EACO-RAG: Towards Distributed Tiered LLM Deployment using Edge-Assisted and Collaborative RAG with Adaptive Knowledge Update

Jiaxing Li[1], Chi Xu[1], Lianchen Jia[2], Feng Wang[3], Cong Zhang[4], Jiangchuan Liu[1]

[1]Simon Fraser University, Burnaby, BC, Canada [2]Tsinghua University, Beijing, China
[3]University of Mississippi, University, MS, USA [4]Jiangxing Intelligence Inc., Shenzhen, China

{jla641, chix, jcliu}@sfu.ca, jlc21@mails.tsinghua.edu.cn, fwang@cs.olemiss.edu, vcongzc@gmail.com

Abstract

Large language models (LLMs) have demonstrated impressive capabilities in language tasks, but they require high computing power and rely on static knowledge. To overcome these limitations, Retrieval-Augmented Generation (RAG) incorporates up-to-date external information into LLMs without extensive fine-tuning. Meanwhile, small language models (SLMs) deployed on edge devices offer efficiency and low latency but often struggle with complex reasoning tasks. Unfortunately, current RAG approaches are predominantly based on centralized databases and have not been adapted to address the distinct constraints associated with deploying SLMs in edge environments. To bridge this gap, we propose Edge-Assisted and Collaborative RAG (EACO-RAG), a lightweight framework that leverages distributed edge nodes for adaptive knowledge updates and retrieval. EACO-RAG also employs a hierarchical collaborative gating mechanism to dynamically select among local, edge-assisted, and cloud-based strategies, with a carefully designed algorithm based on Safe Online Bayesian Optimization to maximize the potential performance enhancements. Experimental results demonstrate that EACO-RAG matches the accuracy of cloud-based knowledge graph RAG systems while reducing total costs by up to 84.6% under relaxed delay constraints and by 65.3% under stricter delay requirements. This work represents our initial effort toward achieving a distributed and scalable tiered LLM deployments, with EACO-RAG serving as a promising first step in unlocking the full potential of hybrid edge–cloud intelligence.

1 Introduction

In recent years, Large Language Models (LLMs) have made remarkable strides in natural language comprehension and generation, drawing significant interest from both academia and industry [Chang *et al.*, 2024; Wei *et al.*, 2022; Zhang *et al.*, 2024b]. They are revolutionizing real-world applications by enabling more intelligent, adaptive, and scalable solutions. Their integration not only enhances user experiences with accurate, real-time, and context-aware responses but also improves system efficiency and unlocks new capabilities in autonomous processing [Wu *et al.*, 2022], recommendations [Lyu *et al.*, 2023; Ren *et al.*, 2024], and decision-making [Yang *et al.*, 2023; Li *et al.*, 2022].

On the other hand, Retrieval-Augmented Generation (RAG) enhances LLMs by incorporating retrieval mechanisms from external knowledge bases. Although LLMs benefit from vast pre-trained knowledge, their static nature limits its performance when information evolves rapidly. RAG addresses this limitation by providing up-to-date context during inference, which improves response accuracy. Modern LLMs scale to billions or even trillions of parameters, demanding extensive computing resources [Li *et al.*, 2024b; Guo *et al.*, 2025]. Given the high cost of retraining, RAG offers an efficient alternative by dynamically integrating current knowledge. Consequently, RAG-based methods are increasingly adopted in healthcare, education, and legal services, with a projected compound annual growth rate of 44.7% from 2024 to 2030 [Grand View Research, 2025].

In industrial deployments, there is a trend to shift toward solutions with lower latency and reduced hardware costs. This trend drives the adoption of small-scale LLMs, or small language models (SLMs), for edge and on-device applications [Zhang *et al.*, 2024a; Qu *et al.*, 2024]. Nevertheless, SLMs often struggle with tasks that require extensive external knowledge, leading to higher hallucination rates [Chen *et al.*, 2024; Yang *et al.*, 2018]. As such, paring RAG with SLMs can compensate for their limited capacity by retrieving relevant external information, and thus enhance factual accuracy and contextual relevance while safeguarding data privacy and reducing delay—features crucial for industrial applications with different Quality of Service (QoS) requirements.

However, current RAG solutions are not yet optimized for edge distributed deployments of SLMs. They often rely on centralized databases and face several key challenges. First, large-scale retrieval can introduce redundant information, leading to increased latency and inference costs [Hofstätter *et al.*, 2023; Yu *et al.*, 2024]. Second, irrelevant or misleading retrieval degrades output quality [Chen *et al.*, 2024]. Third, complex or multi-hop queries may exceed the capacity of lightweight databases and models, necessitating deeper retrieval and reasoning methods [Zhao *et al.*, 2024]. Finally, most existing RAG systems cannot dynamically adapt to ever-changing user interests across regions and over time.
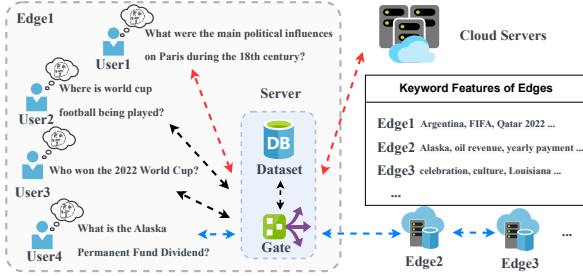
Figure 1: EACO-RAG's adaptive retrieval. Each edge maintains a dynamic local dataset of popular topics. The collaborative gating mechanism selects retrieval sources from local, edge, or cloud datasets to adapt to evolving user interests and knowledge distributions. Black, blue, and red arrows denote local, edge-assisted, and cloud communications, respectively.

To address these limitations and facilitate highly distributed, scalable deployments, we propose Edge-Assisted and Collaborative RAG (EACO-RAG). Rather than relying on a single, monolithic knowledge base, EACO-RAG employs a distributed approach across multiple edge nodes, enabling broader and more contextually relevant retrieval beyond immediate local trends. Moreover, each node adaptively updates its local knowledge sets, ensuring timely accuracy in response to evolving information. A hierarchical collaborative gating mechanism is also designed to determine whether to perform retrieval and generation locally, via edge assistance, or in the cloud. This approach allows SLMs at the edge to work in tandem with powerful cloud resources, thereby reducing costs while maintaining accuracy and satisfying delay requirements. Figure 1 illustrates a toy example of how EACO-RAG adapts retrieval across different levels.

Through these optimizations, EACO-RAG substantially lowers inference costs and retrieval delays while maintaining high accuracy. Our experiments show that under relaxed delay constraints, EACO-RAG achieves accuracy comparable to cloud-based 72B LLM+GraphRAG while reducing cost by up to 84.6%. Even when maintaining similar accuracy and delay to LLM+GraphRAG, it still reduces cost by up to 65.3%. These results underscore the potential of EACO-RAG as a promising method toward distributed tiered LLM deployments, bridging edge and cloud to unlock more efficient, scalable, and adaptable AI systems.

The contributions of this work are summarized as follows:

- To the best of our knowledge, this paper is the first effort to systematically propose and investigate an edge-assisted distributed RAG architecture, which leverages adaptive knowledge updates and collaboration across edge nodes and cloud resources to offer a cost-efficient solution for large-scale distributed environments.

- We design dynamic update and edge-assisted distributed mechanisms that enable each edge node to adjust its local knowledge set. This ensures the real-time validity and effectiveness of edge data, also avoids the limitations of relying on a single dataset.

- We develop a context-aware collaborative gating mechanism that leverages Safe Online Bayesian Optimization
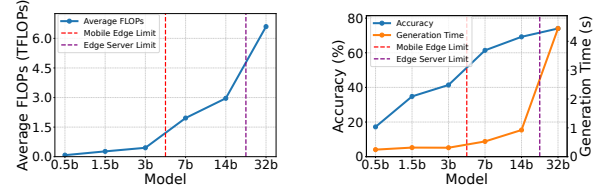

Figure 2: Performance trade-offs in LLM-only applications using Qwen2.5 [Yang *et al.*, 2024], evaluated on the TriviaQA dataset [Joshi *et al.*, 2017]. **Left:** Model size vs. inference cost, showing the relationship between LLM parameters and TFLOPs. **Right:** Model size vs. accuracy and delay, illustrating the impact of LLM parameter on accuracy and generation latency.

Approach	Input Token	Output Token	Inference Cost
LLM-only	16.01 ± 5.01	27.21 ± 14.83	˜0.65 TFLOPs
Naive RAG	3632 ± 28.95	26.59 ± 19.81	˜22.98 TFLOPs
GraphRAG	9017 ± 2529	142.7 ± 91.58	˜58.57 TFLOPs

Table 1: Comparison of token utilization and inference computational cost among LLM-only, Naive RAG and GraphRAG (with default parameters) using a 3B LLM. The total computational cost, measured in TFLOPs, is estimated based on [Pope *et al.*, 2023].

for cost-effective decision-making and dynamic adaptation to evolving user demands and QoS constraints.

- We conduct extensive experiments evaluating EACO-RAG, demonstrating its outperforms centralized RAG systems in terms of retrieval relevance, adaptability to evolving information, scalability to different scenarios and efficiency in balancing accuracy, delay, and cost.

2 Background and Motivation

Recent advancements in LLMs have explored both scaling up models for enhanced reasoning and optimizing smaller models for efficiency [Yang *et al.*, 2024; Guo *et al.*, 2025]. Large-scale language models, such as OpenAI o1[1], o3[2], and DeepSeek R1[3], push the boundaries of accuracy and complex reasoning by leveraging extensive computational resources. While these models achieve unprecedented performance, their high inference costs and delay make them less practical for real-time and resource-constrained applications.

At the same time, SLMs delpoy on edge have gained significant attentions, particularly when integrated with external knowledge sources, as seen in Naive RAG and RAG variants like GraphRAG [Edge *et al.*, 2024]. As shown in Figure 2, leveraging SLMs for on-device inference can reduce delay and operational costs, making them suitable for dynamic industrial settings with varying QoS requirements. However, existing RAG implementations face critical challenges. Naive retrieval mechanisms can introduce irrelevant or misleading information, degrading model performance. Furthermore, cloud-based retrieval solutions, such as GraphRAG, add latency and increase token consumption, as shown in Table 1. The overhead from retrieving large volumes of text as context

---

[1]https://openai.com/o1/

[2]https://openai.com/index/openai-o3-mini/

[3]https://api-docs.deepseek.com/news/news250120

Query Type	Example Queries
Time	Who won the 2022 World Cup?
	Who are the candidates for the 2024 U.S. election?
	What are the breakthroughs in Tesla's latest autopilot?
Location	What is the Alaska Permanent Fund Dividend?
	What are the Mardi Gras traditions in New Orleans?
	When is Vermont's maple syrup season, how to join?

Table 2: Examples of real-world queries with temporal and spatial variations, highlighting challenges in adapting retrieval models to dynamic user interests.

can significantly raise the input-output token ratio, increasing TFLOPs consumption.

Moreover, real-world queries exhibit both temporal and spatial variations, posing further challenges for retrieval models. Table 2 illustrates how user interests fluctuate over time, with queries often reflecting evolving events, such as sports results, political elections, or technological advancements. Since LLMs are trained on datasets available up to a fixed cutoff date, even large models are inherently limited by outdated information. For instance, the powerful o1 model, without external updates, cannot recognize entities like o3 or R1. A retrieval system that fails to update its knowledge base frequently may struggle to provide accurate responses to time-sensitive queries. Similarly, spatial variations affect query relevance, as users from different locations seek information specific to their regional context, such as local policies, cultural traditions, or seasonal events. Without mechanisms for adapting to these dynamic changes, retrieval models risk conflating information from different regions or failing to retrieve relevant data, leading to incomplete or misleading responses. Existing RAG methods often lack the ability to dynamically adjust to these shifts in information demand, resulting in sub-optimal retrieval quality and inference accuracy. In our experiments, we measured accuracy by comparing generated responses to ground truth using GPT-4o [Langchain-AI, 2023].

To address these limitations, we propose EACO-RAG, a dynamic system designed to adapt to changing real-world conditions. Our approach integrates three core components: (1) dynamic knowledge updates at the edge, ensuring dataset relevance and accurate as contents of user interests evolve over time; (2) edge-assisted and collaboration, enabling retrieval across other relevant edge nodes rather than being limited to the local dataset, thereby expanding the scope of knowledge sets on the edge; and (3) an adaptive collaborative gating mechanism that selects the optimal retrieval strategy to meet various QoS demands.

3 EACO-RAG Design

3.1 Overview

EACO-RAG integrates edge nodes with cloud resources, using lightweight models and adaptive retrieval strategies to reduce costs while maintaining high QoS. It features an edge-assisted RAG architecture that dynamically updates local knowledge bases, ensuring timely and relevant information. The system efficiently utilizes edge databases and SLMs for query answering. By enabling collaboration among edge
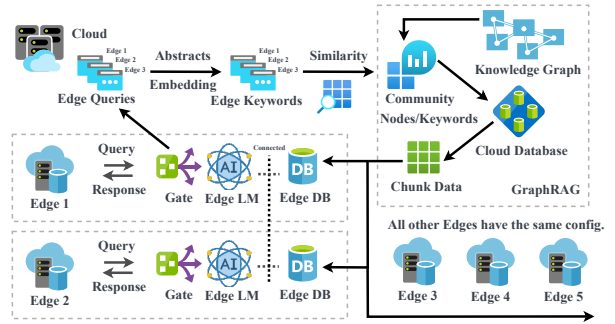

Figure 3: Workflow of the EACO-RAG system design.

nodes and incorporating a collaborative gating mechanism, EACO-RAG improves scalability and adaptability, allowing seamless operation across diverse real-world scenarios.

3.2 Why GraphRAG

A key enhancement of EACO-RAG is its integration with GraphRAG, a structured retrieval framework that improves accuracy by maintaining strong contextual relationships among knowledge elements. Unlike conventional RAG methods, GraphRAG organizes information into nodes, edges, and communities. Nodes represent discrete knowledge units, edges capture relationships, and communities group semantically related concepts. This structure enables more precise and context-aware retrieval, making GraphRAG particularly effective for multi-hop reasoning and knowledge-intensive tasks. However, its extended retrieval time and reliance on long-context processing make it more suitable for cloud deployment. Running GraphRAG directly on edge devices would be computationally restrictive and inefficient in utilizing edge resources.

EACO-RAG addresses these challenges by selectively extracting and distributing relevant knowledge from GraphRAG to the edge. This balances retrieval efficiency and contextual depth. The system dynamically updates the most relevant data chunks from GraphRAG communities, enabling localized knowledge adaptation without maintaining a full graph structure. Strong intra-community alignment in GraphRAG ensures that even lightweight mechanisms, like Naive RAG, operate with well-structured and semantically coherent data. This reduces ambiguity in concept interpretation and mitigates confusion from polysemous terms. For example, the term "model" may refer to a fashion model, a scientific model, or a prototype. By anchoring retrieval within relevant communities, EACO-RAG ensures that each edge node receives only the most contextually appropriate data. This minimizes definitional ambiguity while optimizing retrieval efficiency on the edges.

3.3 System Design

As illustrated in Figure 3, the cloud periodically collects and processes queries from users across various edge nodes, maintaining a knowledge graph that organizes nodes and communities based on evolving information trends. These nodes serve as keywords to extract relevant data chunks from recent queries at each edge location, allowing the system to capture both spatial and temporal shifts in user interests. The extracted data is then dynamically distributed to the appropriate edge nodes, reducing the dependency on large local databases while mitigating retrieval costs and resource overhead.

During query processing, EACO-RAG dynamically selects the optimal retrieval source by evaluating the overlap between an incoming query and the keyword-indexed knowledge at each edge node. This allows retrieval to extend beyond a single location, incorporating relevant insights from nearby nodes rather than being restricted to the most locally popular data. As information needs evolve over time, edge databases are continuously updated, ensuring that responses remain contextually relevant across both spatial and temporal dimensions.

Recognizing the various complexity of queries, EACO-RAG distinguishes between those that can be efficiently processed at the edge and those requiring escalation. While lightweight models and edge databases efficiently handle straightforward queries, more complex or multi-hop reasoning tasks may exceed their capacity due to limited local resources. To address this, the system escalates complex queries to the cloud-based knowledge graph or high-parameter models, improving accuracy while maintaining responsiveness and adaptability across diverse real-world scenarios.

At the core of this framework is the collaborative gating mechanism, which optimizes query processing by balancing cost, accuracy, and delay. For simple, well-covered queries, the system relies on SLMs at the edge for fast responses. When queries demand deeper reasoning or involve less common topics, retrieval is progressively escalated to more comprehensive knowledge sources, ensuring a balance between computational efficiency and response quality. For complicated queries that also demands high precision and reduced delay, additional cloud resources are allocated to execute large-parameter language models.

To optimize the balance between QoS and cost-effectiveness, EACO-RAG formulates decision-making as a contextual multi-armed bandit problem. Safe Online Bayesian Optimization continuously refines decision policies based on real-time query distribution and retrieval performance. By dynamically coordinating retrieval and inference across edge and cloud resources, EACO-RAG effectively reduces operational costs while ensuring accurate and timely responses, making it well-suited for dynamic and evolving application environments.

4 Modelling and Optimization

4.1 Modelling with Contextual Multi-Armed Bandit

Our goal is to minimize operational costs, including resource and time costs. Resource costs come from model inference and retrieval, based on input and output tokens. Time costs account for network and loading delays. The system must satisfy two QoS constraints: (i) accuracy must exceed $QoS^\rho_{\min}$, and (ii) response time must stay within $QoS^h_{\max}$.

Context: At each time step t, the context is represented as $c_t := [d_t, s_t, q_t] \in \mathcal{C}$. Here, d_t denotes network delays, which include both cloud and edge delays, helping assess network availability. The term s_t consists of two values: the highest keyword overlap ratio between the query and the edge datasets, along with the corresponding edge dataset. The query complexity q_t is represented as a set that captures whether the query requires single-hop or multi-hop reasoning, its length, and the number of entities it contains [Yang *et al.*, 2018].

Control Policies: The system's control policy at time t is denoted as $x_t := [r_t, g_t] \in \mathcal{X}$. Here, r_t selects the retrieval source, which can be none, edge-assisted naive retrieval, or cloud knowledge graph-based retrieval. The term g_t determines the response generation location, either local SLM or cloud LLM. These policies dynamically adapt to real-time context to optimize cost while maintaining accuracy and delay constraints.

GPU Model	FP64 (Double Precision)
NVIDIA GeForce RTX 4090	1.29 TFLOPS
NVIDIA Tesla P100	4.70 TFLOPS
NVIDIA Tesla V100	7.80 TFLOPS
NVIDIA A100 Tensor Core	9.70 TFLOPS
NVIDIA H100 Tensor Core	60.00 TFLOPS

Table 3: Double-precision (FP64) peak performance (in TFLOPS) of various server-side NVIDIA GPUs.

The total cost function is defined as:

$$u_t(c_t, x_t) = \delta_1 \cdot u_r^t(c_t, x_t) + \delta_2 \cdot u_d^t(c_t, x_t), \quad (1)$$

where δ_1 and δ_2 weight resource and time costs. $u_r^t(c_t, x_t)$ represents computational costs, while $u_d^t(c_t, x_t)$ accounts for time costs. For ease of parameter adjustments, we unify the unit of resource and time costs by scaling the time cost with the peak TFLOPs of different GPUs depending on c_t and x_t as shown in Table 3, which turn out to also better reflect real-world situations as the time cost is usually minimal for edge devices but significant for cloud computing.

The optimization problem is formulated as:

$$
\begin{aligned}
\min_{\{x_t\}_{t=1}^T} \quad & \sum_{t=1}^T u_t(c_t, x_t) \\
\text{s.t.} \quad & \rho_t(c_t, x_t) \geq QoS^\rho_{\min}, \quad \forall t \leq T, \\
& h_t(c_t, x_t) \leq QoS^h_{\max}, \quad \forall t \leq T,
\end{aligned}
\quad (2)
$$

where T is the total number of decision steps. $\rho_t(c_t, x_t)$ denotes answer accuracy and $h_t(c_t, x_t)$ represents overall delay at step t. $QoS^\rho_{\min}$ and $QoS^h_{\max}$ denote the minimum accuracy and maximum delay, respectively. The goal is to minimize total cost while meeting accuracy and delay constraints at each step.

This formulation achieves cost efficiency while meeting the essential QoS constraints of accuracy and delay. The QoS constraints can be adjusted to suit different scenarios and applications, enabling various gate instances to address diverse requirements. In addition, the system adapts to changing contexts through online learning, which supports real-time decision-making. In the following subsection, we present our Safe Online Bayesian Optimization solution, which further details how the collaborative gate mechanism makes adaptive decisions in real time.

4.2 Safe Online Bayesian Optimization

We propose a Bayesian online optimization approach for decision-making in dynamic environments, outlined in Algorithm 1. It uses Gaussian Processes (GPs) to model system correlations and quantify uncertainty, enabling adaptive optimization while reducing cost and maintaining performance and delay constraints.

Function Approximation. The algorithm use GPs to estimate cost and constraint functions, following established methods [Williams and Rasmussen, 2006; Duvenaud, 2014]. Each function is modeled as $\text{GP}(\mu(x), k(x, x'))$, where $\mu(x)$ is the mean function and $k(x, x')$ captures covariance. Based on observed data, the algorithm iteratively update posterior distributions for cost, accuracy, and delay to refine estimates.

Safe Set Identification. The safe set S_t consists of control policies satisfying system constraints at time t:

$$
\begin{aligned}
S_t = \{x \in \mathcal{X} \mid & \mu_t^{(1)}(c_t, x) - \beta\sigma_t^{(1)}(c_t, x) \geq QoS^\rho_{\min} \\
& \wedge \mu_t^{(2)}(c_t, x) + \beta\sigma_t^{(2)}(c_t, x) \leq QoS^h_{\max}\}.
\end{aligned}
\quad (3)
$$

Here, $\mu_t^{(i)}(c_t, x)$ and $\sigma_t^{(i)}(c_t, x)$ represent the GP-predicted mean and uncertainty for function i (accuracy or delay). The parameter β adjusts the confidence bound, balancing exploration and safety.

Algorithm 1: Collaborative Gating SafeOBO Algorithm

---

1 **Inputs:** Control space $\mathcal{X}$, Safe seed set S_0, kernel k, $QoS^\rho_{\min}$ (minimum accuracy), $QoS^h_{\max}$ (maximum delay), exploration parameter β, cost weights δ_1, δ_2

2 **Initialization:** $Z_0 = \emptyset, y_0^{(0)} = \emptyset, y_0^{(1)} = \emptyset, y_0^{(2)} = \emptyset$;

3 **for** $t = 1, \ldots, T_0$ *(Exploration phase)* **do**

4   **Observe** context c_t;

5   **Randomly select** x_t from $\mathcal{X}$ (warm-up step);

6   **Observe** $h_t(c_t, x_t)$ (response time), $\rho_t(c_t, x_t)$ (accuracy), $u_r^t(c_t, x_t)$ (resource cost), $u_d^t(c_t, x_t)$ (delay cost);

7   **Compute total cost:**
     $u_t(c_t, x_t) = \delta_1 \cdot u_r^t(c_t, x_t) + \delta_2 \cdot u_d^t(c_t, x_t)$;

8   **Update** GP posteriors:

9          $y_t^{(0)} \leftarrow y_{t-1}^{(0)} \cup u_t(c_t, x_t)$ (update cost posterior);

10         $y_t^{(1)} \leftarrow y_{t-1}^{(1)} \cup \rho_t(c_t, x_t)$ (update accuracy posterior);

11         $y_t^{(2)} \leftarrow y_{t-1}^{(2)} \cup h_t(c_t, x_t)$ (update response time posterior);

12 **end**

13 **for** $t = T_0 + 1, \ldots, T$ *(Exploitation phase)* **do**

14   **Observe** context c_t;

15   **Compute** $\mu_{t-1}^{(i)}(c_t, x)$ and $\sigma_{t-1}^{(i)}(c_t, x)$ for all $i = 0, 1, 2$, using the posterior from the previous iteration;

16   **Estimate the safe set:**

17
$$S_t = S_0 \cup \{x \in \mathcal{X} \mid \mu_t^{(1)}(c_t, x) - \beta\sigma_t^{(1)}(c_t, x) \geq QoS^\rho_{\min}$$

18      $\wedge \mu_t^{(2)}(c_t, x) + \beta\sigma_t^{(2)}(c_t, x) \leq QoS^h_{\max}\}$;

19   **Select** $x_t = \arg\min_{x \in S_t} \mu_t^{(0)}(c_t, x) - \beta_t\sigma_t^{(0)}(c_t, x)$;

20   **Observe** $h_t(c_t, x_t)$ (response time), $\rho_t(c_t, x_t)$ (accuracy), $u_r^t(c_t, x_t)$ (resource cost), $u_d^t(c_t, x_t)$ (delay cost);

21   **Compute total cost:**
     $u_t(c_t, x_t) = \delta_1 \cdot u_r^t(c_t, x_t) + \delta_2 \cdot u_d^t(c_t, x_t)$;

22   **Update** GP posteriors:

23         $y_t^{(0)} \leftarrow y_{t-1}^{(0)} \cup u_t(c_t, x_t)$ (update cost posterior);

24         $y_t^{(1)} \leftarrow y_{t-1}^{(1)} \cup \rho_t(c_t, x_t)$ (update accuracy posterior);

25         $y_t^{(2)} \leftarrow y_{t-1}^{(2)} \cup h_t(c_t, x_t)$ (update response time posterior);

26 **end**

Exploration and Exploitation. The algorithm begins with an exploration (warm-up) phase, randomly selecting decisions to build an initial dataset of cost, accuracy, and response time. Once sufficient data is collected, the system shifts to an exploitation phase, optimizing decisions within the safe set to minimize total cost:

$$x_t = \arg\min_{x \in S_t} \mu_t^{(0)}(c_t, x) - \beta_t\sigma_t^{(0)}(c_t, x). \tag{4}$$

Here, $\mu_t^{(0)}(c_t, x)$ and $\sigma_t^{(0)}(c_t, x)$ denote the predicted mean and uncertainty of the cost function.

This strategy ensures cost efficiency while meeting accuracy and delay constraints. As new data continuously refine decision-making and update the GP models, the system adapts to changing contexts in real time, further enhancing its overall adaptability.

5 Prototype Implementation

Our implementation employs a dual-layer architecture consisting of cloud and edge components. At the edge, a RTX 4090 GPU simulates multiple edge nodes running the SLM, local RAG, and EACO-RAG. The cloud is simulated using an A800 GPU emulating an $8 \times$ H100 setup, supporting a 72B model with graphRAG-based retrieval and achieving approximately 1-second query response time.

To maintain relevant local knowledge, the system dynamically updates a repository of 1,000 local data chunks, triggering updates when the cloud accumulates 20 new QA pairs. Keywords extracted from these queries guide updates via a first-in-first-out (FIFO) policy, ensuring efficient storage and retrieval performance.

Recall that beyond local inference, EACO-RAG also incorporates an edge-assisted retrieval mechanism, selecting datasets from other edge nodes when local coverage is insufficient. The system evaluates the overlap ratio, defined as the proportion of query keywords present in the target dataset, to determine the most relevant retrieval edge data chunks. To identify valid query keywords, a lightweight semantic embedding model ('all-MiniLM-L6-v2') is used to map the query to related keywords, considering those with a similarity score above 50% as valid matches. A similar strategy is applied to GraphRAG-based adaptive knowledge updates, where recent edge queries are processed through the embedding model to identify relevant keywords in GraphRAG. The system then selects the top-k communities containing the highest number of similar keywords or nodes and distributes up to 500 data chunks from these communities to the edge.

Additionally, the collaborative gating mechanism is also implemented to dynamically select the optimal inference path—ranging from local small-model inference to full cloud-based retrieval with the 72B model—based on query complexity and system constraints. For domain-specific retrieval, knowledge graphs are constructed from the corresponding knowledge sources associated with the datasets used for performance evaluation (i.e., 139 Wikipedia pages for Wiki QA dataset and the seven canonical Harry Potter books for Harry Potter QA dataset as specified in next section). This adaptive approach optimizes response quality while balancing computational efficiency and resource utilization.

6 Performance Evaluation

6.1 Datasets Selection

To evaluate the effectiveness of EACO-RAG, we configure two distinct datasets. The first dataset, referred to as Wiki QA, is based on Wikipedia and is designed to represent general-domain questions. The second dataset focuses on the Harry Potter series and is intended to emulate more specialized, industrial scenarios.

Wiki QA Dataset: We selected 139 popular Wikipedia pages from the Natural Questions (NQ) dataset [Kwiatkowski *et al.*, 2019], each containing more than 10 question-answer pairs. To enhance the dataset's complexity and diversity, we integrated additional QA pairs from TriviaQA [Joshi *et al.*, 2017] and HotpotQA [Yang *et al.*, 2018] corresponding to these pages, resulting in a comprehensive dataset of 571 QA pairs.

Harry Potter QA Dataset: Sourced from [Candurkar, 2023], this dataset contains 1,180 high-quality question-answer pairs covering various aspects of the Harry Potter series. We filtered the dataset to retain more reliable pairs. Compared to the Wiki QA dataset, the Harry Potter QA dataset features more challenging questions that require specific background knowledge, effectively simulating complex industrial scenarios.

By employing these two datasets, we aim to rigorously test EACO-RAG's performance across both general and specialized domains, demonstrating its versatility and effectiveness in handling a range of query complexities.

6.2 Overall Performance

To evaluate the effectiveness of EACO-RAG, we compare EACO-RAG with several retrieval and generation baselines, including standalone SLMs, naive RAG-based edge retrieval, and cloud-based GraphRAG retrieval using both 3B and 72B parameter LLMs, as shown in Table 4. EACO-RAG dynamically selects among these

	Wiki QA			Harry Potter QA		
	Accuracy (%)	**Delay** (s)	**Cost** (TFLOPs)	**Accuracy** (%)	**Delay** (s)	**Cost** (TFLOPs)
3b LLM-only	28.72	0.30 ± 0.07	0.60 ± 0.16	31.69	0.31 ± 0.08	0.65 ± 0.20
3b LLM+Naive RAG	61.57	0.88 ± 0.11	23.10 ± 0.34	52.54	1.00 ± 0.18	23.62 ± 0.38
3b LLM+GraphRAG	76.01	3.01 ± 1.21	60.02 ± 17.45	63.47	2.82 ± 1.32	58.99 ± 16.69
72b LLM+GraphRAG	94.39	0.97 ± 0.64	711.43 ± 309.52	77.12	1.03 ± 0.84	739.79 ± 402.18
EACO-RAG (Cost-Efficient)	**94.92**	1.27	**109.40**	**78.00**	1.74	**139.43**
EACO-RAG (Delay-Oriented)	94.17	**0.75**	247.03	76.28	**0.79**	496.19

Table 4: Performance comparison of EACO-RAG and baseline approaches under our dual-layer (edge–cloud) architecture on Wiki QA and Harry Potter QA tasks, highlighting trade-offs among accuracy, delay, and cost.

Warm-up Steps	Accuracy (%)	Delay (s)	Cost (TFLOPs)
	Wiki QA		
EACO-RAG-300	**94.92**	1.27	**109.40**
EACO-RAG-200	89.66	1.26	140.06
EACO-RAG-100	87.22	1.49	346.29
	Harry Potter QA		
EACO-RAG-500	**78.00**	1.74	**139.43**
EACO-RAG-300	77.35	1.12	402.19
EACO-RAG-100	74.44	1.31	511.60

Table 5: Effect of different warm-up steps on EACO-RAG's gating decisions for Wiki QA and Harry Potter QA, showing how initial exploration influence accuracy, delay, and computational cost.

Model	Accuracy (%)	Delay (s)	Cost (TFLOPs)
Qwen2.5 7B	94.57	1.48	93.83
Qwen2.5 3B	94.92	1.27	109.40
llama3.2 3B	93.35	1.07	272.72
Qwen2.5 1.5B	91.42	0.95	167.67

Table 6: EACO-RAG performance of various SLMs on Wiki QA.

strategies based on query context and system constraints. We evaluate EACO-RAG under two different settings. In a cost-efficient setting, where delays up to 5s are acceptable, the gate prioritizes lower costs by favoring local inference and edge retrieval. In a delay-oriented setting, where responses must be under 1s, the gate relies more on cloud-based inference when network conditions allow, leading to higher computational costs.

Results demonstrate that EACO-RAG effectively leverages edge-assisted retrieval to significantly reducing costs while maintain accuracy and delay. Under relaxed latency constraints with strict accuracy requirements, it achieves performance comparable to a 72B LLM+GraphRAG system while reducing costs by 84.6% on the Wiki QA dataset. Similarly, on the more challenging Harry Potter dataset, it attains similar accuracy with an 81.2% cost reduction. In the delay-oriented setting, cost reductions are 65.3% for WikiQA and 32.9% for Harry Potter QA. This strong performance is attributed to EACO-RAG's ability to dynamically select integrated strategies that optimize the trade-off between cost and QoS.

Table 7 further takes a closer look at two examples on how the collaborative gating mechanism selects the optimal answer path for each query. In Question 1, the query is recognized as a simple single-hop problem, and the context indicates that the edge dataset fully covers the relevant entities with low delay; consequently, the gate selects the corresponding edge dataset and local 3B SLM. In contrast, in Question 2 the query is identified as a complex multi-hop problem based on contextual cues, and the context reveals that edge-assisted retrieval is insufficient—thus, the gate opts for cloud-based collaborative retrieval and generation.

These results confirm the effectiveness of our method, demonstrating that EACO-RAG can significantly lower costs across different constraints while maintaining robust performance.

6.3 Impact of Warm-up Steps

The number of warm-up steps (T_0) in EACO-RAG significantly influences the gate's strategy selection. An increased amount of warm-up data enhances the gate's contextual understanding, enabling the system to better differentiate between queries that can be answered locally and those requiring GraphRAG or cloud-based LLM inference. We evaluated various warm-up sizes and measured their impact on accuracy, delay, and cost. As shown in Table 5, for Wiki QA—with relatively simple queries—the gate begins favoring edge-based responses at 100 warm-up steps and effectively distinguishes query by 300 steps. In contrast, the Harry Potter dataset, which contains more specialized and context-dependent queries, requires more warm-up data before the system can shift more queries to edge-based inference to reduce costs.

6.4 Comparison of Different SLMs

In our previous experiments, we deployed a uniform 3B SLM at the edge for adaptability and scalability across edge devices with varying computational capabilities. To further evaluate EACO-RAG with SLMs of varying sizes and origins, we tested several open-source SLMs on the WikiQA dataset, including Qwen2.5 7B, Qwen2.5 1.5B, and llama3.2 3B. As shown in Table 6, replacing the edge SLM with Qwen2.5 7B reduced overall cost despite its higher computational expense, as the gate identified more queries that could be resolved at the edge. In contrast, using Qwen2.5 1.5B significantly lowered delay but increased cost, indicating that more queries were escalated to cloud-based retrieval.

Notably, the llama3.2 3B model underperformed compared to Qwen2.5 3B. This difference is due to distinct training approaches. Qwen2.5 3B benefits from large-scale pre-training and targeted fine-tuning [Qwen Team, 2024], which enhances its knowledge and contextual reasoning [Kaplan et al., 2020]. Llama3.2 3B relies on pruning and distillation [Meta AI, 2024], resulting in a lighter model with faster inference but reduced reasoning ability [Jiao et al., 2019]. As contextual understanding is critical because EACO-RAG relies on contextual understanding for retrieval-augmented generation in both edge-assisted and cloud-based methods, in this case, LLaMA3.2 underperforms compared to Qwen2.5 on EACO-RAG.

6.5 Ablation Study and Hyperparameter Analysis

In this study, we remove the collaborative gating mechanism and exclude cloud retrieval and generation. This isolates the effects of adaptive knowledge update and edge-assisted retrieval on accuracy. Without cloud retrieval, delay and cost differences between local and edge responses become negligible. We focus solely on evaluating the accuracy impact of these two components under various

Question 1	What is the name of the spell used to unlock doors?
Process	Context:{Single-hop; 15 tokens; 3 entities (spell, unlock, door); Edge4:[100% match, 20 ms delay]; Cloud:[300 ms delay]} $\Rightarrow$ Gate $\Rightarrow$ Decision:{Edge4 dataset + 3B SLM}
Output	Alohomora (Correct)

Question 2	What impact does Harry's friendship with Hermione have on his understanding of empathy and compassion?
Process	Context:{Multi-hop; 21 tokens; 4 entities (Harry, Hermione, empathy, compassion); Edge6:[25% match, 32 ms delay]; Cloud:[350 ms delay]} $\Rightarrow$ Gate $\Rightarrow$ Decision:{Cloud GraphRAG + 72B LLM}
Output	Harry's friendship with Hermione deepens his empathy and compassion through her intellect, support, and loyalty, helping him consider others' perspectives and act with kindness. (Correct)

Table 7: Illustrative examples of QA processing in EACO-RAG using the collaborative gate mechanism.
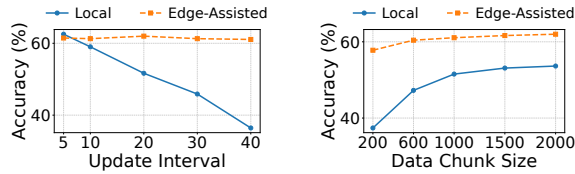

Figure 4: Accuracy comparison under different hyperparameter settings in the ablation study. **Left:** shows how varying the local adaptive update trigger interval influences accuracy. **Right:** illustrates the effect of different chunk sizes in the edge dataset.

hyperparameter settings.

We first examine the effect of the local adaptive update trigger interval. We compare naive RAG relying solely on the local update database with and without edge-assisted retrieval. As shown in Figure 4(a), when relying solely on local dataset, the update interval has a strong impact on accuracy. Incorporating edge-assisted retrieval reduces this sensitivity. Frequent local updates may even outperform edge-assisted retrieval due to stronger contextual correlations from the knowledge graph, but they consume more resources. We also study the impact of local chunk size. Figure 4(b) indicates that larger chunk sizes yield higher accuracy for both methods. With edge-assisted retrieval, convergence occurs at 600 data chunks; without edge-assisted, over 1000 data chunks are needed. However, on resource-constrained edge devices, a larger chunk size adds to the retrieval burden. In summary, more frequent updates and larger chunk sizes improve accuracy but increase resource overhead, while edge-assisted retrieval reduces sensitivity to both parameters. These findings also demonstrate that adaptive knowledge update and edge-assisted retrieval effectively address issues of outdated datasets and limited diversity from region-specific edge datasets.

7 Related Work

Retrieval-Augmented Generation Enhancements. RAG improves language models by integrating relevant text from knowledge bases [Lewis *et al.*, 2020]. Extensions such as Adaptive RAG [Jeong *et al.*, 2024], Corrective RAG [Yan *et al.*, 2024], and Self-RAG [Asai *et al.*, 2023] addressed limitations in retrieval strategies and query complexity by adapting operations based on query difficulty. Mallen et al. [Mallen *et al.*, 2023] classified query complexity via entity frequency to guide binary decisions on retrieval sufficiency, while Qi et al. [Qi *et al.*, 2021] employed fixed operations (retrieval, reading, reranking) that require specialized LM training. Despite these advances, traditional methods often rely on centralized frameworks. In contrast, EACO-RAG leverages distributed edge computing to dynamically update local databases, reducing delays and communication overhead.

Edge-Enabled Cost Optimization and Resource Allocation. Reducing LLM deployment costs has attracted significant research attention. Techniques such as model quantization [Xiao *et al.*, 2023; Park *et al.*, 2023], pruning [Ma *et al.*, 2024], and distillation enabled smaller models to mimic larger ones, while caching strategies [Stogiannidis *et al.*, 2023; Zhu *et al.*, 2023; Gill *et al.*, 2024; Li *et al.*, 2024a] and key-value state reuse [Liu *et al.*, 2024; Yao *et al.*, 2024] further lowered computational expenses. Additionally, approaches like FrugalGPT [Chen *et al.*, 2023] and model multiplexing [Bang, 2023; Kim *et al.*, 2023] dynamically adjusted model size based on query complexity. On the edge computing front, research has focused on optimizing resource allocation by balancing delay, energy, and processing power through task offloading, resource scheduling [Naouri *et al.*, 2021], and edge-cloud collaboration [Gu *et al.*, 2023; Xiong *et al.*, 2020]. EACO-RAG offers a holistic solution that integrates both edge and cloud resources and introduces inter-node collaboration for dynamic knowledge sharing and updating, thereby optimizing retrieval and generation while minimizing operational costs.

8 Further Discussion

The limitations of EACO-RAG stem from its constrained experimental scope. In our study, the collaborative gating mechanism only selects among four retrieval and inference strategies. In real-world applications, a broader range of adaptive strategies may emerge, further enhancing efficiency and accuracy. It is also interesting to have more large-scale studies and deployments in the future to fully assess the system's capabilities. In addition, the current gating design in our prototype implementation relies on general heuristics. Yet, query features can vary greatly for different datasets. Therefore, in practical industrial settings, the gate could further incorporate dataset-specific characteristics to provide richer contextual cues. This integration can better improve decision-making, adaptability, and optimization across diverse environments.

9 Conclusion

This paper presented EACO-RAG, an edge-assisted and collaborative RAG system that uses adaptive knowledge update to compensate query evolving across both region and time, and minimizes cost while maintaining high accuracy and low delay. By dynamically selecting among local, edge-assisted, and cloud-based strategies, our method achieved accuracy comparable to cloud-based RAG at significantly lower inference costs. Our experiments showed that our approach delivered substantial gains across various parameter settings, datasets, and parameters, demonstrating its adaptability and scalability. Future work will focus on improving edge dataset knowledge management for graph-based retrieval, further optimizing and validating the system in larger-scale scenarios.

References

[Asai *et al.*, 2023] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*, 2023.

[Bang, 2023] Fu Bang. Gptcache: An open-source semantic cache for llm applications enabling faster answers and cost savings. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software*, pages 212–218, 2023.

[Candurkar, 2023] Sara Candurkar. Harry potter trivia ai dataset. https://huggingface.co/datasets/saracandu/harrypotter-trivia-ai-new, 2023. Accessed: 2025-02-07.

[Chang *et al.*, 2024] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

[Chen *et al.*, 2023] Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

[Chen *et al.*, 2024] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 17754–17762, 2024.

[Duvenaud, 2014] David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, Apollo - University of Cambridge Repository, 2014.

[Edge *et al.*, 2024] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[Gill *et al.*, 2024] Waris Gill, Mohamed Elidrisi, Pallavi Kalapatapu, Ali Anwar, and Muhammad Ali Gulzar. Privacy-aware semantic cache for large language models. *arXiv preprint arXiv:2403.02694*, 2024.

[Grand View Research, 2025] Grand View Research. Retrieval-augmented generation (rag) market report. https://www.grandviewresearch.com/industry-analysis/retrieval-augmented-generation-rag-market-report, 2025. Accessed: 2025-02-07.

[Gu *et al.*, 2023] Huixian Gu, Liqiang Zhao, Zhu Han, Gan Zheng, and Shenghui Song. Ai-enhanced cloud-edge-terminal collaborative network: Survey, applications, and future directions. *IEEE Communications Surveys & Tutorials*, 2023.

[Guo *et al.*, 2025] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[Hofstätter *et al.*, 2023] Sebastian Hofstätter, Jiecao Chen, Karthik Raman, and Hamed Zamani. Fid-light: Efficient and effective retrieval-augmented text generation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023.

[Jeong *et al.*, 2024] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403*, 2024.

[Jiao *et al.*, 2019] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

[Joshi *et al.*, 2017] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[Kaplan *et al.*, 2020] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[Kim *et al.*, 2023] S. Kim, K. Mangalam, J. Malik, M. W. Mahoney, A. Gholami, and K. Keutzer. Big little transformer decoder. *arXiv preprint arXiv:2302.07863*, 2023.

[Kwiatkowski *et al.*, 2019] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, pages 453–466, 2019.

[Langchain-AI, 2023] Langchain-AI. Auto evaluator: An evaluation toolkit for language models. https://github.com/langchain-ai/auto-evaluator, 2023.

[Lewis *et al.*, 2020] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 2020.

[Li *et al.*, 2022] Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 2022.

[Li *et al.*, 2024a] Jiaxing Li, Chi Xu, Feng Wang, Isaac M von Riedemann, Cong Zhang, and Jiangchuan Liu. Scalm: Towards semantic caching for automated chat services with large language models. *arXiv preprint arXiv:2406.00025*, 2024.

[Li *et al.*, 2024b] Jing Li, Zhijie Sun, Xuan He, Li Zeng, Yi Lin, Entong Li, Binfan Zheng, Rongqian Zhao, and Xin Chen. Locmoe: A low-overhead moe for large language model training. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 6377–6387. International Joint Conferences on Artificial Intelligence Organization, 2024.

[Liu *et al.*, 2024] Yuhan Liu, Hanchen Li, Yihua Cheng, Siddhant Ray, Yuyang Huang, Qizheng Zhang, Kuntai Du, Jiayi Yao, Shan Lu, Ganesh Ananthanarayanan, et al. Cachegen: Kv cache compression and streaming for fast large language model serving. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pages 38–56, 2024.

[Lyu *et al.*, 2023] Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Christopher Leung, Jiajie Tang, and Jiebo Luo. Llm-rec: Personalized recommendation via prompting large language models. *arXiv preprint arXiv:2307.15780*, 2023.

[Ma *et al.*, 2024] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 2024.

[Mallen *et al.*, 2023] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

[Meta AI, 2024] Meta AI. Llama 3.2: Revolutionizing edge ai and vision with open innovation. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/, 2024. Accessed: 2025-02-10.

[Naouri *et al.*, 2021] Abdenacer Naouri, Hangxing Wu, Nabil Abdelkader Nouri, Sahraoui Dhelim, and Huansheng Ning. A novel framework for mobile-edge computing by optimizing task offloading. *IEEE Internet of Things Journal*, 2021.

[Park *et al.*, 2023] Gunho Park, Minsub Kim, Sungjae Lee, Jeonghoon Kim, Beomseok Kwon, Se Jung Kwon, Byeongwook Kim, Youngjoo Lee, Dongsoo Lee, et al. Lut-gemm: Quantized matrix multiplication based on luts for efficient inference in large-scale generative language models. In *The Twelfth International Conference on Learning Representations*, 2023.

[Pope *et al.*, 2023] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of Machine Learning and Systems*, 2023.

[Qi *et al.*, 2021] Peng Qi, Haejun Lee, Tg Sido, and Christopher Manning. Answering open-domain questions of varying reasoning steps from text. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

[Qu *et al.*, 2024] Guanqiao Qu, Qiyuan Chen, Wei Wei, Zheng Lin, Xianhao Chen, and Kaibin Huang. Mobile edge intelligence for large language models: A contemporary survey. *arXiv preprint arXiv:2407.18921*, 2024.

[Qwen Team, 2024] Qwen Team. Qwen2.5-llm: Extending the boundary of llms. https://qwenlm.github.io/blog/qwen2.5-llm/, 2024. Accessed: 2025-02-10.

[Ren *et al.*, 2024] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. Representation learning with large language models for recommendation. In *Proceedings of the ACM on Web Conference 2024*, 2024.

[Stogiannidis *et al.*, 2023] Ilias Stogiannidis, Stavros Vassos, Prodromos Malakasiotis, and Ion Androutsopoulos. Cache me if you can: An online cost-aware teacher-student framework to reduce the calls to large language models. *arXiv preprint arXiv:2310.13395*, 2023.

[Wei *et al.*, 2022] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[Williams and Rasmussen, 2006] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

[Wu *et al.*, 2022] Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. Autoformalization with large language models. *Advances in Neural Information Processing Systems*, 2022.

[Xiao *et al.*, 2023] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, 2023.

[Xiong *et al.*, 2020] Xiong Xiong, Kan Zheng, Lei Lei, and Lu Hou. Resource allocation based on deep reinforcement learning in iot edge computing. *IEEE Journal on Selected Areas in Communications*, 2020.

[Yan *et al.*, 2024] Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

[Yang *et al.*, 2018] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[Yang *et al.*, 2023] Rui Yang, Ting Fang Tan, Wei Lu, Arun James Thirunavukarasu, Daniel Shu Wei Ting, and Nan Liu. Large language models in health care: Development, applications, and challenges. *Health Care Science*, 2023.

[Yang *et al.*, 2024] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[Yao *et al.*, 2024] Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. Cacheblend: Fast large language model serving with cached knowledge fusion. *arXiv preprint arXiv:2405.16444*, 2024.

[Yu *et al.*, 2024] Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. Evaluation of retrieval-augmented generation: A survey. *arXiv preprint arXiv:2405.07437*, 2024.

[Zhang *et al.*, 2024a] Mingjin Zhang, Jiannong Cao, Xiaoming Shen, and Zeyang Cui. Edgeshard: Efficient llm inference via collaborative edge computing. *arXiv preprint arXiv:2405.14371*, 2024.

[Zhang *et al.*, 2024b] Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K. Gupta, and Jingbo Shang. Large language models for time series: A survey. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024.

[Zhao *et al.*, 2024] Ruilin Zhao, Feng Zhao, Long Wang, Xianzhi Wang, and Guandong Xu. Kg-cot: Chain-of-thought prompting of large language models over knowledge graphs for knowledge-aware question answering. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 6642–6650. International Joint Conferences on Artificial Intelligence Organization, 2024.

[Zhu *et al.*, 2023] Banghua Zhu, Ying Sheng, Lianmin Zheng, Clark Barrett, Michael I. Jordan, and Jiantao Jiao. On optimal caching and model multiplexing for large model inference. *arXiv preprint arXiv:2306.02003*, 2023.