Segmented Angular Pre-Processing for Accurate and Efficient In-Memory Vector Similarity Search

Chi-Tse Huang[1], Jen-Chieh Wang[1], Hsiang-Yun Cheng[2], and An-Yeu (Andy) Wu[1]
[1] Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan
[2]Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
rickhuang@access.ee.ntu.edu.tw; jack@access.ee.ntu.edu.tw; hycheng@citi.sinica.edu.tw; andywu@ntu.edu.tw

2025 62nd ACM/IEEE Design Automation Conference (DAC) | 979-8-3315-0304-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/DAC63849.2025.11133320

Abstract*—Vector similarity search (VSS) is a fundamental operation in modern AI applications, including few-shot learning (FSL) and approximate nearest neighbor search (ANNS). Cosine similarity is widely regarded as the optimal metric for VSS. However, VSS incurs substantial energy and computational overhead, primarily due to frequent vector transfers and the complexity of cosine similarity calculations in high-dimensional spaces. Prior research has explored the use of ternary content addressable memories (TCAMs) for parallel in-memory VSS to reduce vector movement. Exact-Match TCAM (EX-TCAM) enables exact bit-matching, and Best-Match TCAM (Best-TCAM) supports Hamming distance calculations, both of which are spatial metrics and computationally efficient. As a result, existing TCAM-based VSS approaches have focused on developing frameworks to efficiently support more complex spatial metrics such as the L_∞ and L_1 norms. However, these spatial metrics exhibit notable discrepancies compared to angular metrics like cosine similarity. To overcome this limitation, we propose Seg-Cos, a TCAM-based framework that directly approximates cosine similarity within TCAM for angular VSS. Seg-Cos introduces a dedicated pre-processing technique and encoding scheme that segments vectors and encodes them as circular ranges based on their angles and magnitudes. Seg-Cos is the first angular VSS framework compatible with both EX-TCAM and Best-TCAM, enabling accurate and energy-efficient VSS in the angular domain. Simulation results demonstrate that Seg-Cos improves energy efficiency by 1.41× and achieves up to 2.2% higher accuracy over prior EX-TCAM-based methods in FSL. In ANNS, Seg-Cos enhances recall rate by 10% to 52% and improves energy efficiency by 2× compared to previous Best-TCAM approaches with L_1 norm.

I. Introduction

Vector similarity search (VSS) serves as a foundational operation in modern artificial intelligence (AI) applications, including few-shot learning (FSL) [1], [2] and approximate nearest neighbor search (ANNS) [3]. These applications leverage VSS to address challenges such as catastrophic forgetting in FSL [1], [2] and hallucinations in generative AI [4]. By comparing vectors in high-dimensional spaces, VSS enables critical tasks, including classifying novel instances and retrieving relevant information from large databases. A key metric for these tasks is cosine similarity, which effectively measures angular relationships between vectors.

Despite its efficacy, the deployment of VSS is hindered by the substantial energy consumption [5], [6] associated with the frequent transfer of vectors between off-chip memories and processing units in traditional von Neumann systems. In pursuit of energy-efficient solutions in VSS, researchers have been exploring in-memory search (IMS) [7]–[14]. IMS aims to minimize vector movement and accelerate the VSS process by performing parallel searches directly within memory. Recently, two primary IMS-based VSS approaches have been developed using ternary content-addressable memories (TCAMs): 1) Exact-Match TCAM (EX-TCAM) [7]–[10] and 2) Best-Match TCAM (Best-TCAM) [11]–[14]. EX-TCAM locates stored vectors that exactly match a given query vector, whereas Best-TCAM extends EX-TCAM by enhancing its sensing circuitry to identify the closest stored vector. As a result, EX-TCAM achieves significantly lower power consumption per search request compared to Best-TCAM. EX-TCAM-based VSS operates in an iterative manner to support the L_∞ norm [7]–[10] using range encoding [7], as illustrated in the left part of Fig. 1(a). On the other hand, Best-TCAM-based VSS [11]–[14] enables single-pass identification of the nearest vector. Best-TCAM originally supports the Hamming distance [11], [12] and extends to the L_1 norm with thermometer encoding [13], [14], as shown in the right part of Fig. 1(a).

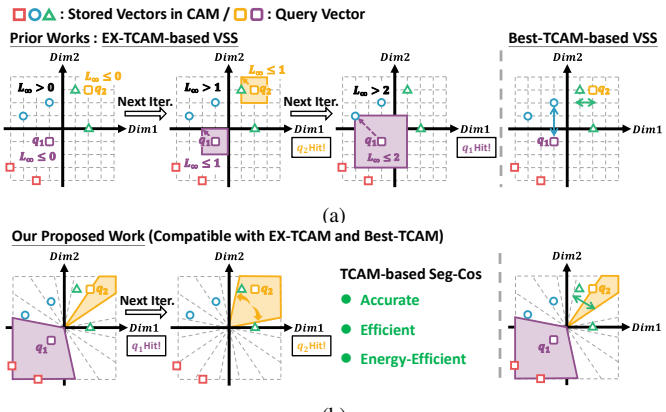


Fig. 1. Comparison between (a) TCAM-based prior works with spatial metrics and (b) the proposed work for angular similarity metrics

While TCAM-based VSS offers advantages, existing frameworks are primarily designed for spatial metrics. In contrast, cosine similarity, an angular metric widely recognized as optimal for software-based VSS [1], [2], achieves superior accuracy compared to TCAM-based VSS [8]–[11], which relies on spatial metrics. To bridge this gap, prior research [10] has explored fine-tuning embedding models for spatial metrics. However, this approach incurs significant training overhead, making them impractical for many real-world applications. Thus, enabling the execution of angular similarity directly within TCAM has become a critical issue. Integrating

angular measurements into TCAM-based VSS presents three key design challenges. First, the underlying metric of TCAM-based VSS is Hamming distance, an additive metric where each dimension contributes independently. In contrast, cosine similarity evaluates the angular relationship through a multiplicative interaction. This difference makes it challenging to support angular similarity with TCAM. Second, cosine similarity is more sensitive to dimensions with larger absolute values, whereas TCAM-based VSS treats all dimensions equally, failing to account for non-uniform effects. Third, existing encoding schemes are designed for spatial metrics, which cannot accommodate the circular distance calculations necessary for angular measurements.

To overcome the above challenges, we propose Seg-Cos, a training-free framework for TCAM-based angular VSS without requiring modifications to existing hardware. Seg-Cos introduces a novel segmented cosine similarity metric that integrates seamlessly into existing TCAM-based VSS. Our approach optimizes quantization, vector representation, and encoding schemes, thereby improving accuracy by 2.2% in FSL, recall by 10% to 52% in ANNS, and energy efficiency by 1.41× in FSL and 2× in ANNS compared with prior TCAM-based VSS supporting L_∞ norm and L_1 norm. The key contributions of this work are as follows:

1) **Developing Segmented Cosine Similarity based on the Re-interpretation of Hamming Distance:** Seg-Cos incorporates two-dimensional cosine similarity into the flow of Hamming distance, making it the first TCAM-based angular VSS framework compatible with both EX-TCAM and Best-TCAM.
2) **Incorporating Characteristics of Cosine Similarity into Range Generation:** Seg-Cos develops magnitude-aware range generation to account for the non-uniform effects in cosine similarity to enhance the functionality and accuracy of TCAM-based Seg-Cos.
3) **Circular Encoding with Range Representation for Angular Distance Measurement:** Seg-Cos enables circular range representation and distance calculation for TCAM-based VSS and avoids the extra codeword overhead induced by the circular structure.

II. Background

A. Vector Similarity Search (VSS)

Vector similarity search (VSS) is a foundational operation in modern AI applications. These applications represent entities—such as words, sentences, images, or videos—as embeddings, i.e., vectors in a learned feature space. For instance, convolutional neural networks (CNNs) transform images into vector embeddings, while sentence transformers encode textual data as vectors. VSS enables the comparison of high-dimensional vectors to identify vectors that are closest to a query, supporting tasks such as few-shot learning (FSL) and approximate nearest neighbor search (ANNS). In FSL, VSS classifies query data by identifying its closest class. In ANNS, VSS efficiently retrieves the approximately nearest

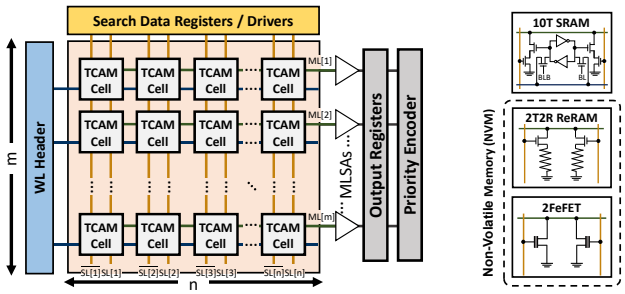


Fig. 2. (a) The TCAM architecture. (b) The configuration of bit cells.

vectors from large-scale databases, which is essential for applications like retrieval-augmented generation (RAG) [4] and recommendation systems. Both cases heavily rely on metrics like cosine similarity $\mathrm{COS}(\vec{q}, \vec{s})$, which measures the cosine of the angle between vectors $\vec{q}$ and $\vec{s}$.

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

While cosine similarity is a standard metric in VSS, quantifying the angular difference imposes a significant computational burden due to complex floating-point operations. Additionally, the frequent transfer of vectors in von Neumann architectures further degrades energy efficiency. As alternatives, computationally efficient spatial metrics, such as L_∞ and L_1 norms, which measure the maximum or summation of difference across dimensions as described in Eq. (2) and Eq. (3), have been proposed to simplify calculations of metrics.

$$\|\vec{q}, \vec{s}\|_\infty = \max_i |q_{[i]} - s_{[i]}|. \tag{2}$$

$$\|\vec{q}, \vec{s}\|_1 = \sum_i |q_{[i]} - s_{[i]}|. \tag{3}$$

To reduce data transfer overhead, researchers have focused on incorporating these computationally efficient spatial metrics into ternary content-addressable memory (TCAM) combined with different encoding schemes.

B. Ternary Content-Addressable Memory (TCAM)

Ternary content-addressable memory (TCAM) is a specialized memory architecture designed to execute parallel search operations rapidly. Unlike traditional random access memory (RAM), which retrieves data based on specific memory addresses, TCAM compares query data against all stored entries simultaneously, identifying matched memory words in a single clock cycle. Every TCAM cell can represent one of three states: '1', '0', or 'X' ("don't care"), with the "don't care" state capable of matching any input voltage, thereby supporting more flexible search functionalities. The architecture of a standard $n \times m$ TCAM array with n columns and m rows is showcased in Fig. 2(a). Match-lines (MLs) are interconnected among bit cells of n-bit words and linked to match-line sensing amplifiers (MLSAs). Search-lines (SLs) are shared across rows of the TCAM array. The search procedure involves the pre-charging phase for MLs and the assertion of the query data onto SLs. TCAM cells that do not align

with the query data (mismatched cells) will discharge more significantly, causing the voltage levels on the MLs to drop. MLSAs then determine the state of the MLs by evaluating the discharge rate and extent of the voltage drops. Exact-Match TCAM (EX-TCAM) utilizes a pair of inverters [15] as sensing circuits for identifying exact match and mismatch. On the other hand, Best-Match TCAM employs more sophisticated sensing mechanisms, such as analog-to-digital converters (ADC) [11], [13], [14] or winner-take-all (WTA) circuits [12]. These components can read out the distances or detect the columns that exhibit the smallest discharge currents.

The cell structure of TCAM (shown in Fig. 2(b)) can be built upon a variety of devices, including CMOS-based SRAM, as well as emerging non-volatile memories (NVMs) (e.g., resistive RAM (ReRAM) [17], ferroelectric field effect transistor (FeFET) [18] and spin-transfer torque magnetic RAM (STT-MRAM) [19]). Emerging NVM-based TCAMs offer superior density and improved search energy efficiency over the CMOS-based SRAM [20], presenting a promising choice for implementing TCAM to enhance energy efficiency.

C. In-Memory Search (IMS)-based VSS

Several studies [7]–[14] have explored VSS using TCAM, focusing on EX-TCAM-based and Best-TCAM-based VSS. IMS-based VSS generally consists of three main components: pre-processing, encoding scheme, and search process. While the pre-processing flow is typically consistent across IMS-based VSS supporting spatial metrics, the key difference between EX-TCAM-based and Best-TCAM-based VSS lies in their search processes and encoding schemes.

The pre-processing flow in IMS-based VSS consists of two stages: quantization and range Generation. In the quantization stage, vectors are converted from floating-point to fixed-point values, typically using uniform quantization. The range generation stage, introduced by BORE [10], processes these fixed-point values to generate ranges, which enhances the pre-processing step with additional functionality.

For EX-TCAM-based VSS [8], [9], upon receiving a query q, the system expands a hyper-cube centered on q until it encompasses a stored vector, as illustrated in Fig. 1(a). This process identifies the vector with the smallest L_∞ norm distance. It utilizes range encoding (e.g., RENÉ [7]) to encode ranges as ternary codewords for threshold matching. Range encoding preserves the correspondence between overlaps of ranges and match outcomes of codewords. The match outcomes between codewords determine if the stored vectors are inside (exact match) or outside (mismatch) the hyper-cube. A "hit" occurs when any stored vector is within the hyper-cube; otherwise, the search continues with expanded ranges.

On the other hand, Best-TCAM-based VSS preforms a one-pass search process, directly comparing the query with stored vectors without iterative range expansion. Best-TCAM-based VSS supports the Hamming distance [11] without encoding. Furthermore, SAPIENS [13], [14] extend the functionality of Best-TCAM by adopting thermometer encoding to calculate the L_1 norm distance based on the Hamming distance.

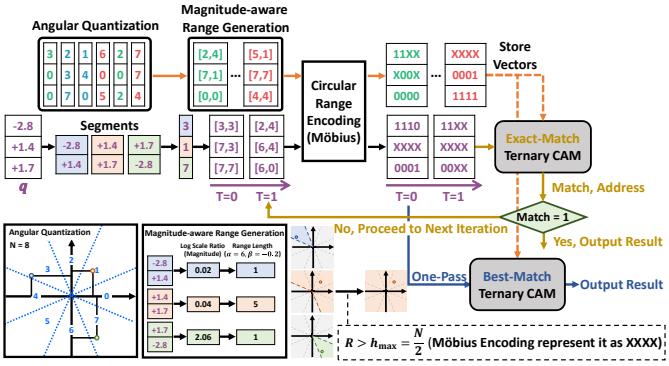
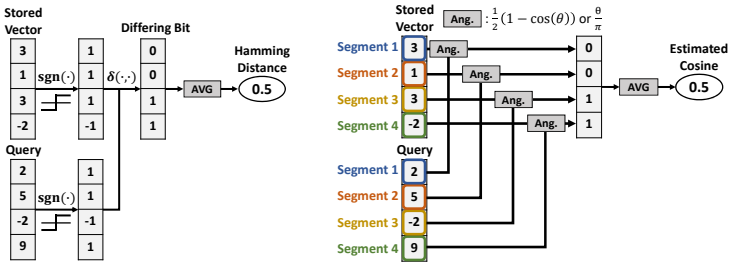


Fig. 3. Processing flow of TCAM-based Seg-Cos



(a) (b)

Fig. 4. (a) Conventional and (b) our perspective to interpret Hamming distance

III. Seg-Cos: Segmented Angular Vector Similarity Search for In-Memory Search

In this paper, we propose segmented cosine similarity, a TCAM-compatible metric based on Hamming distance, to approximate cosine similarity. We also design Seg-Cos, a framework that supports segmented cosine similarity for angular VSS within TCAM-based systems. The processing flow and technical enablers of Seg-Cos are illustrated in Fig. 3. In Section III-A, we define segmented cosine similarity for integrating the angular similarity measurements into TCAM-based VSS. In Section III-B, we develop pre-processing techniques, including angular quantization and magnitude-aware range generation, by leveraging the characteristic of cosine similarity. In Section III-C, we propose a circular encoding that enables circular distance calculations without inducing extra overhead compared to non-circular encoding.

A. Proposed Segmented Cosine Similarity Measurement

To enable efficient angular VSS in TCAM, we design Seg-Cos by reinterpreting the Hamming distance as an angular metric. This insight allow us to simplify the calculation of cosine similarity, make it compatible with TCAM. The Hamming distance is paired with a bipolar sign function $\mathrm{sgn}(\cdot)$ to quantize vectors into 1-bit precision for VSS, as illustrated in Eq. (4a). It measures bit-wise differences and is commonly regarded as a spatial metric, as shown in Fig. 4(a).

$$\mathrm{HAM}(\vec{q},\vec{s}) = \frac{1}{D}\sum_{i=1}^{D}\frac{1}{2}|\mathrm{sgn}(q_i) - \mathrm{sgn}(s_i)| \tag{4a}$$

$$= \frac{1}{D}\sum_{i=1}^{D}\frac{1}{2}(1-\mathrm{COS}(q_i,s_i)) = \frac{1}{D}\sum_{i=1}^{D}\frac{\theta_{q_i,s_i}}{\pi} \tag{4b}$$

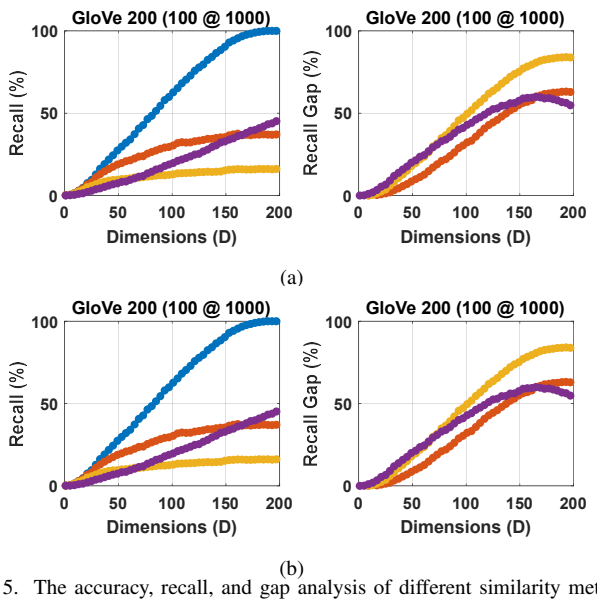


Fig. 5. The accuracy, recall, and gap analysis of different similarity metrics compared with cosine similarity in (a) miniImageNet and (b) GloVe 200

Previous works have explored the encoding and search process for IMS-based VSS to conduct more complex spatial metrics (e.g., L_∞ norm and L_1 norm) based on the Hamming distance measurement. Our motivational experiments, shown in Fig. 5, compare the accuracy and recall of cosine similarity with these spatial metrics across varying dimensions D. As D increases, the gaps between cosine similarity and L_∞ norm, as well as L_1 norm, widen and eventually saturate. In contrast, the gap between cosine similarity and Hamming distance initially grows but later diminishes. This trend indicates that Hamming distance behaves distinctly from other spatial metrics.

To explain this phenomenon, we introduce a novel perspective on interpreting Hamming distance, as depicted in Fig. 4(b). We treat the Hamming distance as a process that divides the D-dimensional vectors $\vec{s} = (s_1, s_2, ..., s_D)$ and $\vec{q} = (q_1, q_2, ..., q_D)$ into D one-dimensional segments, calculate an angular metric, such as cosine distance or angular difference, for each segment pair (s_i, q_i), and then average these values to estimate the angular metrics between the entire vectors, as illustrated in Eq. (4b). Notably, in one-dimensional space, all angular metrics yield the same result and are equivalent up to an affine transformation, as the comparison reduces to a binary distinction: whether the directions of s_i and q_i are the same or opposite. When $D = 1$, the Hamming distance is proportional to the cosine similarity. As D increases, discrepancies between cosine similarity and segment-wise angular metrics grow because each segment remains one-dimensional while D expands. At the same time, the number of segments also increases. As a result, initially, the accuracy gap widens as the growing discrepancy outweighs the benefits of additional segments. However, with further increases in D, the larger number of segments improves the estimation, reducing the gap between Hamming distance and cosine similarity.

Building on this reinterpretation of the Hamming distance, we consider a general case where each vector $\vec{v}$ is divided into D overlapping segments, each with D_{seg} dimensions, denoted as $\vec{v}_{\text{seg},i} = (v_i, v_{(i+1) \bmod D}, ..., v_{(i+D_{\text{seg}}-1) \bmod D})$. We analyze the cosine similarity in Eq. (1) and express it as a weighted synthesis of segment-wise similarities in Eq. (5). This formulation reveals that segment pairs with larger magnitude products $|\vec{q}_{\text{seg},i}||\vec{s}_{\text{seg},i}|$ contribute more significantly to cosine similarity and are more representative of it.

$$\text{COS}(\vec{q}, \vec{s}) = \sum_{i=1}^{D} \frac{|\vec{q}_{\text{seg},i}||\vec{s}_{\text{seg},i}|}{D_{\text{seg}}|\vec{q}||\vec{s}|} \text{COS}(\vec{q}_{\text{seg},i}, \vec{s}_{\text{seg},i}) \tag{5}$$

However, due to the limited operations supported by TCAM, the weighted aggregation involving coupled coefficients from both the query and stored vectors becomes impractical. To address this, we apply two inequalities (Cauchy–Schwarz and Jensen's inequality) to separate the coupling through a logarithmic transformation, as shown in Eq. (6). This decoupling allows for independent pre-processing while preserving the effectiveness of the metric.

$$\begin{aligned} &\ln\left(1 - \text{COS}(\vec{q}, \vec{s})\right) \\ \geq &\ln\left(\sum_{i=1}^{D} \frac{|\vec{q}_{\text{seg},i}||\vec{s}_{\text{seg},i}|}{D_{\text{seg}}|\vec{q}||\vec{s}|}\left(1 - \text{COS}(\vec{q}_{\text{seg},i}, \vec{s}_{\text{seg},i})\right)\right) \\ \geq &\frac{1}{D}\sum_{i=1}^{D} \ln\left(1 - \text{COS}(\vec{q}_{\text{seg},i}, \vec{s}_{\text{seg},i})\right) - \ln \frac{|\vec{q}||\vec{s}|}{|\vec{q}_{\text{seg},i}||\vec{s}_{\text{seg},i}|} \end{aligned} \tag{6}$$

To simplify the angular computation, we apply the first-order Taylor expansion of $\ln(1-\cos\theta)$ around $\theta = \phi$, as shown in Eq. (7). Since classification results are invariant to scaling and shifting of similarity and distance, we replace $\frac{\sin\phi}{1-\cos\phi}$ with $\frac{1}{\alpha}$ and multiply the entire expression with α, as shown in Eq. (8). To further optimize this metric, we introduce β as a parameter that controls clamping effects in Eq. (9), mitigating the error induced by Jensen's inequality. For the EX-TCAM-based Seg-Cos, we select the MAX aggregation function, as shown in Eq. (10a). For the Best-TCAM-based Seg-Cos, we choose the AVG aggregation function, as depicted in Eq. (10b).

$$\ln\left(1 - \cos\theta\right) \approx \frac{\sin\phi}{1 - \cos\phi}(\theta - \phi) + \ln\left(1 - \cos\phi\right) \tag{7}$$

$$\theta_i(\vec{q}, \vec{s}, \alpha, \beta) = \theta_{\text{seg},i} - \alpha \times (\ln \frac{|\vec{q}|}{|\vec{q}_{\text{seg},i}|} + \ln \frac{|\vec{s}|}{|\vec{s}_{\text{seg},i}|} + \beta) \tag{8}$$

$$\max(\theta_i(\vec{q}, \vec{s}, \alpha, 0), \alpha\beta) = \max(\theta_i(\vec{q}, \vec{s}, \alpha, \beta), 0) + \alpha\beta \tag{9}$$

$$\text{Seg-Cos}_{\text{MAX}}(\vec{q}, \vec{s}) = \max_{i:1 \leq i \leq D} \max(\theta_i(\vec{q}, \vec{s}, \alpha, \beta), 0) \tag{10a}$$

$$\text{Seg-Cos}_{\text{AVG}}(\vec{q}, \vec{s}) = \frac{1}{D}\sum_{i=1}^{D} \max(\theta_i(\vec{q}, \vec{s}, \alpha, \beta), 0) \tag{10b}$$

The choice of D_{seg} is crucial for the Seg-Cos metric. When $D_{\text{seg}} = 1$ and $\alpha = 0$, the metric reduces to the Hamming distance. When $D_{\text{seg}} = D$ and $\alpha\beta \leq 0$, it corresponds to the angular difference between vectors. Although increasing

D_{seg} allows the metric to capture higher dimensional characteristics and improve accuracy, setting D_{seg} beyond two introduces multiple degrees of freedom (DoF) in angular orientation, significantly increasing the complexity of angular representation. Meanwhile, a two-dimensional space is the smallest space capable of representing the full range of angular differences, offering substantial improvements over a one-dimensional space. Therefore, we select $D_{\text{seg}} = 2$ to preserve a single DoF, ensuring both computational simplicity in pre-processing and significant accuracy gains.

B. Pre-processing of IMS-based Segmented Cosine Similarity

The pre-processing flow of vectors in IMS-based VSS consists of two stages: **Quantization** and **Range Generation**. BORE [10] utilized ranges to skip redundant search iterations in EX-TCAM-based VSS. Seg-Cos extends the role of the **Range Generation** stage by embedding logarithmic magnitude ratios in the ranges. These two stages enable Seg-Cos to integrate segmented cosine similarity into TCAM-based VSS, as defined in Eq. (10a) and Eq. (10b), where angular orientations are subtracted and logarithmic magnitude ratios are summed to produce the final similarity measure.

Quantization. To enable the calculations of two-dimensional angular difference within TCAM, we design a quantization method for representing two-dimensional angular orientation. With only a single DoF in two-dimensional space, we use uniform angular divisions for quantization. Specifically, the two-dimensional plane is divided into N equal sections, each spanning an angle of $\frac{2\pi}{N}$ radians. As illustrated in the quantization stage of Fig. 3, the vector is first segmented, and each segment is projected onto the two-dimensional plane. The orientation of each segment is then mapped to one of these angular sections, indexed from 0 to $N-1$. Consequently, this allows us to calculate the two-dimensional angular difference using the circular distance between quantized values.

Range Generation. The length of range for each segment is generated based on the logarithmic ratio of the full vector magnitude to the segment magnitude, as shown in Eq. (11) and Eq. (12). Since the minimum range length is 1, the half-span is clamped to ensure it remains non-negative. In EX-TCAM-based VSS, the query vector must expand for distance measurement, while the stored vectors remain fixed. If the range encoding cannot accommodate this expansion, we introduce γ as a knob to shift part of the range from the query vector to stored vectors in advance. In Best-TCAM-based VSS, where the roles of the query and stored vectors are functionally identical, γ is set to zero. With Eq. (11) and Eq. (12), segment pairs with larger magnitude products have narrower ranges to maintain precision, while pairs with smaller products have wider ranges to reduce their influence on the final similarity measure.

$$R_{\vec{q}}(\vec{v}_{\text{seq},i}) = 2 \times \max(\lfloor \alpha \times (\ln \frac{|\vec{v}|}{|\vec{v}_{\text{seg},i}|} + \frac{\beta}{2}) \rceil - \gamma, 0) + 1 \quad (11)$$

$$R_{\vec{s}}(\vec{v}_{\text{seq},i}) = 2 \times \max(\lfloor \alpha \times (\ln \frac{|\vec{v}|}{|\vec{v}_{\text{seg},i}|} + \frac{\beta}{2}) \rceil + \gamma, 0) + 1 \quad (12)$$

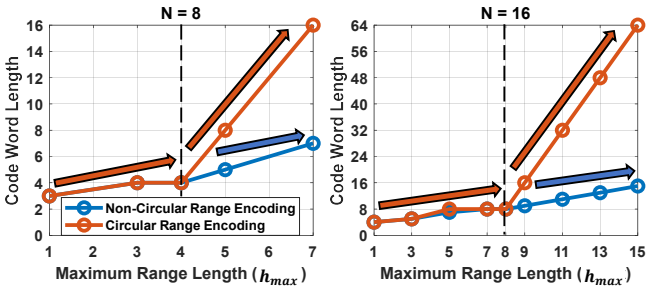


Fig. 6. Codeword length of range encoding solved by CP

C. Encoding for TCAM-based Segmented Cosine Similarity

Previous works have introduced encoding schemes for non-circular distance calculation and range representation, such as RENÉ for value-to-range searches and BORE for range-to-range searches. The codeword length (CL) of them grows linearly with the maximum range length ($h_{\max}$) [7]:

$$CL = \log_2 N + h_{\max} - \log_2 h_{\max} - 1. \quad (13)$$

To develop encoding supporting circular distance calculations for the Seg-Cos framework, we first utilized OR-Tools with constrained programming, extending the approach of BORE to circular range encoding. As shown in Fig. 6, our experiment reveals a key trend: while the codeword length of non-circular range encoding increases linearly with $h_{\max}$, circular range encoding exhibits a two-piece linear growth pattern. The first piece spans $h_{\max}$ from 1 to $\frac{N}{2}$, with CL increasing linearly from $\log_2 N$ (binary encoding) to $\frac{N}{2}$. The second piece extends $h_{\max}$ from $\frac{N}{2}$ to $N-1$, where CL grows linearly from $\frac{N}{2}$ to $\frac{N^2}{4}$ with a steeper slope. Notably, in this second piece, every one-unit increase in $h_{\max}$ results in an extra $\frac{N}{2}$ codewords, which is equivalent to the total codeword length required for the entire first piece. To avoid excessive codeword length overhead for supporting circular range representation, we choose $h_{\max}$ as $N/2$ for the design of circular encoding.

Based on the above analysis, we propose a circular encoding scheme called "Möbius Encoding," designed for $h_{\max} = \frac{N}{2}$. Möbius encoding converts range $[s, e]$ into a ternary codewords $(c_0, c_1, ..., c_{\frac{N}{2}-1})$ as defined in Eq. (14) with $CL = \frac{N}{2}$, where $[i, j]^{\complement}$ denotes the complement range of the range $[i, j]$, expressed as $[(j+1) \bmod N, (i-1) \bmod N]$. The codewords are determined by evaluating the relationship between the range $[s, e]$ and reference half-circle ranges. Specifically, the codeword c_i is determined as follows: $c_i = 0$ if $[s, e]$ lies entirely outside the reference half-circle range, $c_i = 1$ if $[s, e]$ is fully contained within the reference half-circle range, and $c_i = \text{X}$ if range $[s, e]$ overlaps with the reference half-circle range but is not completely contained.

$$c_i = \begin{cases} 0, & \text{if } [s,e] \subset [i+1, (i+\frac{N}{2}) \bmod N]^{\complement} \\ 1, & \text{if } [s,e] \subset [i+1, (i+\frac{N}{2}) \bmod N] \\ \text{X}, & \text{otherwise} \end{cases} \quad (14)$$

TABLE I
SETTING AND STATISTICS OF TCAM SIMULATION.

Platform	Eva-CAM [16]	NeuroSim [27]
TCAM Type	EX-TCAM	Best-TCAM
Device	2FeFET [23]	
Technology	22nm	
On/off Ratio	1.6×10^5	
Area (μm^2)	1698.575	6090.125
Search Latency (ns)	1.069	13.8432
Search Energy (pJ)	1.934	56.715

IV. EXPERIMENTAL RESULTS

A. Settings of Dataset and Simulation

We conducted experiments on FSL tasks to assess the effectiveness of EX-TCAM-based Seg-Cos. Our small-scale experiments utilize Conv4 [2] with 32 dimensions on the Omniglot dataset [17]. For the large-scale experiments, we employed ResNet18 [21] with 128 dimensions on the mini-ImageNet dataset [2]. Our experiments followed the standard 32-way 1-shot Omniglot setting [7], [8]. For the mini-ImageNet, we ventured into a more challenging 8-way 4-shot configuration [22]. For our experiments on ANNS, we aimed to evaluate the effectiveness of the Best-TCAM-based Seg-Cos with GloVe [24] and NYTimes [25]. The GloVe dataset consists of millions of word embeddings, each with dimensions of either 100 or 200. The NYTimes dataset contains 300,000 embeddings, each with dimensions of 256. The key performance metric for these ANNS experiments is the recall rate, specifically "recall 100@1000," which follows the evaluation setting used in [26], and measures the proportion of the top 100 true neighbors in retrieved 1000 candidates returned by VSS.

To estimate the search energy in our experiments, we utilized the Eva-CAM [16] for EX-TCAM and NeuroSim [27] for Best-TCAM with 22nm 2FeFET [23] TCAM models. The TCAM arrays are configured to 128 × 32 for all scenarios. All simulated statistics are listed in Table I.

B. Energy-Accuracy Trade-off for EX-TCAM VSS in FSL

We compare the effectiveness of EX-TCAM-based Seg-Cos against prior EX-TCAM-based VSS methods [7]–[10] in FSL scenarios, including a low-dimension case (32-way 1-shot omniglot, $D = 32$) and a high-dimension case (8-way 4-shot miniImageNet, $D = 128$). Fig. 7(a) depicts the Pareto fronts, showcasing the energy-accuracy trade-offs with pre-trained controllers for cosine similarity. Each point on the curves corresponds to a different quantization bit-width ranging from 4 bits ($N = 16$) to 6 bits ($N = 64$). Seg-Cos pushes the Pareto front to a better trade-off and maximizes accuracy and energy efficiency in the high-dimension scenario. Among the prior approaches, RENÉ [7]–[9] lacks support for range-to-range searches, making BORE [10] more energy-efficient by reducing iterations and codeword length. While EX-TCAM-based Seg-Cos exhibits similar trends to BORE [10] in the low-dimensional scenario, it excels by 2.2% accuracy gain in high-dimensional settings, where the increasing number of segments provides better estimations of cosine similarity.

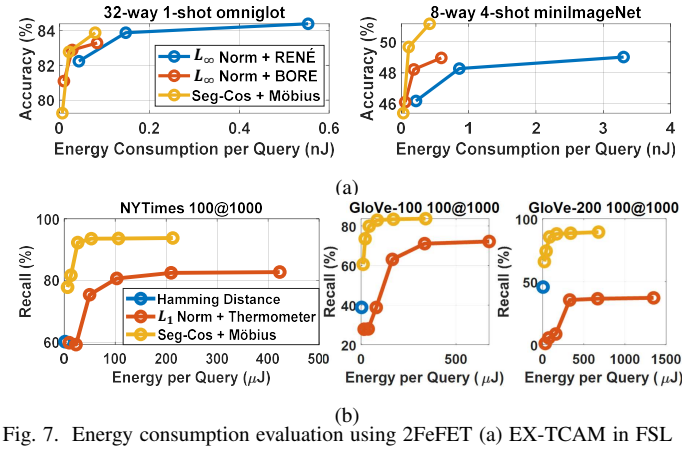


Fig. 7. Energy consumption evaluation using 2FeFET (a) EX-TCAM in FSL and (b) Best-TCAM in ANNS with our proposed Seg-Cos and prior works

C. Energy-Recall Trade-off for Best-TCAM VSS in ANNS

We compare Best-TCAM-based Seg-Cos against prior Best-TCAM-based works with Hamming distance [11] and L_1 norm [13], [14] with thermometer encoding in ANNS scenarios. Fig. 7(b) depicts the Pareto fronts illustrating the energy-recall trade-offs, where cosine similarity serves as the ground truth with a recall rate of 100%. The points on the curves represent the different quantization bits of vectors ranging from 2 bits ($N = 4$) to 7 bits ($N = 512$). Seg-Cos pushes the Pareto front to a better trade-off and enhances the recall rate by 10% to 52% across different datasets compared to Best-TCAM-based L_1 norm implementations [13], [14]. While Seg-Cos extends Hamming distance and consequently consumes more energy, it improves energy efficiency by 2× over Best-TCAM-based VSS with L_1 norm deu to shorter codeword length in Möbius encoding compared to thermometer encoding.

V. CONCLUSION

In this paper, we propose Seg-Cos, a framework that extends TCAM-based VSS to the angular domain, overcoming the limitations of traditional TCAM-based VSS with spatial metrics. Through the use of angular quantization, magnitude-aware range generation, and Möbius encoding, Seg-Cos bridges the gap between TCAM-based VSS and cosine similarity. Our approach integrates seamlessly with existing TCAM-based systems, enabling improvements in accuracy and energy efficiency without hardware modifications. Experimental results demonstrate that Seg-Cos improves energy efficiency by up to 1.41× and 2.2% higher accuracy over prior EX-TCAM works in FSL. In ANNS scenarios, Seg-Cos enhances recall by 10% to 52% and energy efficiency by 2× over prior Best-TCAM works supporting L_1 norm. By addressing the challenges of cosine similarity implementation within TCAM architectures, Seg-Cos represents an advancement in in-memory search.

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 author is sponsored by the Novatek Ph.D. Fellowship program.

REFERENCES

[1] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," *Int. Conf. Machine Learning (ICML)*, pp. 1842–1850, 2016.

[2] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, D, Wierstra, "Matching networks for one shot learning," *Advances in Neural Information Processing Syst.*, 2016.

[3] W. Li, *et al*., "Approximate nearest neighbor search on high dimensional data—experiments, analyses, and improvement," *IEEE Trans. Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1475-1488, 2019.

[4] P. Lewis, *et al*., "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Syst. (NeurIPS)*, pp. 9459-9474, 2020.

[5] A. Ranjan, *et al*., "X-mann: A crossbar based architecture for memory augmented neural networks," *IEEE/ACM Design Autom. Conf. (DAC)*, pp. 1–6, 2019.

[6] H.-W.Hu, *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.

[7] Bremler-Barr, Anat, *et al*., "Encoding Short Ranges in TCAM Without Expansion: Efficient Algorithm and Applications," *IEEE/ACM Trans. Netw. (TN)*, vol. 26, no. 2, pp. 835–850, 2018.

[8] 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.

[9] Laguna, Ann Franchesca, *et al*., "Ferroelectric FET based in-memory computing for few-shot learning," *Great Lakes Symp. VLSI (GLSVLSI)*, pp. 373–378, 2019.

[10] Huang, Chi-Tse, *et al*., "BORE: Energy-Efficient Banded Vector Similarity Search with Optimized Range Encoding for Memory-Augmented Neural Network," *IEEE Design, Autom. & Test in Europe Conf. & Exhibition (DATE)*, 2024.

[11] Karunaratne, Geethan, *et al*., "Robust high-dimensional memory-augmented neural networks.," *Nature communications*, vol. 12, no. 1, pp. 2468, 2021.

[12] 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.

[13] 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.

[14] 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.

[15] Yin, Xunzhao, *et al*., "Ferroelectric ternary content addressable memories for energy-efficient associative search," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 4, pp. 1099–1112, 2022.

[16] 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.

[17] D. Fujiki, *et al*., "In-memory data parallel processor," *ACM SIGPLAN Notices*, vol. 53, no. 2, pp. 1–14, 2018.

[18] 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 .

[19] 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.

[20] Li, Shuangchen, *et al*., "Nvsim-cam: a circuit-level simulator for emerging nonvolatile memory based content-addressable memory," *IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, vol. 13, no. 1, pp. 1-7, 2016.

[21] Kaiming, He, *et al*., "Deep residual learning for image recognition," *Proceedings of the IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1685–1694, 2016.

[22] 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.

[23] 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.

[24] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning., "Glove: Global vectors for word representation," *Conf. empirical methods in natural language processing (EMNLP)*, pp. 1532-1543, 2014.

[25] Bache, K. & Lichman, M. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013

[26] Lee, Yejin, *et al*., "Anna: Specialized architecture for approximate nearest neighbor search." *IEEE Inter. Symp. High-Performance Computer Architecture (HPCA)*, pp. 169-183, 2022.

[27] Peng, Xiaochen, *et al*., "DNN+NeuroSim: An end-to-end benchmarking framework for compute-in-memory accelerators with versatile device technologies," *IEEE Inter. Electron Devices Meeting (IEDM)*, pp. 1532-1543, 2019.