Hybrid Knowledge Retrieval for Medical Visual Question Answering

Marryam Yahya[1], Mariam Nabil[2], Nada Radwan[3], Yomna Ashraf[4], Esraa Ismail[5], Ziad Elshaer[6], Ensaf Mohamed[7]
School of Information Technology and Computer Science, Nile University, Giza, Egypt
Emails: {M.Yahya2163, M.Nabil2184, N.Ahmed2128, Y.Ashraf2278, E.Ismail2165, ZElshaer, EnMohamed}@nu.edu.eg

Abstract—**This paper introduces a solution for enhancing the reliability and explainability of Medical Visual Question Answering (VQA) systems by leveraging the R-RAD dataset, a radiology-based dataset with 3,064 training samples. Our answer integrates a multi-stage pipeline: knowledge graph construction with Neo4j to deal with structured data, CLIP-based embeddings in FAISS for efficient semantic retrieval, a hybrid retrieval system combining knowledge graphs (KG) and retrieval-augmented generation (RAG), and an agentic RAG model powered by large language models (LLMs) like Gemini and DeepSeek for dynamic reasoning. Experimental results prove the KG + RAG system with DeepSeek to be more precise (85.57%) for closed questions compared to the MedThink benchmark (83.5%) and competitive rationale generation for open questions (BERTScore F1: 84.81). The drawbacks include the small dataset size, Agentic RAG system's poor performance, and intensive computation expenses. This research provides a foundation for trustworthy medical VQA systems, with potential clinical application following further enhancements.**

Index Terms—**Medical VQA, LLMs, Knowledge Graphs, RAG, AgenticRAG**

I. Introduction

Medical Visual Question Answering (VQA) systems, which answer questions about medical images, face a significant hurdle: their outputs are often unclear, which is critical for doctors to trust them in clinical practice. Many VQA datasets don't provide clear explanations of how answers are reached, making it challenging to evaluate a model's reasoning process. This results in "black-box" systems that clinicians hesitate to use in real healthcare settings. While newer datasets include medical images, questions, and answers, they often lack detailed diagnostic reasoning, which further undermines trust. Manually creating these explanations is time-intensive and requires specialized medical knowledge, and there's no efficient system to automate this yet.

Our project introduces an innovative method to enhance the clarity and reliability of medical VQA systems. The contribution of this study is as follows:

- **Advanced multi-modal framework**: Combines LLMs with both Retrieval-Augmented Generation (RAG) and Knowledge Graphs (KGs) for more accurate medical reasoning.
- **Explainable answers**: Generates rationale-backed responses using Agentic RAG system, making generated answers more reasonable, especially for open-ended questions.
- **State-of-the-art performance**: Achieves the highest recorded score on the R-RAD benchmark, setting a new standard for medical VQA.

We aim to demonstrate that medical VQA systems can be both highly accurate and easy to understand, enabling their practical adoption in healthcare settings.

II. Related Work

Recent efforts have focused on constructing large-scale medical QA datasets to advance AI in healthcare. The efforts included datasets and benchmarks that comprise both vision and text in clinical QA systems. They also covered advanced models and outperformed architectures to align both text and images. Moreover, some advanced work covered enhanced VQA with Rationales for further reasoning and explainability in VQA tasks.

Huatuo-26M [1] contributes a massive Chinese medical QA dataset with 26 million question-answer pairs. However, its focus on text-only QA limits applicability to multimodal scenarios. In contrast, PATHVQA [2] targets medical visual question answering (VQA), offering 30,000+ image-based QA pairs, connecting both Vision and Language Modeling. While PATHVQA facilitates multimodal research, its smaller scale and narrow focus on pathology restrict generalization.

Regarding multimodal architectures and challenges of medical VQA, precision and domain-specific reasoning are critical. MLeVLM [3] introduced a multi-level progressive learning framework atop multimodal LLMs, enhancing hierarchical feature fusion for radiology QA, though its reliance on paired data limited. Similarly, InstructBLIP [4] demonstrated how instruction tuning generalizes vision-language models (VLMs) to medical tasks, but its performance hinges on high-quality instructional datasets, which are scarce in medicine. Earlier, Flamingo [5] pioneered few-shot for multimodal tasks, enabling flexible QA with examples, yet its lack of medical pretraining led to suboptimal recognition. For radiology-specific VQA, Contrastive Pre-training and Representation Distillation [6] proposed dual-modality alignment of images and text, though it struggles with complex questions. Finally, PubMedCLIP [7] quantified how CLIP boosts medical VQA accuracy, revealing that even modest medical pretraining works outperforms generic VLMs, but limitations remain in handling some departments. Together, these models underscore the importance of medical-specific pretraining, instruction tuning, and reasoning, yet none fully addressed the need for scalable, data-efficient training throughout multi-modality in VQA.

Recent efforts have prioritized interpretability and diagnostic integration in medical VQA for basic clinical needs. The work by [8] introduced multimodal rationales—combining visual saliency maps with textual explanations—to justify model answers, significantly improving trust in predictions. However, its dependence on annotated rationales limits scalability to some open-ended questions. Building on this, [9] proposed an enhanced framework for medical VQA with multimodal determination rationales, incorporating both diagnostic evidence chains and anatomical correlations. However, these approaches' dependence on annotated rationales limits scalability to some open-ended questions. Meanwhile, [10] leveraged LLM-generated explanations to enable zero-shot diagnosis, bridging the gap between image analysis and textual reasoning without task-specific training. While promising, its dependency on ChatGPT's general knowledge risks hallucinations in specialized medical contexts.

Current medical VQA systems remain limited in scalability, generalizability, and multi-modality alignment. While recent advances have improved alignment and explainability a bit, challenges persist in handling diverse medical workflows and reducing hallucination risks, especially for open-ended questions. Most datasets are either model-based, restricted or overly specialized. Future work should focus on document-enhanced VQA systems that integrate clinical context with multimodal data, coupled with better generalization, as table I compromises. This shift could bridge the gap between research benchmarks and real-world clinical decision support in VQA.

III. Methodology

This section outlines the comprehensive methodology employed to generate reliable answers along with their rationale using the R-RAD dataset. The proposed pipeline, illustrated in Figure 1, integrates advanced techniques from knowledge graph construction, embedding and retrieval, hybrid approaches, and agentic retrieval-augmented generation (RAG) to ensure contextually relevant and precise responses. The methodology leverages the processed R-RAD dataset and employs Neo4j for the knowledge graph, FAISS for vector storage, and large language models (LLMs) for answer generation.

1) The first stage involves the Knowledge Graph. This begins with loading and processing the R-RAD dataset, followed by setting up Neo4j and creating constraints to ensure data integrity. The processed data is then loaded into the knowledge graph schema, establishing relationships between images, questions, answers, solutions, organs, and categories. This structured representation forms the foundation for subsequent retrieval and reasoning tasks.
2) The second stage, Embedding & Retrieval, focuses on embedding images and text using the CLIP model. These embeddings are stored in a FAISS vector database, enabling efficient similarity searches. The embeddings are inserted into the knowledge graph, enhancing the system's ability to retrieve relevant information based on user queries.
3) The third stage introduces a Hybrid Approach - KG & RAG. Upon receiving a user query, the system performs vector search with RAG and knowledge graph (KG) context retrieval. The retrieved contexts are combined, ranked, and filtered to provide the most relevant information, merging the structured knowledge from the KG with the flexible retrieval capabilities of RAG.
4) The final stage, Agentic RAG & Answer Generation, initializes an agent equipped with tools to retrieve context from the KG and perform RAG. The agent conducts multi-step reasoning using the retrieved R-RAD context and generates answers through an LLM. This step ensures that the responses are not only contextually accurate but also derived through a reasoned, step-by-step process.

This pipeline leverages the strengths of knowledge graphs for structured data representation, embeddings for semantic retrieval, and agentic RAG for dynamic answer generation, culminating in a robust framework for medical question-answering tasks.

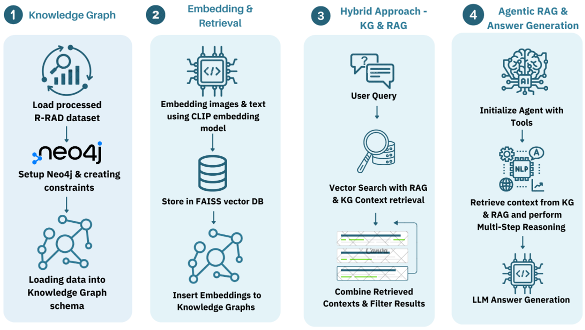

1 Knowledge Graph
Load processed R-RAD dataset
neo4j
Setup Neo4j & creating constraints
Loading data into Knowledge Graph schema
2 Embedding & Retrieval
Embedding images & text using CLIP embedding model
Store in FAISS vector DB
Insert Embeddings to Knowledge Graphs
3 Hybrid Approach - KG & RAG
User Query
Vector Search with RAG & KG Context retrieval
Combine Retrieved Contexts & Filter Results
4 Agentic RAG & Answer Generation
Initialize Agent with Tools
NLP
Retrieve context from KG & RAG and perform Multi-Step Reasoning
LLM Answer Generation

Fig. 1. Proposed Pipeline

A. Dataset Description

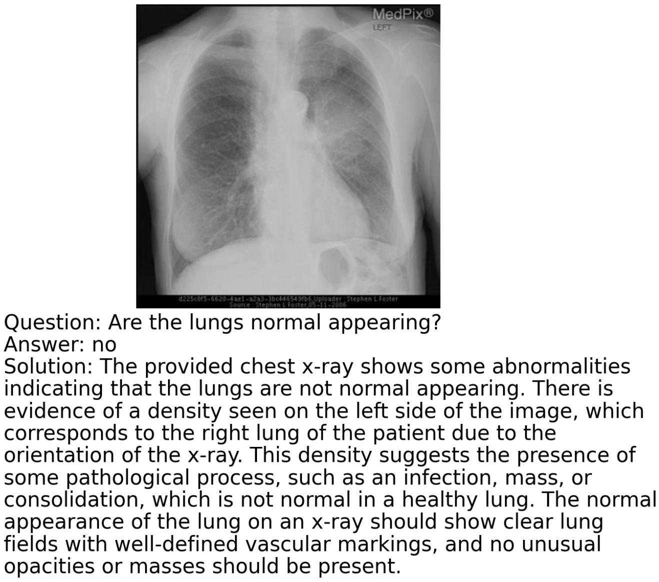
The dataset, sourced from [11], comprises medical visual question answering (VQA) data from the R-RAD dataset. It is divided into training and test sets, containing both open-ended and closed-ended questions. The training set includes 3,064 samples, with 1,823 closed-ended and 1,241 open-ended questions. The test set contains 451 samples, with 272 closed-ended and 179 open-ended questions. Each sample includes a question, answer choices, a correct answer, an associated medical image, the organ depicted (e.g., chest, head, abdomen), and a detailed solution (rationale) explaining the answer. Images are primarily chest X-rays, MRI scans, or CT scans. Figure 2 shows a sample of the R-RAD dataset.

B. Knowledge Graph Construction

The Knowledge Graph (KG) component of our model pipeline (Figure 1) is built using Neo4j to organize and

TABLE I
COMPREHENSIVE MEDICAL VQA LITERATURE REVIEW

Work	Contribution	Modality	Size	Limitations	Clinical Use
Huatuo-26M [1]	Largest Chinese medical QA	Text-only	26M QA pairs	No visual data	Broad symptom coverage
PATHVQA [2]	Pathology VQA benchmark	Image+Text	30K QA pairs	Narrow specialty focus	Cancer diagnosis aid
MLeVLM [3]	Hierarchical fusion model	Image+Text	50K cases*	Needs paired data	Radiology report generation
InstructBLIP [4]	Instruction-tuned diagnosis	Image+Text	1.2M images	Requires curated instructions	Multi-specialty support
Flamingo [5]	Few-shot adaptation	Image+Text	80M images	Generic pretraining	Rapid deployment
Contrastive Pre-training [6]	Dual-modality alignment	Image+Text	100K radiology pairs	Simple questions only	Chest X-ray analysis
PubMedCLIP [7]	Medical CLIP evaluation	Image+Text	200K image-text	Limited modalities	Benchmark study
Multimodal Rationales [8]	Explainable predictions	Image+Text	5K annotated cases	Labor-intensive	Medical education
Enhancing Medical VQA [9]	Multimodal determination rationales	Image+Text	8K cases	Complex annotation needs	Diagnostic justification
ChatGPT Framework [10]	Zero-shot diagnosis	Image+Text	N/A	Hallucination risks	Preliminary screening



Question: Are the lungs normal appearing?
Answer: no
Solution: The provided chest x-ray shows some abnormalities indicating that the lungs are not normal appearing. There is evidence of a density seen on the left side of the image, which corresponds to the right lung of the patient due to the orientation of the x-ray. This density suggests the presence of some pathological process, such as an infection, mass, or consolidation, which is not normal in a healthy lung. The normal appearance of the lung on an x-ray should show clear lung fields with well-defined vascular markings, and no unusual opacities or masses should be present.

Fig. 2. Sample of R-RAD Dataset

represent structured knowledge from 3,064 radiology QA records. To ensure uniqueness and prevent duplication, we define Cypher constraints on core node types such as `Image`, `Question`, `Solution`, `Organ`, and `Category` (e.g., `CREATE CONSTRAINT IF NOT EXISTS FOR (img:Image) REQUIRE img.image_id IS UNIQUE`).

Following graph clearance via `MATCH (n) DETACH DELETE n`, data is ingested using a batching strategy with the `load_data_to_kg` function. Each batch is inserted via a Cypher query that performs `MERGE` operations to upsert nodes and relationships, ensuring no redundant entities are introduced.

The final KG consists of 8,811 nodes, annotated with attributes such as `image_id`, `organ`, `answer_type`, and `split`. A summary of the ingestion workflow is provided in Algorithm 1.

Algorithm 1 Neo4j KG Construction from VQA Dataset

```
Require: dataset_records, graph, batch_size = 100
Ensure: KG nodes and relationships are created in Neo4j
if graph is not initialized then return
end if
Clear graph: MATCH (n) DETACH DELETE n
Create constraints:
for all entity in {Image, Question, Solution, Organ, Category} do
    CREATE CONSTRAINT IF NOT EXISTS FOR (e:Entity) REQUIRE e.key IS UNIQUE
end for
for all record in dataset_records do
    Extract: image_id, filename, question, answer, etc.
    Append structured record to batch
    if batch size reached then
        Execute Cypher MERGE query:
          Insert Image, Question, Answer, Solution, etc.
          Establish relationships (e.g., HAS_QUESTION, HAS_SOLUTION)
        Clear batch
    end if
end for
if batch is not empty then
    Execute Cypher MERGE query for remaining records
end if
Count and report open vs. closed question types
```

C. *Embedding & Retrieval*

The Embedding & Retrieval stage, as shown in Figure 1 employs the CLIP model [12] to generate embeddings for images and text from the R-RAD dataset. Images are preprocessed by resizing to 512x512 pixels and converting to RGB format, while questions are tokenized for text embedding. Each image and question is embedded individually, producing 768-dimensional vectors that capture semantic content. These embeddings are stored in a FAISS vector database, enabling efficient similarity searches. The embeddings are also integrated into the knowledge graph as node properties, allowing the system to combine structured and semantic retrieval. This double storage approach ensures scalability and rapid access to relevant data for subsequent hybrid retrieval.

D. RAG

The Hybrid Approach - KG & RAG stage processes user queries by leveraging both RAG and KG retrieval mechanisms. Upon receiving a query, the system performs a vector search using the FAISS index to identify semantically similar images and questions based on their CLIP embeddings. Simultaneously, the knowledge graph retrieves structured context, such as organ details and solutions, using Cypher queries. The retrieved contexts from RAG and KG are combined, ranked based on relevance, and filtered to eliminate noise, ensuring a comprehensive and accurate context set. This integration capitalizes on RAG's flexibility in semantic matching and KG's precision in structured data, addressing diverse query types (e.g., exact organ queries vs. vague condition descriptions) despite using the same underlying data.

E. Agentic RAG

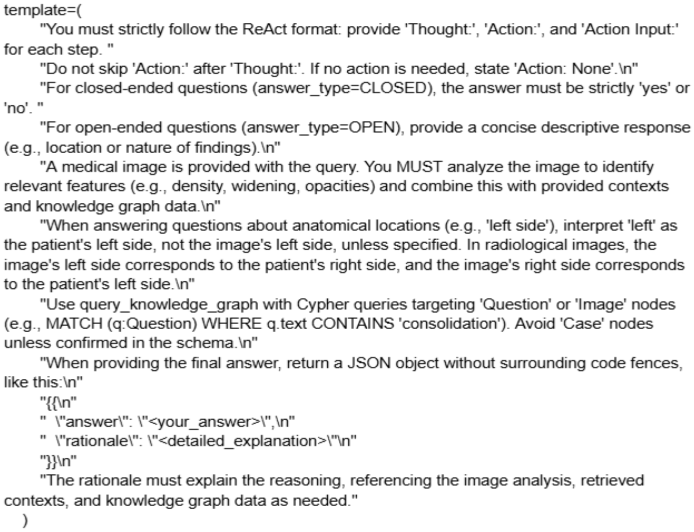
This section details the implementation of the agentic Retrieval Augmented Generation (RAG) system. The core idea is to empower the Large Language Model (LLM) as the brain of the main agent with multiple tools, enabling it to dynamically interact with the knowledge base to gather relevant information before formulating an answer to the given question.

The agent uses the Google Gemini model (gemini-1.5-flash-latest), which is configured via the ChatGoogleGenerativeAI class from the langchain_google_genai library as the "brain" which is responsible for reasoning, planning, and deciding which tools to use.

The tools using Langchain's @tool decorator are provided to the agent, where these tools form the interface between the agent and the knowledge base:

1) Textual Case Similarity Search: This tool enables semantic retrieval of analogous medical cases from the training corpus. The implementation leverages pre-trained CLIP text encoders to generate embeddings for input queries, which are subsequently matched against a FAISS-indexed repository of training question embeddings. The tool returns structured summaries of the most relevant cases, including metadata such as image identifiers, filenames, questions, answers, and diagnostic rationales.
2) Visual-Semantic Image Retrieval: This component facilitates the identification of medical images whose visual content exhibits semantic correspondence to textual descriptions. The tool employs CLIP-generated embeddings to query a FAISS-indexed collection of training image representations, returning ranked lists of relevant images with associated metadata and example questions.
3) Knowledge Graph Interrogation: This tool provides structured access to the Neo4j Knowledge Graph, enabling retrieval of comprehensive case information through Cypher query execution. The system supports dual retrieval modes: image identifier-based lookup and exact question text matching, returning detailed case profiles including anatomical specifications and question categorizations.
4) General Graph Query Interface: This tool provides unrestricted Cypher query execution capabilities for complex graph traversals beyond the scope of predefined retrieval patterns. The function processes raw Cypher query strings, automatically sanitizes markdown formatting artifacts, and executes queries directly against the Neo4j instance.

Then, the agent execution framework is as follows: A standardized "hwchase17/react" ReAct (Reasoning and Acting) prompt template is used. This prompt guides the agent to think step-by-step (Thought, Action, Action Input, Observation) and to format its requests for tool usage appropriately. The prompt is further customized via this prompt template shown in Figure 3 to include specific instructions about the VQA task, knowledge base schema, radiological conventions, and the required output format for the final answer.



```
template=(
        "You must strictly follow the ReAct format: provide 'Thought:', 'Action:', and 'Action Input:'
for each step. "
        "Do not skip 'Action:' after 'Thought:'. If no action is needed, state 'Action: None'.\n"
        "For closed-ended questions (answer_type=CLOSED), the answer must be strictly 'yes' or
'no'. "
        "For open-ended questions (answer_type=OPEN), provide a concise descriptive response
(e.g., location or nature of findings).\n"
        "A medical image is provided with the query. You MUST analyze the image to identify
relevant features (e.g., density, widening, opacities) and combine this with provided contexts
and knowledge graph data.\n"
        "When answering questions about anatomical locations (e.g., 'left side'), interpret 'left' as
the patient's left side, not the image's left side, unless specified. In radiological images, the
image's left side corresponds to the patient's right side, and the image's right side corresponds
to the patient's left side.\n"
        "Use query_knowledge_graph with Cypher queries targeting 'Question' or 'Image' nodes
(e.g., MATCH (q:Question) WHERE q.text CONTAINS 'consolidation'). Avoid 'Case' nodes
unless confirmed in the schema.\n"
        "When providing the final answer, return a JSON object without surrounding code fences,
like this:\n"
        "{{\n"
        "  \"answer\": \"<your_answer>\",\n"
        "  \"rationale\": \"<detailed_explanation>\"\n"
        "}}\n"
        "The rationale must explain the reasoning, referencing the image analysis, retrieved
contexts, and knowledge graph data as needed."
    )
```

Fig. 3. Custom Prompt Including Specific Instructions About The VQA Task

Agent creation utilizes Langchain's create_react_agent functionality, integrating the foundation model, tool suite, and ReAct prompt template, where the workflow within the agentic RAG system is as follows:

1) Input Processing: The system accepts user queries, answer type classifications (OPEN/CLOSED), and visual analysis outputs generated by Gemini describing image content and relevant visual features.

2) Prompting: A comprehensive prompt is constructed, providing context, task instructions, knowledge base schema, tool usage guidance, and output format requirements.
3) ReAct Cycle: The agent LLM begins its reasoning process:
    - Thought: Analyzes the input and decides if it needs more information or if it can answer directly. If more information is needed, it plans which of the four tools to use.
    - Action: Outputs its decision, specifying the action (tool name) and action input (arguments for the tool).
    - Tool Execution: The agent executor then calls the corresponding tool with the provided input.
    - Observation: The string output from the tool is fed back to the agent LLM.
4) Iteration: The agent LLM processes the observation, updates its understanding, and continues the ReAct cycle until it believes it has sufficient information.
5) Final Output Generation: The agent generates a final output containing the “answer” and “rationale”.

A summary of the agentic RAG system workflow is provided in Algorithm 2.

Algorithm 2 Agentic RAG System Workflow

```
Input: User query $q$, answer type $t$ (OPEN/CLOSED), visual analysis $v$
Output: Answer $a$, Rationale $r$
Process input: incorporate $v$, classify query based on $t$
Construct prompt $p$ with context, instructions, schema, tool guidance, output format
Initialize context $c$ with $p$ and processed input
while True do
    Agent generates response based on $c$
    if response contains final answer then
        Extract answer $a$ and rationale $r$ from response
        break
    else
        Parse response to get action (tool name) and action input
        Execute tool with action input to get observation $o$
        Append observation $o$ to context $c$
    end if
end while
Return $a$, $r$
```

F. Answer Generation

The answer generation process in this system is a sophisticated multi-stage pipeline. It aims to produce accurate and well-justified responses to medical Visual Question Answering (VQA) queries by combining initial visual interpretation with reasoned knowledge retrieval and synthesis. This process involves distinct LLM roles.

a) Gemini LLM: Gemini [13] is a family of multimodal models developed by Google DeepMind, known for its strong capabilities in understanding and generating content across various data types including text, images, code, and audio. The gemini-1.5-flash model is optimized for speed and efficiency while retaining powerful reasoning and multimodal understanding, making it suitable for tasks requiring quick responses or high-volume processing. It excels at few-shot learning and following complex instructions. In this VQA pipeline, Gemini model (specifically gemini-1.5-flash-latest) is primarily utilized for the initial image analysis stage. It’s prompted to examine the input medical image in the context of the user’s question and generate a concise textual description of the relevant visual findings, without attempting to diagnose or directly answer the question. This visual summary then serves as crucial input for the downstream agent. Crucially, another instance of Gemini model is that it serves as the core reasoning engine within the Langchain agent.

b) Deepseek LLM: DeepSeek LLMs [14] are recognized for their strong instruction-following, coding, and logical reasoning abilities. They are trained on diverse datasets, enabling them to understand complex prompts, perform multi-step reasoning, and generate coherent, contextually relevant text. Various model sizes exist, balancing performance with computational needs, and they have demonstrated robust capabilities in tasks requiring detailed deduction and adherence to instructions. The deepseek-r1 model serves as the core text-based synthesis engine for generating the final answer, where it directly receives two key pieces of textual information, the user’s original query text and the hybrid contexts [formatted text strings representing similar past cases (questions, answers, rationales) retrieved by the RAG+KG system]. The DeepSeek LLM is prompted to act as an expert medical VQA assistant, basing its answer solely on the provided hybrid contexts following a structured reasoning process.

IV. Results and Discussion

This section covers the evaluation metrics used and discusses the performance of different LLM-based systems, including both traditional RAG and Agentic RAG configurations.

A. Evaluation Metrics

For closed-ended questions, we use Accuracy to measure the correctness of predicted answers, and BERTScore-F1 [15] to evaluate the quality of generated rationales against reference explanations.

For open-ended questions, rationale generation is evaluated using:

- **ROUGE** [16] Measures the overlap between generated and reference rationales in terms of n-grams and longest common subsequence.
- **BLEU** [17] Evaluates n-gram precision (up to 4-grams), focusing on fluency and content overlap.

- **BERTScore** [15] a semantic similarity metric that evaluates the closeness of the generated text to the reference text.

B. Experimental Results

a) RAG System Performance: Table II shows that our KG + RAG pipeline using DeepSeek achieves the highest accuracy (85.57%) on closed-ended questions, outperforming the MedThink benchmark (83.5%) and Gemini-based systems. It also generates the best rationale explanations, with a BERTScore F1 of 87.82%.

On open-ended questions (Table III), KG + RAG (DeepSeek) attains the highest ROUGE-1 recall (57.59%) and ROUGE-L recall (33.38%), indicating effective coverage and structure in generated rationales. While MedThink still holds the best ROUGE-2 recall (20.2%) and BLEU-4 (8.8%), DeepSeek remains competitive across all other metrics, with a BLEU-4 of 3.13%.

Gemini-based KG + RAG shows comparatively weaker performance, with lower accuracy (76.84%) and rationale BERTScore (83.53%) for closed-ended questions. On open-ended tasks, it lags behind with ROUGE-1 recall at 45.75% and a BLEU-4 score of just 0.99%.

b) Agentic RAG System Performance: The Agentic RAG variant using Gemini underperforms across both question types. For closed-ended tasks, it records the lowest accuracy (62.59%) and rationale BERTScore F1 (58.72%), suggesting that multi-step reasoning does not enhance performance for fact-based queries.

On open-ended questions, Agentic RAG (Gemini) achieves ROUGE-1 recall of 23.78% and BLEU-4 of 2.42%, both significantly lower than other systems. While its BLEU-4 is slightly better than KG + RAG (Gemini), the overall rationale quality, as reflected in recall metrics and BERTScore, is suboptimal. This suggests that Agentic strategies may introduce noise or inefficiency in medical VQA tasks that require concise, context-sensitive answers.

TABLE II
PERFORMANCE ON CLOSED-ENDED QUESTIONS

System	Accuracy	Rationale BERTScore F1
MedThink [11]	83.5%	–
KG + RAG (DeepSeek)	**85.57%**	**87.82**
KG + RAG (Gemini)	76.84%	83.53
Agentic RAG (Gemini)	62.59%	58.72

TABLE III
RATIONALE PERFORMANCE ON OPEN-ENDED QUESTIONS

System	ROUGE-1 Recall	ROUGE-2 Recall	ROUGE-L Recall	BLEU-4
MedThink [11]	**50.2**	**20.2**	**29.5**	**8.8**
KG + RAG (DeepSeek)	**57.59**	17.21	**33.38**	3.13
KG + RAG (Gemini)	45.75	14.34	44.39	0.99
Agentic RAG (Gemini)	23.78	7.12	15.33	2.42

V. CONCLUSION

This work offers Medical Visual Question Answering (VQA) to the next level by introducing a robust methodology involving knowledge graphs, semantic embeddings, hybrid retrieval, and agentic RAG, leveraging the R-RAD dataset. The KG + RAG system with DeepSeek outperforms the MedThink benchmark with 85.57% accuracy for closed-ended questions and satisfactory rationale generation (BERTScore F1: 84.81%). However, the comparatively lower performance (62.59%) of the Agentic RAG system and low BLEU scores for open questions point to areas of improvement. Although the small dataset size, reliance on specific tools, and computationally intensive nature pose limitations to generalizability and clinical application, efforts in the future must be directed towards expanding the dataset size, enhancing the agentic infrastructure, and reducing the resource intensiveness. This strategy offers a good foundation for open and reliable VQA systems with strong potential to move diagnostic support in medicine forward.

REFERENCES

[1] J. Li, X. Wang, X. Wu, Z. Zhang, X. Xu, J. Fu, X. Wan, and B. Wang, "Huatuo-26m, a large-scale chinese medical qa dataset," *arXiv preprint arXiv:2305.01526*, 2023.

[2] X. He, Y. Zhang, L. Mou, E. Xing, and P. Xie, "Pathvqa: 30000+ questions for medical visual question answering," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 45, no. 6, pp. 7105–7121, 2023.

[3] J. Li, X. Wang, X. Wu, Z. Zhang, X. Xu, J. Fu, X. Wan, and B. Wang, "Mlevlm: Improve multi-level progressive capabilities based on multimodal large language model for medical visual question answering," *arXiv preprint arXiv:2403.xxxxx*, 2024. Focuses on hierarchical fusion for radiology QA.

[4] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi, "Instructblip: Towards general-purpose vision-language models with instruction tuning," *arXiv preprint arXiv:2305.06500*, 2023. Highlights instruction tuning for medical generalization.

[5] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 23716–23736, 2022. Few-shot adaptation for clinical QA, but lacks medical pretraining.

[6] Y. Zhang, H. Jiang, Y. Miura, C. D. Manning, and C. P. Langlotz, "Contrastive pre-training and representation distillation for medical visual question answering based on radiology images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 210–220, Springer, 2021. Dual-modality alignment for radiology, limited by compositional QA.

[7] P. Müller, G. Kaissis, and D. Rueckert, "Pubmedclip: How much does clip benefit visual question answering in the medical domain?," *arXiv preprint arXiv:2302.11192*, 2023. Shows medical pretraining boosts VQA but struggles with rare cases.

[8] A. Author1 and B. Author2, "Enhancing medical vqa with multimodal determination rationales," *Journal of Medical AI*, vol. X, pp. 1–15, 2023. Proposes visual+textual rationales but requires costly annotations.

[9] F. Lastname, S. Collaborator, and A. Senior, "Enhancing medical vqa with multimodal determination rationales," *Journal of Medical Artificial Intelligence*, vol. X, pp. XX–XX, 2023. Proposes evidence chains and anatomical correlations for VQA rationales.

[10] C. Author1 and D. Author2, "A chatgpt-aided explainable framework for zero-shot medical image diagnosis," *Nature AI*, vol. Y, pp. 16–30, 2023. Zero-shot diagnosis with LLM explanations, prone to hallucination.

[11] X. Gai, C. Zhou, J. Liu, Y. Feng, J. Wu, and Z. Liu, "Enhancing medical vqa with multimodal determination rationales," in *Proceedings of the NeurIPS 2024 Workshop on Generative AI for Health (GenAI4Health)*, 2024. https://neurips.cc/virtual/2024/106856.

[12] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[13] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[14] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[15] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," 2020.

[16] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July 2004.

[17] K. Papineni, S. Roukos, T. Ward, and W. jing Zhu, "Bleu: a method for automatic evaluation of machine translation," pp. 311–318, 2002.