Generative AI Final Report

Nicolas Smits
AI Engineering - Chemical Engineering
nsmits@andrew.cmu.edu

John Speaks
Information Technology Strategy
jspeaks@andrew.cmu.edu

Darren Chen
Mathematical Sciences
darrenc2@andrew.cmu.edu

1 Introduction

Retrieval-augmented generation (RAG) and vision-language models (VLMs) have significantly improved grounded question-answering (QA) [1, 2, 3, 4]. However, traditional text-centric approaches struggle with multimodal documents, including lecture slides and technical notes, where critical information is embedded in figures, tables, and complex layouts [5, 6]. We propose a multimodal RAG pipeline designed to retrieve and reason over academic materials in their native visual format. Using the REAL-MM-RAG benchmark [7], we analyze how various retrieval and fine-tuning strategies influence answer quality, groundedness, and stability.

Our primary architecture, Image-RAG, retrieves relevant document pages as images to preserve visual context for a Qwen3-VL generator [8], utilizing a frozen ColQwen3 retriever [9] to isolate generation performance. We further enhance the model via Low-Rank Adaptation (LoRA) [10]. This approach allows us to systematically compare configurations ranging from non-RAG baselines and OCR-based text-RAG to our fine-tuned pipeline.

We hypothesize that while text-RAG may perform adequately on dense prose, Image-RAG will demonstrate superior reasoning over visually structured technical content. We expect a performance trend where RAG-based methods significantly reduce hallucinations compared to the baseline, with LoRA fine-tuning and teacher distillation providing the final necessary alignment to maximize groundedness. Ultimately, this work seeks to identify the optimal design for a robust, multimodal academic QA assistant.

2 Dataset and Task

For our project, we will experiment with the effects of RAG and RAG-specific fine tuning on answer quality and hallucinations on multi-modal question-answer tasks. We will compare how VLMs are able to respond to these VQA tasks with different levels of RAG support ranging from an out of the box VLM as a baseline, VLM + ideal context (image and OCR), VLM + RAG context (image and OCR), and VLM + RAG context finetuned for the task (with the given dataset and one we specialy generate. We will evaluate these VLMs on a number of metrics across question quality and groundedness.

We plan to use the REAL-MM-RAG benchmark dataset [7] to train the components of our model. The dataset consists of over 8000 pages of documents and 5000 queries about the content of the document. We chose this dataset because it is multi-modal and its prompt style matches the goals of our project. Instead of direct questions about specific slides (search tasks), prompts are formatted to knowledge seeking and more generally to be answered with the system files.

There are a variety of metrics used for these kinds of tasks falling into the major categories: quality/accuracy of answers and groundedness in the file content. Building on previous work from Wasserman 2025 [7], we will explore the following evaluation metrics for each component of the system:

Statistical Scores

- **BLEU**: The precision of n-gram overlaps between the generated output and ground truth answer.
- **ROUGE-1**: The recall of word unigrams between the generated and ground truth solution (sees if correct answer content is present).
- **ROUGE-2**: The recall of word bigrams to evaluate how similarly the model generates in terms of local word order compared to ground truth.
- **ROUGE-L**: The longest common subsequence to capture structural similarity.

Model Scores

- **BERTScore**: The cosine similarity in the BERT representation of a response and the ground truth for correctness.

LLM-as-a-Judge (See Appendix B)

- **Accuracy**: Measures factual consistency between the generated response and the ground truth answer.
- **Relevance**: Evaluates how directly the response addresses the question and avoids irrelevant information.
- **Groundedness**: Assesses if claims made in the generated response are supported by information in the context.

- **Faithfulness**: Measures how well the answer does not hallucinate and remains consistent with the context.

Our set of metrics combines traditional n-gram approaches for linguistic exactness matching the intended document answers with more high level metrics to allow some flexibility in model style. With these metrics we can observe how strictly the model uses the context and how much variation is caused by hallucination versus style choices.

3 Related Work

Retrieval-Augmented Generation: Retrieval-Augmented Generation (RAG) grounds language models in external knowledge, mitigating the tendency of parametric-only models to hallucinate [1, 2]. By retrieving relevant documents during generation, these systems significantly improve factual reliability [1, 2]. Late-interaction methods such as ColBERT [11] further enhance retrieval through fine-grained, token-level matching.

However, most RAG systems remain limited to text-only corpora. This assumption is problematic for multimodal documents, where critical information is embedded in layouts, figures, and tables. Common workarounds rely on OCR pipelines, which are often lossy and discard important visual structure in the conversion process [7]. The REAL-MM-RAG benchmark [7] highlights these limitations, particularly for tasks requiring reasoning over visual elements and multi-page context. To address this, we extend RAG to operate directly on page images, bypassing intermediate text extraction.

Multimodal Document Retrieval and QA: Prior work on document QA, including DocVQA and InfographicVQA [5, 6], demonstrates the importance of jointly reasoning over visual and textual content. REAL-MM-RAG further emphasizes robustness to query variation and increasing task complexity in multimodal retrieval settings [7].

Recent multimodal retrievers such as ColQwen3.5 [9] extend late-interaction retrieval to vision-language inputs by modeling interactions between query embeddings and visual patch representations. This enables direct retrieval over document page images without relying on OCR. We adopt this paradigm and use a frozen retriever to isolate the effects of generation and supervision.

Vision-Language Models for Grounded QA: Vision-language models such as Flamingo [3] and the Qwen-VL family [12, 13, 14, 8] achieve strong performance on multimodal reasoning tasks by combining visual encoders with large language models, enabling joint processing of images, text, and structured content [4, 15]. These models can capture fine-grained visual details and align them with linguistic representations, making them well-suited for document understanding [3, 12].

However, they are typically used in closed-book settings, relying on internal knowledge rather than external evidence, which can lead to ungrounded outputs and hallucinations [16]. This limitation becomes more pronounced in document QA tasks that require precise grounding in specific visual elements or multi-page context [7]. Our approach integrates VLMs with retrieval, ensuring that generation is explicitly conditioned on retrieved page images.

Fine-Tuning and Extensability: Parameter-efficient methods such as LoRA [10] enable adapting large models with minimal overhead by introducing low-rank updates while keeping the base model frozen. This makes fine-tuning feasible in resource-constrained settings and allows rapid experimentation [10].

However, performance depends heavily on the quality and alignment of supervision. Recent work shows that teacher distillation provides stronger and more stable training signals than raw annotations, particularly for reasoning-intensive tasks [17, 18]. In multimodal QA, dataset-provided answers may be incomplete, inconsistent, or weakly grounded in the underlying evidence [7].

We employ teacher-generated, evidence-conditioned supervision, where a stronger model produces answers grounded in the retrieved content on an analogous slides document QA task (our target domain) compared to the finance and technology slides in REAL-MM-RAG [1]. Finetuning on this data improves both the explanatory quality and the faithfulness of generated responses, aligning training with grounded QA objectives [17]. However, we investigate the transferability of finetuning on distilled labels to an adjacent domain to evaluate the feasibility of extensibility based across document QA tasks.

4 Methods

Our system is intended addresses grounded question answering over multi-modal documents, motivated by the goal of building a QA assistant for academic lecture content. We train and evaluate on the REAL-MM-RAG benchmark [7] as a proxy (corpus of multi-modal documents from financial reports, technical presentations, technical materials). We hypothesize that strong performance on the aforementioned corpus transfers to the academic lecture note QA domain. Accordingly, we assess the QA performance improvements derived from varying retrieval pipeline implementations, finetuning, and label distillation on the REAL-MM-RAG dataset to infer optimal configurations an academic QA assistant.

Baseline Configuration: No-Context VLM To establish a lower bound for performance, our baseline utilizes the Qwen3-VL-8B-Instruct model in a "closed-book" setting. In this configuration, the generator is provided only with the user query and is forced to rely entirely on its internal parametric knowledge. This baseline serves to quantify the degree of hallucinations when the model lacks access to the document corpus and justifies the necessity of the RAG architecture.

Idealized Modality Analysis (Oracle RAG) Before implementing the full retrieval pipeline, we define ideal (Oracle) methods to determine the performance ceiling of the generator.

- **Ideal OCR Text:** The generator is provided with the ground-truth document page as raw text extracted via Tesseract OCR [19]. This measures the model's ability to reason over structured text without visual noise.
- **Ideal Image:** The generator receives the ground-truth document as a high-resolution page image. This evaluates the VLM's native ability to parse layouts, figures, and tables compared to the OCR-text alternative.

Multimodal Retrieval Architecture The core of our proposed system is an end-to-end RAG pipeline. We implement and compare two distinct retrieval strategies to supply context to the generator:

- **Text-based Retrieval:** Document pages are converted to text via OCR, chunked, and embedded using a dense retriever (all-MiniLM-L6-v2) in a ChromaDB vector store.
- **Visual Late-Interaction Retrieval:** We utilize ColQwen3.5 [9] to index full page images. This model computes relevance scores through late interaction between query token embeddings and visual patch embeddings, bypassing the lossy OCR step.

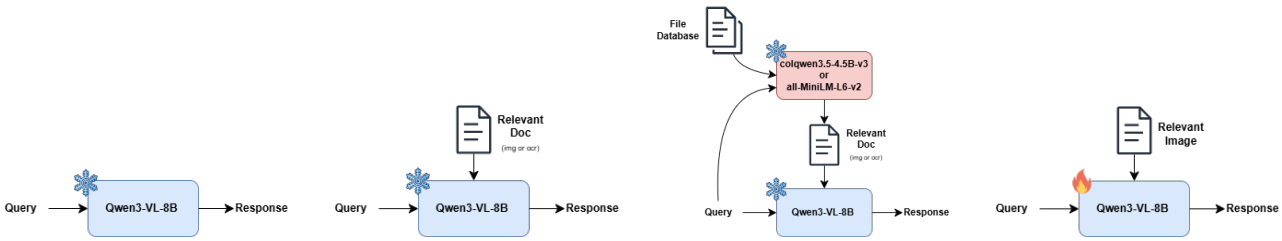
In both cases, the top-k most relevant context pieces are retrieved and prepended to the generator's prompt.

Fine-Tuning Implementation: We perform parameter-efficient fine-tuning on an NVIDIA A10 GPU using QLoRA. To maintain memory efficiency, the base Qwen3-VL-8B model may be loaded with quantization using the `bitsandbytes` library. We inject LoRA adapters into the transformer's attention blocks, enabling the model to adapt to the specific layouts and linguistic style of academic documents without the computational overhead of full-parameter updates. This configuration should allow for stable training on a single AWS g5.xlarge instance while preserving the multimodal reasoning capabilities of the base model.

Extensibility Test Our final experiment investigates the extensibility of our model finetuned on financial/technological report/slides data to the originally intended target task of QA on lecture slides and academic content. This serves to demonstrate whether QA performance can be transferred between domains, envisioning a scenario where a document QA RAG system could be finetuned on a large document data set, then deployed on a smaller adjacent target dataset. As stated in the motivation of this investigation, the ultimate goal is to asses the generation and reasoning performance over lecture slide content specifically. Therefore, in order to test performance in this domain, a small dataset of representative slides is curated. This is achieved by human-in-the-loop curation of all $\approx 1,300$ slides, where slides dense with subject matter content are selected for QA generation. The subset of ≈ 650 slides is then passed to Qwen3-VL-30B-Instruct model (larger than the finetuned model) on an High RAM NVIDIA A100 GPU, to generate QA paris using the slide image as grounding. Said QA pairs are generated using a slightly modified version of the prompt used in the REAL-MM-RAG study (wording changed to reference lecture slides rather than reports) in order ensure a degree of control between testing experiments, with the data being the primary changed variable [1] (see appendix for system prompt). Generation performance between the out-of-the-box Qwen3-VL-8B-Instruct and it's REAL-MM-RAG finetuned counterpart are compared to asses the feasibility and generalization capacity across domains for adjacent document QA tasks.

5 Experiments

All of our experiments are run using a Qwen3-VL-8B-Instruct model running on an AWS g5.xlarge EC2 instance with an Nvidia A10G GPU. Our baseline (experiment A) is the model provided with only the test set questions and no relevant (figure 1a). Our ideal setups (experiments B, C, F, H, and I) skip the retrieval stage of RAG and provides the relevant document (either as an image or transcribed text) directly to the model to provide an upper bound of system performance (figure 1b). We also have experiments D, E, G, J, and K run with a true functioning retrieval system pictured in figure 1c to anchor our exploration in real use cases.



(a) No Context (b) Idealized RAG (c) True RAG (d) Finetuning

Figure 1: System setups for each experiment type.

Table 1 has the averaged performance of each model across all documents with letter labels. First we compared the baselines (A, B, and C) and found that both image and text context had statistically significant improvements from the no context

Table 1: Evaluation Metrics Across Model Setups (Mean and 95% CI)

Setup	BLEU	R-1	R-2	R-L	BERT Score	LLM Acc.	LLM Rel.	LLM Grnd.	LLM Faith.
A. Baseline (no context)	0.2150 [0.1974, 0.2327]	0.4754 [0.4602, 0.4905]	0.3075 [0.2900, 0.3250]	0.4138 [0.3977, 0.4299]	0.9121 [0.9093, 0.9149]	1.9658 [1.8689, 2.0627]	4.8240 [4.7751, 4.8730]	1.0228 [1.0005, 1.0452]	1.0228 [1.0005, 1.0452]
B. Ideal OCR context	0.3706 [0.3483, 0.3930]	0.6673 [0.6515, 0.6831]	0.5176 [0.4975, 0.5377]	0.6021 [0.5844, 0.6198]	0.9400 [0.9372, 0.9428]	4.3211 [4.2196, 4.4227]	4.9585 [4.9317, 4.9854]	4.5902 [4.5089, 4.6714]	4.6595 [4.5826, 4.7364]
C. Ideal image context	**0.4845** [0.4615, 0.5074]	**0.7704** [0.7562, 0.7846]	**0.6461** [0.6272, 0.6649]	**0.7143** [0.6978, 0.7309]	**0.9577** [0.9553, 0.9602]	**4.5994** [4.5212, 4.6777]	**4.9725** [4.9527, 4.9923]	**4.8399** [4.7882, 4.8917]	**4.8129** [4.7554, 4.8703]
D. RAG OCR	0.2615 [0.2414, 0.2815]	0.5256 [0.5081, 0.5431]	0.3808 [0.3606, 0.4010]	0.4718 [0.4534, 0.4902]	0.9160 [0.9128, 0.9191]	2.9575 [2.8166, 3.0983]	4.7663 [4.7060, 4.8267]	4.3871 [4.2863, 4.4879]	4.5192 [4.4264, 4.6119]
E. RAG Image	**0.4374** [0.4143, 0.4605]	**0.7348** [0.7193, 0.7503]	**0.6023** [0.5826, 0.6221]	**0.6756** [0.6582, 0.6930]	**0.9519** [0.9492, 0.9546]	**4.5324** [4.4486, 4.6126]	**4.9892** [4.9761, 5.0024]	**4.8239** [4.7640, 4.8839]	**4.8636** [4.8128, 4.9143]
F. Ideal Finetuned	**0.5245** [0.5009, 0.5481]	**0.7846** [0.7703, 0.7989]	**0.6696** [0.6503, 0.6889]	**0.7407** [0.7243, 0.7572]	**0.9617** [0.9593, 0.9642]	**4.4978** [4.4170, 4.5786]	4.9624 [4.9392, 4.9857]	**4.8100** [4.7513, 4.8686]	4.8361 [4.7813, 4.8908]
G. RAG Finetuned	0.4736 [0.4499, 0.4973]	0.7485 [0.7329, 0.7641]	0.6241 [0.6039, 0.6443]	0.7007 [0.6834, 0.7180]	0.9559 [0.9532, 0.9586]	4.4323 [4.3460, 4.5186]	**4.9791** [4.9653, 4.9929]	4.7941 [4.7328, 4.8554]	**4.8871** [4.8398, 4.9344]
H. Custom Data Ideal Default	0.3303 [0.3137, 0.3470]	0.7012 [0.6902, 0.7123]	0.5096 [0.4944, 0.5249]	0.6183 [0.6047, 0.6319]	0.9354 [0.9330, 0.9377]	**4.8107** [4.7632, 4.8582]	4.9826 [4.9646, 5.0006]	4.8503 [4.8020, 4.8986]	4.8764 [4.8334, 4.9193]
I. Custom Data Ideal Finetuned	**0.3622** [0.3440, 0.3804]	**0.7065** [0.6945, 0.7185]	**0.5463** [0.5304, 0.5622]	**0.6441** [0.6299, 0.6582]	**0.9403** [0.9379, 0.9427]	4.5537 [4.4908, 4.6166]	4.9185 [4.8859, 4.9510]	**4.8515** [4.8035, 4.8994]	4.8735 [4.8290, 4.9180]
J. Custom Data RAG Default	0.2982 [0.2815, 0.3149]	0.6688 [0.6568, 0.6808]	0.4751 [0.4591, 0.4910]	0.5848 [0.5705, 0.5991]	0.9297 [0.9272, 0.9323]	4.7406 [4.6899, 4.7913]	**4.9993** [4.9890, 5.0097]	4.8343 [4.7849, 4.8837]	4.9278 [4.8905, 4.9650]
K. Custom Data RAG Finetuned	0.3228 [0.3056, 0.3400]	0.6811 [0.6690, 0.6933]	0.4924 [0.4760, 0.5088]	0.6064 [0.5922, 0.6207]	0.9336 [0.9310, 0.9361]	4.6359 [4.5835, 4.6882]	4.9891 [4.9772, 5.0011]	4.8044 [4.7491, 4.8598]	**4.9516** [4.9164, 4.9867]

baseline model across all metrics. Additionally, baseline C performed better than B across all metrics. This is expected because the model should not have the world knowledge to solve these questions without context, and the VLM is better able to understand tables and 2D formats as images and not transcribed text versions.

In experiments D and E, we ran image and text RAG inference to get a more realistic picture of a RAG system with Qwen3 out of the box. We found that image RAG performed better than text RAG across all metrics, and also had only slight performance drops compared to the ideal (C). Text RAG suffered from larger performance drops relative to ideal (B) likely due to indexing being harder on tables and figures transcribed into text leading to more RAG failures.

For experiments F and G, we first finetuned the VLM in the ideal setting (see figure 1d). We then ran inference on both the ideal setting and the RAG setting using the model to see how its effects change the upper bound performance and real performance of the system. Compared to the non-finetuned equivalents C and E, we saw increases in all metrics besides LLM-as-a-judge metrics for F and G.

Our final set of experiments H, I, J, and K explored running the similar experiments to C, F, E, and G respectively, but on the new 10-x23 slides dataset we created. We observe that finetuning still leads to improvements in our non-LLM metrics for this dataset even though it is not in the same domain as the finetuning training data.

6 Code Overview

Our codebase is organized into two main directories: `scripts/`, which contains the core pipeline modules (retrieval, generation, OCR, data generation, and evaluation), and `experiments/`, which contains the notebooks orchestrating each of the eleven experiments described in Section 5. All notebooks in `experiments/` were written without AI.

In `scripts/vlm.py`, lines 35–124 implement the Qwen3-VL-8B-Instruct inference path along with the optimizations needed to fit generation within 24 GB of VRAM, and contain the system prompt that was iteratively tightened to guide the model towards the question response format. Lines 161–171 configure 4-bit NF4 quantization via `bitsandbytes` with the compute dtype set independently of the storage dtype, and lines 212–255 wrap the quantized base with PEFT LoRA adapters held in a higher precision than the base weights, which is what makes mixed-precision QLoRA numerically stable on a single A10G. It should be noted that a lot of these optimizations were made redundant once the laborious switch from a AWS VM to Colab was made towards the end of this project.

In `scripts/rag.py`, lines 34–48 initialize a Qdrant multi-vector collection so the per-patch ColQwen3.5 embeddings are preserved natively for late-interaction MaxSim scoring rather than mean-pooled; because the VM ran out of RAM when these embeddings were held in memory alongside the VLM, retrieved embeddings are serialized to disk as NumPy arrays for the generator to read back in. Lines 128–149 implement top-k retrieval over this index. In `scripts/ocr.py`, lines 79–96 embed Tesseract-extracted page text with `all-MiniLM-L6-v2`, lines 179–180 add the chunked text to ChromaDB, and lines 224–244 perform top-k cosine-similarity retrieval; for both `rag.py` and `ocr.py`, AI assistance was used for the file-handling and data-serving paths into the vector stores, while the retrieval logic and VLM integration were written by the team. Finally, lines 92–168 of `scripts/generate_custom_data.py` contain the teacher prompt used by Qwen3-VL-30B to synthesize QA pairs from each lecture slide; this prompt required substantial iteration because the model repeatedly produced questions referring to the slide as an artifact ("what is the main topic of this slide?"), and the final version accumulates an explicit list of forbidden phrasings and a fallback to produce `NO QUESTION` when no slide-independent question can be formed (full prompt in Appendix A).

7 Timeline, Human Work, & AI Use

7.1 Time Breakdown

Table 2 summarizes approximate hours spent on each stage of the project, aggregated across all team members. The largest share went to writing and modifying code (data pipelines, finetuning scripts, evaluation harnesses, and the QA generation pipeline) and to running experiments, where wall-clock time was dominated by finetuning runs and inference sweeps over the held-out test sets and the curated lecture-slide set.

Activity	**Approx. Hours**
Reading papers (REAL-MM-RAG, RAG, ColPali, related VLM work)	6
Reading code documentation	4
Compiling and running existing code (and environment setup)	8
Writing/modifying code (data, finetuning, evaluation, QA generation pipelines)	30
Writing scripts to run experiments (sweep configs, batch jobs)	10
Running experiments (finetuning runs, inference, QA generation, eval)	30
Compiling and analyzing results (metrics, qualitative inspection)	12
Report writing and revision	10
Total	**110**

Table 2: Approximate hours per project stage, aggregated across team members.

7.2 Work Partition

The work was distributed across team members as summarized in Table 3. Most contributions involved collaboration between two or more members; the table identifies the primary owner of each component.

Member	**Primary Contributions**
Nicolas	Dataset preparation and preprocessing the REAL-MM-RAG financial-report data; ColQwen retriever integration and indexing; ideal-RAG inference notebooks for both base and finetuned models; debugging of model loading issues on Qwen3-VL variants; lecture-slide curation and QA generation pipeline (shared with Darren); Report writing.
John	Finetuning pipeline implementation and LoRA configuration, technical-report, and technical-slides splits; evaluation harness (BLEU, ROUGE, BERTScore, paired significance testing); LLM-as-a-judge batch job creation using Together AI API; analysis of cross-domain transfer results. Report writing
Darren	QA generation pipeline (shared with Nicolas); Report writing.

Table 3: Primary contributions per team member.

7.3 AI Use

AI assistance was used primarily for data-ingestion logic, file processing, and OS-level glue (e.g., serving embeddings and images into the vector stores and across the VM filesystem), as well as for debugging runtime errors during pipeline integration. All AI-generated code was manually reviewed, modified, and tested by the team to ensure correctness and that we understood every line we kept.

8 Research Log

We originally planned to run five different experimental setups instead of the eleven setups we ended up exploring. Generally, this change was an expansion of our exploration done to check our results against performance upper bounds. Originally, we had no idealized context inference experiments (B and C in table 1), but decided to explore this route as

additional baselines to get an idea of a default upper bound performance of the VLM without any noise from the RAG model. We kept measuring this ideal upper bound across different experiments as a way to better measure the performance of the RAG system (experiments F, H, and I).

We did also have to scale back some explorations. ideally, we would have explored and finetuned the full RAG and VLM system, but the compute we were able to access through AWS could not fit the RAG and VLM models for finetuning in GPU memory. Because of this, we used AWS for inference and Google Colab credits from Colab Pro for finetuning. After finetuning only one model, we ran out of Colab credits and could not further explore the different finetuning trials. We had also originally planned to generate enough custom data to be able to finetune another model in addition to the finetuning on the REAL-MM-RAG benchmark data in order to compare how much potential improvement we could get from tangential data versus specific domain specific data. Unfortunately, we did not have sufficient data or computation for this experiment. Using all of the course lecture slides, we did not have sufficient samples for finetuning, and we had also run out of Google Colab compute credits.

We also changed our LLM-as-a-Judge metric collection. First, instead of collecting only two metrics (accuracy and groundedness), we added relevance and faithfulness to measure additional aspects of quality and how much the VLM adheres to the context. We also changed our scoring plan from a simple boolean 0 or 1 for non-compliance and compliance, we used a 1-5 scale graded on a rubric we provided to the judge LLM. This captured more nuance and allowed the judge to not overly punish almost correct answers with a 0. Lastly, we also changed the model we intended to use from GPT based to the new Gemma-4 model. This was because we did not have access to OpenAI API credit that we initially thought we would, and the Gemma model offered improved performance compared to the GPT models we were looking at at a reduced inference rate on other API services like TogetherAI.

9 Conclusion

Our investigation found that not only is context important to answering specific questions, but the correct mode of context is. Across all experiments, image based retrieval and context performed better than text equivalents due to the better spatial reasoning possible with images. We also found that while finetuning enhanced statistical metrics, LLM-as-a-judge metrics decreased as a result. This may be because finetuning shifts the LLM answers to better match the expected format, but not necessarily produce better answers. Future exploration could explore other approaches to improving responses like in context learning, or explore how modifications to the LLM-as-a-judge contribute to scores to determine the best learning and evaluation modes for consistent improvements and results.

10 Thought-Experiment on Compute

In total, we used over 150 hours of AWS g5.xlarge EC2 instance time (equip with a 24GB A10G Nvidia GPU), 20 hours of Google Colab A100 GPU processing (300 credits), and 15.4 million Gemma-4 tokens in TogetherAI. This comes to $165, $30 and $5.39, respectively, or about $200 overall. While we did not use our full $450 allotment, this was because we were not able to access any more powerful GPUs than an A10G due to AWS account limits, and we did use up all of our Colab Pro credits and pay out of pocket for the Gemma inference we needed.

If we had $1000 of compute credit and could actually spend it on A100 GPUs, this would afford us more hours for finetuning, both by allowing finetuning in the first place, but also by reducing inference/train time. From AWS, the cheapest available A100 VM is a p4d.24xlarge instance with 8 x A100 GPUs for $21.96 per hour. This would give us 45 hours of compute. Using this time, we would expand our exploration to look at finetuning both the retrieval process and VLM instead of just the VLM. We would have also looked at generating more data in order to also train an in domain model to explore how much of the potential finetuning benefits can come from a tangential dataset even if you have no domain dataset. We also could stop clipping contexts and use a larger model with the additional GPU space.

References

[1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*, 2020.

[2] Jack Rae et al. Improving language models by retrieving from trillions of tokens. *DeepMind*, 2022.

[3] Jean-Baptiste Alayrac et al. Flamingo: a visual language model for few-shot learning. *DeepMind*, 2022.

[4] Junnan Li et al. Blip-2: Bootstrapping language-image pre-training. *arXiv preprint arXiv:2301.12597*, 2023.

[5] Minesh Mathew et al. Docvqa: A dataset for vqa on document images. In *WACV*, 2021.

[6] Minesh Mathew et al. Infographicvqa. In *WACV*, 2022.

[7] Navve Wasserman, Roi Pony, Oshri Naparstek, Adi Raz Goldfarb, Eli Schwartz, Udi Barzelay, and Leonid Karlinsky. REAL-MM-RAG: A real-world multi-modal retrieval benchmark. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31660–31683, Vienna, Austria, July 2025. Association for Computational Linguistics.

[8] Qwen Team. Qwen3 technical report, 2025.

[9] athrael soju. Colqwen3.5: Visual document retrieval with hybrid linear-attention models, 2026.

[10] Edward J. Hu et al. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[11] Colbert: Efficient and effective passage search via contextualized late interaction over bert. *arXiv preprint arXiv:2004.12832*, 2020.

[12] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

[13] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[14] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[15] Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, Siyang Qin, Reeve Ingle, Emanuele Bugliarello, Sahar Kazemzadeh, Thomas Mesnard, Ibrahim Alabdulmohsin, Lucas Beyer, and Xiaohua Zhai. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*, 2024.

[16] Potsawee Manakul et al. Selfcheckgpt: Zero-resource black-box hallucination detection. *EMNLP*, 2023.

[17] Cheng-Yu Hsieh et al. Distilling step-by-step: Outperforming larger language models. *arXiv preprint arXiv:2305.02301*, 2023.

[18] Lucie Magister et al. Teaching small language models to reason. *arXiv preprint arXiv:2212.08410*, 2022.

[19] Tesseract github repo, 2026.

11 Appendix A

The following system prompt was used with the teacher VLM to generate question-answer pairs from each lecture slide. The placeholders {args.num_questions} and {prompt_instruction} are substituted at runtime with the requested number of QA pairs and a document-type-specific instruction respectively.

```
You are given a single page (slide) from a lecture deck. The slide
may contain text, figures, equations, tables, and diagrams.

Your task is to produce {args.num_questions} question-answer pairs
that an interested learner might ask to test their understanding of
the topic shown on this slide. The questions must be self-contained
and answerable by someone who can see the slide but who is not aware
that any 'slide', 'page', 'document', 'figure', 'dialogue',
'example', or 'lecture' is being referenced. Write questions about
the SUBJECT MATTER itself, not about the slide as an artifact. Under
absolutely no circumstances use references such as 'in the
dialogue', 'on the slide', 'in this slide' etc., or questions
containing an explicit reference to a slide. All of the questions
should be questions about the topic the slide is on, but should in
no way reference the slide. For example, a question containing 'on
the slide discussing ...' is absolutely forbidden under all
circumstances.

Substituting words for slide is absolutely forbidden. An example of
such a workaround which is not allowed is the following: 'What is
the primary focus of the content presented in the document?'. The
preceding question is an example of a question which is not about
the subject matter on a document but rather about the document
itself. Only ever write questions about the subject matter and
concepts without reference to the document in the question, this is
absolutely crucial.
If you cannot make a relevant question without explicitly
referencing the slide, write your question as 'NO QUESTION'.

Each question must:
- Refer to the actual concepts, techniques, theorems, names, or
  quantities shown
- Be answerable in 1-3 sentences from information uniquely supported
  by this slide
- Use specific terms (e.g. 'Euclid's proof', 'softmax',
  'transformer', 'a CNN with 3 layers') rather than generic ones
  ('this proof', 'the model', 'the architecture')
- Where useful, combine information from multiple parts of the slide

{prompt_instruction}

Examples of GOOD questions for a slide explaining Euclid's proof of
infinite primes:
- How does Euclid's proof construct a new number to derive a
  contradiction?
- Why must the constructed number Q have a prime factor larger
  than P?
- What assumption does Euclid's proof of the infinitude of primes
  start from?

Examples of BAD questions (DO NOT WRITE QUESTIONS LIKE THESE):
- What is the main topic being introduced on this slide? [refers to
  the slide as an artifact]
- What proof is being discussed in the dialogue? [refers to slide
  framing]
- What can you learn from this figure? [refers to a visual element]
- What is the title? [trivial / about the slide, not the content]
- How does the model work? [too generic, no specific name]

Return the output EXCLUSIVELY as a valid JSON array of objects.
Each object must have exactly two string keys: "question" and
"answer". Do not include markdown code fences, do not include any
text before or after the JSON array.
```

Below is the QA pair data generation prompt from the original REAL-MM-RAG paper [1].

```
You are given a single page from a document. This page may contain text, figures, tables, and diagrams. The document is about
{Document_short_description}. Make sure to mention the company or product name when the question pertains to specific information about it.

Your task is to produce 10 question-answer pairs. Each question should represent a plausible inquiry that a person (who has not seen the page)
might ask about the information uniquely presented on this page. The questions should not reference this specific page directly (by page
number, pointing to a specific paragraph or figure, and never refer to the document using phrases like 'in the document'), nor should they
quote the text verbatim. They should use natural language reflecting how someone might inquire about the page's content without direct access.

Please focus mainly on tables and figures rather than just text. Try to formulate questions that require data from multiple locations on the
page. The answer should be uniquely supported by the content of this page (i.e., the user could not find the answer elsewhere in the report or
infer it without seeing this particular page). Ensure that the questions are not too general. For example, refer to the company name instead of
using "the company" generically, as different documents may refer to different companies.

Good questions:
What types of properties does Wihlborgs specialize in?
How many women are in Wihlborgs' group management?
What was the occupancy rate of Wihlborg properties in 2021?
Was the net letting positive in the first quarter of 2022?
Which key factor contributed the most to the change in property value in 2022?
What's the difference in the number of properties in South Copenhagen between 2021 and 2022?
What is the proportion of women on the board of directors?
How many dividends were paid in 2020 in total equity?
What portion of Wihlborg's sponsorship in 2018 was community-oriented?

Bad questions:
What is the title of the page?
What is the main purpose of this page?
What can you learn from the figure?
What is the name of the CEO mentioned in the document?
What is IBM's clear path to growth according to the document?
What is the basic earnings per share for 2023?

Make sure to include references to the company and product names when needed.

Output format:
Return as a list of objects:
[
    { "query": "...", "answer": "..." },
    { "query": "...", "answer": "..." },
    { "query": "...", "answer": "..." },
    { "query": "...", "answer": "..." },
    { "query": "...", "answer": "..." }
]
```

12 Appendix B

This is the scoring rubric prompt used by Gemma 4 for the LLM-as-a-judge measures.

```
You are an expert AI evaluator grading the performance of a Retrieval-Augmented Generation (RAG)
system.
You will be provided with a Question, a Ground Truth Answer, the Generated Answer, and the
Retrieved Context (which may be text or images).

Your task is to evaluate the Generated Answer across four categories on a scale of 1 to 5.

### Categories:
1. Answer Accuracy: How well does the Generated Answer match the Ground Truth?
   - 1: Completely incorrect or contradicts the ground truth.
   - 3: Partially correct but missing key details.
   - 5: Perfectly matches the ground truth semantics and facts.
2. Answer Relevance: How directly does the Generated Answer address the Question?
   - 1: Off-topic or fails to answer the question.
   - 3: Answers the question but includes unnecessary filler.
   - 5: Directly and concisely answers the question.
3. Groundedness: To what extent are the claims in the Generated Answer supported by the Retrieved
Context?
   - 1: None of the claims are found in the context.
   - 3: Some claims are supported, but others are missing from the context.
   - 5: Every claim in the answer is explicitly backed by the context.
4. Faithfulness: Does the Generated Answer avoid hallucinations and remain true to the context?
```

```
   - 1: Hallucinates external facts or directly contradicts the context.
   - 3: Mostly faithful, but makes minor logical leaps not supported by the context.
   - 5: Strictly adheres to the context with zero hallucinations or contradictions.

### Output Format
You must output ONLY valid JSON in the following format. Do not include markdown blocks or any other
 text.
{
  "accuracy_reason": "<brief justification>",
  "accuracy_score": <int>,
  "relevance_reason": "<brief justification>",
  "relevance_score": <int>,
  "groundedness_reason": "<brief justification>",
  "groundedness_score": <int>,
  "faithfulness_reason": "<brief justification>"
  "faithfulness_score": <int>,
}
```