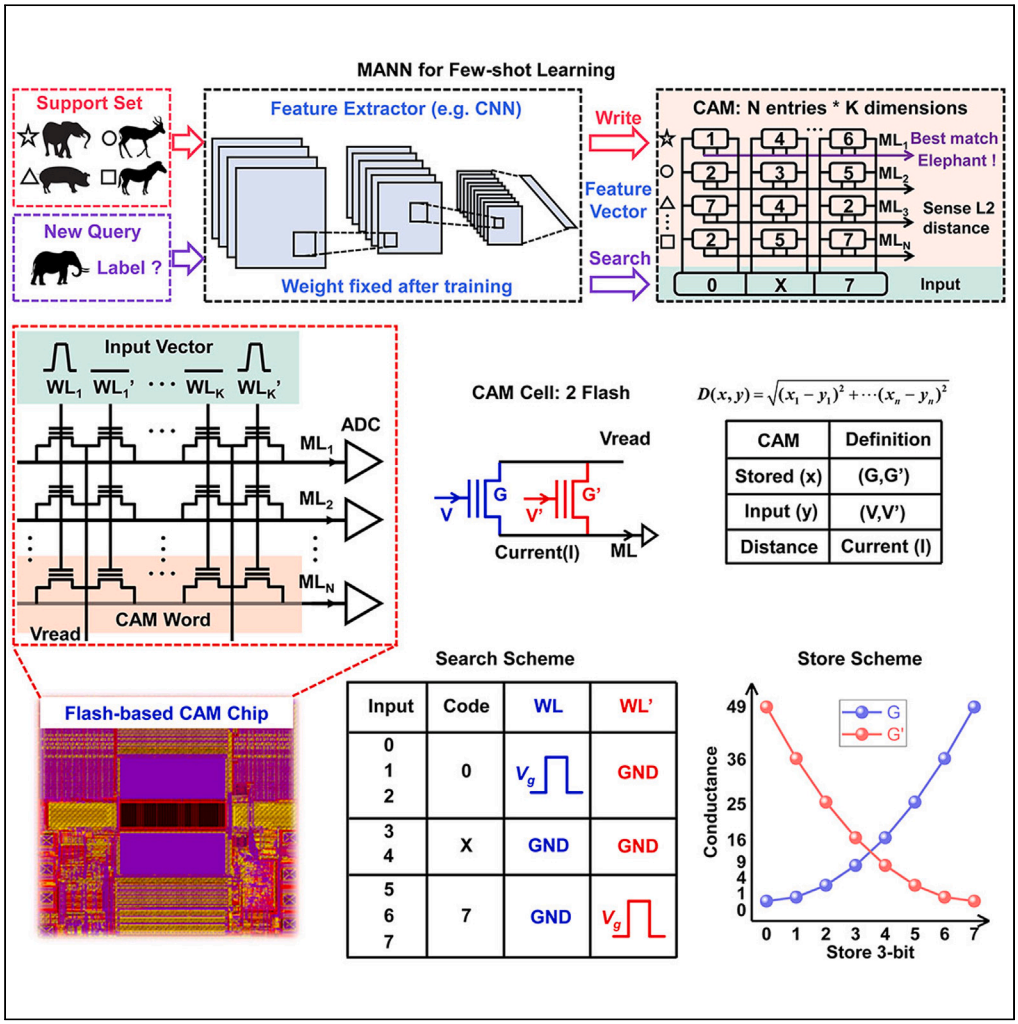
iScience

CellPress
OPEN ACCESS

Article

Flash-based content addressable memory with L2 distance for memory-augmented neural network



Haozhang Yang, Peng Huang, Ruiyi Li, ..., Lifeng Liu, Xiaoyan Liu, Jinfeng Kang

phwang@pku.edu.cn

Highlights

Program the conductance values of devices to quadratic relation for L2 distance

1Mb Flash-based multi-bit CAM chip demonstration

Excellent robustness against environmental disturbance

Significantly reduced search latency and energy compared with GPU

Yang et al., iScience *26*, 108371
December 15, 2023
https://doi.org/10.1016/j.isci.2023.108371

iScience

CellPress
OPEN ACCESS

Article

Flash-based content addressable memory with L2 distance for memory-augmented neural network

Haozhang Yang,[1,2] Peng Huang,[1,2,3,*] Ruiyi Li,[1,2] Nan Tang,[1,2] Yizhou Zhang,[1,2] Zheng Zhou,[1,2] Lifeng Liu,[1,2] Xiaoyan Liu,[1,2] and Jinfeng Kang[1,2]

SUMMARY

Memory-augmented neural network (MANN) has received increasing attention as a promising approach to achieve lifelong on-device learning, of which implementation of the explicit memory is vital. Content addressable memory (CAM) has been designed to accelerate the explicit memory by harnessing the in-memory-computing capability. In this work, a CAM cell with quadratic code is proposed, and a 1Mb Flash-based multi-bit CAM chip capable of computing Euclidean (L2) distance is fabricated. Compared with ternary CAM, the latency and energy are significantly reduced by 5.3- and 46.6-fold, respectively, for the MANN on Omniglot dataset. Besides, the recognition accuracy has slight degradation (<1%) even after baking for 10^5 s at 200°C, demonstrating the robustness to environmental disturbance. Performance evaluation indicates a reduction of 471-fold in latency and 1267-fold in energy compared with GPU for search operation. The proposed robust and energy-efficient CAM provides a promising solution to implement lifelong on-device machine intelligence.

INTRODUCTION

Lifelong learning is a key issue for intelligent edge devices, in which on-the-fly learning with only a few examples is required.[1–3] Deep neural networks (DNNs) have achieved impressive results on many data-intensive tasks, such as image classification[4] and speech recognition.[5] But it struggles to implement flexible on-device learning with very few data because of the catastrophic interference in straightforward gradient-based solutions.[6] As we all know, this kind of rapid adaptation is a celebrated aspect of biological brains, which could learn novel behavior based on a few scraps of input information together with past experience.[7] Inspired by the biological brain, recent works have proposed neural networks with an explicit memory, i.e., memory-augmented neural network (MANN), which consists of a feature extractor (e.g., convolutional neural network, CNN) and an explicit memory to store and retrieve features.[8–10] The access to explicit memory is a content-based attention mechanism, which needs to compute the distance or similarity between the input feature and all the stored features. Lifelong on-device learning occurs by storing never-before-seen extracted features into the explicit memory, and inference occurs by distance-based retrieval of the stored features. Accelerating the explicit memory is of vital importance to MANN because operations related to this consume 80% of total execution time.[11,12]

To improve the performance of MANN, one promising alternative is to realize the explicit memory with content addressable memory (CAM), which can compare the input search vector with all the stored vectors in parallel and calculate pairwise distances *in situ*, harnessing the in-memory-computing capability.[13–18] Past improvements have been realized by employing emerging non-volatile memories (NVMs), including resistive random-access memory (RRAM),[13–15] spintronic device (MTJ),[16] and ferroelectric field-effect transistor (FeFET)[17,18] to constitute a ternary CAM (TCAM) and compute Hamming or Manhattan (L1) distances in MANN. We know that the memory states in the biological brain are high-precision or even analog, and the processing function is much more complex than linear. Therefore, developing multi-bit CAM design with a more powerful distance function is an essential path to imitate the sophisticated capabilities of our brain and move toward advanced lifelong machine intelligence.[19–22]

Among the distance functions used in machine learning algorithms, Euclidean (L2) distance is very powerful and excels in classification, clustering, and retrieval tasks.[23] The implementation of L2 distance for multi-bit CAM demands two physical variables following the quadratic relation, which is very difficult for the NVM devices. The I_d-V_g function in saturation region of transistors is the closest to the quadratic relation of L2 distance. This kind of L2 distance function has been proposed[24] and implemented utilizing FeFET.[25] However, with the technology scaling down, it will deviate further from quadratic due to mobility degradation and velocity saturation of short-channel effects.[26,27]

[1]School of Integrated Circuits, Peking University, Beijing 100871, China
[2]Beijing Advanced Innovation Center for Integrated Circuits, Beijing 100871, China
[3]Lead contact
*Correspondence: phwang@pku.edu.cn
https://doi.org/10.1016/j.isci.2023.108371

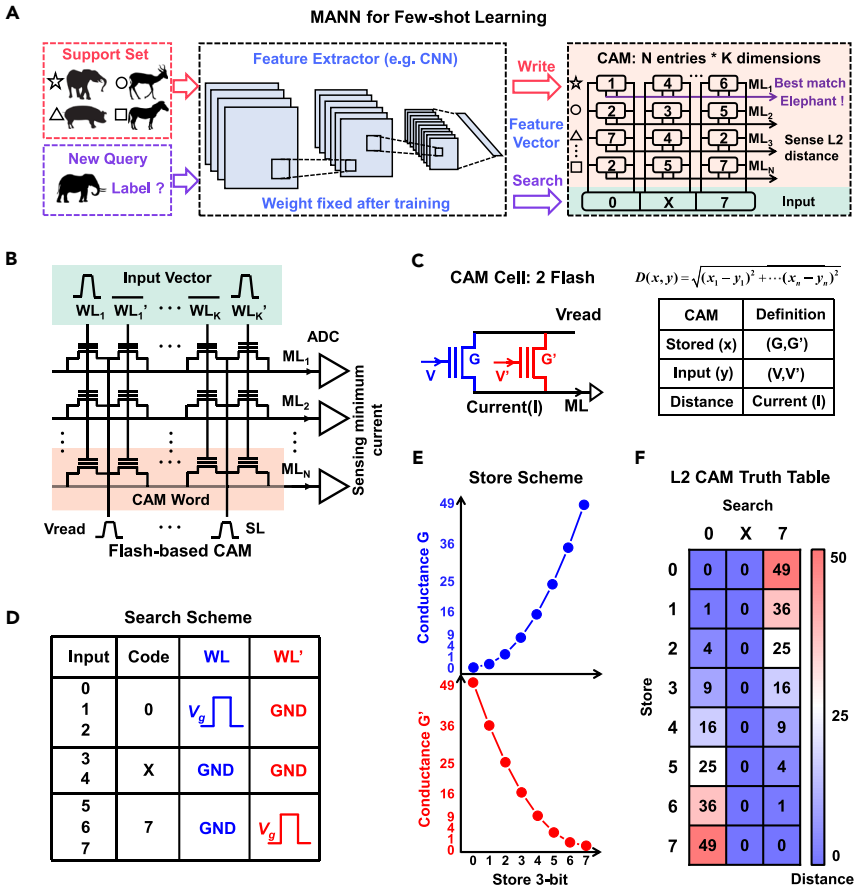


Figure 1. Flash-based multi-bit-storage and ternary-search CAM design with L2 distance

(A) The major components and working process of MANN for few-shot learning. The feature extractor is offline trained and then fixed. Classes of support set can be learned and embedded into CAM for predicting query.

(B) The schematic of proposed Flash-based CAM array. The Flash transistors in a row constitute a CAM word, and the search vector is applied to WL. Search result is sensed from the current of ML, where a higher current represents a larger distance.

(C) CAM cell structure, of which one CAM cell consists of two Flash transistors. Stored content (x) is represented as conductance pair G and G′ and search input (y) as the gate voltages applied to WL and WL'.

(D) Search scheme. The original multi-level input is coded as ternary with two input voltages.

(E) Store scheme. The G values are selected according to the quadratic relation.

(F) L2 CAM truth table.

To address this, a novel CAM cell with quadratic coding scheme is proposed, which directly programs the conductance values of devices to quadratic relation for L2 distance, and demonstrated with Flash memory. A 1Mb Flash-based multi-bit-storage and ternary-search CAM chip capable of computing L2 distance was fabricated. Compared with ternary CAM with Hamming distance, the latency and energy are significantly reduced, and the recognition accuracy of MANN with fabricated CAM on Omniglot dataset is largely increased for different few-shot learning tasks. Besides, we adopted baking for 10^5 s at 200°C to investigate the immunity of Flash-based CAM to environmental disturbance, which is essential to lifelong learning. The results show that the accuracy decrease of our fabricated CAM is slight (<1%). Compared with GPU for search operation, the fabricated CAM achieves more than two orders of magnitude reduction in both search latency and energy. The robust and energy-efficient CAM provides a promising solution to implement lifelong on-device machine intelligence.

RESULTS

Flash-based CAM design with L2 distance

Figure 1A shows the major components and working process of MANN. The MANN commonly includes a CNN or RNN as feature extractor and an explicit memory for distance-search, which is implemented by CAM for acceleration.[28] Firstly, the feature extractor is meta-trained offline and deployed to the edge devices. The weight parameters are fixed and will no longer change in the following phases. Next, for the few-shot learning phase, several never-before-seen classes, with one or few examples for each class, are employed as support set and sent to the feature extractor. The extracted features are written into the CAM, realizing on-device adaptation of MANN. For the inference

phase, the feature extractor takes input new query and generates features for distance-search, which is performed in CAM in parallel. For the CAM word, a mismatch case will result in the increase of current in match lines (MLs), and the current will be larger for the larger degree of mismatch (distance). Therefore, by sensing the sum currents of each ML to obtain the minimum, we can identify the closest class as the predicted label for input query.

To imitate the high-precision property and sophisticated processing capability of our brain, we propose a Flash-based CAM design that is multi-bit-storage, ternary-search, and capable of computing L2 distance (1), as illustrated in Figure 1B.

$$D(x, y) = \sqrt{(x_1 - y_1)^2 + \cdots (x_n - y_n)^2} \quad \text{(Equation 1)}$$

The Flash transistors in a row constitute a CAM word, which represents the previously learned knowledge and corresponds to the x in Equation 1. The search vectors will be coded and transformed into a series of word line (WL) voltages, which represent the encountering problems and corresponds to the y in Equation 1. By applying a constant read voltage (V_{read}) to SLs, we can sense the sum currents of each ML in parallel to obtain the L2 distance $D(x, y)$ and identify the closest class as the predicted label for input query. It should be noted that the elimination of the square root operation in L2 distance will not decrease any accuracy in the application of MANN, which just requires comparing distance. The structure of CAM cell is shown in Figure 1C. One CAM cell consists of two Flash transistors, of which stored content (x) is represented as conductance pair G and G′ and search input (y) as the gate voltages applied to WL and WL′. The values of conductance pair are selected according to quadratic relation in L2 distance. Search result (D) is sensed from the current of ML, where a higher current represents a larger distance.

Take 3-bit as an example to illustrate the computing principle of the proposed CAM in detail. As shown in Figure 1D, for the eight states of 3-bit input, the original "0, 1, 2" are all coded to "0" for CAM input, "5, 6, 7" coded to "7", and "3, 4" coded to the wildcard "X". For the "0" value, WL is applied with V_g and WL′ is connected to ground. For the "7" value, WL is connected to ground and WL′ is applied with V_g. For the wildcard "X", both of the WLs are connected to ground. The 3-bit store scheme is shown in Figure 1E. The values of conductance pair are selected according to quadratic relation with the 8 states. For storing "0", G is set as 0, i.e., the off-state of Flash transistor, and G' is set as 49, which means a fixed conductance step multiplied by 49. The step can be selected according to the conductance range of realistic devices. With the store state increasing, the value of G increases, and G' decreases by quadratic relation. Under this setting, the output current will represent the L2 distance between the search input and stored content. The truth table is shown in Figure 1F.

Experimental demonstration of the proposed CAM

To validate the proposed CAM design, we fabricated a CAM chip with 65-nm NOR Flash technology. The test platform for the following study and the layout of 1Mb Flash-based CAM are shown in Figure 2A (See STAR Methods). In Flash transistor, the threshold voltage (V_{th}) is modulated by the number of electrons in the floating gate, which are injected from channel or swept out by tunneling.[29] According to the need for on-the-fly learning and the design of L2 CAM, it is critical to program the Flash transistors to the preset states accurately and efficiently. The program linearity will deteriorate as the V_{th} increases under constant bias voltages. To address this and improve program efficiency, we propose a two-stage program scheme, and by selecting the step size carefully, an excellent linearity can be achieved (See details in Figure S1). What's more, parallel program is another effective approach to speed up on-the-fly learning. Therefore, we adopt a parallel scheme that is along with different BLs, as shown in Figure S2. By jointly employing the two schemes, we can get a 19.6-fold reduction in total program cycles, which provides the basis for rapid and precise on-device learning (See Figure S3), and the stability analysis of the proposed programming method is shown in Figure S4.

Then we experimentally implement the proposed 3-bit-storage and ternary-search CAM design with L2 distance in our chip for proof-of-concept. First, eight different states of the 3-bit scheme are encoded into corresponding conductance pairs and programmed to 400 CAM cells, each state with 50 cells. For each stored state, all three kinds of code ("0", "X", and "7") are input to the CAM cell for distance calculation. The measured currents on ML of a single CAM cell are shown in Figure 2B. According to the aforementioned coding scheme, there is at most one transistor in the CAM cell that will turn on, and the measured currents represent the square of the absolute difference between search and store, which exhibits a good consistency with the numerical L2 distance.

After verifying the functionality of CAM cell, we further investigate the performance of CAM array. Considering the neural network structure employed in the following study, here we set the dimension of CAM word as 64, which is the total number of CAM cells per ML. To investigate the most difficult case to sense, we define 1-bit mismatch is that there is only one CAM cell mismatch in the CAM word, and the absolute difference between input and storage is "1" for this mismatch cell, such as search input is "0" when the stored content is "1". The define is similar to Liu et al.[30] These cases correspond to the minimum currents in mismatch cases. The possibility of a CAM cell outputting an L2 distance larger than one is originated from the setting of unit one in multi-bit scheme. For TCAM, the match case is distance "0", and mismatch case is distance "1". While for multi-bit CAM (e.g., 3-bit), we assume the state step as unit "1" in search vector and stored content, and the full range is from "0" to "7". The L2 distance of CAM cell is the square of difference value between search and storage, and therefore may be larger than "1". Precisely, "1" is the minimum L2 distance except the match case in our method. The sensing margin between fully match and 1-bit mismatch in the 64-dimension word is more than 400nA, as shown in Figure 2C, which is friendly to the analog to digital converter (ADC) circuit design.[31] The analysis of distance distribution of CAM cell in few-shot learning tasks is shown in Figure S4. Moreover, the different degree of mismatch shows a good linear relationship with the ML current, as shown in Figure 2D, of which the number of mismatch bits represents the number of CAM cells with distance "1". Therefore, the degree of mismatch can be reliably obtained and distinguished in the CAM array by simply comparing the output currents. This benefits from the tight current distribution and sufficient ON/OFF ratio of the

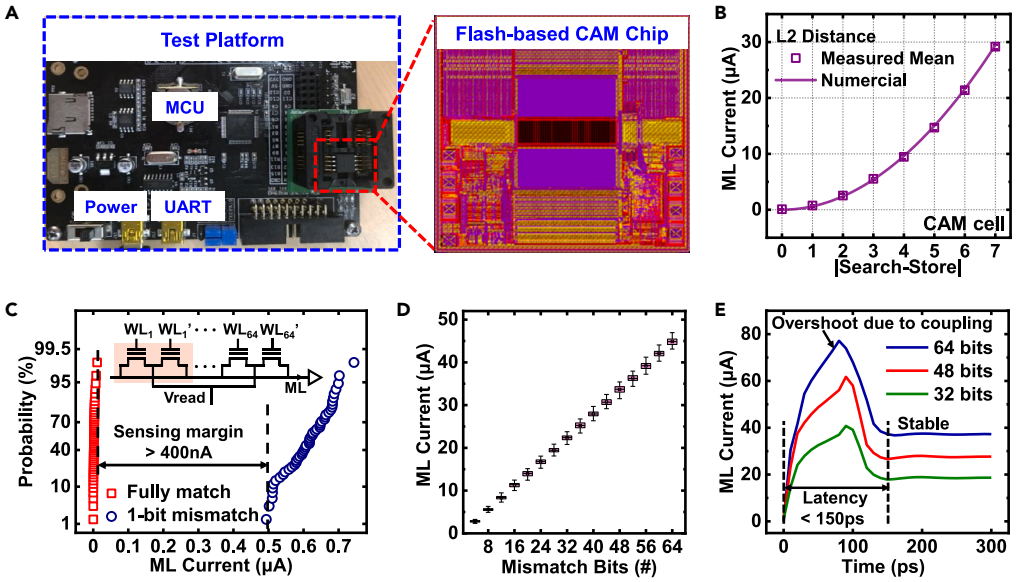


Figure 2. Experimental demonstration of the proposed CAM

(A) Test platform and the layout of 1Mb Flash-based CAM chip.

(B) The measured currents in ML of single CAM cell among 400 samples. It shows a good consistency with numerical L2 distance. The operating conditions are V_g = 6 V and V_{read} = 0.8 V. The minimum current step is 0.6 μA.

(C) The measured sensing margin between fully match and 1-bit mismatch case in 64-dimension CAM word. The margin is sufficient enough for the ADC circuit design.

(D) The measured ML current with the degree of mismatch in 64-dimension word. The good linearity is conducive to the reliable distinction. Data are represented as mean ± SEM.

(E) Simulations of the transient response in Flash-based CAM. The compact model of NOR Flash transistor is built based on the BSIM3v3 SPICE model calibrated by experimental data.

Flash transistor. Due to the parasitic effects in the test platform, the measured transient response cannot represent the intrinsic speed of CAM inside the fabricated chip. Therefore, we adopt HSPICE simulations to analyze the transient response of the proposed Flash-based CAM, as reported in the literature.[13,17,32] We first build a compact model of NOR Flash transistors based on the BSIM3v3 SPICE model calibrated by experimental data (See Figure S5). The simulation results are shown in Figure 2E, which indicate that the ML currents can reach stable after 150 ps at most for different mismatch bits in 64-dimension CAM word. The search latency will increase as the word width increases due to the parasitic resistance and capacitance in MLs (See Figure S6), which is a common characteristic in CAM design.[33] From this point of view, multi-bit capability in CAM is also crucial to reducing the word width and search latency. As for the overshooting, we think it is induced by the coupling effect of capacitance between the gate and source of Flash transistors. During working process, the source of Flash transistors is biased with a constant read voltage (V_{read} = 0.8V), and the drain terminals are connected to ground. When a query is input to the Flash-based CAM for distance search, a large search voltage (V_g = 6V) is applied to the gate terminals, inducing an abrupt rising step. The high-frequency component of gate voltage is injected into the source terminals through the coupling capacitance between gate and source (C_{gs}) of Flash transistors. Therefore, the source voltage will be pulled up, which is higher than V_{read}, in the initial stage of switching transient and generates the current overshoot. When gate voltage stabilizes at 6V, the injection from gate to source is terminated, making output current stable. As for the potential impact, we define the ML current that exceeds the final stable state is the portion of overshoot, and it takes up about 1/3 of the CAM array's energy consumption in search operations. Thus far, we have experimentally demonstrated the proposed CAM design capable of computing L2 distance reliably and fast both in single cell and array.

Experiments of few-shot learning

To evaluate the performance of our proposed L2 CAM design, we conduct the MANN experiments for few-shot learning. We employ a four-layer CNN as the feature extractor. The specific network structure and algorithm setting are shown in Figure S7. Note that the weight parameters are fixed after offline meta-training and will no longer change during the few-shot learning. For the benchmark, we choose the widely used Omniglot dataset[34] for demonstration and evaluation. In this dataset, there are 1623 handwritten characters (classes) from different alphabets, and each character contains 20 examples from different people. We randomly choose 1200 classes for meta-training of feature extractor and the rest of 423 classes for configuring few-shot learning tasks. For an *n*-way *k*-shot task, the model will first be fed into *n* different classes, each class with *k* examples, and implement on-device few-shot learning by writing the extracted feature vectors into Flash-based CAM. After that, the model should be able to classify new samples from the above *n* classes.

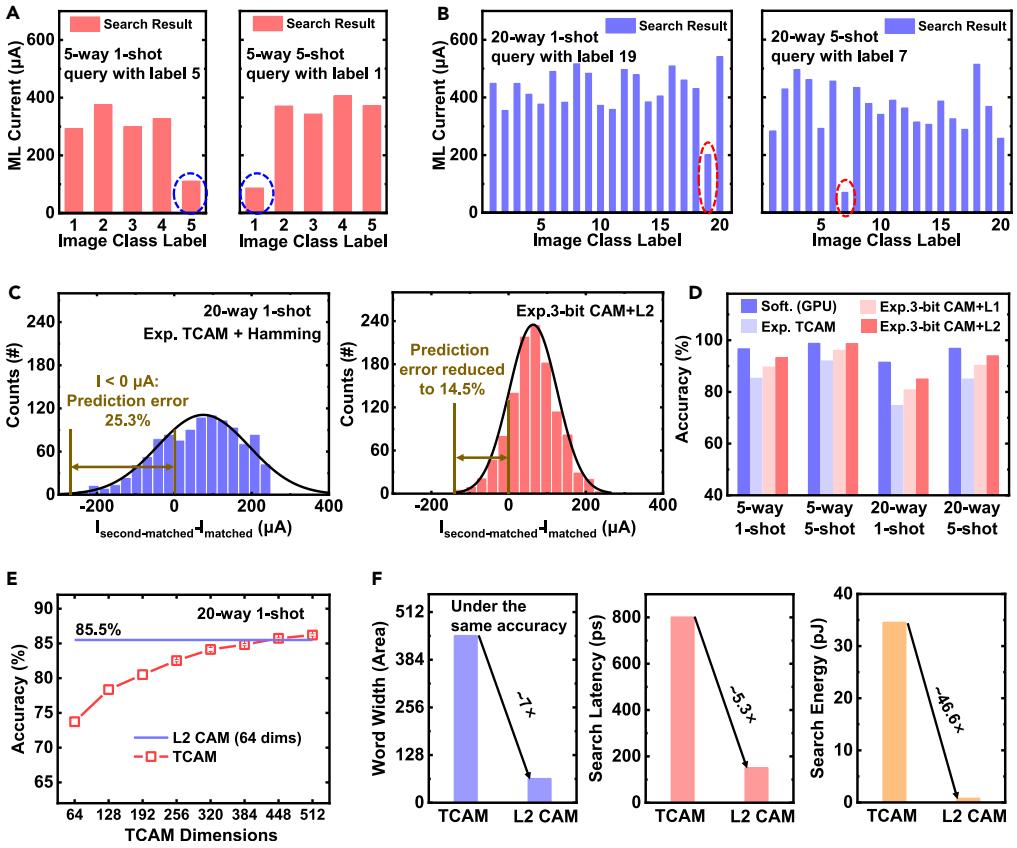


Figure 3. Experimental results of CAM-based few-shot learning

(A) The measured sum currents representing L2 distance between query vectors and stored contents from 5-way tasks.

(B) The measured sum currents representing L2 distance between query vectors and stored contents from 20-way tasks. The results exhibit a reliable distinction for predicting label, and the distinction is naturally improved due to the increasing learning examples.

(C) The distribution of current difference between the second-closest matched and the closest matched case in TCAM with Hamming distance and our proposed CAM with L2 distance. The prediction error is greatly reduced by using the proposed method.

(D) The accuracy comparison of GPU (full precision) and CAM experiments (@64-dimension features), which indicates the superiority of multi-bit-storage property and L2 distance.

(E) The inference accuracy increases with TCAM dimensions. To achieve the same accuracy with our L2 CAM, the word width of TCAM needs to be extended to 448 dimensions. Data are represented as mean ± SEM.

(F) Performance comparison between TCAM and the proposed L2 CAM.

In our experiments, during few-shot learning, the 64-dimension features of support set are first extracted by 4-layer CNN, quantized to 3-bit according to the scheme of L2 distance, and then stored in the Flash chip. The conductance distributions of feature vectors in different tasks are shown in Figure S8. During inference, the extracted query features are coded and transformed into a series of WL voltages for computing L2 distance with all the stored features in parallel. The predicted label is given by sensing ML for minimum current (See STAR Methods). Figures 3A and 3B show the measured sum currents representing L2 distance from four randomly selected queries in 5-way 1-shot, 5-way 5-shot, 20-way 1-shot, and 20-way 5-shot tasks, respectively. The results exhibit a reliable distinction for predicting labels. Naturally, the 5-shot tasks show a larger distinction than the 1-shot tasks. It is because the stored features of 5-shot tasks are the average results across 5 examples and can better reflect the characteristics of classes. We compare the distribution of current difference between the second-closest matched and the closest matched case in TCAM with Hamming distance and our proposed L2 CAM, respectively. The results are shown in Figure 3C, which are measured from 1200 samples. The network using Hamming distance for comparison is the same as the network using L2 distance, which is a 4-layer CNN, each layer including 64 channel kernels. The only difference in the two experiments is that for Hamming distance, the output vectors are quantized to ternary, whereas for L2 distance, they are quantized to 3-bit according to our method. The negative value of current difference represents the prediction error of CAM-based MANN. We can see that it is reduced (25.3%–14.5%) by using the proposed method. Furthermore, we also compare the overall inference accuracy of four methods: software in GPU, measured on TCAM with Hamming distance, 3-bit-storage and ternary-search CAM with L1 distance, and L2 distance, respectively, as illustrated in Figure 3D. We also compare the classification accuracy of our L2 CAM-based MANN against the state-of-the-art hardware-based few-shot learning works with Omniglot dataset, as shown in Table S1. For a fair comparison, all the CAM widths are set as 64 dimensions in this

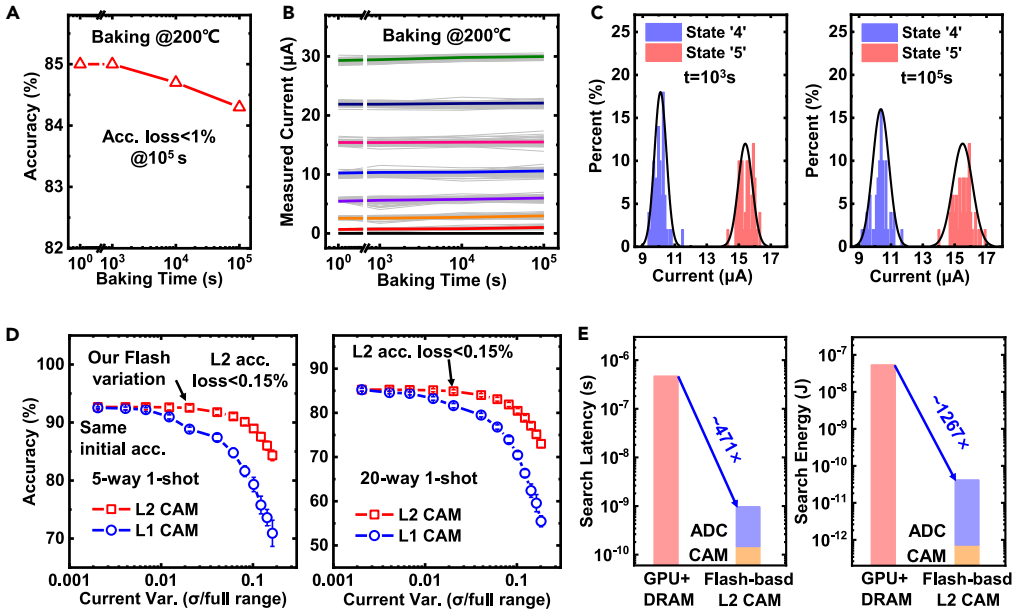


Figure 4. Performance analysis and evaluation of CAM-based few-shot learning

(A) Measured accuracy with the baking time. The accuracy loss is less than 1% after baking 10^5 s at 200°C, which proves the robustness for on-device lifelong learning.

(B) Statistical retention behavior of the 3-bit storage at 200°C. Gray lines are the raw data of 50 cells for each state, and colorful lines are the mean currents.

(C) The measured and fitted evolution of current in state "4" and state "5". The distribution is only slightly spread.

(D) Simulated impacts of current variations on inference accuracy for L1 and L2 CAM, which show the L2 CAM possesses a better immunity to the disturbance, and the variation level in our Flash chip has almost no degradation (<0.15%). Data are represented as mean ± SEM.

(E) Performance comparison with GPU on search latency and search energy.

experiment. The recognition accuracy of L2 CAM is increased by 8.5% on average for different few-shot learning tasks compared with TCAM. Take the 20-way 1-shot task as an example, our proposed 3-bit-storage CAM with L2 distance exhibits a 10.8% accuracy improvement compared with TCAM and a 4.2% improvement for 3-bit-storage CAM with L1 distance, which indicates the superiority of multi-bit storage property and L2 distance. To consider the parasitic effects on inference accuracy, we perform the post-layout simulations, as shown in Figure S9, and the results indicate a <0.3% accuracy drop. On the other hand, to achieve the same accuracy as L2 CAM for 20-way 1-shot, the word width of TCAM needs to be extended to 448 dimensions, as shown in Figure 3E. For other learning tasks (5-way 1-shot, 5-way 5-shot, and 20-way 5-shot), the word width of TCAM also requires 448 dimensions or even larger to achieve comparable accuracy with 64-dimension L2 CAM, as shown in Figure S10. We also compare the overhead of the two cases at the same accuracy, which indicates a reduction of 7-fold in circuit area, 5.3-fold in search latency, and 46.6-fold in search energy for the proposed L2 CAM, as illustrated in Figure 3F. The comparison of recognition accuracy for 5-way and 20-way task with other distance metrics is shown in Figure S11. Although there is a little gap in accuracy with GPU, we can use the p-stable locality sensitive hash (LSH) algorithms to increase the word width of L2 CAM approaching software accuracy,[35] as shown in Figure S12. Besides, our proposed L2 CAM method can also be extended beyond 3-bit to 4-bit or 5-bit, provided that the programming accuracy and state number of the devices are sufficient, as shown in Figure S13.

DISCUSSION

In the field of neuroscience, it is found that environmental factors would influence our cognition and brain functions throughout our life span.[36] Likewise, for our proposed CAM-based MANN, the accuracy of few-shot learning can also be influenced by the reliability of Flash devices under various stresses. To investigate this, we baked the Flash chip at 200°C for a long time (10^5 s) and measured the inference accuracy periodically. The accuracy loss of 20-way 1-shot task is less than 1% after baking for 10^5 s at 200°C, which proves the robustness of on-device lifelong learning, as shown in Figure 4A. This is benefited from the good stability of Flash cells, as shown in Figure 4B, where gray lines are the discrete data of 50 cells for each state and colorful lines are the mean currents of 8 states in 3-bit-storage CAM. The specific evolutions of state "4" and "5" are also shown in Figure 4C, which only exhibit a slight spread. Furthermore, the accuracy can also be affected by the device variation, which arises from the fabrication of Flash chips. To analyze this, we simulated the impacts of current variations on inference accuracy for L1 and L2 CAM, respectively, as shown in Figure 4D. We can see that our proposed L2 CAM exhibits a better immunity to the variation than L1 CAM, which is benefited from the sophisticated function relationship and quadratic coding of L2 distance, similar to the anti-noise mechanism of the complex coding scheme in biological brains.[37] Under a quadratic coding scheme, the magnitude of sum current in ML is dominated by the larger state, which could cover the fluctuation of smaller states and therefore reduce the influence of variation. Besides, the

Table 1. Comparison of Flash-based CAM with other reported works

	SRAM[45]	RRAM[13]	MTJ[32]	FeFET[17,20]		PCM[46]	This work
CAM design	16T	2T-2R	15T-4MTJ	2FeFET	2FeFET	2T-2PCM	2Flash
Tech. node (nm)	65	180	40	45	28	90	65
Bit precision	Ternary (0, 1, X)				2-bit	Bipolar (−1, +1)	3-bit
Cell area per bit (μm^2)	11.43	0.54	10.76	0.15	0.40	0.41	0.03[a]
Search latency (ps)[b]	520	N/A	170	355	700	N/A	150
Energy (fJ/bit/search)	1.04	1.15	0.17	0.4	1.1	2.5	0.33
Distance metric	Hamming distance				Sigmoid	Dot product	L2
Accuracy (5-way1-shot)	N/A	92.8%	N/A	93.1%	93.2%	97.2%	93.3%

[a]The cell area is calculated by $10F^2$ for NOR Flash transistor.
[b]All the search latency in compared references were evaluated by simulation.

variation level in our Flash chip ($\sigma < 0.6\ \mu A$) shows almost no degradation to the accuracy (<0.15%) under different few-shot learning tasks with L2 CAM. These results together illustrate the robustness of our proposed Flash-based CAM design with L2 distance.

In comparison to the designs based on other emerging devices, Flash memory is based on the mature technology and hence can be integrated for a large scale, which possesses a high parallelism and large capacity for search. Besides, the retention of Flash-based CAM is excellent, which shows less than 1% accuracy loss after baking for 10^5 s at 200°C, ensuring the robustness of on-device lifelong learning. What is more, it is also with a higher density benefited from the compact 2T structure. As for the high programming voltage in scaled technology nodes, the memory updates of Flash are quite few in the few-shot learning problems, which only involves several known samples. Besides, we can also thin the tunneling layer in gate dielectric to decrease the programming voltage, which can satisfy the applications that need to frequently change the stored contents with a relaxed retention demand.[38,39]

As the dimension of stored vectors increases, the level of calculated distances (output currents) will increase and the proportion of mismatch currents to the summed current decrease, requiring ADC with a higher bit number to detect the difference, which is a common matter in CAM-based distance search. Our proposed L2 CAM could enlarge the detectable mismatch difference compared with the TCAM method, especially for large dimensional vectors, as shown in Figure S14.

As for the learning process, it is quite occasional in few-shot learning applications because the known samples are very few, typically ranging from several to tens.[40–43] During few-shot learning, the feature vector of known sample is first extracted by the CNN encoder, quantized according to the L2 coding scheme, and then written into the identified Flash-based CAM entry, which means one sample (shot) corresponds to one memory update. With our proposed programming method, the Flash devices are firstly erased to a high current state and then programmed to the target states, which guarantees no erasing operation during programming. That is to say, one memory update involves one cycling operation. The cycling endurance performance of our Flash devices is shown in Figure S15, which exceeds 3×10^3 cycles. Therefore, the endurance performance of our devices is quite adequate for the few-shot learning applications.

Compared with the implementations by employing conventional von Neumann computing architecture, the CAM-based MANN can compare input search vector with all the stored vectors in parallel and calculate pairwise distances *in situ*, which takes out the frequent data movement between explicit memory and processing unit. The MANN model in our experiment is run in PyTorch with NVIDIA GeForce GTX 1080Ti. The performance evaluation reveals a 471-fold reduction in search latency and 1267-fold reduction in search energy compared with GPU for search operation (See STAR Methods for details), as illustrated in Figure 4E.[13,44] The end-to-end speedup of the whole MANN workload for few-shot learning tasks is 4.7x with GPU backed by our proposed Flash-based L2 CAM (See Figure S16). For the quantization problem, first, the current range is determined through statistics analysis on the target datasets, which just needs to cover the match case and the least mismatch case. The excessive part will be directly cut off before ADC, which narrows the current range. Second, the quantization focus of CAM-based few-shot learning is to find the relative magnitude relationship between the sum currents of different MLs. Although the output L2 distance of a single cell exceeds 4-bit, the sensing margin of a CAM word will be amplified by accumulating the mismatch of all CAM cells on the same ML. The analyses are consistent with the simulation results as shown in Figure S17, which indicates that 4-bit ADC for quantization is already sufficient, with a negligible (<0.3%) accuracy loss. Therefore, we choose 4-bit ADC for performance benchmarking. Table 1 compares our Flash-based L2 CAM with other reported works.[13,17,20,32,45,46] The results show advances referring to the cell area (0.03 μm^2/bit), search latency (150 ps), and search energy (0.33 fJ/bit/search), which are most critical metrics for CAM. Besides, the benchmarking also considers bit precision, and distance metric, which can impact the classification accuracy of MANN in few-shot learning problems.

To sum up, a Flash-based multi-bit-storage and ternary-search CAM capable of computing L2 distance was first fabricated. Compared with TCAM capable of computing Hamming distance, the search performance of MANN based on our fabricated CAM achieves a 5.3-fold reduction in latency and 46.6-fold reduction in energy at the same recognition accuracy and an 8.5% accuracy improvement when searching with the same word width. Besides, our fabricated CAM shows excellent immunity to the variation of devices and high temperatures. The successful demonstrations of the robust and energy-efficient CAM open new possibilities for implementing lifelong on-device machine intelligence.

Limitations of the study

Due to the parasitic effects in the measurement platform, the measured transient response cannot represent the intrinsic speed of CAM inside the fabricated chip. Therefore, we adopt HSPICE simulations to analyze the transient response of the proposed Flash-based CAM. Besides, although the experiments of Flash-based 64-dim L2 CAM on Omniglot dataset shows remarkable performance, demonstrations for more complex tasks (e.g., miniImageNet) is a common challenge for CAM-based distance search, where the overall architecture and circuit topology still need to be carefully studied.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS
- METHOD DETAILS
  - Experimental platform
  - Measurement setup
  - Performance evaluation and comparison
- QUANTIFICATION AND STATISTICAL ANALYSIS

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2023.108371.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2205100, the National Natural Science Foundation of China Program under Grant 62022006, Grant 62034006, the 111 Project (B18001).

AUTHOR CONTRIBUTIONS

P.H. defined the research question and directions. H.Z.Y. conceived the method and performed the experiments. H.Z.Y. and X.Y.L. performed the simulation. R.Y.L., N.T., and L.F.L. conducted the data analysis. Y.Z.Z. and Z.Z. performed the benchmarking. P.H. and J.F.K. supervised the project. H.Z.Y. wrote the manuscript. All authors analyzed the results and implications and commented on the manuscript at all stages.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: February 14, 2023
Revised: October 7, 2023
Accepted: October 26, 2023
Published: October 31, 2023

REFERENCES

1. Dhar, S., Guo, J., Liu, J.J., Tripathi, S., Kurup, U., and Shah, M. (2021). A survey of on-device machine learning: An algorithm and learning theory perspective. ACM Trans. Internet Things *2*, 1–49.
2. Chen, Z., and Liu, B. (2016). Lifelong Machine Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning (Morgan and Claypool Publishers).
3. Silver, D.L., Yang, Q., and Li, L. (2013). Lifelong machine learning systems: Beyond learning algorithms. In 2013 AAAI Spring Symposium Series.
4. Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. Commun. ACM *60*, 84–90. https://doi.org/10.1145/3065386.
5. Graves, A., Mohamed, A.R., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947.
6. McCloskey, M., and Cohen, N.J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. Psychol. Learn. Motiv. *24*, 109–165.
7. Friston, K. (2003). Learning and inference in the brain. Neural Netw. *16*, 1325–1352.
8. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. (2016). Meta-learning with memory-augmented neural networks. In International Conference on Machine Learning, pp. 1842–1850.
9. Vinyals, O., Blundell, C., Lillicrap, T., and Wierstra, D. (2016). Matching networks for one shot learning. Adv. Neural Inf. Process. Syst. *29*.
10. Lemke, C., Budka, M., and Gabrys, B. (2015). Meta-learning: a survey of trends and technologies. Artif. Intell. Rev. *44*, 117–130.
11. Karunaratne, G., Schmuck, M., Le Gallo, M., Cherubini, G., Benini, L., Sebastian, A., and Rahimi, A. (2021). Robust high-dimensional memory-augmented neural networks. Nat. Commun. *12*, 2468.

12. Stevens, J.R., Ranjan, A., Das, D., Kaul, B., and Raghunathan, A. (2019). Manna: An accelerator for memory-augmented neural networks. In Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, pp. 794–806. https://doi.org/10.1145/3352460.3358304.
13. Mao, R., Wen, B., Kazemi, A., Zhao, Y., Laguna, A.F., Lin, R., Wong, N., Niemier, M., Hu, X.S., Sheng, X., et al. (2022). Experimentally realized memristive memory augmented neural network. Nat. Commun. *13*, 6284.
14. Li, H., Chen, W.C., Levy, A., Wang, C.H., Wang, H., Chen, P.H., Wan, W., Philip Won, H.-S., and Raina, P. (2021). One-shot learning with memory-augmented neural networks using a 64-kbit, 118 GOPS/W RRAM-based non-volatile associative memory. In 2021 Symposium on VLSI Technology, pp. 1–2.
15. Wu, T.F., Li, H., Huang, P.C., Rahimi, A., Rabaey, J.M., Wong, H.S.P., Shulaker, M.M., and Mitra, S. (2018). Brain-inspired computing exploiting carbon nanotube FETs and resistive RAM: Hyperdimensional computing case study. In 2018 IEEE International Solid-State Circuits Conference-(ISSCC) (IEEE), pp. 492–494. https://doi.org/10.1109/ISSCC.2018.8310399.
16. Ranjan, A., Jain, S., Stevens, J.R., Das, D., Kaul, B., and Raghunathan, A. (2019). X-MANN: A crossbar-based architecture for memory augmented neural networks. In Proceedings of the 56th Annual Design Automation Conference (DAC), pp. 1–6. https://doi.org/10.1145/3316781.3317935.
17. Ni, K., Yin, X., Laguna, A.F., Joshi, S., Dünkel, S., Trentzsch, M., Müller, J., Beyer, S., Niemier, M., Hu, X.S., and Datta, S. (2019). Ferroelectric ternary content-addressable memory for one-shot learning. Nat. Electron. *2*, 521–529.
18. Laguna, A.F., Yin, X., Reis, D., Niemier, M., and Hu, X.S. (2019). Ferroelectric FET based in-memory computing for few-shot learning. In Proceedings of the 2019 on Great Lakes Symposium on VLSI, pp. 373–378. https://doi.org/10.1145/3299874.3319450.
19. Hu, X.S., Niemier, M., Kazemi, A., Laguna, A.F., Ni, K., Rajaei, R., Sharifi, M.M., and Yin, X. (2021). In-memory computing with associative memories: a cross-layer perspective. In 2021 IEEE International Electron Devices Meeting (IEDM), p. 25-2. https://doi.org/10.1109/IEDM19574.2021.9720562.
20. Kazemi, A., Sharifi, M.M., Laguna, A.F., Müller, F., Yin, X., Kämpfe, T., Niemier, M., and Hu, X.S. (2022). FeFET multi-bit content-addressable memories for in-memory nearest neighbor search. IEEE Trans. Comput. *71*, 2565–2576.
21. Yang, H., Huang, P., Han, R., Liu, X., and Kang, J. (2023). An ultra-high-density and energy-efficient content addressable memory design based on 3D-NAND flash. Sci. China Inf. Sci. *66*, 142402.
22. Li, C., Graves, C.E., Sheng, X., Miller, D., Foltin, M., Pedretti, G., and Strachan, J.P. (2020). Analog content-addressable memories with memristors. Nat. Commun. *11*, 1638.
23. Dokmanic, I., Parhizkar, R., Ranieri, J., and Vetterli, M. (2015). Euclidean distance matrices: essential theory, algorithms, and applications. IEEE Signal Process. Mag. *32*, 12–30.
24. Kazemi, A., Sahay, S., Saxena, A., Sharifi, M.M., Niemier, M., and Hu, X.S. (2021). A flash-based multi-bit content-addressable memory with Euclidean squared distance. In 2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), pp. 1–6. https://doi.org/10.1109/ISLPED52811.2021.9502488.
25. Kazemi, A., Müller, F., Sharifi, M.M., Errahmouni, H., Gerlach, G., Kämpfe, T., Imani, M., Hu, X.S., and Niemier, M. (2022). Achieving software-equivalent accuracy for hyperdimensional computing with ferroelectric-based in-memory computing. Sci. Rep. *12*, 19201.
26. Takeuchi, K., and Fukuma, M. (1994). Effects of the velocity saturated region on MOSFET characteristics. IEEE Trans. Electron Devices *41*, 1623–1627.
27. Esseni, D., and Abramo, A. (2003). Modeling of electron mobility degradation by remote Coulomb scattering in ultrathin oxide MOSFETs. IEEE Trans. Electron Devices *50*, 1665–1674.
28. Ren, Y., Lin, R., Ran, J., Liu, C., Tao, C., Wang, Z., and Wong, N. (2021). BATMANN: A Binarized-All-Through Memory-Augmented Neural Network for Efficient In-Memory Computing. In 2021 IEEE 14th International Conference on ASIC (ASICON) (IEEE), pp. 1–4. https://doi.org/10.1109/ASICON52560.2021.9620292.
29. Bez, R., Camerlenghi, E., Modelli, A., and Visconti, A. (2003). Introduction to flash memory. Proc. IEEE *91*, 489–502.
30. Liu, L., Sharifi, M.M., Rajaei, R., Kazemi, A., Ni, K., Yin, X., Niemier, M., and Hu, X.S. (2022). Eva-cam: a circuit/architecture-level evaluation tool for general content addressable memories. In 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1173–1176. https://doi.org/10.23919/DATE54114.2022.9774572.
31. Shibata, H., Kozlov, V., Ji, Z., Ganesan, A., Zhu, H., and Paterson, D. (2017). A 9GS/s 1GHz-BW Oversampled Continuous-Time Pipeline ADC Achieving-161 dBFS/Hz NSD. In 2017 IEEE International Solid-State Circuits Conference-Digest of Technical Papers, p. 278. https://doi.org/10.1109/ISSCC.2017.7870369.
32. Wang, C., Zhang, D., Zeng, L., Deng, E., Chen, J., and Zhao, W. (2019). A novel MTJ-based non-volatile ternary content-addressable memory for high-speed, low-power, and high-reliable search operation. IEEE Trans. Circuits Syst. I. *66*, 1454–1464.
33. Chang, M.F., Lin, C.C., Lee, A., Chiang, Y.N., Kuo, C.C., Yang, G.H., Tsai, H.J., Chen, T.F., and Sheu, S.S. (2017). A 3T1R nonvolatile TCAM using MLC ReRAM for frequent-off instant-on filters in IoT and big-data processing. IEEE J. Solid-State Circuits *52*, 1664–1679.
34. Lake, B., Salakhutdinov, R., Gross, J., and Tenenbaum, J. (2011). One shot learning of simple visual concepts. In Proceedings of the Annual Meeting of the Cognitive Science Society, *33*.
35. Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V.S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the Twentieth Annual Symposium on Computational Geometry, pp. 253–262. https://doi.org/10.1145/997817.997857.
36. Kramer, A.F., Bherer, L., Colcombe, S.J., Dong, W., and Greenough, W.T. (2004). Environmental influences on cognitive and brain plasticity during aging. J. Gerontol. A Biol. Sci. Med. Sci. *59*, M940–M957.
37. Montijn, J.S., Meijer, G.T., Lansink, C.S., and Pennartz, C.M.A. (2016). Population-level neural codes are robust to single-neuron variability from a multidimensional coding perspective. Cell Rep. *16*, 2486–2498.
38. Fang, H.K., Chang-Liao, K.S., Chou, K.C., Chao, T.C., Tsai, J.E., Li, Y.L., Huang, W.H., Shen, C.H., and Shieh, J.M. (2020). Impacts of electrical field in tunneling layer on operation characteristics of Poly-Ge charge-trapping Flash memory device. IEEE Electron. Device Lett. *41*, 1766–1769.
39. Hong, S.H., Jang, J.H., Park, T.J., Jeong, D.S., Kim, M., Hwang, C.S., and Won, J.Y. (2005). Improvement of the current-voltage characteristics of a tunneling dielectric by adopting a Si3N4/ SiO2/ Si3N4 multilayer for flash memory application. Appl. Phys. Lett. *87*. https://doi.org/10.1063/1.2093932.
40. Lake, B.M., Salakhutdinov, R., and Tenenbaum, J.B. (2015). Human-level concept learning through probabilistic program induction. Science *350*, 1332–1338.
41. Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011). The Caltech-UCSD Birds-200–2011 dataset (Technical report California Institute of Technology). https://www.vision.caltech.edu/datasets/cub_200_2011/.
42. Vinyals, O., Blundell, C., Lillicrap, T., and Wierstra, D. (2016). Matching networks for one shot learning. Adv. Neural Inf. Process. Syst. *29*.
43. Yu, M., Guo, X., Yi, J., Chang, S., Potdar, S., Cheng, Y., Tesauro, G., Wang, H., and Zhou, B. (2018). Diverse few-shot text classification with multiple metrics. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, *Volume 1*Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1206–1215. https://doi.org/10.18653/v1/N18-1109.
44. Van der Plas, G., Decoutere, S., and Donnay, S. (2006). A 0.16 pJ/conversion-step 2.5 mW 1.25 GS/s 4b ADC in a 90nm digital CMOS process. In 2006 IEEE International Solid-State Circuits Conference-Digest of Technical Papers, p. 2310. https://doi.org/10.1109/ISSCC.2006.1696294.
45. Choi, W., Lee, K., and Park, J. (2018). Low cost ternary content addressable memory using adaptive match line discharging scheme. In 2018 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–4. https://doi.org/10.1109/ISCAS.2018.8351461.
46. Karunaratne, G., Schmuck, M., Le Gallo, M., Cherubini, G., Benini, L., Sebastian, A., and Rahimi, A. (2021). Robust high-dimensional memory-augmented neural networks. Nat. Commun. *12*, 2468.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
MATLAB R2020a	MathWorks	https://ww2.mathworks.cn/
Origin 2020b	OriginlLab Corporation	https://www.originlab.com/
PyTorch	Open Source	https://pytorch.org/
Flash-based MANN	Github	https://github.com/ElijahBele/Flash-based_L2_CAM.git
Other		
Semiconductor parameter analyzer	Keysight	https://www.keysight.com/cn/zh/home.html
Cortex-M3 MCU	ARM	https://developer.arm.com/
GeForce GTX 1080Ti	NVIDIA	https://www.nvidia.cn/

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to the lead contact, Peng Huang (phwang@pku.edu.cn).

Materials availability

This study did not generate any new unique reagents.

Data and code availability

- All data reported in this paper will be shared by the lead contact upon request.
- The code used in this study are available at https://github.com/ElijahBele/Flash-based_L2_CAM.git.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS

This study does not use experimental models typical in the life sciences.

METHOD DETAILS

Experimental platform

The experimental platform is built based on a prototype CAM chip, which was fabricated in standard 65-nm NOR Flash technology. The Flash array is organized as the crossbar architecture with 1M devices. In addition to the Flash devices, the chip integrates the circuits for addressing, erasing, programming, and communication (serial peripheral interface, SPI). The chip is interfaced with a PCB test board, which employs an ARM Cortex-M3 MCU for system control and timing management, and a semiconductor parameter analyzer (Keysight B1500A) for electrical characterization.

Measurement setup

To read a Flash device, the corresponding WL and BL are first selected. After that, a source measurement unit (SMU) on the B1500A applies a read voltage to the Flash device and senses the flowing current via an on-chip readout circuit. Because the readout circuit only allows for a relatively small current limit, the search operation of one CAM word is performed by reading the CAM cells in sequence (open 2 WLs at a time), and summing the currents outside the chip. The search between different CAM words is also performed in sequence limited by the pin number and test equipment. The whole experimental platform is operated by a host computer and MATLAB software.

Performance evaluation and comparison

For the evaluation of performance metrics, the cell area ($0.03\mu m^2$/bit) is calculated by the $10F^2$ (common bit line structure) for NOR Flash transistors. The search latency (150ps) is based on the simulation results of HSPICE. The search energy (0.33 fJ/bit/search) is the average results of search vectors from few-shot learning tasks.

For the comparison with GPU-based solution, the evaluation of our method mainly considers two parts: Flash-based CAM array and quantization circuit. The part of Flash-based CAM array is calculated according to the above analysis. For the mismatch quantization on ML current, it is similar to the quantization of activation in NVM-based hardware neural network. First, the current range is determined through statistics analysis on the target datasets and the excessive part will be directly cut off before ADC, which narrows the current range for quantization. Second, although the output L2 distance of a single cell exceeds 4-bit, the quantization focus of CAM-based few-shot learning is to determine the relative magnitude relationship between the sum currents of different MLs. The sensing margin is also the mismatch accumulation of all CAM cells on the same ML, which is much larger. Besides, our L2 distance can further amplify the margin thanks to the quadric coding scheme. Therefore, a 4-bit ADC is already sufficient for the quantization of ML current, with an accuracy loss less than 0.3%. We assume one ML equipped with an ADC.[44] The total latency is 0.15 ns + 0.8 ns = 0.95 ns for one image search in 20-way 1-shot task. Likewise, for the energy estimation of search operation, we also calculate these two parts. By reading the Flash conductance matrix out and simulating the response in HSPICE, the energy consumption of Flash array is 0.74 pJ. For the consumption in ADC, one quantization operation is 2 pJ. Therefore, the total energy consumption is 0.74 pJ + 2 pJ × 20 = 40.74 pJ for one image search in 20-way 1-shot task. For GPU-based method, we run the MANN workload in PyTorch with NVIDIA GeForce GTX 1080Ti. The task is run for multiple iterations to provide a better average hardware performance. The estimations of latency (447 ns) and energy consumption (51.7 nJ) for one image search in GPU with DRAM scheme are acquired using PyTorch Profiler and the NVIDIA System Management Interface.[13]

QUANTIFICATION AND STATISTICAL ANALYSIS

The statistical analysis details of each experiment can be found in the corresponding figure legend, and this study does not include quantification.