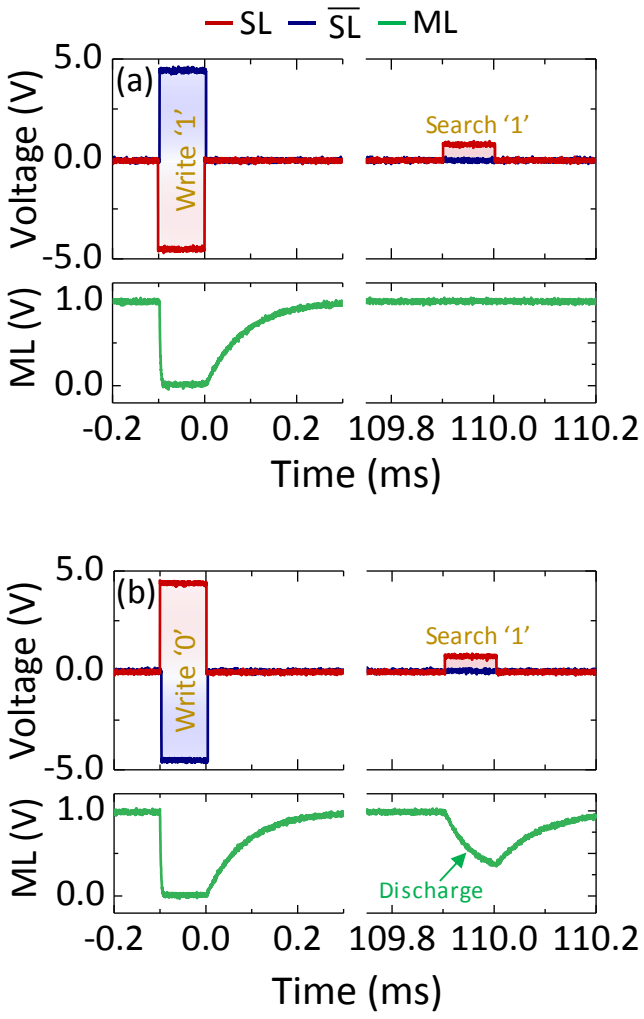
The polarization switching dynamics in 8nm HfO_2 FeFET is limited by ferroelectric reverse domain nucleation. In this model, the switching time follows an exponential dependence on the applied pulse amplitude[1] ($\tau \propto \exp((V_o/V_{APP})^2)$), where V_o is a fitting parameter and V_{APP} is the applied pulse ampliutde. When plotting the logarithm of the applied pulse width as a function of the $(1/V_{APP})^2$, we can obtain a linear relationship. Supplementary Fig.1 shows that the measured data can be nicely fitted by a linear straight line. Therefore, this confirms that the switching kinetics in HfO_2 FeFET follows the nucleation limited model.



Supplementary Fig.1. The relationship between the required pulse width and the inverse of the square of the applied pulse amplitude for a given FeFET memory window.

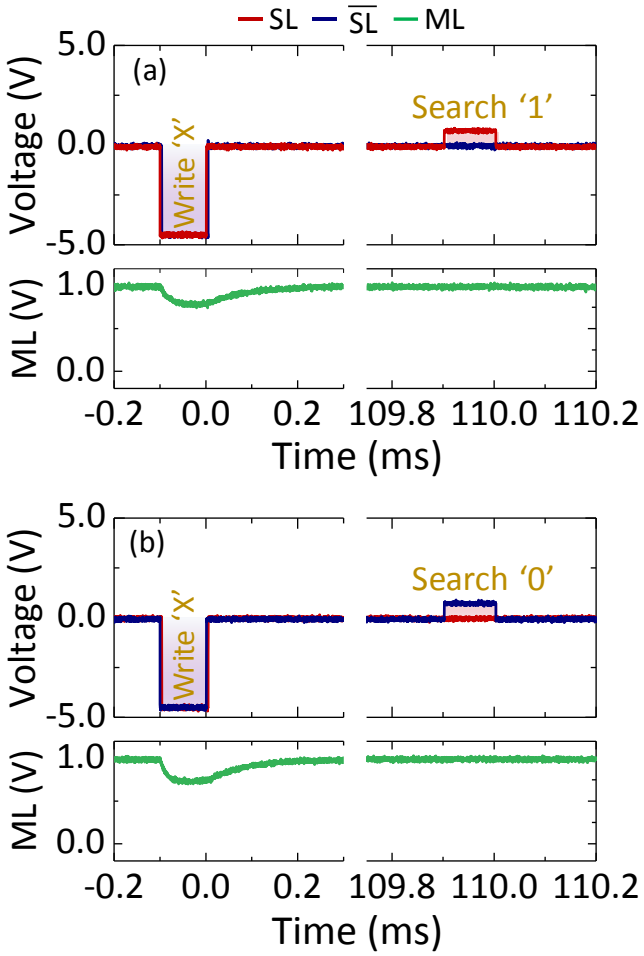
Single 2FeFET TCAM Cell Characteristics

The main text introduces the search bit '0' operation in a single 2FeFET TCAM cell. Here we further show the search bit '1' operation in a single TCAM cell, where SL is high and $\overline{SL}$ is low during search operation. Supplementary Fig.2a and Fig.2b show the cases where information bit '1' and bit '0' are stored and bit '1' is searched. When the search data matches the stored data, the match line remains high as the discharge current is small; when the mismatch occurs, the match line discharges as a high current flow through the FeFET in the low-V_{TH} state.



Supplementary Fig.2. Search logic '1' operation in a single 2FeFET TCAM cell.

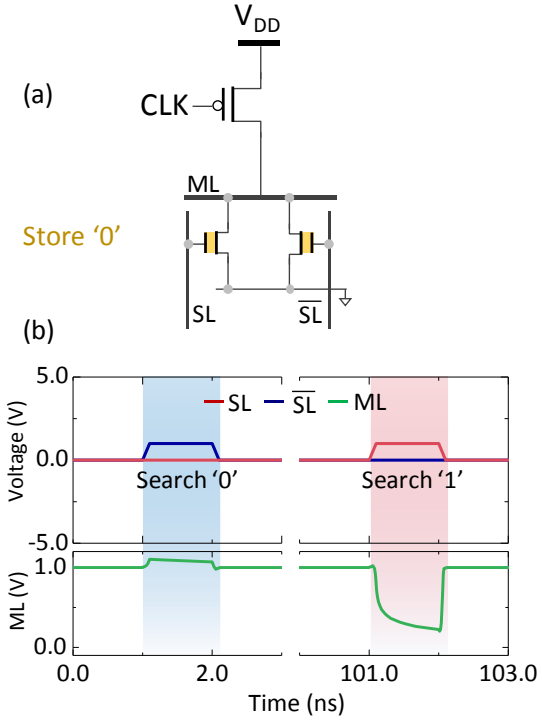
The third state in TCAM is the don't care state 'X', which is a wildcard for data search where the output always results in a match. The don't care condition can be implemented in the search pulse or the write pulse. In the former approach, search pulses with amplitudes of 0 V (or even negative voltage) are applied to both SL and $\overline{SL}$, and therefore the leakage current is minimal and the match line will remain high, indicating a match. The latter case is shown in Supplementary Fig.3, where both FeFETs in the TCAM cell are written to high-V_{TH} states, representing the don't care state 'X'. The waveforms show that during the search for a logic '1' (Supplementary Fig.3a) and a logic '0' (Supplementary Fig.3b), the match line remains high, indicating a match.



Supplementary Fig.3. The operations of a single TCAM cell with the ternary don't care state 'X'. (a) search bit '1' and (b) search bit '0' when the two FeFETs are written into high-V_{th} states.

Single 2FeFET TCAM Cell Simulation

We simulated a single 2FeFET TCAM cell operation in Spice[2]. The FeFET model used in the simulation is based on an experimentally calibrated multi-domain Preisach model[3]. In this model, the ferroelectric is assumed to be composed of independent domains whose polarization-voltage relationship for each domain is described by a single retangular hysteron. We assume a distribution of the coercive field pertaining to each hysteron. The overall response of the ferroelectric FET can be obtained by summing the contributions from all the individual domains by considering their coercive field distribution. The model adopts the nucleation limited domain switching dynamics model, which allows for an accurate description of the FeFET switching dynamics. We apply this model in the simulation of a single 2FeFET TCAM cell to demonstrate the search operation speed. For all the simulation results shown in the paper, parasitic capacitances on the metal lines (45nm node) are included.

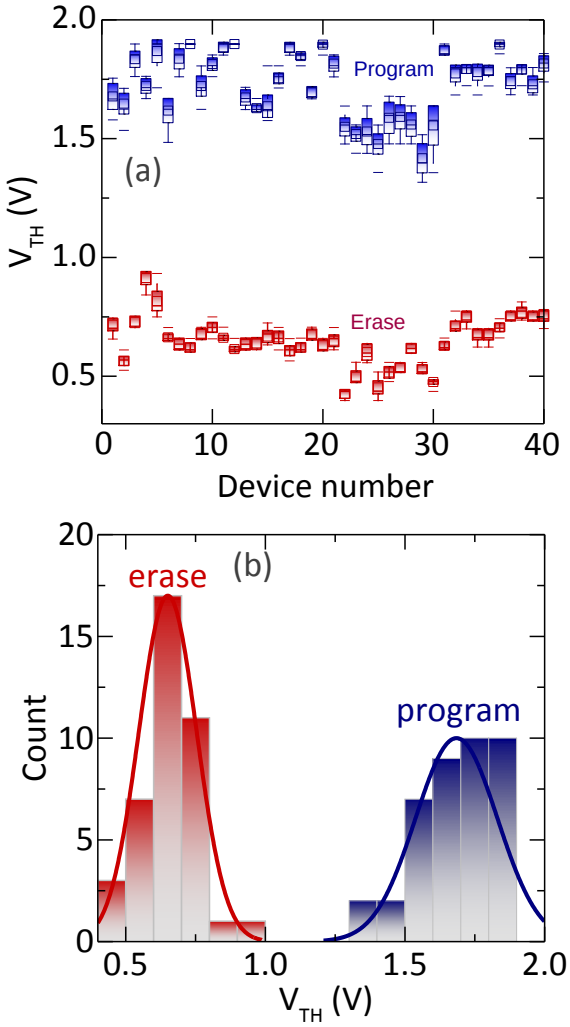


Supplementary Fig.4. Simulation of the search operation in a 2FeFET TCAM cell. (a) The simulated cell structure which contains a pull up (PMOS) device to pre-charge the ML. (b) The waveforms of search pulses and the ML voltage.

In the simulation, as shown in Supplementary Fig.4a, the single TCAM cell is composed of a pull up (PMOS) device, which pre-charges the ML. The TCAM cell is written to store a logic '0' state initially - i.e., the low-V_{TH} state for the left FeFET and the high-V_{TH} state for the right FeFET. Then the voltages corresponding to the search bits '0' and bit '1' are sequentially applied. The simulation results exhibit behavior identical to the experiments, as shown in Supplementary Fig.4b. The ML stays high when the search data matches the stored data and discharges otherwise. More importantly, it shows that the search operations are intrinsically fast and can be completed in less than 1 ns as it only involves a single transistor read operation.

FeFET Variation

It is important to examine the variation of FeFET in a TCAM array, including the cycle-to-cycle variation within a single device and device-to-device variation within an array. Supplementary Fig.5a shows such variation data for 40 FeFET devices. For each device, 50 program/erase cycles are performed to examine the cycle-to-cycle variation. In Supplementary Fig.5b, the device-to-device variation is shown for the first program and erase cycle.



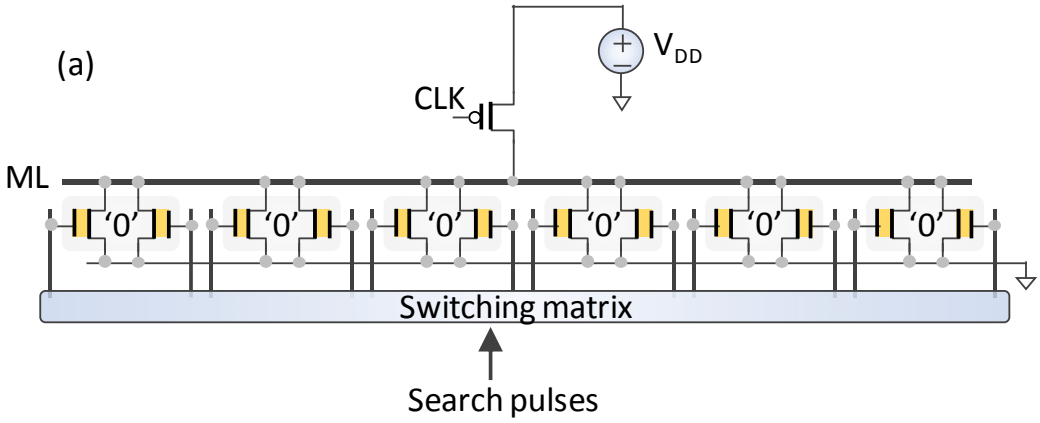
Supplementary Fig.5. Device-to-Device and cycle-to-cycle variation of FeFET devices. (a) Cycle-to-cycle variation for 40 devices. For each device, 50 program/erase cycles were measured.

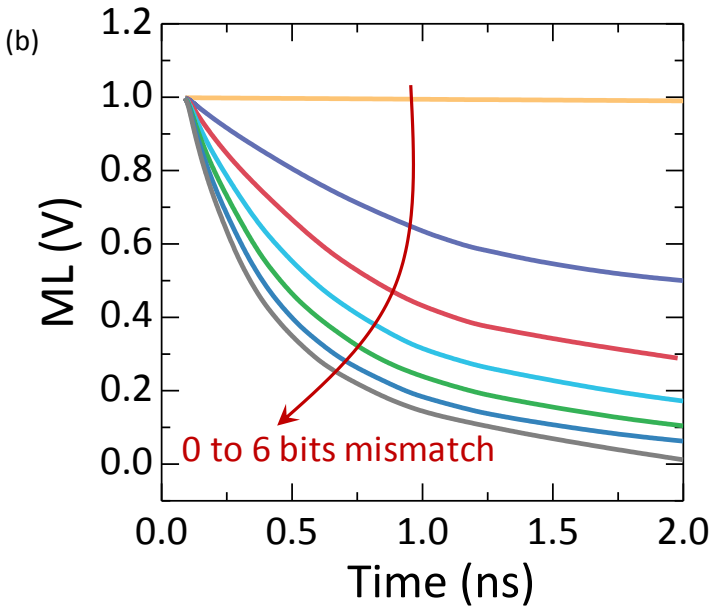
Erase pulse of 4 V, for 10 μs and a program pulse of -4V, for 10 μs are applied. (b) Device-to-device variation for the 40 devices at the first program/erase cycle. The box plot indicates one standard deviation of the cycle-to-cycle variation.

These data show that the distributions corresponding to the high-V_{TH} state and the low-V_{TH} state in the FeFETs are well separated, allowing reliable error-free sensing of the device state after considering the variation. This suggests that the polarization switching in FeFET is a well-controlled deterministic phenomenon. This makes FeFET a promising candidate for embedded nonvolatile memory applications.

Simulation of the Degree of Match Computation in a TCAM Array

We performed the simulation of the degree of match calculation in a FeFET TCAM array with integrated perheral sensing circuitry. To match the experiments, we simulated a 1x6 TCAM array, as shown in Supplementary Fig.6a. We replace the passive resistor used in the experiment with a pull up active transistor. For the simulation, all the TCAM cells are first programmed to store bit '0' initially. Then the match line is pre-charged by enabling the PMOS transistor and then kept floating. Search vectors, i.e., '000000', '000001', '000011', '000111', '001111', '011111', '111111', are then applied, similar to the experiment.



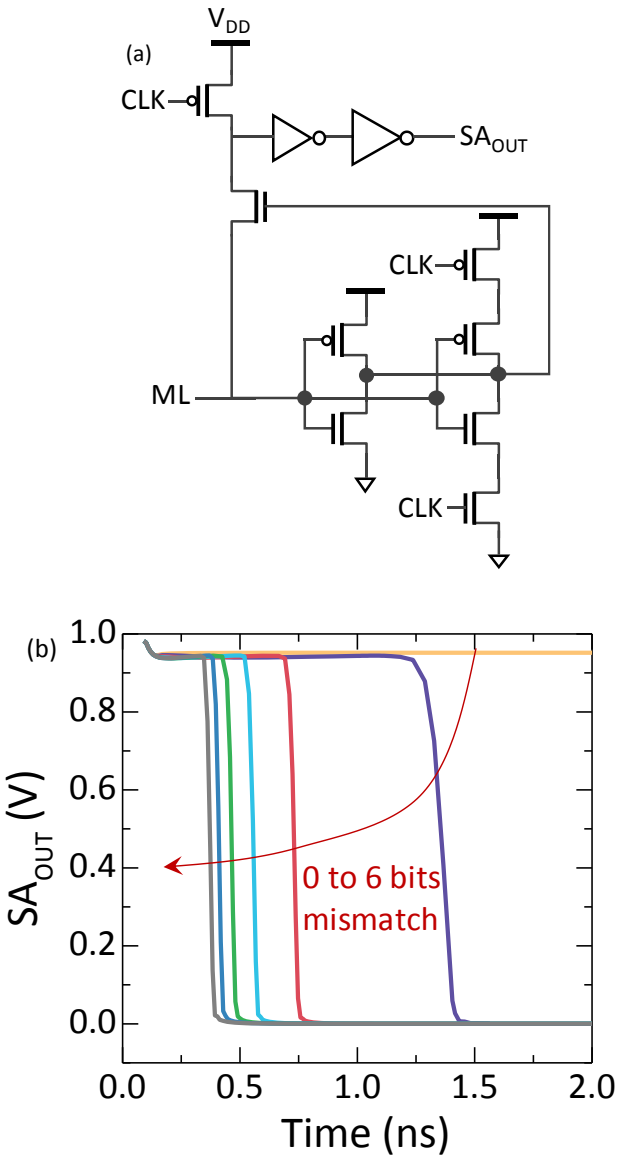


Supplementary Fig.6. Simulation-based studies for sensing the degree of match in a TCAM array. (a) The array prototype we simulated is composed of 1x6 TCAM cells. In the simulation, we replace the resistor in the experiment with a PMOS transistor. (b) Depending on the degree of match between the search vector and the stored vector, the match line discharge rate will be different.

Supplementary Fig.6b shows the match line waveform during the search operation as a function of search pattern. It suggests that, with the increase in the number of mismatched bits (number of bit '1' in the pattern in this case), the match line discharge rate increases. The simulation results are consistent with the measured data. Note that, the operation speed shown here is the transistor read speed, which is intrinsically fast, similar to what is shown in Supplementary Fig.4.

Simulation of Sense Amplifier to Detect the Degree of Match

To detect the small signal margin of the match line (ML) in Supplementary Fig.6, an additional step is necessary. A commonly used clocked self-referenced sense amplifier (CSRSA) for this application is shown in Supplementary Fig.7a[4,5]. It dynamically modulates threshold voltages of the ML during the pre-charge and evaluation stages, and transfers the small signal swing of the ML to a large sensing window in SA_{out}, resulting in improved output development time and better sensing and noise tolerance.

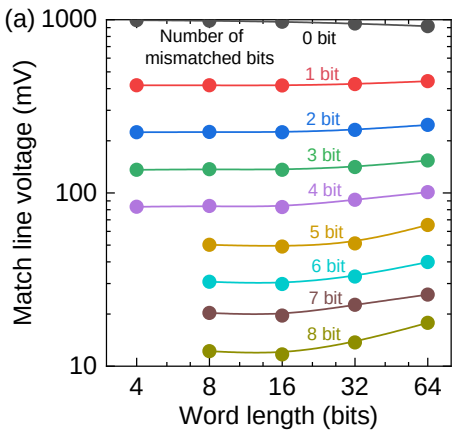


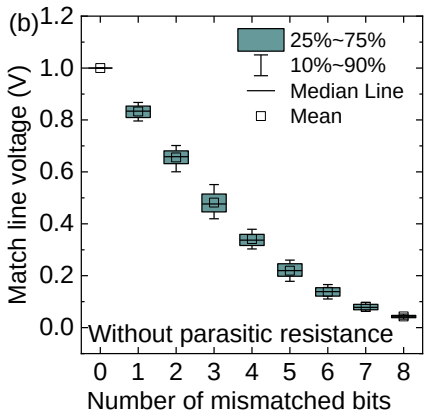
Supplementary Fig.7. Simulation of the sense amplifier for detection of the degree of match in the TCAM architecture: (a) circuit diagram of a clocked self-referenced sense amplifier (CSRSA)[4,5] ,and (b) its simulated waveforms showing the output of CSRSS with different degrees of match.

We simulated the operation of the sense amplifier by replacing the pull up device in Supplementary Fig.6a with CSRSA, and applying the same search vectors in Supplementary Fig.6. The sense amplifier output waveform is shown in Supplementary Fig.7b. It shows that the sense amplifier is able to translate the discharge rates of the ML with different degrees of match to the sensing window which can be easily detected and distinguished.

The Effect of Array Size, Variation, and Parasitics on Sensing Degree of Match

It is important to study Hamming distance calculation capability as a function of the TCAM array size. That said, an initial question to consider is what the maximum Hamming distance that needs to be detected must be. Is it necessary to detect all N different values in an 1xN array? For one-shot learning, we are only interested in the nearest neighbor. As LSH aims to bring similar entries closer and distinct entries farther apart, this means that we may not need to search for entries whose Hamming distance is too far away as they are likely irrelevant to the query. For example, in our initial studies with the Omniglot dataset presented in this paper, we limited the Hamming distance to be at most 16 bits. Thus, any memory entry whose LSH signature was more than a 16-bit Hamming distance away was considered be too far to be a nearest neighbor. (One would expect that for different datasets, different Hamming distance thresholds may be used, but initial results suggest that it is not necessary to detect all the number of mismatched bits in an 1xN array.)





Supplementary Fig.8. (a) Hamming distance calculation capability as a function of the TCAM array size (namely the word length). Hamming distances from 0 bits up to 8 bits are shown here as a representative study. (b)The effect of FeFET device-to-device variation on the sensing of the Hamming distance. The variation is extracted from the measured FeFET variation shown in Supplementary Fig.5. To generate the distribution, 50 different 1x8 TCAM arrays are simulated.

Detecting the number of mismatched bits was also considered from a hardware perspective via simulation-based case studies. Again, as a representative example, Supplementary Fig.8a shows the match line voltage as a function of the TCAM word size for different Hamming distances between the query and a stored vector. This study suggests that Hamming distance detection of up to 8 bits can be well supported in an 1x64 TCAM array.

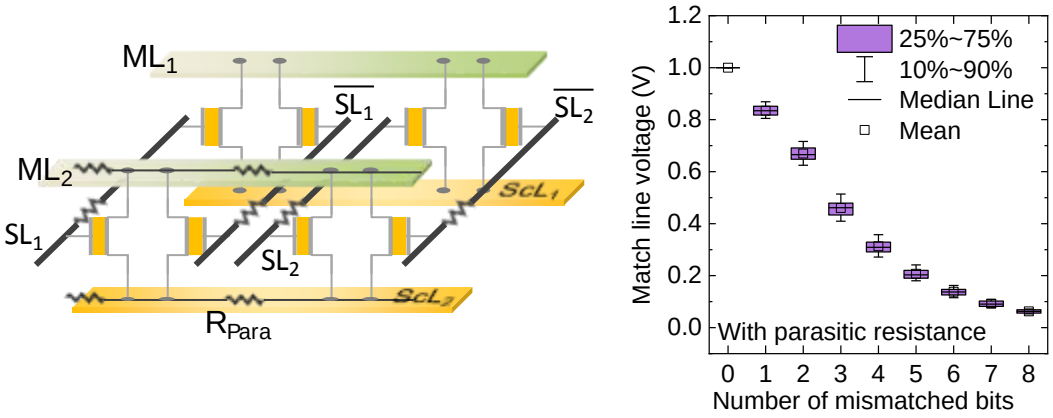
Besides word length, another important factor should be considered is device-to-device variation in an FeFET. We simulated the impact of FeFET device variations on Hamming distance sensing. The results are shown in Supplementary Fig.8b. We take the measured device-to-device variations in Supplementary Fig.5 and simulate the variation of the match line voltage for various numbers of mismatched bits. At this early stage of FeFET development, device variation is not optimized. However, we can reliably differentiate all the 8 levels in a

1x8 TCAM array. Extending beyond 1x8 array is challenging right now, but it is likely with future process optimization.

Furthermore, one significant advantage of a Hamming distance calculation is that it can be done in segment. As a simple example, if all 64 levels need to be differentiated in a 1x64 array, we can divide the entire TCAM array into 8 subarrays, where each subarray contains 1x8 TCAM cells. Each subarray can reliably detect Hamming distance.

Therefore, though the current FeFET technology is at its early development stage, the ferroelectric TCAM can well support the MANN application in one-shot learning. With future optimization of this technology, the variation can be further suppressed, which can extend the current Hamming distance detection capability to larger arrays.

The effect of parasitics on the operation of FeFET TCAM array is presented below. All the simulation results in this work have already included the parasitic capacitors on the metal lines. It does not affect the function of the TCAM array, therefore, we mainly focus on the effect of parasitic resistance. The line resistance on the search lines (SLs) is not a concern for a practical TCAM array. For a FeFET with 8nm HfO_2, the gate leakage current is on the order of pA at 1V. Therefore, the gate resistance is around 10^{12} Ω. Copper line resistance can be estimated based on 22nm technology node, where the copper line resistivity ρ is 3.5μΩ·cm[6]. Assume a line aspect ratio of 4, The line resistance between two consecutive rows (distance about 2F, F is the feature size) is 0.8 Ω, which is 12 orders of magnitude smaller than the gate resistance. For any practical array size (e.g., number of rows smaller than 10^6), the parasitic resistances on the SLs are not a concern.

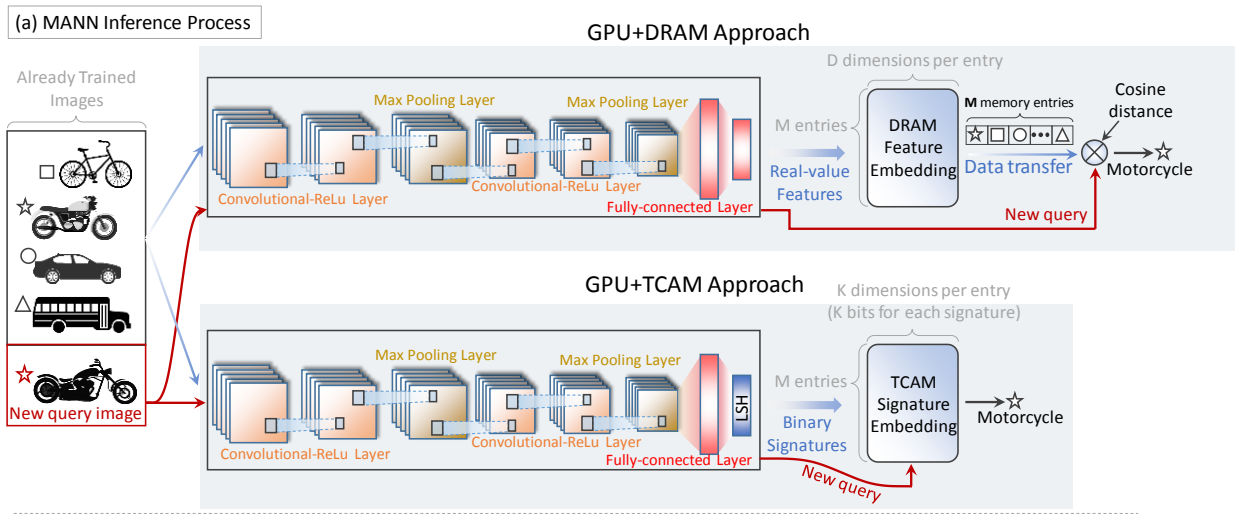


Supplementary Fig.9. (a) Schematics of 2x2 FeFET TCAM array. Also shown in the schematic are the parasitic resistances on the match lines (MLs), source lines (ScLs), and search lines (SLs). (b) Same simulation results as Supplementary Fig.8b but with the parasitic resistances included.

During the write operation, the match lines (MLs) and source lines (ScLs) are grounded. Therefore, the parasitic resistances on the MLs and ScLs are not a problem during the write operation. For the search operation, the effect of parasitic resistances is estimated as follow. The gate line distance between two FeFETs is 4F. This would lead to an 8F distance between two neighboring TCAM cells. Then the resistance of the metal line can be calculated to be 3.2 Ω. We include this resistance on the MLs and SLs between neighboring TCAM cells. The same simulation done in Supplementary Fig.8b is performed with parasitic resistance included and shown in Supplementary Fig.9b. This shows that the Hamming distance calculation function is not affected by this small parasitic resistance.

The Neural Network Implementation

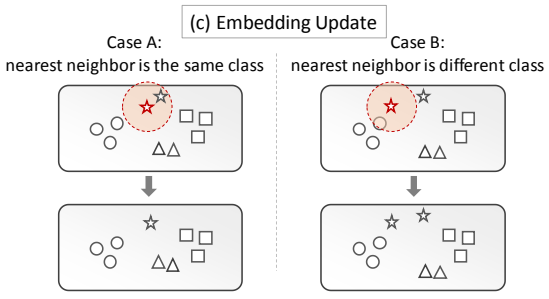
As shown in Supplementary Fig.10a, the conventional neural network to extract features from the input images in our studies is composed of two, 3x3 convolutional-ReLU layers with 64 filters, a max-pooling layer, another two, 3x3 convolutional-ReLU layers with 128 filters, and a max-pooling layer. In the conventional approach using a GPU backed by DRAM, two fully connected layers are also used, whose output is the feature vector that is stored in the memory as a floating point number, as shown in Supplementary Fig.10a. In the TCAM approach, we replace the last fully connected layer with a locality sensitive hashing (LSH) function, which translates the real-valued feature vectors into binary signature vectors that are stored in the TCAM memory. It is to be noted that the LSH does not add overhead to both the storage and computation.



(b) Computation for one memory search operation

	GPU+DRAM Approach	GPU+TCAM Approach
Computation	$\cos(\theta) = \frac{\sum_{i=1}^{D} f_{1i} \cdot f_{2i}}{\sqrt{\sum_{i=1}^{D} f_{1i}^2} \cdot \sqrt{\sum_{i=1}^{D} f_{2i}^2}}$	Hamming Distance
Data transfer	**M** entries	**0** entries
Multiply	**M * D**	**0**
Add	**M * (D - 1)**	**0**
Search	**0**	**1**

D dimensional feature vector: (f_{11}, f_{12}, ..., f_{1D})



Supplementary Fig.10. Inference mode operation of the memory augmented neural network for one/few-shot learning. (a) The system architecture and information flow during inference process for both GPU backed by conventional DRAM and GPU backed by TCAM. The feature extraction is performed with a convolutional neural network. For the GPU backed by TCAM, we replace the last fully connected layer with a LSH layer to translate the real valued feature vectors to binary signature vectors. (b) The total amount of computation for one memory search operation. For GPU backed by DRAM, it involves data transfer operation and vector multiply-accumulate operation; whereas the TCAM needs one memory search operation. (c) The rule for updating the feature sets in the memory. Two cases are shown, one is when the nearest neighbor has the same label as the query and the other one is when the two are different. In the former case, we replace the nearest neighbor with a normalized mean, while in the latter case we add the new query to the memory if it is not full or replace the least recently used term.

Given a random hyperplane, *r*, and a query vector, *q*, *q* is assigned to hash buckets depending on which side of hyperplane *r* vector *q* is located. The hash buckets are in one-hot binary form. The hashed values obtained from the hashing using different hyperplanes are concatenated to form the signature. For the given simulations, there are 4 hash buckets per random hyperplane *r*. There are 16 random hyperplanes used for the experiment. Comparing LSH signatures is accomplished using a Hamming Distance calculation. The greater the Hamming Distance between the two signatures, the farther the distance between the points represented by the signature.

For a single memory search operation, we need to calculate the distance between the query and all the stored memory entries. In the GPU backed by DRAM approach, that means that we need to transfer all the memory entries (**M** entries) to the compute units to calculate the cosine distance. The total computation involves a data transfer of M entries and **M * D** multiply operations as well as **M * (D-1)** add operations for cosine distance calculation (where D is the dimension of the feature vector), as shown in Supplementary Fig.10b. However, for the GPU backed by TCAM approach, we only need a single memory search, which significantly reduces both the energy and latency. Sometimes, there can be multiple matched memory entries returned with the same Hamming distance to the query. For all the matched entries, if they belong to the same class, it does not matter which one is chosen. The class label is simply returned. Otherwise, the L2 distance between the matches and the query are calculated to select the item with the minimum distance to the query.

The embedding in the memory is updated based on the following rules, as shown in Supplementary Fig.10c. If the nearest neighbor entry has the same label as the support image to be written, the nearest neighbor entry will be updated based on the new information. For example, in Case A in Supplementary Fig.10c, the query and its nearest neighbor have the same class (Δ). The nearest neighbor is then updated by computing the normalized mean of the query and the nearest neighbor feature vector. If the nearest neighbor entry has a different label from the support image to be written, the least recently used (LRU) memory entry is then replaced. In Case B in Supplementary Fig.10c, the nearest neighbor (o) and the query (Δ) have different classes, if the memory is not full, the query is added to the memory, otherwise the LRU memory entry is replaced.

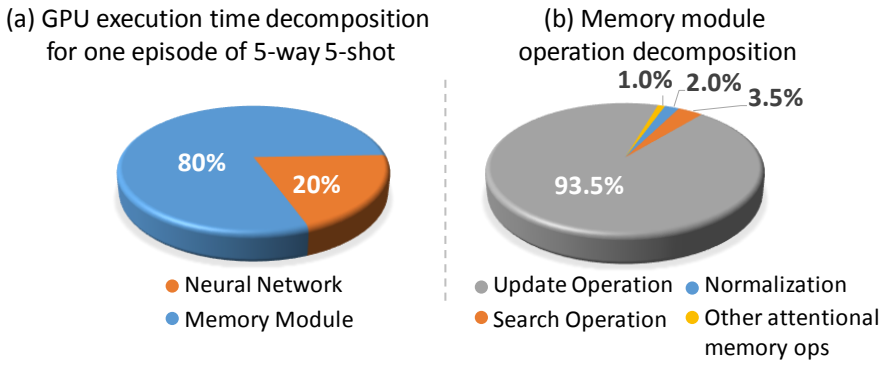
For training, a batch of query images are used to compute for the triplet loss function that will be used to compute the gradients to adjust the weights of the neural network during backpropagation. The goal of the triplet loss function is to move the entries with the same label closer to each other and those with different labels away from each other. The triplet loss is based on the squared distance of the query from the nearest entry with the same label (n_+) and the nearest entry with a different label (n_-) and is obtained by the following equation:

$$Loss = (q - A[n_+])^2 - (q - A[n_-])^2 + \alpha$$

Where A[n+]/A[n-] is the signature for the n+/n- entry, respectively. For inference, the hash signature of the query image is compared with all the memory entries, the label of the nearest neighbor is then returned as the label of the query image.

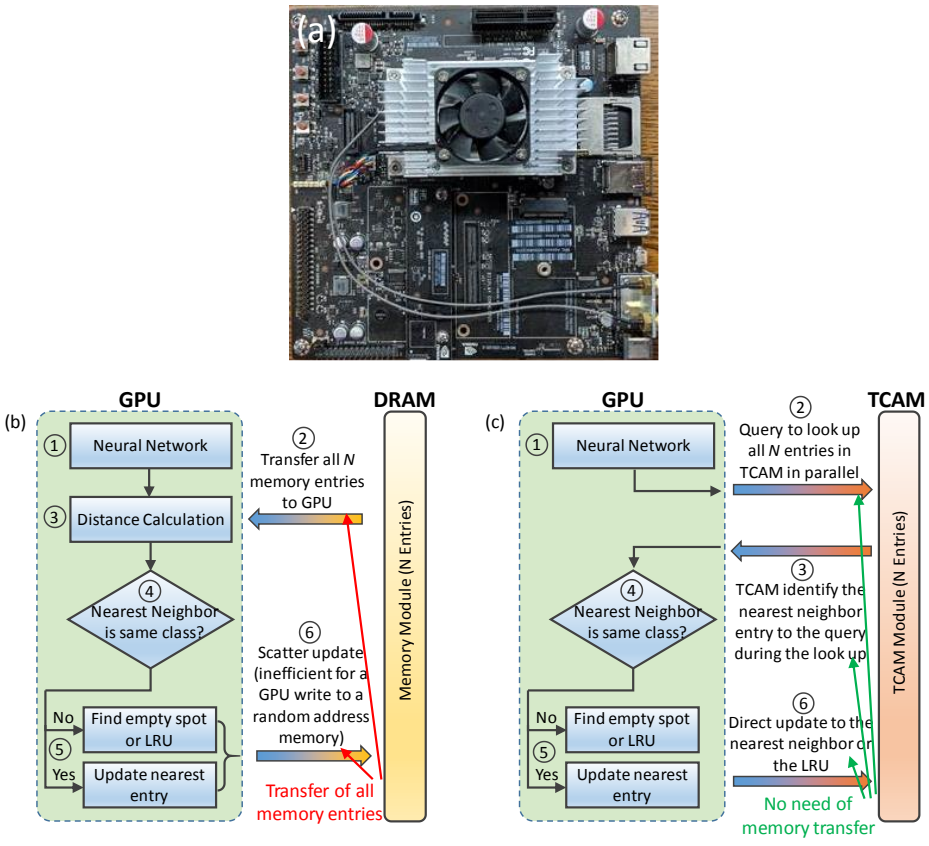
Benchmarking the Performance of MANN for One and Few-Shot Learning

As shown in Supplementary Fig.11, for the one and few shot learning tasks performed in this work, for the GPU backed by DRAM implementation, 80% of the time is spent on performing operations related to the memory module during inference. It is expected that this time will increase for more complex MANN tasks, relative to the one/few shot learning in this work (i.e., the Omniglot dataset). Therefore, accelerating the operation in the memory module with ferroelectric TCAMs will provide even greater improvements in overall system performance when compared to this initial case study. Supplementary Fig.12 shows the hardware platform to implement MANN systems and the system flow diagrams for one-shot learning inference applications for both GPU backed by DRAM and GPU backed by TCAM approaches. It shows significant reduction of the memory transfer by replacing the DRAM memory with TCAM attentional memory.

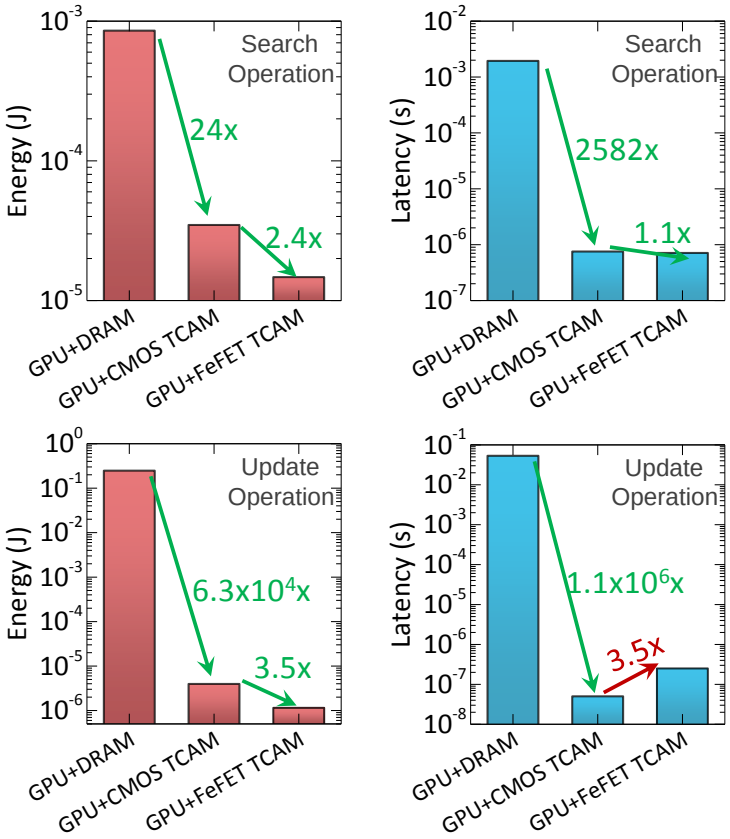


Supplementary Fig.11. Decomposition of execution time during the inference process. (a) The total execution time includes time spent in the neural network feed forward pass and in the memory module for one episode in 5-way 5-shot learning in a GPU backed by a conventional DRAM memory module. (b) Decomposition of the execution time spent in the memory module, including the update operation, and search operation.

Within the 80% of the time spent related with the memory module, 93.5% of the time is spent performing memory update operations. This is because the memory update operation is inefficient in a GPU, as shown in Supplementary Fig.12b. After the memory search operation, the memory index that must be updated is identified (Supplementary Fig.10c), (either the nearest neighbor or the LRU entry). Given the index, the GPU is extremely inefficient in performing a scatter update (i.e., an operation like A[index]=B, where A is the memory array and B is the new value written to the memory entry). Writing to a random address entry cannot be effectively parallelized in a GPU. However, with the TCAM approach, the memory entry can be directly written after the entry is identified through the memory search operation, due to its content-based addressing feature, as shown in Supplementary Fig.12c.



Supplementary Fig.12. (a) The Hardware setup of a MANN system implemented on a GPU backed by DRAM platform. (b) and (c) Flow diagram of a MANN system implemented on a GPU backed by DRAM and a GPU backed by TCAM platform, respectively, during the inference phase of one-shot learning.

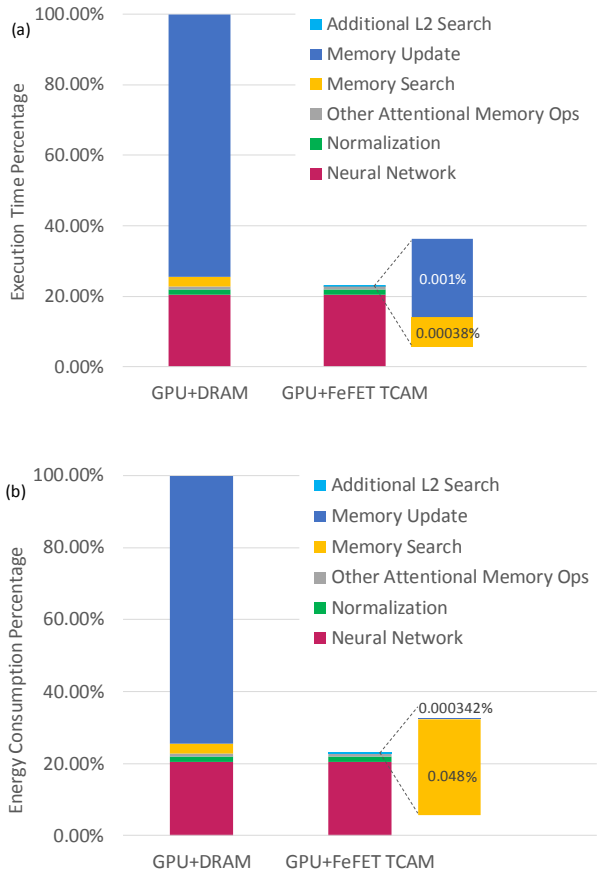


Supplementary Fig.13. Energy and latency benchmarking of memory search and memory update operations for one episode in 5-way 5-shot learning for three different architectures: GPU+DRAM, GPU+CMOS TCAM, and GPU+FeFET TCAM.

For the **search** operation, when a 16T CMOS TCAM array is employed, we can observe 24x and 2582x reductions in energy and latency respectively compared with a GPU backed by DRAM. We further observe a 2.4x and 1.1x reduction in energy and latency respectively when replacing the 16T CMOS TCAM with 2FeFET TCAMs.

For the **update** operation, we observe significant reductions in energy and latency by using CMOS TCAMs when compared with a GPU backed by DRAM. This is because of the extremely inefficient memory update operation in a GPU, as explained in Supplementary Fig.11 and Supplementary Fig.12. When replacing the CMOS TCAM with a 2FeFET TCAM, we observe an additional 3.5x reduction in energy, but also incur a 3.5x penalty in the latency. This is because the FeFET write is electric field driven, which is extremely energy-efficient, but slower than the SRAM write in CMOS TCAM. Overall, FeFET TCAM has the same energy-latency product for the update operation compared with CMOS TCAM.

Based on the improvement achieved with the 2FeFET TCAM for both the memory update and search operations, the overall system performance improvement can be calculated using Amdahl's law, with the decomposition shown in Supplementary Fig.11 (80% of time spent in memory module). Supplementary Fig.14 shows the execution time and energy consumption decomposition of a single MANN test episode in 5-way 5-shot learning for both a GPU backed by DRAM and a GPU backed by FeFET TCAM platforms. This suggests savings of ~4.5x in both energy *and* latency with the 2FeFET TCAM approach compared with the GPU+DRAM approach. Note that, this improvement can be increased for more complex MANN tasks which will require more frequent memory operations.



Supplementary Fig.14. (a) Execution time and (b) energy consumption breakdown for various operations in one episode of 5-way 5-shot learning.

To compute the GPU latency and energy consumption, an NVIDIA TX2 is used to run the individual operations. The energy is obtained externally using NI USB-6216 BNC[7,8]. The simulation is run for 1000 and 2000 iterations. The result of the 1000 iteration run is subtracted from the result of the 2000 iteration run to remove the initialization overhead. The result is then divided by 1000. This is repeated 10 times to provide a better idea of the average hardware performance across multiple runs.

Reference

1. H. Mulaosmanovic et al. Switching kinetics in nanoscale hafnium oxide based ferroelectric field-effect transistors. ACS Appl. Mater. Interfaces 9, 3792−3798 (2017)
2. Cadence Virtuoso Spectre Circuit Simulator (2011)
3. K. Ni et al. A circuit compatible accurate compact model for ferroelectric FETs. *IEEE Symposia on VLSI Technology & Circuits*, 131–132 (2018)
4. I. Arsovski, R. Wistort. Self-referenced sense amplifier for a cross-chip-variation immune sensing in high-performance content-addressable memories. IEEE Custom Integrated Circuits Conference, 453-456 (2006)
5. 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, 896-907 (2014)
6. A. Chen, A comprehensive crossbar array model with solutions for line resistance and nonlinear device characteristics. IEEE Trans. Electron Dev. 60, 1318-1326 (2013).
7. FLUKE, "80i-110s AC/DC current clamp," http://www.fluke.com/fluke/iden/accessories/current-clamps/80i-110s.htm?pid=55352, 2018, [Online; accessed October-2018].
8. National Instruments, "NI USB-6216 BNC," http://sine.ni.com/nips/cds/view/p/lang/en/nid/207100, 2018, [Online; accessed October-2018]