www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 14 Issue 06 June 2025, Page No. 27355-27361
ISSN: 2319-7242 DOI: 10.18535/ijecs/v14i06.5163

Architectural Features of Extended Retrieval Generation with External Memory

Gartman Ievgen

CEO and founder, Bridge.Digital
Austin, TX, USA

Abstract
This article examines the RoCR framework, a Retrieval-Augmented Generation (RAG) system optimized for edge deployment in latency-sensitive environments such as real-time search, product recommendation, and dynamic content generation in eCommerce platforms. RoCR leverages Compute-in-Memory (CiM) architectures to enable fast, energy-efficient inference at scale. At the core of the solution is the CiM-Retriever, a module optimized for performing max inner product search (MIPS). Two architectural variants of the generator are analyzed—decoder-only (RA-T) and encoder–decoder with kNN cross-attention—both demonstrating improved accuracy across various tasks while maintaining scalability to millions of documents. The aim of this study is to analyze the architectural characteristics of RAG systems enhanced with external memory modules, focusing on their applicability to eCommerce-scale tasks requiring sub-second response times and contextual relevance. The methodology is based on a review of recent scientific publications, enabling an in-depth exploration of the system-level design of RAG solutions leveraging memory augmentation. The insights from this analysis will be particularly relevant to AI practitioners and system architects working on scalable, high-performance retrieval systems for domains such as personalized retail, product search, and dynamic user engagement optimization. Moreover, the results are of interest to hardware-software co-design specialists and architects of scalable distributed platforms focused on integrating external memory modules in the context of cognitive and neural network applications.

Keywords: Retrieval-Augmented Generation, Compute-in-Memory, Edge LLM, noise-aware training, contrastive learning, external memory, non-volatile crossbars.

1. Introduction

In recent years, interest in deploying large language models (LLMs) on edge devices has grown considerably, driven by increasing demands for data privacy and low-latency processing. Traditional fine-tuning methods based on gradient descent and parameter updates are often too resource-intensive for edge environments—even when using advanced accelerators. For instance, training a mid-sized model with Pocketngine requires approximately 90 hours [1].

In response, the Retrieval-Augmented Generation (RAG) approach has gained traction. RAG enhances generation quality by dynamically retrieving contextually relevant information, a critical capability for serving up-to-date product data, user reviews, or FAQs in eCommerce systems without re-training the underlying LLM [2]. However, applying RAG at the edge introduces two major challenges: high retrieval latency and scalability bottlenecks as profile data accumulates [1].

The purpose of this article is to examine the architectural characteristics of extended RAG systems that incorporate external memory for edge-based deployment.

The scientific novelty of this research lies in a structured architectural analysis of the RoCR framework, designed to accelerate Retrieval-Augmented Generation on edge devices equipped with Compute-in-Memory (CiM) architectures. This includes evaluating the effectiveness of the CiM-Retriever, methods for contrastive and noise-aware embedding training, the reshape module, and two generator

variants (decoder-only and encoder–decoder with kNN cross-attention). The analysis considers throughput, latency, energy efficiency, and scalability to millions of documents.

The core hypothesis posits that contrastive embedding training with simulated CiM noise—combined with noise-aware optimization and a reshape module—can produce embeddings that preserve relevance order under max inner product search (MIPS) on CiM hardware. This would enable low-latency, highly scalable RAG systems that approach cloud-level performance.

The study methodology is based on a comparative review of existing research in this area.

2. Literature Review

The body of literature on the architectural characteristics of extended retrieval generation with external memory can be broadly divided into three thematic groups. The first includes applied research on the development of retrieval-augmented generative systems across different domains [1, 2, 3]. The second focuses on foundational work in retrieval and representation methods underpinning such architectures [4, 5, 6, 9]. The third group comprises studies on model interpretability and user interaction within extended retrieval systems [7, 8, 10].

A retrieval-augmented generation engine for landscape design, introduced by Shelby L. and da Silva R. V. M. A. [1], combines vector-based retrieval from an external design database with a generative model that produces design solutions based on retrieved content. The authors emphasize semantic alignment between the query and retrieved data. A similar approach is explored by Sarto S. et al. [2] in automatic image captioning, where external memory is used to store visual features and key descriptive fragments. These are then fed into a transformer-based generative network, improving both the accuracy and diversity of the captions. Qin R. et al. [3] demonstrate how such retrieval-augmented systems can be implemented in heterogeneous edge computing-in-memory architectures through hardware–software co-design optimized for high throughput and energy-efficient memory access.

Foundational studies provide the methodological basis for these applied systems. Radford A. et al. [4], in their work on CLIP, show that contrastive learning of multimodal embeddings yields transferable visual models suitable for downstream retrieval-augmented generation tasks. Malkov Y. A. and Yashunin D. A. [5] introduce hierarchical navigable small world graphs (HNSW) for approximate nearest neighbor search, enabling sub-millisecond retrieval times in large vector indexes applicable to RAG systems with external memory. Gao T., Yao X., and Chen D. [6] demonstrate improvements in semantic alignment using simple augmentation and contrastive negation techniques. A comprehensive survey on open-domain QA by Zhu F. et al. [9] synthesizes current "retrieve-and-read" approaches, focusing on hybrid sparse-dense retrieval architectures, ranking, and reading mechanisms—all directly relevant to external-memory RAG models.

Research on interpretability and user experience complements technical development. Cornia M., Baraldi L., and Cucchiara R. [7] present an empirical analysis of transformer-based image captioning models, using attribution and attention visualization techniques to examine how different memory layers influence final outputs. Qiu Y. et al. [8] explore how professional knowledge in landscape architecture can be abstracted through digital platforms, analyzing mechanisms of access, application, and distribution—elements that could be incorporated into the external memory of generative systems. Fruchard B. et al. [10] study user preferences and efficiency in image tagging and browsing, which has implications for the interface design of systems where retrieval and memory need to be presented intuitively.

Overall, the literature reveals a wide range of approaches, from domain-specific retrieval-augmented generation systems to core algorithmic advances in retrieval and interpretability. However, several tensions emerge. On the one hand, the performance and energy efficiency focus of edge architectures [3] often conflicts with the needs of semantically rich retrieval tasks, which rely on complex embedding models [4, 6]. On the other hand, while interpretability-focused methods [7] are valuable, they are rarely integrated into deployed retrieval-generation pipelines using external memory. There is also limited discussion of dynamic memory updates based on user feedback [10] and the management of professional ontological content [8].

In describing practical examples of companies leveraging external memory to augment search-result generation, sources [11–13] were employed, with all information drawn from the organizations' official websites.

Moreover, the field currently lacks unified metrics that jointly assess retrieval efficiency, generative accuracy, and usability—hindering comparative analysis and slowing the development of more robust and adaptive systems.

3. Components and Architectural Levels of RAG with External Memory

The retriever module is responsible for identifying the most relevant documents from external profile memory based on the current user query—without updating LLM parameters. The underlying architecture is typically built on models like BM25 or BERT, which are used to search large datasets for relevant matches. Figure 1 illustrates the components of a RAG architecture that integrates external memory.

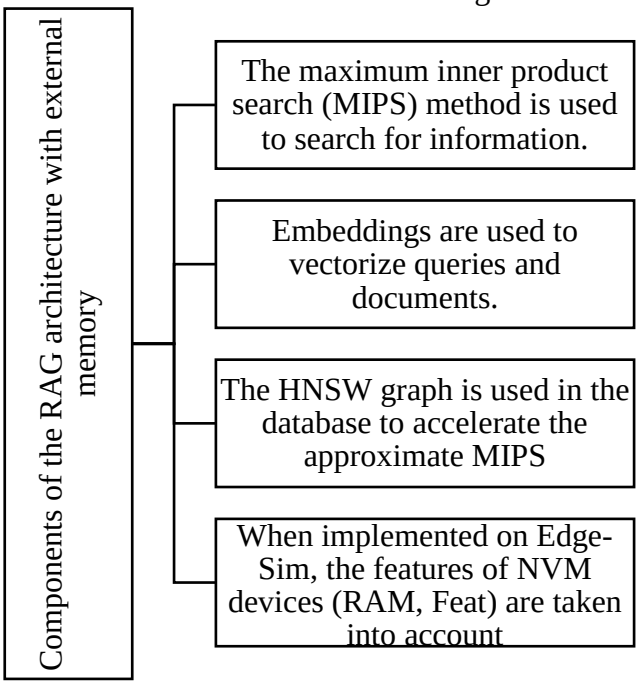

Figure 1. Components of the RAG architecture with external memory [1]

A closer look at the architecture components in Figure 1 reveals that information retrieval is performed using max inner product search (MIPS). For each query x and document d, the dot product of their embeddings $E(x)\cdot E(d)$ is computed, and the top-k documents with the highest scores are selected [1].

Embeddings are generated using sentence embedding models (e.g., all-MiniLM-L6-v2) for textual inputs and CLIP encoders for visual inputs, with spatial vector averages used for aggregation [1, 2].

To accelerate approximate MIPS over millions of profile documents, the system employs a Hierarchical Navigable Small World (HNSW) graph, which provides amortized query time of approximately O(log N) based on database size.

When implemented on Edge-CiM platforms, the system accounts for the specific characteristics of non-volatile memory (NVM) devices such as RRAM or FeFET, including noise levels (σ_v), int8 precision constraints, and 64×64 crossbar dimensions. The retriever module stores the index directly in NVM and performs MIPS in-memory, minimizing data transfer between processing units and memory [2, 8].

The generator (LLM) can be deployed in two architectural variants. In the RA-T (decoder-only) model, retrieved tokens are filtered to remove stop words and duplicates, then prepended to the decoder input sequence. A self-attention mechanism processes both the retrieved tokens and newly generated words. In the RA-T (encoder–decoder) model, retrieved documents are first encoded by a separate Transformer encoder. During generation, each decoder layer performs both self-attention over the generated text and kNN-based cross-attention over the retrieved representations. These outputs are combined via a learnable gate $\alpha \in (0, 1)$ [1].

Table 1 compares the key components of RA-T (decoder-only) and RA-T (encoder–decoder) architectures.

Table 1. Comparison of RA-T (decoder-only) and RA-T (encoder–decoder) architecture components [1, 2]

Component	**RA-T (decoder-only)**	**RA-T (encoder–decoder)**

Component	RA-T (decoder-only)	RA-T (encoder–decoder)
Input	[⟨SOS⟩, retrieved, $y_{1:t-1}$]	Query $y_{1:t}$ and hidden states of retrieved after encoding
Self-attention	Over entire sequence (retrieved + generated)	Over $y_{1:t}$
Cross-attention	–	kNN cross-attention over retrieved embeddings, gated with α
Segment embeddings	Two types: for retrieved and generated tokens	Positional + separate retrieved branch
Training	Cross-entropy → Reinforcement Learning (CIDEr)	Cross-entropy → Reinforcement Learning (CIDEr)

(retrieved: stop-word and duplicate-filtered retrieved token set)

The two-stage training protocol begins with cross-entropy minimization. In each iteration, the decoder is fed the ground-truth prefix token sequence and learns to minimize the negative log-likelihood of the correct tokens, conditioned on the previous tokens, retrieved information, and input image. In the second stage, a self-critical reinforcement learning method is applied: beam search generates candidate hypotheses, rewards are computed via CIDEr scores, and the loss gradient is scaled by the difference between the actual and average rewards multiplied by the log probability gradient of the generated output.

During the cross-entropy stage, Gaussian noise with zero mean and dataset-specific variance is added to the document and query embeddings. This noise-aware optimization increases robustness on Edge-CiM hardware by simulating the noise characteristics of non-volatile memory [4].

Contrastive learning is used to construct discriminative vector representations. If explicit labels are available, triplets are formed directly. Without labels, stochastic techniques such as varying dropout levels are used to create positive and negative pairs. In both cases, a triplet loss function encourages the model to bring embeddings of positive pairs closer and push negative ones apart, using a margin-based objective [5, 7].

The embedding transformation module is implemented as an autoencoder that converts 32-bit vectors into an int8 format suitable for in-memory computing on 64×64 CiM matrices, preserving data quality while reducing hardware costs.

In summary, the RoCR architecture combines a high-efficiency document retriever, a flexible LLM decoder with external memory integration, and advanced training strategies. This combination enables fast and accurate search across millions of documents, scalability for large-scale data environments, and resilience to hardware-level noise on Edge-CiM platforms.

4. Hardware-Software Integration of External Memory

Efficient deployment of the RAG framework on edge devices requires tight integration between software components (retriever and generator) and the MIPS accelerator implemented using Compute-in-Memory (CiM) architectures with non-volatile memory (NVM).

In Compute-in-Memory systems, weight matrices are stored directly in the crossbars of non-volatile memory (e.g., PCM, RRAM, FeFET), at the intersection of rows and columns. The input vector is transmitted along horizontal lines, and output currents are read from vertical lines—these currents are proportional to the dot product between each stored weight row and the input vector. This approach eliminates repeated data transfers between memory and compute units, significantly reducing latency and power consumption [6, 8].

CiM architectures are characterized by extremely high throughput, as thousands of multiply-accumulate (MAC) operations can be executed in parallel. Their efficiency is further enhanced by the absence of data transfer overhead and the low current requirements of NVM devices, making CiM-based systems particularly energy-efficient.

However, non-volatile memory technologies are prone to spatial defects and temporal variability, including write noise and aging. These variations are commonly modeled as $v = v_0 + \Delta v$, where $\Delta v \sim N(0, \sigma_v)$. For five representative NVM technologies, the parameter σ_v varies depending on the material and cell design, requiring dedicated error correction mechanisms and adaptive training strategies.

To address these reliability challenges, the RoCR framework employs noise-tolerant training of document and query embeddings. Contrastive learning is implemented using a triplet loss function that encourages proximity between positive examples and distance from negative ones in the embedding space.

Noise-aware training complements the cross-entropy stage by adding Gaussian noise $N(0, \sigma_v)$ to the embeddings, simulating hardware-level variability during optimization. Once embeddings are written to the CiM memory array, this method ensures their representational quality remains robust despite device-level fluctuations [4, 10].

For deployment on Edge-CiM platforms, a reshape module is introduced—an autoencoder that transforms 32-bit embeddings into CiM-compatible format (int8, 64×64 matrix) with minimal distortion. This transformation preserves semantic fidelity while significantly reducing memory footprint and computational load.

This hardware-software co-design approach enables high-throughput, low-latency retrieval and generation at the edge, combining compact, efficient memory usage with reliable semantic performance even under hardware noise conditions.

5. Applications and Optimizations Across Domains

The RoCR framework represents an integrated architecture that combines Retrieval-Augmented Generation (RAG) with CiM-accelerated max inner product search (MIPS). This fusion enables both efficient query segmentation on edge devices (Edge-LLMs) and dynamic access to external knowledge stores, enhancing the system's capacity to support complex tasks in real time.

In the context of landscape architecture, RAG acts as an intelligent design assistant—searching specialized literature and generating high-quality responses to complex project queries. This augments the depth and creativity of design decisions by providing contextual, evidence-based guidance [9].

A closed-domain QA system, *CDQA*, developed by Shelby L. and da Silva R. V. M. A. [1], is based on a RAG pipeline trained on articles and three supporting texts (Foreword, Introduction, Preface). The index consists of document vectors, and relevant fragment retrieval is handled using Elasticsearch combined with the Haystack framework and the BM25 algorithm. The reading module is a BERT model optimized for precise answer extraction. On the JoDLA test set, the system showed performance improvements with Rouge-1 F1 reaching 0.252, Rouge-2 F1 at 0.059, and Rouge-L F1 at 0.215 [1].

In the image captioning domain, Sarto S. et al. [2] proposed transformer-based architectures enhanced with external memory, trained on COCO and Conceptual Captions (CC3M) datasets. Using the COCO index (120,000 images), CIDEr increased from 135.3 to 136.7 (+1.1%). With CC3M (3.1 million images, BLIP captions), a similar CIDEr gain of +1.0% was observed, along with a BLEU-4 score improvement of 2%, reaching 40.8. On the out-of-domain nocaps test set, the RA-T model with the CC3M index boosted CIDEr from 66.7 to 69.5 (+4.2%) over a baseline transformer—demonstrating the robustness and generalizability of the approach.

Table 3. Comparison of Retrieval-Augmented Models Across Domains [1–3]

Dataset	**Model**	**Metric**	**Baseline**	**RAG Variant**
Landscape CDQA	JoDLA Q&A	Rouge-1 F1	ODQA (no RAG)	0.252
		Rouge-2 F1		0.059
		Rouge-L F1		0.215
Image Captioning	COCO Karpathy	CIDEr	Transformer (no retr.)	135.3 → 136.7 (RA-T)

Dataset	Model	Metric	Baseline	RAG Variant
Image Captioning	nocaps OOD	CIDEr	Transformer (no retr.)	66.7 → 69.5 (RA-T)

In this section, we will examine practical case studies of companies using external memory to enhance search-result generation.

The first is Bloomreach, which plans to integrate generative AI – via NVIDIA NIM microservices – into its search and merchandising platform. The implemented architecture includes:

1. Embedding microservices based on NVIDIA NeMo, which convert text fragments (product descriptions, metadata, etc.) into vectors and store them in an external vector store.

2. Retrieval process, performing approximate nearest-neighbor (ANN) search over those vectors, achieving high response speed through optimization with Triton Inference Server and TensorRT.

3. Generative stage, where LLM-generated prompts are appended to the retrieved similarity vectors, enabling the search to return not just product lists but contextualized, explanatory recommendations.

This decoupled retriever–generator approach, with a centralized vector store as external memory, ensures scalability, modularity, and low latency while maintaining high search accuracy [11].

Another example is Wayfair, which in 2024 launched Agent Co-Pilot—an internal digital assistant for sales agents, built on Retrieval-Augmented Generation (RAG). At its core is an LLM-based architecture. These AI models were trained on massive datasets, allowing them to understand and generate human-like text with impressive accuracy. Co-Pilot leverages the LLM's capabilities to analyze customer messages, infer intent, and provide sales agents with a range of useful responses [12].

The final example is Klevu. In December 2024, Klevu introduced Asklo—a RAG solution embedded directly on product detail pages (PDPs), offering interactive Q&A with shoppers. Its architecture includes:

1. Document loaders and a vector index in external memory, into which product descriptions, FAQs, and review snippets are ingested.

2. Retrieval mechanism that selects the most relevant fragments (product snippets, FAQ entries, review highlights) based on vector similarity.

3. Generative module, which – using the retrieved fragments and current dialogue context – produces detailed, personalized responses while preserving session history.

Thanks to this approach, Klevu was able to reduce bounce rates on product pages [13].

The analysis confirms that integrating Retrieval-Augmented Generation with CiM-accelerated MIPS in the RoCR framework not only supports flexible query processing on edge platforms but also enables dynamic, large-scale access to vectorized external knowledge. This expands the functional capabilities of systems operating in resource-constrained or latency-sensitive environments.

6. Conclusion

This study presents RoCR, a hardware-software solution for accelerating and scaling Retrieval-Augmented Generation (RAG) on edge devices. By moving MIPS computation in-memory via CiM crossbars, latency is reduced from minutes or hours to milliseconds, while maintaining high retrieval accuracy with minimal energy consumption.

To enhance embedding robustness under hardware noise, noise-aware and contrastive training methods are recommended. Injecting Gaussian noise that simulates RRAM/FeFET variability, combined with triplet-loss optimization and CDE/CDI evaluation metrics, yields improved performance across a range of datasets.

Architectural flexibility in generation is achieved through two LLM decoder variants: (1) RA-T—a decoder-only model with retrieved-token prefixing; and (2) RA-ED—an encoder–decoder scheme with kNN cross-attention and a learnable adaptive gate. Both approaches show comparable gains in generation quality and can be adapted to diverse hardware constraints. The generalizability of RoCR has been validated across multiple real-world tasks.

In summary, RoCR represents a significant step toward real-time, private, and scalable AI deployments in domains like eCommerce, where adaptive generation and personalized retrieval must be delivered within strict latency and power budgets.

References

1. Shelby L., da Silva R. V. M. A. Retrieval-augmented Generation: Empowering Landscape Architects with Data-driven Design //Journal of Digital Landscape Architecture. 2024. pp. 267-276.
2. Sarto S. et al. Towards retrieval-augmented architectures for image captioning //ACM Transactions on Multimedia Computing, Communications and Applications. – 2024. Vol. 20 (8). pp. 1-22.
3. Qin R. et al. Robust implementation of retrieval-augmented generation on edge-based computing-in-memory architectures //Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design. 2024. pp. 1-9.
4. Radford A. et al. Learning transferable visual models from natural language supervision //International conference on machine learning. PmLR, 2021. pp. 8748-8763.
5. Malkov Y. A., Yashunin D. A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs //IEEE transactions on pattern analysis and machine intelligence. 2018. Vol.42 (4). pp. 824-836.
6. Gao T., Yao X., Chen D. Simcse: Simple contrastive learning of sentence embeddings //arXiv preprint arXiv:2104.08821. 2021. pp. 1-9.
7. Cornia M., Baraldi L., Cucchiara R. Explaining transformer-based image captioning models: An empirical analysis //AI Communications. 2022. Vol. 35 (2). pp. 111-129.
8. Qiu Y. et al. Landscape Architecture Professional Knowledge Abstraction: Accessing, Applying and Disseminating //Land. 2023. Vol. 12 (11). pp.2061.
9. Zhu F. et al. Retrieving and reading: A comprehensive survey on open-domain question answering //arXiv preprint arXiv:2101.00774. 2021. pp. 1-8
10. Fruchard B. et al. User preference and performance using tagging and browsing for image labeling //Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. 2023. pp.1-13.
11. Bloomreach Revolutionizes the Future of Ecommerce Search, Powered by NVIDIA NeMo [Electronic resource] Access mode: https://www.bloomreach.com/en?p=48921 (date of request: 05/14/2025).
12. Agent Co-Pilot: Wayfair's Gen-AI Assistant for Digital Sales Agents [Electronic resource] Access mode: https://www.aboutwayfair.com/careers/tech-blog/agent-co-pilot-wayfairs-gen-ai-assistant-for-digital-sales-agents (date of request: 05/14/2025).
13. Klevu AI Ecommerce Search & Discovery [Electronic resource] Access mode: https://www.klevu.com/ (date of request: 05/14/2025).