BORE: Energy-Efficient Banded Vector Similarity Search with Optimized Range Encoding for Memory-Augmented Neural Network

Chi-Tse Huang*, Cheng-Yang Chang*, Hsiang-Yun Cheng†, An-Yeu (Andy) Wu*, *Fellow, IEEE*

*Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan
†Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
{rickhuang, kevin}@access.ee.ntu.edu.tw, hycheng@citi.sinica.edu.tw, andywu@ntu.edu.tw

***Abstract*—Memory-augmented neural networks (MANNs) incorporate external memories to address the significant issue of catastrophic forgetting in few-shot learning applications. MANNs rely on vector similarity search (VSS), which incurs substantial energy and computational overhead due to frequent data transfers and complex cosine similarity calculations. To tackle these challenges, prior research has proposed adopting ternary content addressable memories (TCAMs) for parallel VSS within memory. One promising approach is to use Exact-Match TCAM (EX-TCAM) with range encoding to find the vector with the minimum L_∞ distance, avoiding the need for sensing circuit modifications as required by Best-Match TCAM (Best-TCAM). However, this method demands multiple search iterations and longer code words, limiting its practicality. In this paper, we propose an energy-efficient EX-TCAM-based design called BORE. BORE skips redundant search iterations and reduces code word length through performing Banded L_∞ distance search with Optimized Range Encoding. Additionally, we consider the characteristics of the similarity metric and develop a distance-based training mechanism aimed at improving classification accuracy. Simulation results demonstrate that BORE enhances energy efficiency by 9.35× to 11.84× and accuracy by 2.95% to 4.69% compared to previous EX-TCAM-based approaches. Furthermore, BORE improves energy efficiency by 1.04× to 1.63× over prior works of Best-TCAM-based VSS.**

I. Introduction

Memory-augmented neural networks (MANNs) [1], [2] offer a promising solution to address catastrophic forgetting, a significant challenge that traditional deep neural networks (DNNs) often encounter when generalizing from prior knowledge to new classes. MANNs integrate DNNs with external memory modules to enable the storage and retrieval of feature vectors (also known as support vectors) extracted by DNNs. It can make accurate class predictions by performing vector similarity search (VSS), which calculates the cosine similarity (i.e., cosine of the angle) between the query and each support vector to identify the most similar one, and has attracted significant attention in recent years. However, VSS involves frequent transfer of vectors between off-chip-memories and computational units, inducing large energy overhead and memory-wall bottleneck in conventional von Neumann-based computing systems [4].

To address the memory-wall bottleneck, researchers have proposed in-memory search (IMS) [5]–[10], which aims to reduce excessive vector movement and accelerate VSS by conducting parallel searches within ternary content-addressable memories (TCAMs). Two main IMS approaches have been explored in the literature: (i) Best-Match TCAM (Best-TCAM) [5]–[8] and (ii) Exact-Match TCAM (EX-TCAM) combined with range encoding (RENÉ) [9]–[11]. Best-TCAM-based designs require modifications of peripheral components and sensing circuits to find the most similar support vector. However, their complex circuitry and insufficient sensing margin result in high power consumption and limited search parallelism within a TCAM array. In contrast, EX-TCAM-based designs eliminate the need for modifications in sensing circuits by performing iterative searches (shown in Fig. 1(a)) using L_∞ distance as the similarity metric. This process involves sequentially expanding search ranges and encoding them for exact matching. Consequently, the search architecture can be simplified as EX-TCAM with simple sensing circuits to identify the stored vector that exactly matches ranges at each iteration, enabling low power consumption per search.

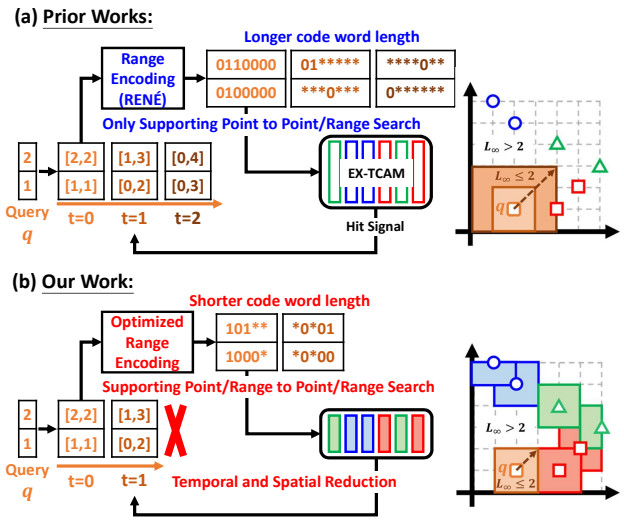

Fig. 1: Comparison between (a) prior works and (b) the proposed BORE.

Despite the notable advantages of EX-TCAM regarding sensing margin and energy efficiency, two significant challenges persist in EX-TCAM-based VSS methods [10], [11]. Firstly, EX-TCAM-based VSS methods require multiple search itera-

979-8-3503-4859-0/DATE24/© 2024 EDAA

tions to determine whether L_∞ distances between the query and any support vectors fall within specific values to identify the nearest-neighbor vector. Although EX-TCAM demonstrates superior energy efficiency per request compared to Best-TCAM, the cumulative energy cost of executing complete L_∞ distance searches can undermine its hardware performance benefits. Secondly, the issue of longer code word lengths in EX-TCAM-based VSS methods leads to a worse trade-off between efficiency and accuracy. Specifically, RENÉ [9] is a general-purpose encoding mainly used for packet processing in networking applications. It encodes individual points and ranges into ternary code words and only supports point-to-point and point-to-range searches. This limitation constrains the search flexibility and extends code word lengths. Consequently, these two challenges deteriorate the overall energy efficiency in the temporal and spatial domains, respectively.

To overcome the above challenges, this paper proposes the banded L_∞ distance search (L_{∞_b}) with the optimized range encoding (BORE). From the temporal domain, we observe that both initial and final iterations of the iterative search process are redundant. Therefore, unlike prior methods [10], [11] that aim to execute complete L_∞ distance searches, BORE skips a substantial portion of redundant search iterations. From the spatial domain, BORE develops the optimized range encoding supporting range-to-range searches to reduce the required code word lengths. In addition, L_∞ distance search typically suffers from a significant accuracy drop compared to software-based cosine similarity search [10], [11]. To bridge the accuracy gap, BORE further replaces the directional-based similarity metric (cosine similarity) with a distance-based one during the training phase, aligning the software training more closely with hardware implementations of inference. Combining these three proposed approaches, BORE improves the overall energy efficiency by 9.35× to 11.84×. The key contributions of this work are as follows:

1) **Banded L_∞ distance search (L_{∞_b}) for reducing search iterations from the temporal domain:** L_{∞_b} shortens the average search process by 1.94× to 3.39× without compromising accuracy significantly.
2) **Optimized range encoding for reducing code word lengths from the spatial domain:** The optimized range encoding demonstrates the code word length reduction from 40% to 68% for executing L_{∞_b} losslessly.
3) **Investigating distance-based training mechanisms for L_∞ and L_{∞_b} distance search:** Controllers trained with distance-based metric perform higher accuracy by 3.25% to 4.76% and reduce search iterations by 1.22× to 1.31× for hardware implementation of L_∞ distance search.

II. BACKGROUND

A. Memory-Augmented Neural Network (MANN) for Few-Shot Learning (FSL)

Few-shot learning (FSL) is a challenging task that aims to classify query images with only a limited number of support images available per class. FSL datasets [2], [13] are typically split into training, validation, and testing sets that contain

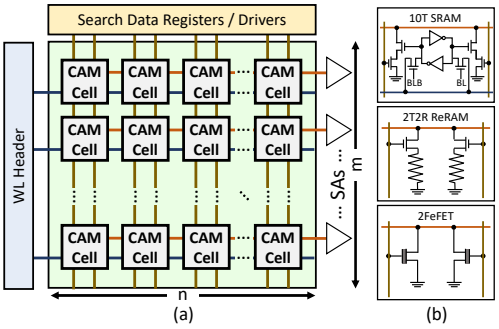

Fig. 2: (a) The EX-TCAM architecture. (b) The configuration of bit cells

distinct, non-overlapping classes. In the training phase, models are exposed to a series of N-way K-shot support images from the training set during different episodes. "N-way K-shot" denotes a task that classifies images from N unseen classes, with K images available as supports for each class. Traditional DNNs often struggle with FSL due to their tendency to overfit on newer data. In contrast, MANNs provide a solution by enhancing DNNs with external memory storage. This augmentation empowers MANNs to make correct predictions by preserving and recalling previously encountered data.

The critical operation in MANNs is the search operation, which involves calculating the cosine similarity between d-dimensional query vector $\vec{q}$ and support vector $\vec{s}$ as Eq. (1). While cosine similarity is a standard metric used in MANNs, measuring closeness by quantifying the angular difference imposes a significant computational burden. As alternatives, hardware-friendly distance metrics like L_1 [5], [6] and L_∞ [9]–[11] as Eq. (2) have been proposed to measure the distances between query and support vectors stored in memory modules.

$$COS(\vec{q},\vec{s}) = \frac{\vec{q}\cdot\vec{s}}{|\vec{q}|\,|\vec{s}|}, \tag{1}$$

$$\|\vec{q},\vec{s}\|_\infty = \max_i |q_i - s_i|. \tag{2}$$

B. Ternary Content Addressable Memory (TCAM)

Ternary Content Addressable Memory (TCAM) is a specialized memory that excels in performing parallel search operations across all memory entries within a single cycle. It can identify memory words that match the search data. The architecture of a conventional EX-TCAM array is showcased in Fig. 2(a), structured into an $n \times m$ configuration. Match-lines (MLs) are shared among bit cells and connected to match-line sensing amplifiers (MLSAs). Both bit-lines (BLs) and search-lines (SLs) are shared across all rows of the TCAM array. Read and write operations in the TCAM array are executed like conventional RAM. A search operation is performed simultaneously across the entire array, comprising two phases: (1) pre-charging the ML and (2) asserting the query data on the SLs. TCAM compares the query vector stored in the search data register and the support vectors stored in CAM cells. These support vectors typically manifest as either resistance or threshold voltages, varying by the device type. The MLSAs evaluate the state (match or mismatch) of the MLs depending on the discharge rate at the end of the comparison cycle.

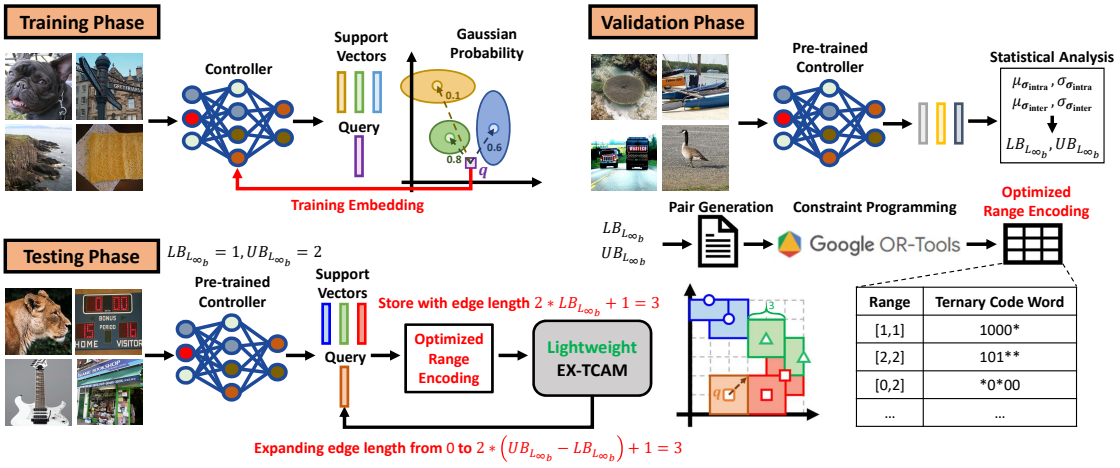

Fig. 3: Overview of the proposed BORE framework.

The CAM cell structure of TCAM (shown in Fig. 2(b)) can be built upon a diverse set of devices, including CMOS SRAM [14] and emerging non-volatile memories (NVMs) (e.g. spin-transfer torque magnetic RAM (STT-MRAM) [15], resistive RAM (ReRAM) [16] and ferroelectric field effect transistor (FeFET) [17]) which offer higher density and search energy efficiency over the CMOS SRAM.

C. Iterative Search with Range Encoding

Prior works [10], [11] have introduced a promising search methodology utilizing EX-TCAM, RENÉ, and an iterative search mechanism for VSS using the L_∞ distance metric. This approach leverages the wildcard state or "don't care" (*) in TCAM, enabling efficient nearest-neighbor searches. The iterative search mechanism transforms a query point into a progressively expanding cube and encodes them by RENÉ to identify the nearest neighbor until a match is detected. Specifically, all support vectors are represented as data points stored in the EX-TCAM. Upon a query q, a sequence of cubes centered on point q with incrementally larger edge lengths is constructed, represented as the cubes filled with colors transitioning from light orange to dark orange in Fig. 1(a). The cube constructed at the t-th iteration aims to check whether the L_∞ distance between query and support vectors is less than or equal to t. If any support points fall within the cube, EX-TCAM signals a "hit." Conversely, if all support points remain outside the cube, TCAM does not signal a "hit," and the search continues. As the range expands, the number of "don't care" bits in the ternary sequences increases, expanding the possibilities of matching ternary sequences, as exemplified by the code words presented in Fig. 1 (a). Thus, RENÉ can be effectively utilized to iteratively search for the nearest neighbor by performing L_∞ distance with EX-TCAM.

III. BORE: Banded Vector Similarity Search with Optimized Range Encoding

This section presents the processing flow and technical enablers of BORE based on Fig. 3. Initially, during the training phase, BORE employs a distance-based metric to enhance accuracy for L_∞ and L_{∞_b} distance search (subsection III-C).

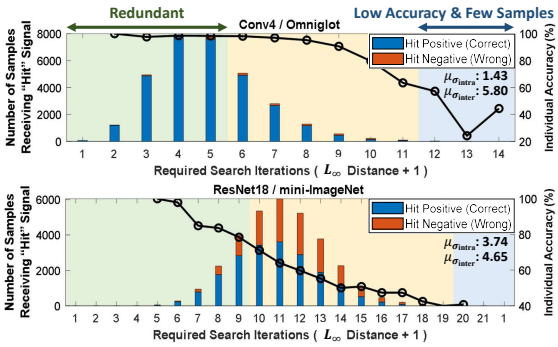

Fig. 4: Hit statistics and accuracy across search iterations on the validation sets, where feature vectors are represented in 5-bit quantization and code word length is 31 bits.

Following model training, BORE leverages statistics derived from validation sets to design the search process (subsection III-A) and range encoding (subsection III-B) to exploit the temporal and spatial redundancies in L_∞ distance search.

A. Banded L_∞ Distance Search (L_{∞_b})

We conducted preliminary experiments to analyze the number of iterations required to receive a "hit" signal (hit count) for each sample, i.e., the nearest neighbor is identified, and VSS is completed. We sampled 32000 data across 1000 different episodes from validation sets. Fig. 4 presents the characteristics of iterations in L_∞ distance search across standard FSL datasets. We observe that the accuracy (represented by the line chart, corresponding to the right y-axis) and hit count (illustrated by the histogram, corresponding to the left y-axis) exhibit different behaviors during these iterations. Accuracy declines as the search proceeds, while the hit count peaks at the fourth and the eleventh iterations for the Omniglot and mini-ImageNet datasets, respectively. Initial iterations (marked in the green background) exhibit higher accuracy than later iterations. This phenomenon is attributed to the fact that the number of required search iterations is one greater than the minimal L_∞ distance between query and support vectors. Thus, initial iterations typically indicate a higher confidence in prediction with lower distance. Moreover, vectors belonging to the same

class are generally closer to each other, which is evidenced by the lower mean value of the intra-class standard deviation ($\mu_{\boldsymbol{\sigma}_{\text{intra}}}$) as compared to the inter-class standard deviation ($\mu_{\boldsymbol{\sigma}_{\text{inter}}}$). These statistical metrics of intra-class ($\boldsymbol{\sigma}_{\text{intra}}$) and inter-class standard deviations ($\boldsymbol{\sigma}_{\text{inter}}$) are all calculated along the dimensions of feature vectors from the validation sets. Conversely, final iterations (marked in the blue background) exhibit fewer hit samples and lower accuracy because query vectors hit in final iterations are dissimilar to all support vectors with larger L_∞ distances between them.

Based on the above analysis, we argue that the middle-stage iterations (marked in the yellow background) play a critical role in the VSS process. Specifically, the initial iteration is to identify the most similar vector from the support vectors of the same class. However, within the context of FSL, it is unnecessary to distinguish the most similar support vector; the primary goal is to predict the correct class. As a result, these initial iterations can be considered redundant and skipped since these vectors can still be correctly classified in later iterations. Similarly, final iterations are also redundant since they have a minimal impact on overall accuracy. Although omitting these final iterations does not significantly affect the average number of search iterations, the reduction in the maximal search range (h_{max}) is particularly advantageous for range encoding, as h_{max} is proven to influence the code word length linearly [9].

According to the above analysis, we propose the banded L_∞ distance search (L_{∞_b}) operating the L_∞ distance search within a specified banded region [$LB_{L_{\infty_b}}$, $UB_{L_{\infty_b}}$] (marked in the yellow background). BORE determines the lower bound ($LB_{L_{\infty_b}}$) and upper bound ($UB_{L_{\infty_b}}$) of L_{∞_b} from statistics of the validation set as Eq. (3) and Eq. (4).

$$LB_{L_{\infty_b}} = \lfloor f_{dis} \times \min(\mu_{\boldsymbol{\sigma}_{\text{intra}}}, \frac{\mu_{\boldsymbol{\sigma}_{\text{inter}}}}{2}) \rfloor \tag{3}$$

$$UB_{L_{\infty_b}} = \lfloor 2 \times f_{dis} \times \min(\mu_{\boldsymbol{\sigma}_{\text{intra}}} + \sigma_{\boldsymbol{\sigma}_{\text{intra}}}, \frac{\mu_{\boldsymbol{\sigma}_{\text{inter}}}}{2}) \rfloor \tag{4}$$

The distribution factor f_{dis} is set as 3.5 to ensure it covers nearly all points from the same class in statistics. The $LB_{L_{\infty_b}}$ is the boundary capturing very similar data within the same distribution, while the $UB_{L_{\infty_b}}$ encompasses data from the same class without extending to other class distributions.

B. Optimized Range Encoding

The existing range encoding RENÉ [9] is initially applied for packet processing in networking applications and only supports point-to-point and point-to-range searches. When utilizing RENÉ to VSS, it becomes necessary to store support vectors as points in TCAM and to iteratively expand the ranges of every dimension in the query vector during L_∞ distance search. Besides, RENÉ is only well-defined when h_{max} is a power-of-2, leading to longer code word lengths in most VSS scenarios.

In L_{∞_b} distance search for VSS, there are two intuitions to reduce the complexity of the encoding. First, since ranges are expanded symmetrically for both L_∞ and L_{∞_b} distance search, only ranges with odd lengths and a few boundary conditions with even lengths are utilized in VSS. Consequently, it becomes possible to eliminate most ranges with even lengths from the domain of the encoder, simplifying the functionality of the range encoding. Secondly, RENÉ cannot fully harness the potential of L_{∞_b}, as it operates vector searches solely within the banded region. For L_{∞_b}, the code word length (CL) required by RENÉ only depends on $UB_{L_{\infty_b}}$, while skipping the initial $LB_{L_{\infty_b}}$ iterations does not contribute to reducing the code word length. Hence, BORE addresses these limitations of RENÉ by eliminating useless ranges and incorporating range-to-range searches, a crucial feature for fully capitalizing on L_{∞_b}. We utilize the OR-Tool [12], a constraint programming (CP) tool developed by Google, to generate the optimized range encoding capable of accommodating all types of searches while minimizing code word length. Below are the detailed steps for constructing the optimized range encoding with the quantization bit-width of vectors as W:

Step 1. Generating all useful pairs for L_{∞_b}: Given that the encoding supports range-to-range searches for L_{∞_b}, the required h_{max} can be reduced from $2 \times UB_{L_{\infty_b}} + 1$ to $2 \times (UB_{L_{\infty_b}} - LB_{L_{\infty_b}}) + 1$. This reduction is achieved by storing support vectors as cubes with edge lengths equal to $2 \times LB_{L_{\infty_b}} + 1$ in the memory. Importantly, this also allows us to skip initial $LB_{L_{\infty_b}}$ iterations. Therefore, we generate all pairs ($\boldsymbol{P}$) from ranges ($\boldsymbol{R}$) with edge lengths less than or equal to h_{max} and eliminate the useless pairs containing even-length ranges which is not corresponding to boundary conditions.

Step 2. Formulate the range encoding as a constrained optimization problem: Every pair $(\boldsymbol{r_1}, \boldsymbol{r_2})$ has a relationship with match or mismatch. Match means that hamming distances between code words of pairs need to be constrained to zero, while those of mismatch pairs need to be larger than zero. Variables $\boldsymbol{V} \in \{0,1\}^{2^W \times CL}$ and $\boldsymbol{M} \in \{0,1\}^{|\boldsymbol{R}| \times CL}$ represent binary code words and wildcard masks. $\boldsymbol{V}$ and $\boldsymbol{M}$ can constitute ternary code words of range r. We use a function $I(r)$ that maps range r into the corresponded index for the encoding module. Without loss of generality, we set $\boldsymbol{V}[0]$ as a zero vector and the ranges starting from the same start-point sp share a binary word $\boldsymbol{V}[sp]$ for reducing the solving space. Thus, it can be formulated as a constrained optimization problem below.

Step 3. Termination: Upon the successful discovery of feasible solutions for $\boldsymbol{V}$ and $\boldsymbol{M}$ by CP, the code word lengths of $\boldsymbol{V}$ and $\boldsymbol{M}$ are reduced by one. Then, BORE repeats to solve the constrained optimization problem in **Step 2.** by CP until it becomes unfeasible. Subsequently, the optimized range encoding with the minimal code word length can be solved.

$$\begin{aligned} \text{Find} \quad & \boldsymbol{V}, \boldsymbol{M} \\ \text{Subject to} \quad & HD(\boldsymbol{V}[sp_1] * \boldsymbol{M}[I(r_1)], \boldsymbol{V}[sp_2] * \boldsymbol{M}[I(r_2)]) \\ & \begin{cases} = 0, & (\boldsymbol{r_1}, \boldsymbol{r_2}) \in \boldsymbol{P_{match}} \\ > 0, & (\boldsymbol{r_1}, \boldsymbol{r_2}) \in \boldsymbol{P_{mismatch}} \end{cases} \end{aligned} \tag{5}$$

C. Distance-based Training Mechanism

The L_∞ distance search suffers from a significant accuracy gap compared to software-based cosine similarity [10], [11]. The discrepancy arises because cosine similarity relies on directional or angular information as Eq. (1), whereas L_∞ distance search leverages distance information as Eq. (2).

TABLE I: Setting and statistics of Eva-CAM Simulation.

Device	2T2R ReRAM [6]	2FeFET [21]
Technology	40nm	22nm
On/off Ratio	30	$1.6 * 10^5$
Area (μm^2)	7328.450	1698.575
Search Latency (ns)	2.199	1.069
Search Energy (pJ)	5.658	1.934

To effectively bridge this accuracy gap, BORE introduces a distance-based training mechanism inspired by the Variational FSL method [18]. During the training phase, BORE replaces the cosine similarity with the geometric mean of Gaussian probability along dimensions as defined in Eq. (6).

$$\ln GAU(\vec{q}, \vec{s_p}, \vec{s_v}) = \frac{1}{d}\sum_{i=1}^{d} \ln\left(\frac{1}{2\pi\sqrt{s_{v_i}}} e^{\frac{-(q_i - s_{p_i})^2}{2 * s_{v_i}}}\right) \quad (6)$$

The Gaussian probability is calculated from the query vector $\vec{q}$, the point value $\vec{s_p}$ and the variance $\vec{s_v}$ of the support vector in a dimension-wise manner. This modification enriches the embedding controllers with distance information, thus enhancing accuracy and generalization on L_∞ and L_{∞_b} distance search.

IV. Experimental Result

A. Settings of Dataset and Simulation

We performed experiments on both small-scale and large-scale FSL tasks to validate the efficacy of BORE. The small-scale task involves Conv4 [2] with 32 dimensions on the Omniglot dataset [13], while the large-scale task includes ResNet18 with 128 dimensions on the mini-ImageNet dataset [2]. The Omniglot dataset contains 1623 classes, with 964 classes in the training set and 659 in the testing set. It consists of characters from worldwide alphabets, with only 20 examples per class. The mini-ImageNet dataset contains 100 classes, with 64 classes in the training, 16 in the validation, and 20 in the testing set. It focuses on diverse objects, with only 600 examples per class. Our experiments employed the standard 32-way 1-shot Omniglot setting, frequently used in prior research [5], [6]. Additionally, we explored a more challenging scenario for mini-ImageNet, the 8-way 4-shot setup from the 5-way 5-shot in other works [20], which involves more classes and fewer shots. To estimate the search energy, we utilized Eva-CAM [19] with 40nm 2T2R ReRAM [6] and 22nm 2FeFET [21] CAM cell models. For all scenarios, we configured the TCAM sizes as 128×32. All simulated statistics are listed in Table I.

B. Pareto Front of Energy-Accuracy Trade-off

We compare the effectiveness of BORE with prior works of EX-TCAM-based VSS. Fig. 5 depicts the Pareto fronts, illustrating the energy-accuracy trade-offs across different datasets with controllers trained with Gaussian probability. The points on the same curve represent the different quantization bits of vectors ranging from 6 bits to 4 bits. BORE pushes the Pareto front to a better trade-off and maximizes energy efficiency while maintaining accuracy.

Notably, RENÉ lacks support for range-to-range searches, so BORE outperforms these blue and red curves by reducing code word length in the spatial domain. It is worth noting that code word length is closely tied to the number of TCAM

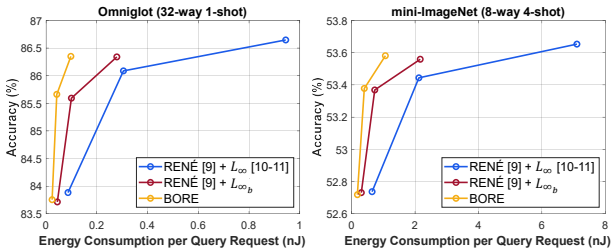

Fig. 5: Energy consumption evaluation using 2T2R ReRAM EX-TCAM with our proposed BORE and prior works of EX-TCAM-based VSS.

TABLE II: Required code word length (CL) across different scenarios.

Encoding	**RENÉ**		**BORE**			
Datasets	**Omniglot / mini-ImageNet**		**Omniglot** (Simple)		**mini-ImageNet** (Difficult)	
	h_{max}	CL	h_{max}	CL	h_{max}	CL
6bit	64	63 bits	19	**22 bits**	29	**31 bits**
5bit	32	31 bits	11	**13 bits**	17	**17 bits**
4bit	16	15 bits	7	**8 bits**	9	**9 bits**

arrays employed for query vector searches, with AND gates as the connection between all EX-TCAMs to simulate parallel search operations. An intriguing observation from Table II is that BORE exhibits the capability to adjust code word length based on the difficulty of datasets. For instance, BORE substantially reduces code word length, ranging from 53% to 68%, for the Omniglot (simple). In contrast, for the mini-ImageNet (difficult), the reduction falls within 40% to 51%. This adaptability underscores the effectiveness of BORE in optimizing performance across different dataset complexities.

C. Analysis of Different Search Methods

To provide a clear understanding of the advantages and specific characteristics of L_{∞_b} distance search, we conduct a detailed analysis of accuracy and hit counts in both scenarios. In Fig. 6, the consistently high cumulative accuracy of L_{∞_b} observed in the first iteration of BORE compared with the sixth iteration and the ninth iteration for Omniglot and mini-ImageNet, respectively. Besides, the convergence to the same average accuracy level in the final iterations indicates that L_{∞_b} has a limited impact on accuracy. Additionally, L_{∞_b} shortens the average search process by $1.94\times$ to $3.39\times$.

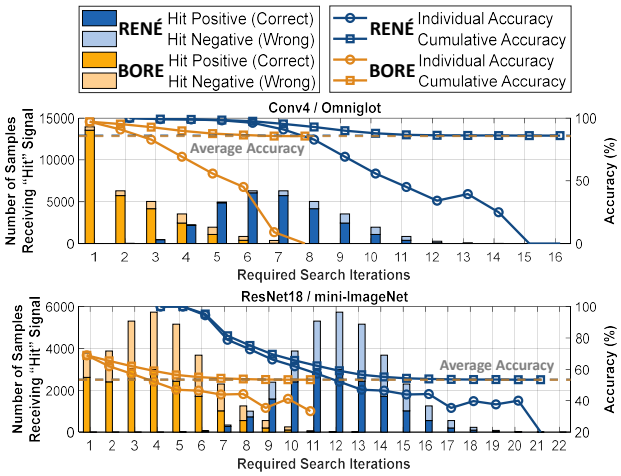

Fig. 6: Comparison of hit counts and accuracy between search mechanisms on testing set in 5-bit quantization. These two scenarios are equipped with the same controllers trained with distance-based metrics.

TABLE III: Accuracy and search iterations between training metrics.

Datasets	Omniglot		mini-ImageNet	
Software Inference (FP-32)				
Training Metrics	Cosine Similarity	Gaussian Probability	Cosine Similarity	Gaussian Probability
Cosine	91.16%	89.03%	62.01%	58.20%
Gaussian	–	90.54%	–	56.46%
Hardware Implementation (RENÉ + EX-TCAM)				
L_∞	83.40% (-7.76%)	86.65% **(-2.38%)**	48.89% (-13.12%)	53.65% **(-4.55%)**
Iterations	12.97	**10.60**	26.10	**19.93**

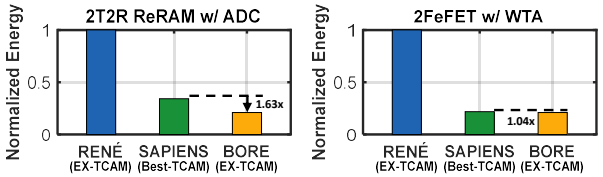

Fig. 7: Comparison with prior works of Best-TCAM-based VSS.

D. Comparisons between Different Training Metrics

From Table III, albeit high accuracy produced by cosine similarity in the software inference, significant accuracy drops exist between L_∞ and cosine similarity, e.g., 7.76% and 13.12% on Omniglot and mini-ImageNet. Gaussian probability classifies data correlated with distance and cluster data in the same class. Although training MANN with Gaussian probability cannot produce the best accuracy in the software inference, the controller trained with Gaussian probability can perform higher accuracy by 3.25% to 4.76% and reduce search iterations by 1.22× to 1.31× for hardware implementation of efficient L_∞.

E. Comparison with Best-TCAM-based VSS

The comparison between BORE and different Best-TCAM-based VSS approaches using analog-to-digital converters (ADCs) [5], [6] and winner-take-all (WTA) sensing circuits [7], [8] is presented in Fig.7. SAPIENS [6] conducts VSS using 2T2R ReRAM Best-TCAM with ADCs, which are costly regarding energy. We also re-implement the SPAIENS [6] using 2FeFET Best-TCAM with WTA proposed by Ref. [8], demonstrating moderate energy efficiency. The results show that BORE outperforms Best-TCAM employing ADCs or WTA, achieving a 1.63× and 1.04× improvement in efficiency, respectively. These improvements are observed using the same devices and CAM array structures as in previous works. It is essential to note that the comparison assumes that Best-Match TCAM with WTA can fully utilize parallelism without experiencing the insufficient sensing margin issue conservatively. This assumption may lead to potentially higher practical energy consumption than the simulated statistics in this comparison.

V. Conclusion

In this paper, we propose BORE to leverage the conventional EX-TCAM architecture effectively. Apart from the prior works, BORE reduces the search complexity from both temporal and spatial domains by simplifying the search process and encoding specialized for L_∞ distance search in VSS. Furthermore, BORE incorporates a distance-based training mechanism, leading to better trade-offs between accuracy and energy consumption. Experimental results show that BORE can improve the energy efficiency by 9.35× to 11.84× and the accuracy by 2.95% to 4.69% compared with prior works of EX-TCAM-based VSS utilizing RENÉ to perform L_∞ distance search. In comparison with Best-TCAM-based works, we also enhance the energy efficiency by 1.04× to 1.63×.

Acknowledgment

This work was supported by the National Science and Technology Council of Taiwan under Grants of MOST 111-2218-E-002-018-MBK and NSTC 112-2218-E-002-025-MBK. The first two authors are also sponsored by the Novatek Ph.D. Fellowship program.

References

[1] Santoro, Adam, *et al.*, "Meta-learning with memory-augmented neural networks," *Int. Conf. Machine Learning (ICML)*, pp. 1842–1850, 2016.
[2] Vinyals, Oriol, *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, 2016.
[3] Ranjan, Ashish, *et al.*, "X-mann: A crossbar based architecture for memory augmented neural networks," *IEEE/ACM Design Autom. Conf. (DAC)*, pp. 1–6, 2019.
[4] Hu, Han-Wen, *et al.*, "ICE: An Intelligent Cognition Engine with 3D NAND-based In-Memory Computing for Vector Similarity Search Acceleration," *IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, pp. 763–783, 2022. vol. 12, no. 1, pp. 19201, 2022.
[5] Li, Haitong, *et al.*, "One-shot learning with memory-augmented neural networks using a 64-kbit, 118 GOPS/W RRAM-based non-volatile associative memory," *IEEE Symp. VLSI Technology*, pp. 1–2, 2021.
[6] Li, Haitong, et al., "SAPIENS: A 64-kb RRAM-based non-volatile associative memory for one-shot learning and inference at the edge," *IEEE Trans. Electron Devices*, vol. 68, no. 12, pp. 6637–6643, 2021.
[7] M. Imani, *et al.*, "SearcHD: A memory-centric hyperdimensional computing with stochastic training," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Syst.*, vol. 39, no. 10, pp. 2422–2433, 2020.
[8] Kazemi, Arman, *et al.*, "Achieving software-equivalent accuracy for hyperdimensional computing with ferroelectric-based in-memory computing," *Scientific reports*, vol. 12, no. 1, pp. 19201, 2022.
[9] Bremler-Barr, Anat, *et al.*, "Encoding Short Ranges in TCAM Without Expansion: Efficient Algorithm and Applications," *IEEE/ACM Trans. Networking (TN)*, vol. 26, no. 2, pp. 835–850, 2018.
[10] Laguna, Ann Franchesca, Michael Niemier, and X. Sharon Hu, "Design of hardware-friendly memory enhanced neural networks," *IEEE Design, Autom. & Test in Europe Conf. & Exhibition (DATE)*, pp. 1583–1586, 2019.
[11] Laguna, Ann Franchesca, *et al.*, "Ferroelectric FET based in-memory computing for few-shot learning," *Great Lakes Symp. VLSI (GLSVLSI)*, pp. 373–378, 2019.
[12] Laurent Perron and Vincent Furnon, OR-Tools v9.7, https://developers.google.com/optimization/.
[13] Lake, Brenden, *et al.*, "One shot learning of simple visual concepts," *the annual meeting of the cognitive science society*, vol. 33, no. 33, 2011.
[14] K. Nii, *et al.*, "13.6 a 28nm 400mhz 4-parallel 1.6gsearch/s 80mb ternary cam," *IEEE Int. Solid-State Circuits Conf. Digest of Technical Papers (ISSCC)*, pp. 240–241, 2014
[15] W. Kang, *et al.*, "In-memory processing paradigm for bitwise logic operations in STT–MRAM," *IEEE Trans. Magnetics (TMAG)*, vol. 53, no. 11, pp. 1–4, 2017
[16] D. Fujiki, *et al.*, "In-memory data parallel processor," *ACM SIGPLAN Notices*, vol. 53, no. 2, pp. 1–14, 2018
[17] X. Yin, *et al.*, "Design and benchmarking of ferroelectric fet based tcam," *IEEE Design, Autom. & Test in Europe Conf. & Exhibition (DATE)*, pp. 1448–1453, 2017
[18] Zhang, Jian, *et al.*, " Variational few-shot learning," *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1685–1694, 2019.
[19] Liu, Liu, *et al.*, "Eva-cam: a circuit/architecture-level evaluation tool for general content addressable memories," *IEEE Design, Autom. & Test in Europe Conf. & Exhibition (DATE)*, pp. 1173–1176, 2022.
[20] Kazemi, Arman, *et al.*, "A flash-based multi-bit content-addressable memory with euclidean squared distance," *IEEE/ACM Int. Symp. Low Power Electronics and Design (ISLPED)*, pp. 1–6,12021.
[21] X. Yin, *et al.*, "An Ultra-Dense 2FeFET TCAM Design Based on a Multi-Domain FeFET Model," *IEEE Trans. Circuits and Syst. II (TCAS-II)*, vol. 66, no. 9, pp. 1577–1581, 2019.