Med-VQA-KGRAG: Enhancement Of Medical Responses By VLMs Using Knowledge Graphs and RAG

Arghyadeep Das
arghyadeepda@umass.edu

Nilesh Nayan
nnayan@umass.edu

Roshita Bhonsle
rbhonsle@umass.edu

Shantanu Todmal
stodmal@umass.edu

1 Problem statement

Recent advancements in large language models (LLMs) have greatly impacted various fields, particularly in medicine. Trained on extensive medical corpora, these models exhibit significant potential in aiding clinical decision-making, diagnosis, patient education, the training of medical students, and medical research. However, despite their impressive generative capabilities, LLMs frequently experience hallucinations, leading to factually incorrect outcomes. In a high-stakes area like medicine, ensuring factual accuracy is crucial. To alleviate these risks, factual grounding mechanisms are essential to improve the reliability of these models. Within the realm of **"open-ended" visual question-answering (VQAs)** for medicine, we propose an approach named **Med-VQA-KGRAG** for factual grounding, which integrates Knowledge Graphs (KGs) and Retrieval-Augmented Generation (RAGs). While the multimodal encoder model embeds the input medical image and question, we hypothesize that the KG will provide a rich, structured, and verified external medical knowledge base to the RAG, thereby enhancing the factual grounding of the Med-VLM's response by dynamically retrieving and citing relevant medical literature and ensuring responses are consistent with established medical knowledge. By integrating multi-modal medical data with external retrieval sources, we aim to improve the trustworthiness and clinical utility of AI-driven medical assistants. Please find our code on GitHub https://github.com/arghyadeep99/Med-VQA-RAG.

2 What you proposed vs. what you accomplished

While the original idea has been implemented as proposed, there were a few different parts of the pipeline that we had to replace to make things work. For VLMs, we proposed to use models like LlaVa-Med (Li et al. (2023) and Med-Flamingo (Moor et al. (2023)). However, we ran into some issues to run them (mostly HuggingFace-side like missing model indexes issues as we learned from the HuggingFace forums). We learnt that BioMed-CLIP is an encoder-only model and it needs to be combined with a decoder LM to be properly used as an end-to-end VLM. We used BioMed-CLIP to instead generate a multimodal vector index for the datasets we used. We finally succeeded in running our pipeline with the following VLMs: LlaVA-RAD, CheXAgent, LLama 3.2 Vision Instruct 11B.

- ☑ Collect and Pre-Process dataset.
- ☑ Build and Train Baseline VLMs on collected dataset and examine its performance.
- ☑ Build a Multimodal Vector Index using FAISS on MIMIC-CXR dataset.
- ☑ Build our own Knowledge Graph on ReXGradient-160K reports and also utilize PrimeKG.
- ☑ Utilize new-generation metrics from DeepEval to analyze our pipeline's results in addition to traditional metrics like ROUGE, BLEU, METEOR, BERTScore, etc. to show their relevance
- ☑ Perform in-depth analysis of our experimental and ablation results to figure out what kinds of examples our approach struggles with.
- ☒ Using MLLM-as-a-Judge for Evaluation of the Generated Answer with respect to Ground Truth - used LLM-as-a-Judge instead due to budget constraints.

3 Related Work

In recent years, retrieval-augmented generation (RAG) has emerged as a powerful paradigm for grounding large-language-model (LLM) outputs in external knowledge sources. The canonical RAG framework Lewis et al. (2021) retrieves dense vector embeddings of text passages and conditions an LLM on those retrieved contexts to improve factuality and relevance. However, purely vector-based retrieval often struggles with compositional or multi-hop queries, and fails to capture structured relationships between entities.

To address these shortcomings, GraphRAG Edge et al. (2025) constructs a hierarchical knowledge graph over text fragments using community-detection algorithms, and then performs breadth-first and depth-first traversals to retrieve semantically linked nodes before passing them to an LLM. While this method significantly enhances reasoning over complex, interlinked concepts, it is currently limited to text-only data and single modalities. Building on this, LightRAG Guo et al. (2024) proposes a hybrid retrieval algorithm that fuses flat-index vector search with graph-based expansion to reduce latency and token usage, outperforming GraphRAG on both efficiency and accuracy benchmarks.

One of the latest works in the KG-RAGs space has been the combination of Graph Neural Networks (GNNs) with RAGs, called **GNN-RAG** (Mavromatis and Karypis (2024)). Due to GNNs' ability to perform dense graph retrievals and its analysis, GNN-RAG has shown great performance in multi-hop, multi-entity retrieval. GNNs excel at graph-based reasoning while LLMs excel in answering due to natural language understanding (NLU) capabilities. We ran into multiple issues trying to replicate this work as one possible RAG approach, right from installation to how to exactly transform existing datasets/knowledge graphs into GNN-RAG, etc.

In **HybridRAG** (Sarmah et al. (2024)), authors have tried to combine vanilla RAG with GraphRAG in parallel to leverage the strengths of both methods on financial "text" data. When the output from both the RAGs is obtained, they just concatenate the GraphRAG output to the vector RAG one and provide it to the LLM for final response generation. However, it does not address multimodal question-answering and focuses on financial datasets. We are using a multimodal retriever for retrieving information from vector embeddings, but later feed the retrieved top-k documents to graph-based RAG and then both of them are fed to the Med-VLM for improved context and knowledge for generating a response. HybridRAG's approach however, is partially used in our ablation studies where we evaluate responses if either only multimodal RAG or only graph-based RAG is used as context for the VLMs.

Parallel efforts have explored purely graph-based retrieval for the biomedical domain. **MMed-dRAG** (Xia et al. (2025)), introduces a versatile multimodal RAG system specifically designed for Med-VLMs to generate more factual responses. It incorporates a domain-aware retrieval mechanism, an adaptive method for selecting the number of retrieved contexts, and a RAG-based preference fine-tuning strategy to improve cross-modality alignment and overall alignment with ground truth. However, MMed-RAG does not use knowledge graphs, while our approach does. Moreover, in our approach, we do not use a "domain-aware" retrieval mechanism since we have only one domain - chest X-rays.

KGARevion (Su et al. (2025)) is a knowledge graph-based agent that answers knowledge-intensive biomedical questions by generating relevant triplets using an LLM and then verifying these triplets against a grounded knowledge graph. This multi-step process strengthens reasoning and adapts to different models of medical inference, outperforming retrieval-augmented generation-based approaches that lack effective verification mechanisms. However, KGARevion works on the domain of text-only medical QA and does not take any image as input.

MedGraphRAG (Wu et al. (2024)) is a graph-based RAG framework for the medical domain, that employs Triple Graph Construction to link user documents to credible medical sources and vocabularies, along with a U-Retrieval technique that combines top-down precise retrieval with bottom-up response refinement. We could not use this method in our pipeline because the books layer is behind a paywall as it is created from purchased medical textbooks. However, we essentially borrow the idea of triple linking as double linking, where we still have the top layer from reports, and second layer as medical definition and vocabularies.

MedRAG (Zhao et al. (2025)) constructs a

four-tier hierarchical diagnostic KG encompassing critical diagnostic differences of various diseases and integrates these differences with similar Electronic Health Records (EHRs) retrieved from a database. Unlike this paper, we do not integrate or enhance our disease nodes with knowledge with EHR records, but build a complete knowledge graph out of external chest reports (ReXGradient-160K) for domain knowledge. We also do not perform hierarchical clustering before building the knowledge graph, since we already have the relations from the PrimeKG knowledge graph.

Since we are specifically working with chest X-rays and their reports, our problem is in some ways related to the report generation problem, though not exactly. For the purpose of this project, we limit our main focus to Visual Question-Answering (VQA) benchmarks. The paper by Liu et al. (2021b) builds a Posterior Knowledge Explorer (PoKE) using the image and word embeddings from the reports by passing them through the transformer layers consisting of multi-head attention and feed forward networks, a Prior Knowledge Explorer (PrKE) uses report embeddings plus graph embeddings in addition to prior knowledge for the image and then finally a Multi-Domain Knowledge Distiller (MKD) distills accurate prior and posterior knowledge and adaptively merges them to generate accurate reports. The paper shares common ideas with our approach in terms of using multimodal vector embedding, graph embedding and vector embedding on a high-level, however the pipeline remains fundamentally different since we don't pass our embeddings at any stage before the VLM to any transformer layers explicitly. In our case, we use a multimodal embedding model to build multimodal embeddings and store them in FAISS (Douze et al. (2024)), which the multimodal retriever uses for fetching top-k relevant documents using Approximate Nearest Neighbor (ANN) (Indyk and Motwani (1998)) via HNSW (Malkov and Yashunin (2018)). These documents are then used as context for the graph retriever to extract information from the knowledge graphs.

These works highlight the growing importance of integrating structured knowledge from knowledge graphs into RAG frameworks to address the challenges of factuality, reasoning, and reliability in medical question answering and other medical AI tasks.

4 Dataset

The different components of our pipeline utilized the following datasets:

4.1 Multimodal Vector Database

To construct the multimodal vector database, we utilized publicly available chest X-ray report generation dataset MIMIC-CXR:

1. **MIMIC-CXR** (Johnson et al. (2019)) is a comprehensive dataset comprising 377,110 chest X-ray images linked to 227,835 imaging studies from 65,379 patients admitted to the Beth Israel Deaconess Medical Center between 2011 and 2016 [Figure 1].

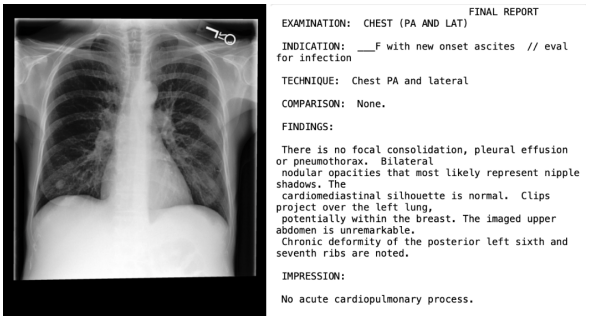

Figure 1: Example of a Image-Report Pair from the MIMIC-CXR dataset

These are well-suited for our task of embedding medical text and images into the FAISS vector database to prepare our multimodal vector index.

4.2 Knowledge Graph

For the knowledge graph (KG) component, we used one each of the following as proposed:

ReXGradient-160K: We used the ReXGradient-160K (Zhang et al. (2025)) reports dataset to prepare our reports' knowledge graph. ReXGradient-160K is the largest publicly available multi-site chest X-ray dataset, containing 273,004 unique chest X-ray images from 160,000 radiological studies, collected from 109,487 unique patients across 3 U.S. health systems (79 medical sites). We used the validation split from this dataset for our knowledge graph construction, which has about 10,000 detailed radiology reports across 6,964 patients. The report has 4 components: *Indication, Comparison, Findings* and *Impression* [Figure 2].

We leveraged the graph creation component of Microsoft GraphRAG (Edge et al. (2025)) to do

Study Description: DG CHEST 1V PORT
Indication: Postop from a ICD placement. Congestive heart failure and atrial fibrillation.
Comparison: 04/30/1995
Findings: New dual lead transvenous pacemaker is seen in appropriate position. No evidence of pneumothorax. Cardiomegaly stable. Decreased symmetric airspace disease, consistent with decreased pulmonary edema. Stable small bilateral pleural effusions and bibasilar atelectasis. Mild asymmetric airspace disease is also seen in the right upper lobe, and pneumonia cannot be excluded.
Impressions: New dual lead transvenous pacemaker is seen in appropriate position. No evidence of pneumothorax. Cardiomegaly stable. Decreased symmetric airspace disease, consistent with decreased pulmonary edema. Stable small bilateral pleural effusions and bibasilar atelectasis. Mild asymmetric airspace disease is also seen in the right upper lobe, and pneumonia cannot be excluded.

Figure 2: Sample Report from ReXGradient-160K Dataset

this from text report documents, where both local and global index search was enabled [Figure 3].

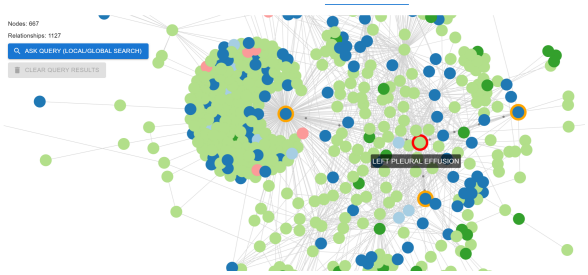

Figure 3: An example from GraphRAG-based KG on ReXGradient-160K

PrimeKG: We used the PrimeKG (Chandak et al. (2023)) knowledge graph, which integrates 20 high-quality resources to describe 17,080 diseases with 4,050,249 relationships representing ten major biological scales, including disease-associated protein perturbations, biological processes and pathways, anatomical and phenotypic scales, and the entire range of approved drugs with their therapeutic action, considerably expanding previous efforts in disease-rooted knowledge graphs. We only used a *subgraph* from this, owing to the online hosting constraints to fully run our pipeline end-to-end that has 35,551 relevant nodes and 367,293 relationships [Figure 4].

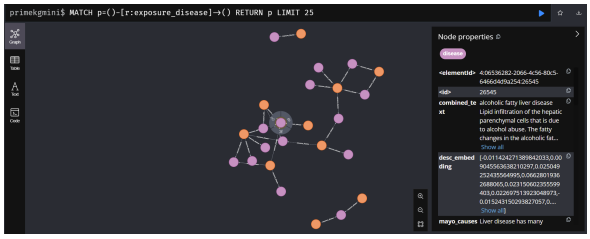

Figure 4: A sample from the PrimeKG on neo4j showing disease-exposure relation, and expanded property details of disease node

4.3 Benchmarking

To evaluate our system's performance, we utilized Medical Visual Question Answering (Med-VQA) datasets like: VQA-RAD and SLAKE. These datasets are specifically designed to assess the capabilities of models in interpreting medical images and question-answering related tasks.

VQA-RAD (Lau et al. (2018)) comprises 3,515 question–answer pairs associated with 315 radiology images. It encompasses a diverse array of clinical questions, thereby serving as a valuable resource for evaluating the accuracy of Med-VQA models. We filtered for chest X-rays and open-ended questions, which resulted in a final dataset over 283 questions [Figure 5].

Q: What is one abnormality that can be seen in the image?
A: Increased opacity in the left retrocardiac region

Q: How was this film taken?
A: PA

Q: The cystic findings in the bilateral apices are consistent with what pathology?
A: emphysema

Figure 5: VQA-RAD Q/A Samples

SLAKE (Liu et al. (2021a)) is a bilingual English-Chinese dataset consisting of 642 images and 14,028 question-answer pairs. It includes both vision-only and knowledge-based questions, covering various medical scenarios. We sampled only the English questions, and then filtered for open ended questions for chest x-ray images, which resulted in 600 samples.

4.4 Data Preprocessing

1. **PrimeKG:** For this, we had to fix the column headers for the relations.csv file to be able to establish relationships when we load it into the neo4j database. Additionally, the knowledge graph was enriched by combining the disease features and drug features into the graph nodes.

2. **ReXGradient-160K, VQA-RAD, SLAKE:** Basic text preprocessing like converting to lowercase, trimming whitespaces, etc. were performed.

5 Baselines

While there is some ambiguity in identifying state-of-the-art models for open-ended visual question answering as every VLM we came across compared themselves on different kinds of tasks and

not specifically for chest X-ray based VQA. We identified that across the spectrum, models like PeFoMed, LlaVa-Med, BioMedCLIP, etc. are supposedly state-of-the-art models. While trying out these models, we faced multiple issues in terms of some models not properly maintained, missing keys or tensors, memory explosion, lack of quantization opportunities, etc. due to our hardware limitations. We still managed to get the following models working on *Vanilla* VQA on both benchmarking datasets and they serve as our baseline: LLaMa Vision Instruct 3.2 (11B), LlaVa-RAD and CheXAgent.

The data type for all the tensors to these models was BF16 as BF16 generally balances precision and recall.

6 Our approach

Our project aims to improve on the existing VQA benchmarks by using a two-stage pipeline [Figure 6].

The first stage is the offline pre-evaluation stage which uses image-question dataset with a multimodal encoder to store the fused embeddings in a vector database (Section 6.1.1). During inference, use different vector retrieval mechanisms (explained in detail in Section 6.1.2) to fetch top-k relevant documents. The data from the top-k documents is sent to the second stage of the pipeline, which consists of a knowledge graph-based RAG. This system consists of richer global data comprising of information on chest x-rays, reports, medical vocabularies, etc. The aim is to use this system to enhance the data received from the multimodal retriever. This enhanced data is then sent to the VLM for final output generation.

The evaluation of final output is done by using Large Language Model (LLM)-as-a-judge framework on new emerging metrics like answer relevancy, Generative Evaluation (GEval), etc., in addition to also using traditional NLP metrics like BLEU (Papineni et al. (2002)), ROUGE (Lin (2004)), METEOR (Banerjee and Lavie (2005)) and BERTScore(Zhang et al. (2020)) calculations.

Most of the previous SOTA works are evaluated using conventional approaches like precision, recall, F1 and traditional NLP metrics. We aim to introduce a few of the emerging approaches for benchmarking a generative model output. We are using *Answer Relevancy* and *GEval* from DeepEval[1] for a robust evaluation of our approach as they are phrase invariant compared to their traditional counterparts.

6.1 Step 1: Multimodal Indexing and Retrieval Framework (MIRF)

6.1.1 Vector Index Creation

The MIMIC-CXR dataset, comprising 227,835 chest X-ray images paired with their corresponding radiology reports was used to create a vector index. For each image-report pair, we generated a joint embedding using the BiomedCLIP model [2]. BiomedCLIP (released by Microsoft and available via the HuggingFace Hub) is a pretrained biomedical vision-language model that integrates a PubMedBERT text encoder with a Vision Transformer (ViT) image encoder. We configured the model with a maximum text sequence length of 256 tokens. In this setup, each X-ray image and its accompanying report are encoded into a shared latent feature space, yielding compatible image and text embedding vectors for the pair.

To handle the large volume of data efficiently, embedding generation was performed in batches rather than processing the entire dataset in one pass. We used a batch size of 5,000 image-report pairs to keep memory usage within acceptable limits. After processing each batch through BioMedCLIP, the resulting embedding vectors were immediately written to disk, and a checkpoint was saved. All computed embedding vectors were indexed for similarity search using the Facebook AI Similarity Search (FAISS) library. We employed FAISS's Hierarchical Navigable Small World (HNSW) algorithm to support approximate nearest neighbor retrieval at scale. The HNSW index was configured to maintain up to 32 neighbor connections per node, and we used inner product as the distance metric for measuring similarity between embedding vectors.

After all batches were processed, these chunked indices were merged into a single unified FAISS index. The merging step consolidated the partial indices without loss of information, resulting in a final index equivalent to one built in a single pass. This chunk-and-merge strategy minimized memory overhead during index construction while ultimately yielding an integrated index for global similarity search across the dataset.

[1] https://github.com/confident-ai/deepeval

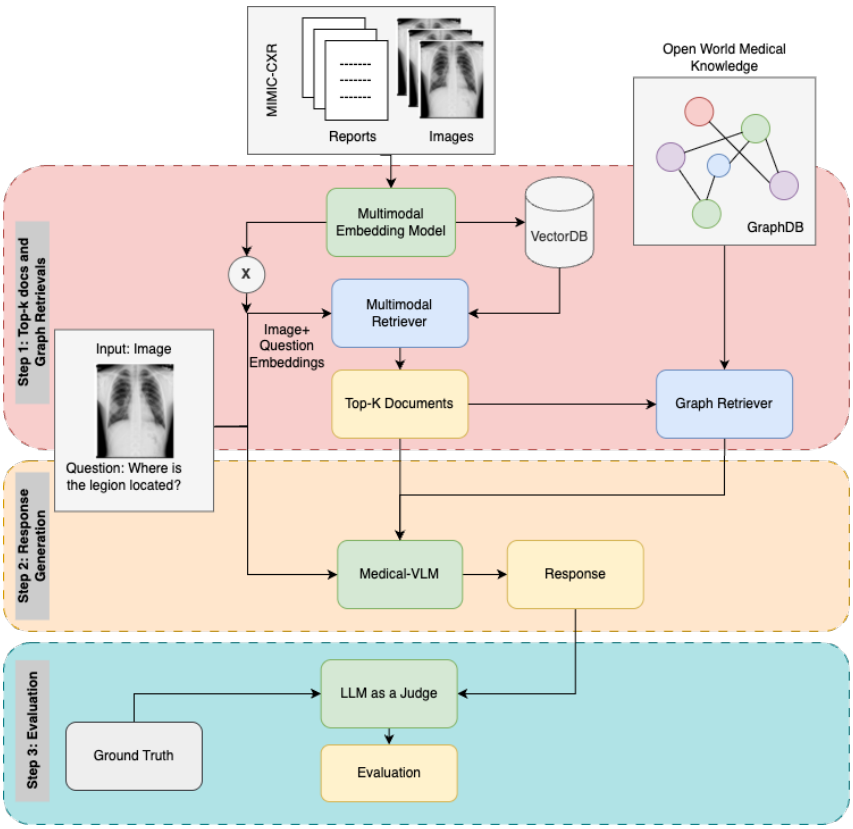

Figure 6: Med-VLM-KGRAG Pipeline Architecture

Setup: This was performed locally on a machine with 8 GB of RAM using the FAISS CPU-only library configuration.

6.1.2 Vector Index Retrieval

Given an input image and its associated query from our benchmarking datasets (VQA-RAD, SLAKE), we compute a joint embedding using the same BioMedCLIP model and preprocessing pipeline employed during indexing. We perform a top-5 nearest neighbor search against the prebuilt FAISS HNSW index. Internally, FAISS returns both the similarity scores (inner products) and the indices of the retrieved vectors. Because we normalized all vectors, these inner products directly correspond to cosine similarity values. The FAISS index stores only raw vectors; to recover the original file paths, we maintain an in-memory mapping (loaded from the JSON document map) between index positions and the corresponding image and report file paths. For each retrieved index, we look up its entry in the document map and construct its result object.

6.1.3 Knowledge Graph Creation

Constructing a Knowledge Graph from Scratch: We used the ReXGradient-160K reports dataset to prepare our reports knowledge graph. ReXGradient-160K is the largest publicly available multi-site chest X-ray dataset, containing 273,004 unique chest X-ray images from 160,000 radiological studies, collected from 109,487 unique patients across 3 U.S. health systems (79 medical sites). We used the validation split from this dataset for our knowledge graph construction, which has about 10,000 detailed radiology reports across 6,964 patients.

We leveraged the graph creation component of Microsoft GraphRAG (Edge et al. (2025)) to achieve this, where both local and global index search was enabled. Due to running GraphRAG locally, we decided to use the validation dataset of 10000 reports for our graphs as the original report dataset was too huge to index using GraphRAG on our local machine.

Leveraging an Existing Knowledge Graph: We were able to successfully integrate the PrimeKG (Chandak et al. (2023)) into our pipeline after dabbling with its edges and nodes CSVs.

It was not straightforward to load PrimeKG on neo4j, and we had to purge a few relations in order to be able to host our knowledge graph online on AuraDB by neo4j due to free tier limitations (200k nodes and 400k relationships max). This was needed because our pipelines were running on Kaggle/Colab Pro, and the neo4j graph database would have otherwise been offline, making it difficult to connect. We also enriched the nodes with information like UMLS definition, OrphaNet definition, symptoms, Preventions, Mayo Clinic Definition, Risk Factors, etc. on diseases and drugs.

6.1.4 Graph Retrieval

Our idea is to combine general medicine knowledge as well as domain specific knowledge (as we learned in the course, training closer to test set helps in LLM domain) in our knowledge graph. Thus, our knowledge graph pipeline has 2 branches of GraphRAG components: one works with neo4j hosted on AuraDB in cloud which comprises of PrimeKG data and the other GraphDB is constructed using Medical Report Data using RexGradient-160K reports dataset which is stored in local. Vector embeddings of graph nodes were created using OpenAI's text-embedding-3-small model (1536-dimensions). PrimeKG is open-source graph dataset consisting of general medical knowledge like drugs, diseases, their relations, etc and RexGradients comprises of X-ray reports, which helps GraphRAG to build knowledge graph on X-ray concepts while the former helps with general medical knowledge.

6.2 Step 2: Medical VLM Inference

We conducted inference on the open-ended question-answer pairs from the VQA-RAD and SLAKE benchmarks using three vision-language models under identical evaluation settings. Both CheXAgent and LLaVA-RAD accept the raw chest X-ray image and question as input. The top-5 documents retrieved via our graph-RAG pipeline are appended directly to the query to provide contextual evidence. In the case of LLaMA Vision-Instruct 3.2 (11 B), we follow the same input procedure but further encapsulate the image, question and retrieved documents within a specialized radiologist-expert prompt, to guide the model toward concise, clinically grounded responses.

The medical VLM used in our project (LLaVa-RAD, ChexAgent and LLaMa 3.2 11B Visual Instruct) required more than 20 GB VRAM on average. This caused CUDA Out of Memory (OOM) issues in the free-tier Google Colab notebook. We resolved this issue partially by shifting our notebooks to Google Colab Pro where we faced another challenge of hitting OOM limit on Nvidia A100 GPUs as well (40 GB VRAM) while running inference on larger datasets like SLAKE. For this, we wrote a custom script to clear CUDA cache on reaching certain percentage of thresholds. This caused slight increase in inference latency but made the inference process stable.

6.3 Ablation Studies

We perform ablation studies on our pipeline by considering the following modifications:

1. **Baseline**: We use the chosen 3 models as it is and run our evaluation pipeline on them using SLAKE and VQA-RAD datasets. Neither RAG nor Graph-based RAG component is involved.

2. **Using only RAG**: We use only the top-k documents fetched using our multimodal retriever and pass it as a context in addition to the question to the model. Example prompt with this ablation:

```
prompt = f"You are an expert
    radiologist. Considering
    the given image and
    provided top-5 relevant
    documents as context,
    answer the following
    question in a single
    short paragraph. Do not
    use bullet points. Be
    precise in your
    responses. Question :
    {{question}}, top-5
    document context:
    {{top_k_documents}}"
```

3. **Using RAG + GraphRAG**: This is our original approach, where we want to use the RAG's top-k documents to be fed into the graph retriever and receive a natural text from GraphRAG for using as context fdr the question. We do not perform an ablation where only GraphRAG is used (without the use of multimodal RAG's top-k documents), or both top-k documents and GraphRAG given as context to the model along with the question, due to context token length max limits and

constraints of OpenAI credit limit. Henceforth anywhere in the paper, if you notice a format of {{model}}+GraphRAG, we essentially mean this RAG + GraphRAG being used. Example prompt:

```
prompt = f"You are an expert
    radiologist. Considering
    the given image and
    provided knowledge as
    context, answer the
    following question in a
    single short paragraph.
    Do not use bullet
    points. Be precise in
    your responses. Question
    : \{\{question\}\},
    top-5 document context:
    \{\{graphrag\_output\}\}"
```

7 Evaluation Metrics

We evaluate the quality of the generated medical responses using both traditional and new-age generation-based metrics. We first define the traditional metrics and provide the relevant definitions:

7.1 ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) measures the overlap between the generated and reference texts based on n-grams, longest common subsequence (LCS), or skip-bigrams.

- **ROUGE-N**: measures n-gram overlap (commonly ROUGE-1 and ROUGE-2).

$$R_N = \frac{\sum_{\text{g}_n \in \text{ref}} \min(\text{C}_{\text{gen}}(\text{g}_n), \text{C}_{\text{ref}}(\text{g}_n))}{\sum_{\text{g}_n \in \text{ref}} \text{C}_{\text{ref}}(\text{g}_n)}$$

- **ROUGE-L**: uses the longest common subsequence (LCS).

$$\text{ROUGE-L} = \frac{LCS(X, Y)}{\text{length}(Y)}$$

- **ROUGE-SUM**: aggregates multiple ROUGE scores (e.g., ROUGE-1, ROUGE-2, ROUGE-L) into a unified measure by summing them with equal or weighted importance.

$$\text{ROUGE-SUM} = \sum_{i=1}^{k} w_i \cdot \text{ROUGE}_i$$

where $\text{ROUGE}_i \in \{\text{ROUGE-1}, \text{ROUGE-2}, \text{ROUGE-L}, \ldots\}$, and w_i is the weight assigned to each component (typically uniform, used here).

7.2 BLEU

BLEU (Bilingual Evaluation Understudy) computes the precision of n-gram overlaps between candidate and reference texts with a brevity penalty (BP) to penalize short outputs.

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

where p_n is the precision of n-grams, w_n is the weight (usually uniform), and BP is the brevity penalty:

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

with c as candidate length and r as reference length.

7.3 METEOR

METEOR aligns unigrams between candidate and reference based on exact, stem, synonym, and paraphrase matches, and incorporates a fragmentation or chunking penalty:

$$\text{METEOR} = F_{\text{mean}} \cdot (1 - p_{chunking})$$

where $F_{\text{mean}} = \frac{10 \cdot P \cdot R}{R + 9P}$ and penalty is a function of alignment fragmentation.

7.4 BERTScore

BERTScore uses contextual embeddings from a pretrained BERT model to compute similarity between candidate and reference tokens. We used BioMedNLP-PubMedBERT specifically.

$$\text{BERTScore}_{\text{F1}} = \frac{2PR}{P + R}$$

where P and R are computed using cosine similarity between token embeddings across reference and candidate:

$$P = \frac{1}{|c|} \sum_{x \in c} \max_{y \in r} \cos(e(x), e(y))$$

$$R = \frac{1}{|r|} \sum_{y \in r} \max_{x \in c} \cos(e(y), e(x))$$

7.5 LLM-as-a-Judge metrics

We used the DeepEval framework to evaluate our generated responses. DeepEval uses LLM-as-a-Judge framework to achieve this. While it provides a plethora of metrics like contextual relevancy, answer relevancy, faithfulness, hallucination, GEval, etc., we used only answer relevancy and GEval as we were constrained by the OpenAI API budget, since others needed to pass context and the token cost was getting too high. The definition of the two metrics as per the official website are:

7.5.1 Answer Relevancy

The Answer Relevancy metric uses LLM-as-a-judge to measure the quality of the pipeline's generator by evaluating how contextually and semantically relevant the actual_output of the LLM application is compared to the provided input. DeepEval's answer relevancy metric is a self-explaining LLM-Eval, meaning it outputs a reason for its metric score.

7.5.2 GEval (Generative Evaluation)

G-Eval is a framework that uses LLM-as-a-judge with chain-of-thoughts (CoT) to evaluate LLM outputs based on fine-grained factuality and consistency. The GEval metric is the most versatile type of metric DeepEval has to offer, and is capable of evaluating almost any use case with human-like accuracy.

8 Results and Discussion

8.1 Evaluation on traditional metrics

Based on Table 1, we observe the following trends:

1. CheXAgent has strong baseline performance across the traditional metrics like ROUGE, BLEU and METEOR.

2. Across all ablations, the RAG+GraphRAG performance is better than only using RAG. However, none of them is able to beat the baseline.

3. The METEOR score is relatively better across all experiments, given how METEOR considers synonyms, paraphrasing, etc. instead of standard n-gram overlaps.

4. The ROUGE-2 scores are generally less than ROUGE-1, given that most ground truth answers are one-word answers, and the generated responses are sentences.

5. LlaMa 3.2 Vision Instruct shows decent performance for both datasets with RAG+GraphRAG across datasets, given that it is the only Instruct model out of the three, while LlaVA-RAD had a considerable performance decline (as per traditional metrics) with the pipeline as compared to baseline.

Why BERTScore is so high across all experiments?

In Table 2, we see that the scores are almost identical and high, showing how our responses managed to capture the essence, in alignment with the baseline. However, digging deep into how exactly BERTScore works, we get an understanding of our results.

BERTScore is fundamentally a soft, embedding-based token alignment metric, not an exact string overlap measure. It produces high scores even when the reference is just one word and the generated result is a longer sentence. This is because of two main reasons:

1. Recall is easy to max out with a single-word reference. Recall in BERTScore is the average, over each reference token, of the maximum cosine similarity to any candidate token. With only one reference token ("chest," say), recall becomes simply the best match between that embedding and any token in the paragraph. If the generated text contains that word (or a close synonym), recall ≈ 1.

2. Precision can also stay high. Precision averages, for each candidate token, the maximum similarity to any reference token.

3. Although most tokens in a long paragraph won't match "chest," many common words ("heart," "lung," "ribs" etc.) will still share reasonable contextual similarity with the reference token in the embedding space, so their cosine scores aren't zero.

4. Unlike BLEU or ROUGE, BERTScore doesn't penalize extra words or longer text. It simply aligns tokens one-to-many (for recall) and many-to-one (for precision) and averages similarities.

A supposed perspective we might get from the performance of our pipeline against the baseline

Table 1: BLEU, METEOR, and ROUGE scores across models and retrieval strategies on SLAKE and VQA-RAD

Dataset	Model Name	Config	BLEU	METEOR	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SUM
SLAKE	LLaVA-RAD	Baseline	0.28	0.47	0.44	0.19	0.21	0.28
		GraphRAG	0.15	0.49	0.42	0.10	0.11	0.21
		RAG	0.10	0.31	0.37	0.09	0.12	0.19
	LLaMA 3.2 Vision Instruct 11B	Baseline	0.43	0.49	0.41	0.34	0.25	0.33
		GraphRAG	0.22	0.47	0.20	0.22	0.13	0.18
		RAG	0.13	0.24	0.17	0.13	0.14	0.15
	CheXAgent	Baseline	0.49	0.52	0.88	0.27	0.87	0.67
		GraphRAG	0.16	0.26	0.23	0.16	0.11	0.17
		RAG	0.09	0.12	0.14	0.15	0.07	0.12
VQA-RAD	LLaVA-RAD	Baseline	0.36	0.32	0.21	0.22	0.19	0.21
		GraphRAG	0.15	0.23	0.11	0.04	0.08	0.08
		RAG	0.10	0.10	0.14	0.01	0.09	0.08
	LLaMA 3.2 Vision Instruct 11B	Baseline	0.39	0.38	0.29	0.28	0.21	0.26
		GraphRAG	0.15	0.27	0.21	0.09	0.14	0.15
		RAG	0.10	0.12	0.18	0.04	0.09	0.10
	CheXAgent	Baseline	0.45	0.46	0.66	0.31	0.32	0.43
		GraphRAG	0.25	0.19	0.29	0.08	0.23	0.20
		RAG	0.07	0.05	0.09	0.06	0.11	0.09

Table 2: BERTScore across models and retrieval strategies on SLAKE and VQA-RAD

	VQA-RAD			SLAKE		
Model	Precision	Recall	F1 Score	Precision	Recall	F1 Score
LlaMa (baseline)	0.87	0.89	0.88	0.86	0.89	0.88
LlaMa+RAG	0.86	0.89	0.88	0.86	0.89	0.87
LlaMa+GraphRAG	0.8657	0.89	0.88	0.86	0.89	0.88
LlavaRAD (baseline)	0.87	0.89	0.88	0.86	0.89	0.88
LlavaRad+RAG	0.877	0.89	0.88	0.87	0.89	0.88
LlavaRad+GraphRAG	0.86	0.89	0.88	0.86	0.89	0.88
CheXAgent (baseline)	0.95	0.95	0.95	0.98	0.98	0.98
CheXAgent+RAG	0.90	0.88	0.89	0.88	0.88	0.88
CheXAgent+GraphRAG	0.88	0.89	0.89	0.87	0.88	0.88

is that the contextual information is not fully utilized based on traditional metrics. However, manually reading the responses v/s ground truth paints a different picture. We observe that our evaluation datasets are mostly consisting of one-word answers even for open-ended labelled questions. While this maybe desirable in a situation where exact answer is required, our core idea was to enhance the original response by providing more medical information around it, and also enabling follow-up questions to be answered in a chat-like environment. Our evaluation datasets are not in a chat style, but rather single question-answer pairs, which makes it difficult to evaluate our true objective. We observe that while quite a few responses contain the ground truth "within" them, the traditional metrics report a lower score because they do not estimate or evaluate the essence of supporting text around the core answer.

8.2 Evaluation using LLM-as-a-Judge

DeepEval's list of metrics like Contextual Precision/Recall/Relevancy, Answer Relevancy, Hallucination, Faithfulness and GEval to be helpful for our use case. However, all of them require passing a "context" for obvious reasons except for GEval and Answer Relevancy in addition to ground truth and generated responses. given our OpenAI credit constraints and cost estimation for passing full context, we decided to only use Answer Relevancy and GEval metrics.

8.2.1 Results using Answer Relevancy

1. We observe a very high score for baseline models across both datasets for CheXAgent, given how CheXAgent responses were mostly one-word answers, and it is not an instruction tuned model so the performance of RAG+GraphRAG and only RAG decreases considerably.

2. For LlaMa 3.2 Vision Instruct model, we observe that our RAG+GraphRAG approach reaches a comparable performance to the baseline, while the only RAG approach continues to lag behind.

3. For LlaVa-RAD, the RAG+GraphRAG version beats the baseline [Figure 7,9].

```
Metric: Answer Relevancy
Question: What is the
    abnormality by the right
    hemidiaphragm?
Expected Output: Minor elevation
    of right hemidiaphragm
Generated Output: The
    abnormality by the right
    hemidiaphragm is likely a
    trace amount of free air, as
    indicated in one of the
    reports. Additionally, there
    may be slight elevation of
    the right hemidiaphragm,
    which could be due to a
    temporary nerve block, as
    noted in another report.
    Overall, the findings
    suggest the presence of
    atelectasis or free air
    under the right
    hemidiaphragm, but no
    pneumothorax is detected.
Score: 0.8571428571
Explanation by LLM as a Judge:
    The score is 0.86 because
    the response effectively
    addresses the abnormality by
    the right hemidiaphragm, but
    includes a mention of
    pneumothorax, which is not
    directly related to the
    query. Despite this, the
    main focus remains relevant,
    maintaining a high score.
```

8.2.2 Results using GEval

1. The baseline beats our approach and ablation considerably across both datasets with the CheXAgent model.

2. For SLAKE, LlaMa 3.2 Vision Instruct model and LlaVa-RAD have a comparable performance with the baseline.

3. For VQA-RAD dataset, the LlaMa 3.2 Vision Instruct and LlaVa-RAD beat the baseline.

4. A key observation about LlaVA-RAD is that the baseline score is generally lower as compared to CheXAgent and LlaMa Vision Instruct 3.2 model [Figure 8,10].

Using DeepEval's metrics that leverage LLM-as-a-Judge framework, we get an empirically better estimate of the performance of our model. DeepEval's framework return a reason for the scores, which on human reading, made sense almost always. A few samples for scores explanation are:

```
Metric: GEval
Question: Which part of the body
    does this image belong to?
Expected Output: Chest
Generated Output: The image is
    centered at the thoracic
    inlet. **There is
    opacification in the left
    lower lung, which could be
    due to atelectasis,
    pneumonia, or pulmonary
    hemorrhage.** The right lung
    is clear. There is no
    pneumothorax or pleural
    effusion. The
    cardiomediastinal silhouette
    has a normal postoperative
    appearance. The right
    jugular infusion port ends
    in the mid SVC.
Score: 0.7274625625
Explanation by LLM as a Judge:
    The actual output describes
    the thoracic region,
    aligning with the expected
    output of 'Chest'. There are
    no factual contradictions or
    critical medical omissions,
    but the response is more
    detailed than necessary.
```

While LLM-as-a-Judge's effective for evaluation still remains an elusive question, the latest trend in the industry around leveraging them is not all in vain, as we see this approach in a reverse way where we try to reason if the evaluation explanation of LLM makes sense, and either accept/reject the score given based on that.

With our approach being more suitable for chat environment, no standard metric is holistic enough to evaluate the responses, LLM-as-a-Judge framework provides a promising evidence

of its effectiveness in evaluating performance of pipelines in such environments. While we didn't get a chance to evaluate our pipeline's effectiveness using metrics like faithfulness, hallucination, etc. due to context tokens cost, we believe that future work holds great hope in terms of being able to evaluate such scenarios better. Moreover, DeepEval's framework has explicit metrics for conversational situations like Conversational GEval, Role Adherence, Knowledge Retention, Conversational Relevancy/Completeness, etc.

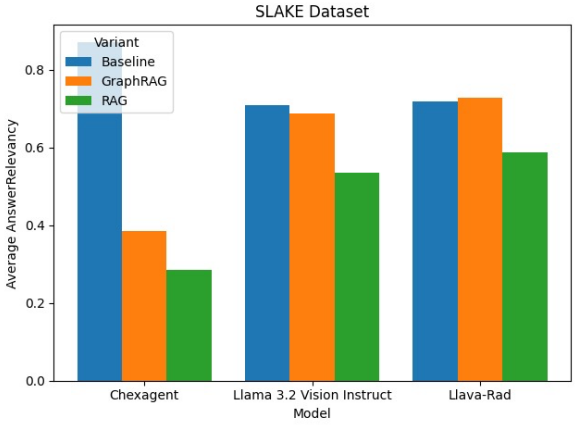

Figure 7: Answer Relevancy Metric comparison across different ablations for SLAKE dataset

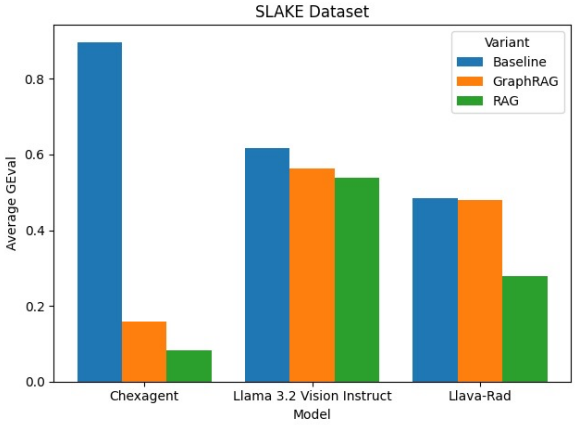

Figure 8: GEval Metric comparison across different ablations for SLAKE dataset

9 Error Analysis

Why are our scores lower for enhancements than baseline?

We believe that since not all models are instruction tuned, so providing context doesn't help in models like CheXAgent. This has instead shown

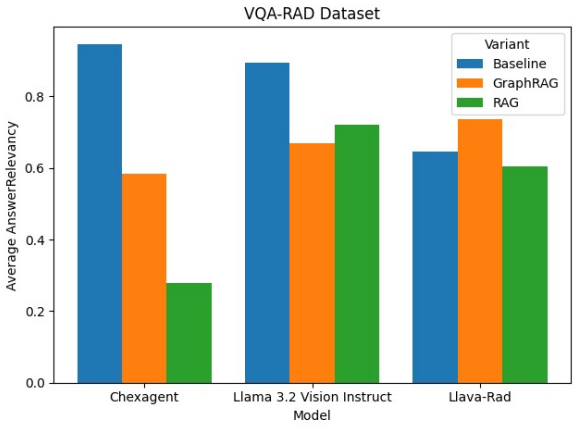

Figure 9: Answer Relevancy Metric comparison across different ablations for VQA-RAD dataset

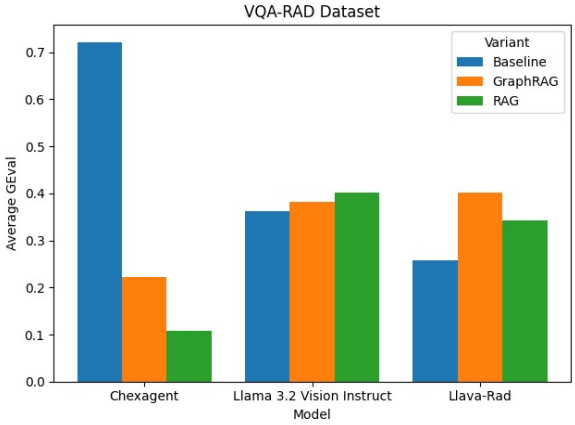

Figure 10: GEval Metric comparison across different ablations for VQA-RAD dataset

worsened performance in comparison to the baseline, both new-age metrics and traditional metrics wise. CheXAgent has been tuned to provide precise answers, so no matter how much context we provide it, the length of responses are not improved and continue to remain short, one-word often.

Even though the CheXAgent baseline performs really well, it only does so because, the ground truth are mostly one word responses even for open-ended problems. This does not satisfy our original objective of generating contextually-rich responses.

We were limited by the choice of models as quite a few SoTA models could not run or faced hardware limitations. The choices of models like LLaVa-RAD and CheXAgent show the limitations of these Medical VLMs not instruction-tuned and thus cannot efficiently understand the retrieved documents and graph-extracted knowledge being

passed to it as additional context. They are unable to reason effectively and determine whether the answer generated is relevant or not. This is overcome by the LLaMa Vision Instruct 3.2 model as it can be prompted to generate a good answer using the context.

Our approach of concatenating the context (retrieved docs) to the query to the Medical VLMS - CheXAgent and LLaVa-RAD fails to generate a good answer and thus more methods of using the context for these models can be explored in the future.

Some Failure Cases: We observe these to be the top generated answers of CheXAgent on the SLAKE dataset:

Model Output	Count
No significant findings.	60
No acute cardiopulmonary process.	34
No abnormalities or diseases are reported.	30
No abnormalities detected.	24
No significant changes were observed.	20

Table 3: Frequency of common report findings in SLAKE

Similarly, for VQA-RAD, `CheXAgent` output analysis reveals the following:

Model Output	Count
No acute cardiopulmonary process.	20
No acute cardiopulmonary process	8
No acute cardiopulmonary disease.	7
No abnormalities detected.	3
No significant changes were observed.	3

Table 4: Frequency of common report findings in VQA-RAD

These generic negations are all semantically equivalent (”no acute cardiopulmonary process” == ”no abnormalities detected”), the model’s output distribution collapses to a very small set of near-synonyms. We believe that because they syntactically share the ”[Negation] + [Anatomical/Clinical Phrase]” pattern, which the model reproduces with minimal risk, reinforcing repetition.

Below is a focused error analysis of the LLaVa-RAD failure on the sample. When prompted:

> “What is the largest organ in the picture?”

(ground truth: Lung), LLaVa-RAD, after GraphRAG fuses in nodes about lung findings (mild atelectasis, pleural effusions, heart size, etc.) responds:

> “The largest organ in the human body is the skin. However, the context provided focuses on the lungs and their conditions, with various findings concerning lung health, including mild atelectasis, pleural effusions, and heart size. These findings are critical for assessing respiratory and cardiovascular health and guiding further medical decisions.”

LLaVa-RAD thus defaults to its broader medical “world knowledge” (e.g. “the skin is the largest organ”) which is incorrect. There are 15/49 such cases in the SLAKE dataset where it predicts this type of answer for the question “What is the largest organ in the picture?”.

10 Contributions of group members

List what each member of the group contributed to this project here. For example:

- **Arghyadeep Das:** Worked on knowledge graph construction, running inference pipelines, prompt engineering, preprocessing, pipeline integration, medical VLM experimentation for shortlisting, report writing.
- **Nilesh Nayan:** Worked on graph retriever server, DeepEval metric generation script, evaluation metrics computations (DeepEval metrics), pipeline integration, running inference pipelines, report writing.
- **Roshita Bhonsle:** Worked on multimodal vector index (Vector database), running inference pipelines, traditional evaluation metrics generation (BLEU, ROUGE, METEOR), MLLM-as-a-Judge exploration, report writing.
- **Shantanu Todmal:** Worked on multimodal vector retriever, medical VLM experimentation for shortlisting, running inference pipelines, evaluation metric computation (BERTScore), graphs.

11 Conclusion

Based on the results we obtained, our proposed Multimodal RAG+GraphRAG approach surpasses the baselines for the following configurations:

- LLaVa-RAD on VQA-RAD dataset according to the AnswerRelevancy Metric
- LLaVa-RAD on SLAKE dataset according to the AnswerRelevancy Metric
- LLaVa-RAD on VQA-RAD dataset according to the GEval Metric
- LLama 3.2 Vision Instruct on VQA-RAD dataset according to the GEval Metric

We believe our results point us to a new direction of improvement, where an initial VLM answers the factual question based on the image, and the subsequent questions by the user are answered a chatbot-style environment can leverage our pipeline. The initial image-question pair sets up the context "domain" for multimodal retrieval, and then the further questions are answered by our pipeline.

While we initially had already hinted at our pipeline being more useful in a chat-like environment, we feel the low scores on direct questions are a mix of reflection of how it might not be necessary to use KG-RAG for simpler VQAs, and at the same time the evaluation datasets being not really relevant for our use case, since our idea was to "enhance" VLM responses, which could be understood in the sense that once the user has asked initial query directly about the image first, it sets the context, based on which our pipeline can use the KG information to answer further questions based on information derived by RAGs. This otherwise might not be possible for non-instruction tuned LMs to do directly, therefore our approach strengthens existing VLMs to leverage our pipeline for surpassing their abilities and answer follow-up questions in chat by allowing remaining questions to be handed off to more robust instruction-tuned LMs for generation based on extracted KG information by RAGs.

For future direction, we would look at redefining our problem statement in a slightly different way and look for chat-based medical datasets if any. Given the time and budget constraints, we were not able to utilize all features of GraphRAG and evaluate its performance as a graph retriever using DeepEval's metrics for RAGs. We would look at improving GraphRAG retrieval responses via auto prompt-tuning feature and do a comparison study of how local v/s global v/s drift search actually works. Another important addition to our research would be to go beyond chest x-rays since we feel that limited us in terms of finding good datasets for our use case. However, for the scope of a course project, we were satisfied with it. We hope that we will be able to run and evaluate the SoTA models like LLaVA-Med, MedFlamingo, etc. so that we can see the impact of our pipeline in them as well.

12 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - ChatGPT 4o

If you answered yes to the above question, please complete the following as well:

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - I am providing a CSV file containing certain metric scores. Give me the LaTeX code to build this as a table. [file attachment]
  - I am writing a research report, where I want to give the definition and formula for ROUGE (and its variants), BLEU, BERTScore, METEOR, and then metrics from DeepEval like Answer Relevancy and GEval. Our work is on improving medical response of medical VLMs using knowledge graphs and RAGs. Kindly give me the LaTeX code for defining and formulating my evaluation metrics.
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?

- For the LaTeX code for table, we got proper result in first go.
- For the definitions and LaTeX code for formulas, had to verify the formulas given by it.

References

Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Chandak, P., Huang, K., and Zitnik, M. (2023). Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67.

Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., and Jégou, H. (2024). The faiss library. *arXiv preprint arXiv:2401.08281*.

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., and Larson, J. (2025). From local to global: A graph rag approach to query-focused summarization.

Guo, Z., Xia, L., Yu, Y., Ao, T., and Huang, C. (2024). Lightrag: Simple and fast retrieval-augmented generation.

Indyk, P. and Motwani, R. (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, page 604–613, New York, NY, USA. Association for Computing Machinery.

Johnson, A. E., Pollard, T. J., Berkowitz, S. J., Greenbaum, N. R., Lungren, M. P., Deng, C.-y., Mark, R. G., and Horng, S. (2019). Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 6(1):317.

Lau, J. J., Gayen, S., Ben Abacha, A., and Demner-Fushman, D. (2018). A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*, 5(1):1–10.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2021). Retrieval-augmented generation for knowledge-intensive nlp tasks.

Li, C., Wong, C., Zhang, S., Usuyama, N., Liu, H., Yang, J., Naumann, T., Poon, H., and Gao, J. (2023). Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36:28541–28564.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Liu, B., Zhan, L.-M., Xu, L., Ma, L., Yang, Y., and Wu, X.-M. (2021a). Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering. In *2021 IEEE 18th international symposium on biomedical imaging (ISBI)*, pages 1650–1654. IEEE.

Liu, F., Wu, X., Ge, S., Fan, W., and Zou, Y. (2021b). Exploring and distilling posterior and prior knowledge for radiology report generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13753–13762.

Malkov, Y. A. and Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836.

Mavromatis, C. and Karypis, G. (2024). Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv e-prints*, pages arXiv–2405.

Moor, M., Huang, Q., Wu, S., Yasunaga, M., Dalmia, Y., Leskovec, J., Zakka, C., Reis, E. P., and Rajpurkar, P. (2023). Med-flamingo: a multimodal medical few-shot learner. In *Machine Learning for Health (ML4H)*, pages 353–367. PMLR.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Sarmah, B., Mehta, D., Hall, B., Rao, R., Patel, S., and Pasquali, S. (2024). Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 608–616.

Su, X., Wang, Y., Gao, S., Liu, X., Giunchiglia, V., Clevert, D.-A., and Zitnik, M. (2025). Kgarevion: An ai agent for knowledge-intensive biomedical qa. *International Conference on Learning Representations, ICLR*.

Wu, J., Zhu, J., and Qi, Y. (2024). Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation. *arXiv e-prints*, pages arXiv–2408.

Xia, P., Zhu, K., Li, H., Wang, T., Shi, W., Wang, S., Zhang, L., Zou, J., and Yao, H. (2025). Mmed-rag: Versatile multimodal rag system for medical vision language models. *ICLR 2025*.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). Bertscore: Evaluating text generation with bert.

Zhang, X., Acosta, J. N., Miller, J., Huang, O., and Rajpurkar, P. (2025). Rexgradient-160k: A large-scale publicly available dataset of chest radiographs with free-text reports. In *arXiv:2505.00228v1*.

Zhao, X., Liu, S., Yang, S.-Y., and Miao, C. (2025). Medrag: Enhancing retrieval-augmented generation with knowledge graph-elicited reasoning for healthcare copilot.