Winning Meta KDD Cup'25 Task 2

Raja Biswas*
rabiswas@nvidia.com
NVIDIA
Singapore

Chris Deotte*
cdeotte@nvidia.com
NVIDIA
USA

Gilberto Titericz Jr*
gtitericz@nvidia.com
NVIDIA
Brazil

Kazuki Onodera*
konodera@nvidia.com
NVIDIA
Japan

Ahmet Erdem*
aerdem@nvidia.com
NVIDIA
Türkiye

Abstract

The KDD Cup 2025 Task 2 focused on building multimodal RAG systems for visual question answering while minimizing hallucinations. Team NVIDIA's winning solution leveraged a fine-tuned Llama-3.2-11B-Vision-Instruct VLM to perform three critical subtasks: generating web search queries, re-ranking contexts, and producing grounded answers. The VLM was trained to output "I don't know" when retrieved information was insufficient.

A fine-tuning datamix comprising 26.5k samples was carefully curated from 2.5k competition examples using NVIDIA NIM llama-4-maverick for synthetic data generation and GPT-4o as LLM-as-a-judge. The datamix curation, VLM fine-tuning process, and hyperparameter selection were optimized using NVIDIA RAGAS Accuracy metric [7] — a blend of three LLM judges (aka council of judges, including Nemotron) that achieves 0.92+ correlation with human judgment . Since the competition's final evaluation was determined by human judges, tuning our pipeline using RAGAS offered a strong advantage.

Pipeline responses were post-processed with an "I don't know" probability threshold optimized using the NVIDIA RAGAS Accuracy metric[7]. Our approach achieved a final human evaluation score of 0.233, securing first place by effectively balancing answer coverage with hallucination prevention.

CCS Concepts

• **Computing methodologies** → **Natural language generation**; *Information extraction.*

Keywords

Visual Question Answering, Retrieval-Augmented Generation, Multimodal LLM, Vision Language Models, Fine-tuning, LLaMA, Hallucination Detection, Multitask Learning, NVIDIA RAGAS, KDD Cup 2025

*All authors contributed equally to this research.

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Raja Biswas, Chris Deotte, Gilberto Titericz Jr, Kazuki Onodera, and Ahmet Erdem. 2025. Winning Meta KDD Cup'25 Task 2. In *Proceedings of* . ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

[1]. Vision Large Language Models (VLLMs) have undergone significant advancements in recent years, empowering multi-modal understanding and visual question-answering (VQA) capabilities behind smart glasses. Despite the progress, VLLMs still face a major challenge: generating hallucinated answers. Studies have shown that VLLMs encounter substantial difficulties in handling queries involving long-tail entities [11]; these models also encounter challenges in handling complex queries that require the integration of different capabilities: recognition, OCR, knowledge, and generation [12].

The Retrieval-Augmented Generation (RAG) paradigm has expanded to accommodate multi-modal (MM) input and demonstrated promise in addressing the knowledge limitation of VLLM. Given an image and a question, an MM-RAG system constructs a search query by synthesizing information from the image and the question, searches external sources to retrieve relevant information, and then provides grounded answers to address the question [2].

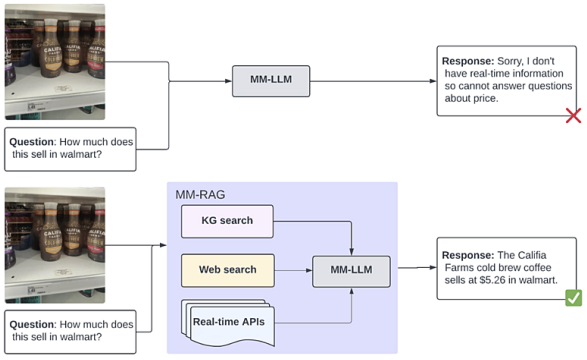

Figure 1: Multi modal RAG pipeline.

Despite its potential, MM-RAG still faces many challenges, such as recognizing the correct subject and comprehending the visual context in the image to understand the question, performing effective searches to retrieve useful information, synthesizing information from different sources to generate coherent and informative

answers, and engaging in smooth multi-turn conversations. A comprehensive benchmark that provides a standardized framework and clear metrics is in pressing need to enable reliable and informative assessment of MM-RAG systems to facilitate and advance innovations.

CRAG-MM is a visual question-answering benchmark that focuses on factual questions, offering a unique collection of image and question-answering sets to enable comprehensive assessment of wearable devices. Specifically, CRAG-MM features a diverse collection of **5k images**, including **3k egocentric ones** captured by **RayBan Meta smart glasses**, covering **13 domains** and reflecting real-world challenges associated with handling egocentric images.

The benchmark includes **4 types of questions**, ranging from simple queries that can be answered by looking at the image only to complex ones that require retrieving information from multiple sources and performing reasoning.

Moreover, CRAG-MM encompasses both **single-turn and multi-turn conversations**, providing a more overarching evaluation of MM-RAG solutions.

2 Meta KDD Cup 2025: CRAG-MM Challenge

[1]. An MM-RAG QA system takes as input an image I and a question Q, and outputs an answer A; the answer is generated by MM-LLMs according to information retrieved from external sources, combined with knowledge internalized in the model. A **Multi-turn MM-RAG QA system**, in addition, takes questions and answers from previous turns as context to answer new questions. The answer should provide useful information to answer the question without adding any hallucination.

Four types of questions were defined in the competition benchmark:

- **Simple questions**: Questions asking for simple facts.
  - **Simple recognition**: This can be directly answered from the image (e.g., "What brand is the milk?" or "Who wrote this book?" where the brand name and the book author are shown on the image).
  - **Simple knowledge**: Requires external knowledge for the answers (e.g., "What's the price of this sofa on Amazon?").
- **Multi-hop questions**: Questions that require *chaining multiple pieces of information* to compose the answer (e.g., "What other movies have the director of this movie directed in the past?").
- **Comparison and Aggregation questions**: Questions requiring *aggregating or comparing multiple pieces of information* (e.g., "Which drinks do not contain added sugar among these?" or "Is this cheaper on Amazon?").
- **Reasoning questions**: Questions about an entity that *cannot be directly looked up* and require *reasoning* to answer (e.g., "Can the dryer be used in Europe?" where the image shows a dryer).

The challenge comprised of **three competition tasks**. **Task #1 and Task #2** contain *single-turn questions*, where the former provides *image-KG-based retrieval*, and the latter additionally introduces *web retrieval*; **Task #3 focuses on multi-turn conversations**.

The content that can be leveraged in QA was provided to ensure fair competition. Additional information on the three tasks are provided in the next section.

Task #1: Single-Source Augmentation

- **Goal**: To test the **basic answer generation capability** of MM-RAG systems.
- Provides an *image mock API* to access information from an underlying **image-based mock KG**. The *mock KG* is indexed by the image and stores structured data associated with the image. Answers to the questions may or may not exist in the mock KG. The *mock API* takes an image as input and returns similar images from the *mock KG*, along with *structured data* associated with each image to support answer generation.

Task #2: Multi-Source Augmentation

- **Goal**: To test how well the **MM-RAG system synthesizes information from different sources**.
- In addition to **Task #1**, this task provides a *web search mock API as a second retrieval source*. The *web pages* may provide *useful information* for answering the question but also *contain noise*.

Task #3: Multi-Turn QA

- **Goal**: To test **context understanding for smooth multi-turn conversations**.
- This task tests the system's ability to conduct *multi-turn conversations*. Each conversation contains **2–6 turns**. Except for the first turn, *questions in later turns may or may not require the image* for answering.

3 LLM as Judge - NVIDIA RAGAS

The final outcome for competition's task 2 is determined by human judges who evaluate the truthfulness of each teams' answer responses. Specifically correct answers are awarded 1 point, partial answers 0.5 point, incorrect answers −1 point, and "I don't know" responses 0 points. The final truthfulness score is the average point value per response [1].

Single-Turn QA.

- For each question in the evaluation set, the answer is scored as:
- **Perfect** (fully correct) → **Score: 1.0**
- **Acceptable** (useful but with minor non-harmful errors) → **Score: 0.5**
- **Missing** (e.g., "I don't know", "I'm sorry I can't find ...") → **Score: 0.0**
- **Incorrect** (wrong or irrelevant answer) → **Score: -1.0**
- **Truthfulness Score**: The *average score* across all examples in the evaluation set for a given MM-RAG system.

Therefore for each question, it is important to know when to say "I don't know" and when to give a response. Team NVIDIA's task 2 solution can estimate the probability of the response being incorrect by using the probability of the first output token being

the capital letter "I" signifying that the model is considering saying "I don't know". We then employ the following post process:

```
if IDX_prob > IDX_threshold:
    responses[i] = "I don't know"
```

Using local validation, we determine that the optimal threshold to maximize public LB is 0.065. And the optimal threshold to maximize the final human evaluation is 0.166. The public leaderboard score is computed using GPT-4o-mini as judge without partial correct grades. Therefore we can use GPT-4o-mini to compute validation score locally to estimate public leaderboard score without submitting.

Estimating the competition final human evaluation score is more difficult because we need to evaluate local predictions in a similar way that humans would reward partial credit. To accomplish this, we use NVIDIA RAGAS Accuracy Metric[7] with a blend of 3 judges (aka council of judges) that achieves a high human correlation of 0.92+ in internal experiments. Judges used were:

- `meta/llama-3.1-70b-instruct`[3]
- `mistralai/mixtral-8x22b-instruct-v0.1`[4]
- `nvidia/llama-3.3-nemotron-super-49b-v1`[6]

RAGAS NV accuracy metric[7] was developed using a RAG dataset with 2,489 samples from eight datasets and inferred using a standard RAG implementation. The generated responses and golden references were then analyzed by three human judges, who assigned scores of 0 (incorrect), 0.5 (partial), or 1 (correct). The average of the human scores was then compared with the blend of 3 LLM-as-a-judge scores using Pearson's correlation, resulting in a value greater than 0.92, confirming the metric's high alignment with human judgment. Judge prompts can be found in the RAGAS repository[8].

With the ratings from the council of judges we can then estimate a human judgement with:

- $p > 0.7$: Correct
- $0.5 \leq p \leq 0.7$: Partially Correct
- $p < 0.5$: Incorrect

Putting this all together in Figure 2, the x-axis is the threshold for generating an "I don't know" response and y axis is the local validation score. When we define the threshold smaller in our code, then the model is forced to say "I don't know" more and when we define the threshold larger, the model is allowed to answer unfiltered more (and potentially hallucinate more).

The solid orange line is the locally computed NVIDIA RAGAS Accuracy metric[7] which includes partially correct labels. The optimal threshold to maximize NV Accuracy is 0.166 indicated by the dotted orange vertical line. Since local validation estimates the final competition score, we used this threshold in our code for our final solution.

4 Task 2 NVIDIA Final Solution

Our approach leverages multi-task finetuning of the Llama-3.2-11B-Vision-Instruct VLM to handle multiple subtasks in a unified way. Rather than training separate models for each component, we taught a single VLM to perform web query generation, context re-ranking, and answer generation through careful dataset curation and workflow setup, as depicted in Figure 3.

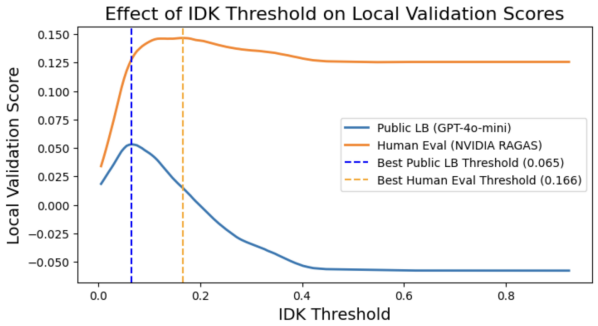

Figure 2: Effect of IDK threshold on local validation scores using two evaluation proxies. The plot shows that an IDK threshold of 0.065 optimizes the GPT-4o-mini public leaderboard score, while 0.166 optimizes the NVIDIA RAGAS based human-eval-aligned score.

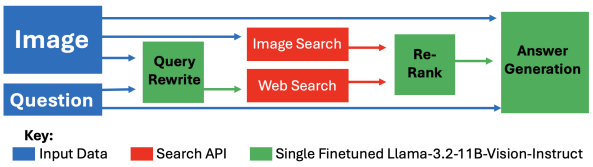

Figure 3: A single finetuned Llama-3.2-11B-Vision-Instruct performs three subtasks in a visual question answering challenge.

The key insight was that by providing better contexts during answer generation and teaching the model when to say "I don't know", we could significantly reduce hallucinations while maintaining high accuracy. This multi-task approach allowed the model to develop a holistic understanding of the RAG pipeline.

4.1 Pipeline Overview

Our task 2 pipeline is illustrated in Figure 3. The system takes a question-image pair as input and processes it through several stages.

Initially, our finetuned VLM analyzes the question and image to generate 8 diverse web search queries (Web Query Generation). For each query, we retrieve the top 4 results from the competition's web search API. In addition, we create crops from the original image and submit them to the competition's image search API. Our finetuned VLM then re-ranks all retrieved contexts from both the web and image search APIs based on their relevance to the original question.

The final answer generation stage combines the original question, image, and top N re-ranked contexts (up to 8) as input to our finetuned VLM. The model has been specifically trained to output "I don't know" when retrieved information deemed insufficient.

Finally, we post-process our response based on the probability that the first generated token will be "I", denoted as IDK Probability. Guided by the local validation as explained in Section 3, we determined that if the IDK probability was greater than 0.166 then we

replace the response with "I don't know" to avoid a score penalty for providing a potentially incorrect answer.

5 Finetuning Llama-3.2-11B-Vision-Instruct

We employed LoRA (Low-Rank Adaptation) for efficient finetuning with the following configuration:

- Rank (r): 16
- LoRA alpha: 32
- LoRA dropout: 0.05
- Target modules: down_proj, o_proj, k_proj, q_proj, gate_proj, up_proj, v_proj
- DoRA: Disabled.

The training setup utilized 8 NVIDIA A100 GPUs with the following hyperparameters:

- Batch size: 1 per device with gradient accumulation steps of 4 (effective batch size: 32)
- Learning rate: 1e-5 with cosine scheduler
- Warmup steps: 32
- Optimizer: AdamW 8-bit
- Precision: bfloat16
- Maximum gradient norm: 16.0
- Maximum token length: 3072.

5.1 Web Query Generation

Web query generation was first of the 3 subtasks we focused on, as shown in Figure 3. Given a question-image pair, we fine-tuned a VLM to provide helpful search strings that could be used with competition's web search API to retrieve helpful contexts. We approached this as a distillation task, leveraging the powerful Llama-4-maverick-17b-128e-instruct model to generate high-quality training data. The key challenge was to (a) include relevant visual details in the search strings and (b) produce diverse queries to optimize recall of relevant information. We addressed these challenges via prompt engineering during synthetic query generation.

5.2 Re-ranker

The re-ranking component was trained as a binary classification task to determine whether a retrieved context is relevant to answering the given question. This approach allowed us to filter out irrelevant or misleading contexts that could cause hallucinations in the final answer. The model was trained to output simple "yes" or "no" responses, with logit probabilities used as relevance scores during inference. The re-ranker component enabled us to sort candidate contexts retrieved from both image and web search API endpoints. We also set a minimum relevance threshold for a context to considered for final answer generation.

5.3 Answer Generation

Answer generation represented the most critical component of our pipeline, where the model synthesizes information from multiple retrieved contexts to produce accurate, grounded responses. This subtask was aimed at:

- teaching the model to respond with "I don't know" when the available external information (retrieved via search APIs) was insufficient,
- learning to be resilient against potentially (very) noisy information sources,
- producing a reliable answer when confident.

This was achieved by a careful data creation strategy as described in Section 6.3. The proportion of "I don't know" (IDK) examples proved crucial - too many IDK training examples made the model excessively conservative, refusing to answer even when sufficient context was available. Through iterative refinement, guided by NVIDIA RAGAS Accuracy Metric[7], we found the optimal balance between hallucination prevention and answer coverage.

6 Train Dataset Creation

Creating a high-quality fine-tuning data mix was critical to the success of the multitask learning strategy. We carefully engineered a data pipeline that leveraged competition search APIs, synthetic data generation, and quality filtering to produce datasets that effectively taught each capability to our model. The data pipeline is illustrated in Figure 4. Fine-tuning examples were derived from the official crag-mm-single-turn-public and crag-mm- multi-turn-public datasets (v0.1.2), which contained 2.5k triplets of (question, image, answer). It was processed in various ways, resulting in a data mix comprising of 26.5k samples across the 3 stated subtasks. This dataset is available as `rbiswasfc/kddcup-sft-datamix` [10].

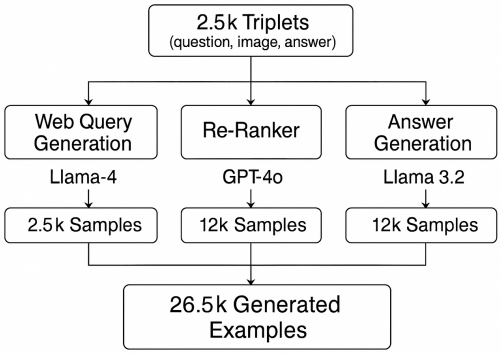

Figure 4: 26.5k finetune train samples were generated from 2.5k competition data

6.1 Web Query Generation Dataset

The Web query generation dataset was created to facilitate the distillation of knowledge from Llama-4-maverick-17b-128e-instruct, a more powerful model capable of generating high-quality and diverse search strings. The purpose of this dataset was to encourage a fine-tuned Llama-3.2-11B model to:

- Extract visual details from images and incorporate them into queries
- Generate queries targeting different aspects of the question
- Balance specificity with generality for optimal retrieval
- Avoid redundant or trivially similar queries.

NVIDIA NIM was used to call the Llama-4-maverick-17b-128e-instruct endpoint during the synthetic web query generation.

Specifically, our synthetic data generation step used the following prompt to generate diverse queries:

```
You will be shown an image and a question about that image.
Your task is to generate 8 short, diverse and effective web
search queries that a person could type into a search engine
(like Google) to find information that would help answer the
question.

Critical constraint: The search engine CANNOT see the image.
You must include visual details in the queries.

Your mission: Generate queries that will return search
results containing the answer to the question.

Process:
1. Understand the visual context and identify the possible
   subject(s) the question could be referring to
2. For each possible subject, create search queries that
  include subject specific keywords and aspects related
   to the question
3. Create queries for each plausible subject

Query optimization:
- Make each query short and self-contained (8-12 words)
- Distribute queries across different plausible
  interpretations from visual context and posed question
- Try to maximize answer retrieval possibility i.e. recall
- Avoid redundant phrasing and trivial re-ordering
- Use varied search approaches

Do not answer the question. Just return 8 search queries
as a numbered list.

Question: {query}

Now, generate search strings that will retrieve the ANSWER
to this question. The search engine cannot see the image -
include visual details.
Provide 8 diverse search queries.
```

As observed, our diversity constraints were primarily prompt-based. We enforced semantic diversity by distributing queries across different plausible interpretations from visual context. We set temperature at 0.2 during generation to encourage variations while maintaining coherence. Coverage diversity was ensured through explicit instruction to target different aspects/subjects of the visual content.

For instance, given the question "What is the average lifespan of this plant?" about an image containing a red Amaryllis flower, the query generation output contains:

- Amaryllis flower average lifespan
- Red Amaryllis plant life expectancy
- Indoor Amaryllis bulb lifespan
- Amaryllis flower longevity
- How long do Amaryllis plants live
- Average lifespan of potted Amaryllis
- Amaryllis plant care and lifespan
- Lifespan of Amaryllis with large red flowers.

6.2 Re-ranker Dataset

The re-ranker dataset leveraged both image and web search API endpoints to create a comprehensive training set for relevance classification. The dataset creation process involved:

- Retrieving potentially relevant candidates from web search and image search api's
- Generating answers using Llama-3.2-11B-Vision-Instruct while providing one retrieved context at a time
- Using GPT-4o to evaluate whether a context led to the correct answer, thereby assigning positive and negative labels
- Sampling positive and negative contexts maintaining a 1:5 ratio

The re-ranking dataset comprised of 2k and 10k positive and negative train examples respectively.

6.3 Answer Generation Dataset

The answer generation dataset was created to encourage our pipeline to produce accurate and grounded responses, naturally leading to reduced hallucinations. For this sub-task, the data pipeline involved the following steps:

(1) Concatenating up to 8 web or image search results to create a VLM context, which represents specific external knowledge available to the model for answering a visual query
(2) If at least one of the above search results lead to the correct answer (as determined during re-ranker dataset creation), then keep the ground truth (GT) answer as target
(3) Else, replace the GT answer with "I don't know" (IDK) as target
(4) Furthermore, we reduced proportion of IDK examples to avoid overly conservative behavior

Our iterative approach to dataset curation, guided by RAGAS evaluation metrics, resulted in a dataset that effectively taught the model when to avoid answering, reducing hallucinations while maintaining high accuracy on answerable questions.

7 Ablation Study

Our pipeline components were highly coupled and optimized jointly to maximize the final human evaluation score. While this makes it difficult to isolate the exact contribution of each component in standalone manner, we provide key insights on their importance.

The "I don't know" (IDK) threshold proved critical for balancing accuracy and hallucination. Operating without a threshold (equivalent to setting it to zero) would lead to attempting every question, resulting in very high hallucination rates. Two representative settings demonstrate this impact:

- IDK threshold = 0.166 (final submission): 50.5% missing rate, 25.7% accuracy, 23.8% hallucination rate (public LB), optimized for human evaluation
- IDK threshold = 0.09: 75.6% missing responses, better public LB truthfulness (0.073), but lower attempt rate (15.8% accurate, 8.6% hallucinations)

Since our fine-tuned model learned to output a well-calibrated IDK probability score, it allowed us to make reliable design choices.

The RAGAS evaluation metrics, which correlate strongly with human judgments, were instrumental in tuning this threshold accurately for the final private leaderboard.

We determined our web query generation strategy through joint optimization with the re-ranker and context filtering. Comparing two approaches:

- Deep strategy: 4 queries with 8 results each
- Broad strategy: 8 queries with 4 results each

The broad approach yielded significantly better validation scores (0.073 vs 0.056), confirming that wider query diversity was more effective for our pipeline. We didn't experiment beyond 8 queries due to diminishing returns and inference compute constraints.

8 Task 2 Alternative Solution

An alternative to finetuning Llama-3.2-11B-Vision-Instruct is to use VLM as zero shot with prompt engineering and then employ a meta model afterward to determine when to say "I don't know". This alternative was able to achieve 85% of the performance as our finetuned model as shown in Table 1 in Section 9

8.1 Meta Model

We trained a random forest classification meta model which determined when to say "I don't know" using features extracted from Llama-3.2-11B-Vision-Instruct response token probabilities and RAG retrieval similarity scores and tf-idf features. The meta model was trained on labels from NVIDIA RAGAS Accuracy Metric[7].

8.2 Feature Extraction

The pipeline works as follows. Starting from the original question and image, the VLM would first use zero shot with prompt engineering to generate web search queries. These queries were used to retrieve similar web text chunks using the competition web search api. Image search was not used. Then the VLM would use the original question, image, and retrieved chunks to zero shot with prompt engineering to generate an answer.

Features were created from VLM response token probabilities and RAG similarity scores. Additionally, the VLM was asked 12 questions about its response, the original question, image, and retrieved text chunks.

Here are some example feature extraction questions. The first 4 are yes/no questions and the next 4 are respond-with-number (1 through 5) questions. Afterward, token probabilities are extracted from the `yes`, `no`, `1`, `2`, `3`, `4`, `5` tokens:

- Is response supported by visual evidence?
- Is response speculative?
- Is response plausible?
- Does query answering require the image?
- How correct is the response?
- How confident does the response sound?
- How detailed is the response?
- How visually grounded is the query?

From these 8 questions and 4 others, token probabilities were extracted as additional features. Finally, TF-IDF was used on the original question and VLM response to create 50 more features.

The random forest meta model would then classify VLM response as either correct or incorrect. A threshold was tuned on local validation scores to optimize NVIDIA RAGAS Accuracy metric[7] including partial credit. All responses below the threshold would be converted into "I don't know" to avoid being penalized by the competition metric.

9 Results

Team NVIDIA's final task 2 solution leveraged a finetuned Llama-3.2-11B-Vision-Instruct model described in Section 4. It's performance results are shown in the first row of Table 1. Our zero shot alternative solution with stage 2 meta model performance results are shown in the second row of Table 1.

Column "GPT-4o-mini" is the local evaluation score using the provided validation data and training on non-validation data. This score does not award partial credit and estimates the public test leaderboard score which is displayed in column "Test Leaderboard". The column "NVIDIA RAGAS" is the local evaluation using the council of judges which does award partial credit and estimates the private test human evaluation displayed in column "Test Human Eval".

Model	**GPT-4o-mini**	**NVIDIA RAGAS**	**Test Leaderboard**	**Test Human Eval**
Finetuned	0.0532	0.1666	0.075	0.233
Zero-Shot + Meta	0.0446	0.1407	0.061	NA

Table 1: Evaluation metrics across different benchmarks.

10 Tasks 1 and 3 Solutions

Both our main task 2 finetuned solution and our alternative zero-shot plus meta model solution can be submitted to tasks 1 and 3 with a few lines of code change.

The difference between task 1 and 2 is the introduction of web search. To submit our task 2 solutions to task 1, we disable the web search. Instead we only perform image search. The remainder of the pipeline stays the same.

The difference between task 2 and 3 is the introduction of multi-turn questions. When submitting our task 2 solutions to task 3, we need to determine how to use question answer history. The simplest approach is to ignore history completely. Other approaches involve incorporating history in various parts of our solution pipeline. For example we can use history to affect query rewrite, use history to affect web search, use history to affect re-rank etc etc.

When submitting to task 3, we chose to only modify the final answer generation step. We appended a list of previous questions without our LLM responses. The remainder of the pipeline including query rewrite, image search, web search, re-rank stays the same. This approach performed the best among the few variations of using history that we tried.

11 Resources

To ensure reproducibility of our results, we provide access to key components of our pipeline:

- **Training Dataset**: The multi-task finetuning is available on HuggingFace at `rbiswasfc/kddcup-sft-datamix` [10]
- **Finetuned Model**: The finetuned Llama-3.2 Vision model is available on HF at `rbiswasfc/aicrowd-kddcup-v9` [5]
- **Solution Code**: We have uploaded our solution code to GitHub at `rbiswasfc/crag-mm`. [9]

12 Conclusion

The Meta KDD Cup'25 Task 2 challenged teams to build robust multimodal RAG systems capable of factual visual question answering under noisy retrieval conditions. Team NVIDIA's winning solution centered around a single multi-task finetuned LLaMA-3.2-11B-Vision-Instruct model capable of generating web queries, re-ranking retrievals, and answering grounded questions — all while learning to say "I don't know" when appropriate.

Crucially, our decision to train the model using NVIDIA RAGAS Accuracy Metric[7], tuned to human judgment, allowed us to bridge the gap between leaderboard metrics and real-world evaluation. This approach helped us outperform alternatives like prompt-based zero-shot models combined with meta classifiers, and secured the top spot with a final human eval score of 0.233 — narrowly edging out other top teams in a highly competitive field.

Our work highlights the power of training holistic multi-task VLMs guided by reliable proxy metrics, and paves the way for scalable real-world systems that can reason, retrieve, and refuse when necessary. As we look to the future, we're excited by the possibilities of expanding this architecture to handle more complex dialog, more diverse domains, and even higher fidelity reasoning grounded in multimodal evidence.

References

[1] 2025. CRAG-MM Challenge - Improve RAG with Real-World Benchmarks. (2025). https://www.aicrowd.com/challenges/meta-crag-mm-challenge-2025

[2] Tianyu Gao, Shuning Lin, Caiming Xiong, Jingjing Liu, Zhiyuan Liu, and Maosong Sun. 2023. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997* (2023). https://arxiv.org/abs/2312.10997

[3] Meta. 2024. *Meta Llama 3.1 70B.* https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct

[4] Mistralai. 2024. *Mistralai Mixtral 8x22B Instruct v0.1.* https://huggingface.co/mistralai/Mixtral-8x22B-Instruct-v0.1

[5] NVIDIA. 2025. *Finetuned Model.* https://huggingface.co/rbiswasfc/aicrowd-kddcup-v9

[6] NVIDIA. 2025. *NVIDIA Nemotron super 49b v1.* https://build.nvidia.com/nvidia/llama-3_3-nemotron-super-49b-v1/modelcard

[7] NVIDIA. 2025. *NVIDIA RAGAS Metrics.* https://docs.ragas.io/en/latest/concepts/metrics/available_metrics/nvidia_metrics/

[8] NVIDIA. 2025. *NVIDIA RAGAS Metrics source.* https://github.com/explodinggradients/ragas/blob/main/src/ragas/metrics/_nv_metrics.py

[9] NVIDIA. 2025. *Solution Code.* https://github.com/rbiswasfc/crag-mm

[10] NVIDIA. 2025. *Training Dataset.* https://huggingface.co/datasets/rbiswasfc/kddcup-sft-datamix

[11] Chen Qiu, Wenhao Zhang, Jindong Wu, Yufei Tan, Xiao Liu, Chunyuan Wang, and Jing Jiang. 2024. SnapNTell: Enhancing Entity-Centric Visual Question Answering with Retrieval Augmented Multimodal LLM. In *Findings of the Association for Computational Linguistics: EMNLP 2024.* https://aclanthology.org/2024.findings-emnlp.14/

[12] Wenhu Yu, Mingyang Zhou, Ziyang Zhu, Menglin Liu, Yuxiang Wang, Yuwei Zeng, Yujia Yu, Zijian Lin, et al. 2023. MM-Vet: Evaluating Large Multimodal Models for Integrated Capabilities. *arXiv preprint arXiv:2308.02490* (2023). https://arxiv.org/abs/2308.02490