

MarsadLab at ImageCLEF 2026 Multimodal Reasoning: LoRA Fine-Tuned Vision Language Models for Multilingual Textual MCQ and Visual Open Question Answering

Working Notes for ImageCLEF 2026 Lab at CLEF 2026

Md. Rafiul Biswas^{1,*}, Wajdi Zaghoulani²

¹Hamad Bin Khalifa University, Doha, Qatar

²Northwestern University in Qatar, Education City, Doha, Qatar

Abstract

We present the MarsadLab submission to the ImageCLEF 2026 Multimodal Reasoning task, covering the Textual Multiple-Choice Question Answering (Textual MCQ) and Visual Open Question Answering (Visual OpenQA) subtasks across six languages: Bulgarian, Chinese, Croatian, English, Italian, and Serbian. For the Textual MCQ subtask, we fine-tuned Qwen2.5-VL-7B-Instruct using 4-bit QLoRA on a language-balanced subset of text-only EXAMS-V examples and performed inference by directly scoring the next-token logits of the candidate answer letters instead of generating free-form responses. This approach guarantees valid single-letter predictions, reduces inference latency, and achieved an official accuracy of 0.6152, ranking third on the leaderboard. For the Visual OpenQA subtask, we employed Qwen3-VL-8B-Instruct with native-language prompting, constrained answer extraction, answer normalization, and five-sample self-consistency decoding, achieving a COMET score of 0.5143 and a sixth-place ranking. Our results demonstrate that parameter-efficient fine-tuning and constrained decoding provide an effective and computationally efficient solution for multilingual multimodal exam reasoning under single-GPU constraints. Error analysis further suggests that OCR robustness for Cyrillic text, particularly in Bulgarian exam pages, and the limited availability of labeled OpenQA training data remain the primary challenges for future improvement.

Keywords

ImageCLEF 2026, Multimodal Reasoning, Multilingual MCQ, Visual Question Answering, Qwen2.5 VL, Qwen3 VL, LoRA Fine Tuning, Self Consistency

1. Introduction

Recent vision language models have shown rapid progress on standard visual question answering benchmarks, but their behavior on real school exam material, in languages other than English, remains less well understood. Exam questions combine dense printed text, mathematical notation, diagrams, tables and culturally specific knowledge. They also frequently use scripts such as Cyrillic and Chinese that demand robust optical character recognition at the encoder side, and they reward exact short answers rather than fluent paraphrases at the decoder side. These properties make multilingual exam question answering a useful stress test for both the perception and the reasoning components of current vision language systems.

The ImageCLEF 2026 Multimodal Reasoning task [1] is part of the broader ImageCLEF 2026 evaluation campaign [2] at CLEF 2026 and is designed to probe exactly this setting. The 2026 edition is built around a fresh dataset of 2025 academic exam questions and requires participants to develop fully open source systems. The task is organized along two axes. On one axis, the system can either be given an image that contains the full exam question together with a small list of candidate answers (Multiple Choice Question Answering, MCQ) or an image that contains only the question, in which case the system must produce a free form textual answer (Open Question Answering, OpenQA). On the other axis, the

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Corresponding author.

✉ mbiswas@hbku.edu.qa (Md. R. Biswas); wajdi.zaghoulani@northwestern.edu (W. Zaghoulani)

🆔 0000-0002-5145-1990 (Md. R. Biswas); 0000-0003-1521-5568 (W. Zaghoulani)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

question text is either rendered directly in the image (visual track) or provided as text accompanying a separate visual element (textual track). The combination yields four subtasks. The benchmark covers six languages, namely Bulgarian, Chinese, Croatian, English, Italian and Serbian, drawn from regional school examinations, and a wide range of school subjects from biology to mathematics.

We participated under the team name `rafiulbiswas` in two of these four subtasks, the Textual MCQ subtask and the Visual OpenQA subtask. Both submissions are constrained to open weight models, as required by the task rules, and are reproducible inside a single A100 GPU environment. For Textual MCQ, we adapted Qwen2.5 VL 7B Instruct [3] via parameter efficient fine tuning on the EXAMS V dataset [4], which is the same multilingual school exam dataset used by the organizers as the MCQ training resource. For Visual OpenQA, we used Qwen3 VL 8B Instruct [5] served through vLLM [6], with native language prompts and self consistency decoding over five samples.

The remainder of the paper is organized as follows. Section 2 reviews recent work on vision language models for multilingual exam question answering. Section 3 describes the data used in our submissions. Sections 4 and 5 describe our Textual MCQ and Visual OpenQA systems in detail. Section 6 reports the official leaderboard results and per language breakdowns. Section 7 discusses the strengths and weaknesses of each approach. Section ?? states the limitations of our submissions and Section 8 concludes.

2. Related Work

Multilingual evaluation of large vision language models on real school exam questions has emerged as a useful stress test for these systems, because exam questions often combine dense text, mathematical notation, diagrams and culturally specific knowledge. The EXAMS V benchmark [4] introduced a multilingual collection of school exam questions across multiple languages and subjects, where the question, options and any accompanying visual element are rendered in a single image. It has since become a standard testbed for evaluating frontier vision language models in a setting that is closer to real classroom material than synthetic VQA benchmarks. The 2025 edition of ImageCLEF previously organized a Multimodal Reasoning task built on similar resources, and the 2026 edition extends this line of work with a new pool of 2025 exam questions and a stricter open source requirement [1].

2.1. Vision Language Models for Exam-Style Reasoning

On the model side, the Qwen2.5 VL family [3] introduced dynamic resolution image processing and improved support for long, text heavy pages, which is well matched to exam images. Qwen3 VL [5] builds on this design with stronger OCR across many languages, longer context windows and better STEM reasoning, but at the cost of higher memory requirements. Other open vision language models, such as InternVL [7], have shown similar gains on dense document understanding, suggesting that the recent improvements in this family of models are broadly transferable to exam style inputs.

2.2. Parameter-Efficient Fine-Tuning at the 7B–8B Scale

For parameter efficient fine tuning of large vision language models, Low Rank Adaptation (LoRA) [8] combined with 4 bit quantization through QLoRA [9] is now a standard recipe that fits 7B to 8B class models on a single A100 GPU. LLM.int8 [10] and the 8 bit optimizers of Dettmers et al. [11] further reduce the memory footprint during training. These developments make full LoRA fine tuning of a 7B vision language model practical inside a single GPU budget, which we exploit in our Textual MCQ submission.

2.3. Decoding Strategies for Open-Ended Question Answering

For open ended generation, self consistency decoding [12] has been shown to improve chain of thought reasoning by sampling several chains and voting on the final answer, and is particularly useful for

Table 1

Textual MCQ training data preparation. EXAMS V splits filtered to text only items in the six target languages, after per language capping at 1,500 examples for training and 100 for validation. The held out EXAMS V test set was not used during training.

Split	Filtered size	After cap (train/val)	Languages
Train	6,689	5,378	BG, ZH, HR, EN, IT, SR
Validation	1,992	600	BG, ZH, HR, EN, IT, SR
Held out test	1,488	1,488	BG, ZH, HR, EN, IT, SR

smaller models whose individual chains are noisier. For multiple choice answering, an alternative to free form generation is to read the answer letter directly from the logits of the candidate tokens, which is faster, cannot emit an invalid letter, and is robust to surface form variation. We adopt the first idea in our Visual OpenQA submission and the second in our Textual MCQ submission.

3. Data

The competition provides separate datasets for the MCQ and OpenQA tracks. For the Textual MCQ subtask, the organizers recommended the EXAMS V dataset [4] as the official training resource and released a private test set under the identifier `SU-FMI-AI/ImageCLEF-MR2026-MCQ-Textual` with 1,653 unlabeled questions. The test set is distributed across the six target languages, with Bulgarian (451), Croatian (335), Chinese (267), English (240), Serbian (187) and Italian (173) questions.

For training the Textual MCQ system, we used the three EXAMS V splits (train, validation, test). We filtered all three splits to keep only items with `type == "text"` and `language` in the set of six target languages. After filtering, we obtained 6,689 training items, 1,992 validation items and 1,488 EXAMS V test items, which we kept aside as a held out evaluation set with gold labels and did not train on. The training distribution is dominated by Croatian (2,811) and Bulgarian (1,356), with smaller amounts in Italian (917), Serbian (799), Chinese (519) and English (287). Table 1 summarizes the filtered splits.

For the Visual OpenQA subtask, the organizers released a labeled training and development dataset with gold label through the Hugging Face repository `SU-FMI-AI/ImageCLEF-MR2026-OpenQA-Visual` and a separate unlabeled test split. The training pool contains 528 questions and the test split contains 439 questions, with each question delivered as a PNG image that includes the question text and any accompanying visual element such as a diagram, graph, table or other figure. The training pool is heavily skewed toward Biology, which accounts for 62 percent of the items, with Chemistry, Physics and smaller amounts of other subjects making up the rest. The language distribution is more balanced, with Chinese (26 percent), Croatian (18 percent), Italian (17 percent), Bulgarian (16 percent), English (13 percent) and Serbian (11 percent). Figure 1 shows the pipeline for two subtasks at a glance.

4. Textual MCQ System

4.1. Model and Quantization

For the Textual MCQ subtask, we fine tuned Qwen2.5 VL 7B Instruct [3]. The model was loaded in 4 bit NF4 quantization through `bitsandbytes` [10], with double quantization enabled and BF16 compute dtype, following the QLoRA recipe of Dettmers et al. [9]. The vision processor was configured with a minimum patch budget of $256 \times 28 \times 28$ pixels and a maximum of $1280 \times 28 \times 28$ pixels, which leaves enough resolution for text heavy exam pages without overshooting the GPU memory budget.

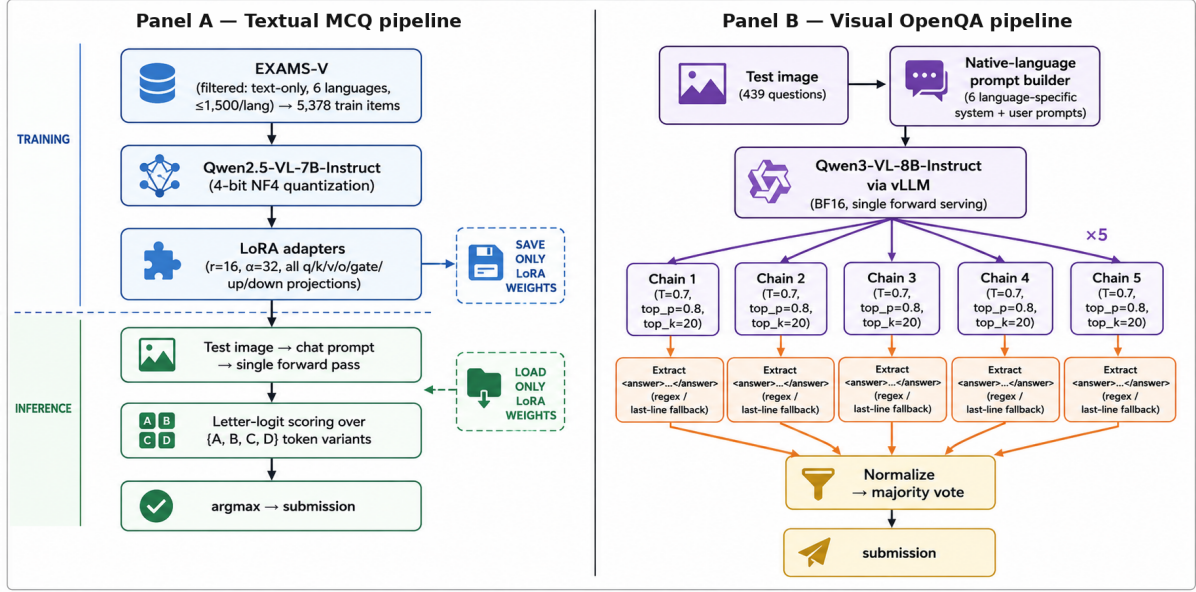


Figure 1: Pipelines for the two subtasks. The MCQ-Textual system (Panel A) uses 4-bit LoRA finetuning and single-pass letter-logit scoring at inference. The Visual OpenQA system (Panel B) uses a zero-shot Qwen3-VL-8B backbone with self-consistency over five sampled chains.

4.2. LoRA Adapters and Trainable Parameters

We attached LoRA adapters [8] of rank 16 with $\alpha = 32$ and dropout 0.05 to all linear projections of the language backbone, namely the four attention projections and the three MLP projections (q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj). After wrapping, the trainable parameter count was 47.6 million out of 8.34 billion, approximately 0.57 percent of the full model. Gradient checkpointing was enabled and the KV cache was disabled at training time.

4.3. Training Data Preparation

To prevent the long tail of the training language distribution from being drowned out by Croatian and Bulgarian, we capped the per language training count at 1,500 examples by sampling uniformly within each language. This produced a final training set of 5,378 items, with Croatian at 1,500, Bulgarian at 1,356, Italian at 917, Serbian at 799, Chinese at 519 and English at 287. A validation subset of 600 examples was constructed in a similar way, with a cap of 100 examples per language.

Each example was converted into a multi turn chat with a system, user and assistant turn. The system message describes the model as an expert multilingual tutor and tells it that it should choose a single best answer. The user message contains the question image and the instruction to respond with only a single capital letter A, B, C or D. The assistant message contains the gold answer letter in upper case. The collator masks the loss on everything except the final assistant token, so the model is trained only to produce the answer letter conditioned on the rest of the chat.

4.4. Optimization

We trained for two epochs with batch size 1 and gradient accumulation 8, giving an effective batch size of 8. The optimizer was paged AdamW 8 bit [11] with learning rate 1×10^{-4} , cosine schedule, warmup ratio 0.03 and weight decay 0.01. The maximum sequence length was 2,048 tokens. Training was carried out in BF16 with TF32 matmul on a single NVIDIA A100 SXM4 80 GB GPU. The training loss dropped from 0.74 at step 25 to a stable band around 0.18 to 0.20 by step 200 and remained there for the rest of training.

4.5. Inference by Letter-Logit Scoring

At inference time, we did not call `generate` on the model. Instead, we built the same chat prompt as during training but with an empty assistant turn, ran a single forward pass, and read off the logits at the last position. For each candidate letter A, B, C and D, we collected the token identifiers that the tokenizer maps to that letter under different surface forms (bare letter, leading space, leading newline), took the maximum logit across those variants, and predicted the letter with the highest score. This approach is strictly faster than sampling, never emits an invalid letter, and was easy to integrate with our held out evaluation pipeline.

4.6. Held-Out Evaluation

To estimate the quality of the fine tuned model before submitting, we evaluated the letter logit predictor on the 1,488 EXAMS V test items that we had kept aside. The overall accuracy was 0.5370 (799 of 1,488). The per language breakdown showed Italian at 0.5793, Chinese at 0.5725, Serbian at 0.5655, Croatian at 0.5517, English at 0.5313 and Bulgarian at 0.3582. Bulgarian was clearly the weakest language in this internal evaluation, and we return to possible reasons in Section 7.

5. Visual OpenQA System

5.1. Model and Serving Stack

For the Visual OpenQA subtask, we used Qwen3 VL 8B Instruct [5], served through vLLM 0.11 [6]. We chose this model for three reasons. First, its native OCR coverage extends to all six dataset languages, which matters because the question text is rendered in the image. Second, its STEM reasoning capacity has been improved over Qwen2.5 VL, which is useful given the math heavy subject mix in the development split. Third, the nine billion parameter footprint in BF16 fits on a single A100 80 GB with comfortable headroom for the KV cache. The model was loaded with tensor parallel size two when two GPUs were available, otherwise with tensor parallel size one. The processor was configured with `min_pixels` of $256 \times 32 \times 32$ and `max_pixels` of $1280 \times 32 \times 32$, which corresponds to between 256 and 1,280 visual tokens per image and balances speed against fidelity on dense exam pages.

5.2. Multilingual Prompting

Because the question text is rendered in the image rather than in the textual context, we wrote system prompts and user instructions in the native language of each item. We covered all six dataset languages with the same two part structure. The system prompt frames the model as an expert tutor, asks it to read the question carefully including any diagram, table or graph, reason step by step, and give the shortest exact answer in the question language. It also requires the final answer to be enclosed in `<answer> . . . </answer>` tags. The user instruction reiterates the formatting requirement and stresses that no explanation should appear inside the tag. We added an optional one line subject hint that nudges the model toward a canonical form of the answer, for example a chemical formula for chemistry, a numeric value with unit for physics, or a short noun phrase for geography.

5.3. Answer Extraction and Normalization

Once a generation was produced, we extracted the final answer with a small cascade. We first looked for content inside `<answer> . . . </answer>` tags. If that failed, we fell back to a multilingual regular expression that matches common final answer prefixes such as *final answer* and *answer*, together with the equivalent forms in the other dataset languages. If both failed, we used the last non empty line of the generation. For evaluation, we normalized both predictions and gold answers by folding LaTeX into plain text equivalents (for example $\frac{a}{b}$ into a/b), removing whitespace, lower casing,

Table 2

Official Textual MCQ leaderboard scores for our submission. Per language accuracy and overall accuracy as reported on the official leaderboard.

Overall	EN	BG	ZH	HR	IT	SR
0.6152	0.7125	0.4346	0.6629	0.6836	0.7341	0.6257

stripping option markers and trimming trailing punctuation. The same normalization was used both for our local development metrics and for self consistency voting.

5.4. Decoding Strategies Compared on Dev

We tested three decoding strategies on the development split, following the sampling parameters recommended by the Qwen3 VL model card.

The first strategy was a zero shot chain of thought baseline with $n = 1$ sample, temperature 0.7, top p 0.8, top k 20, presence penalty 1.5 and a maximum of 1,024 new tokens. The second strategy was self consistency [12], where we sampled $n = 5$ chains with the same parameters as the baseline, extracted a final answer from each chain, and used majority voting over the normalized final answers. When multiple answers tied, we picked the first one in generation order. The third strategy was a subject specialized prompt, which extends the baseline by appending a single line subject hint, for example *chemical formula or compound name* for chemistry items, to the user instruction.

The development split released for this subtask had gold answers replaced by the string `HIDDEN`, so we could not compute meaningful exact match or token F1 numbers locally. We therefore relied on inspection of generations, runtime per sample and the official leaderboard COMET score to choose between the three strategies. For the final test submission, we used self consistency over five samples, since informal inspection of dev generations suggested that voting reduced the rate of degenerate outputs in Bulgarian and Serbian compared to the single sample baseline.

5.5. Submission Generation

For the test split of 439 questions, we ran self consistency inference on the loaded vLLM engine. The full test run took approximately twelve minutes. We then validated that the resulting JSON file contained exactly 439 entries with unique question identifiers and that each entry carried a non empty answer string in the expected schema, before submitting to the leaderboard.

5.6. Experimental Setup

All inference for the Visual OpenQA submission was conducted on an NVIDIA A100 80 GB GPU using Python 3.12, PyTorch 2.5.1, vLLM 0.11.2 and transformers 4.57. The Textual MCQ training and inference were run on the same hardware, with bitsandbytes 0.49 for 4 bit quantization, PEFT 0.19 for LoRA and TRL 1.2 for the supervised fine tuning loop. The full Textual MCQ training run took approximately 3 to 4 hours, and full Textual MCQ test inference required about 7 minutes on the 1,653 question test set.

6. Results

6.1. Textual MCQ Results

Our Textual MCQ submission was ranked third on the official leaderboard with an overall accuracy of 0.6152. The per language breakdown is shown in Table 2. The system was strongest on Italian (0.7341) and Croatian (0.6836), and weakest on Bulgarian (0.4346). The held out accuracy of 0.5370 that we obtained on the EXAMS V test split (Section 4) is therefore a slight underestimate of the leaderboard accuracy, although the relative ordering of the languages was not identical, which is consistent with

Table 3

Official Visual OpenQA leaderboard scores for our submission. COMET per language and corpus level BLEU, ROUGE L and METEOR as reported on the official leaderboard.

COMET	BG	ZH	HR	EN	IT	SR	BLEU	ROUGE-L	METEOR
0.5143	0.4798	0.4848	0.6167	0.4851	0.5031	0.5167	0.0735	0.2013	0.1331

the leaderboard test set being drawn from a fresh distribution of 2025 exam questions rather than from EXAMS V.

6.2. Visual OpenQA Results

Our Visual OpenQA submission was ranked sixth on the official leaderboard with an overall COMET score of 0.5143. The full set of metrics is shown in Table 3. The system scored highest in Croatian (COMET 0.6167) and Serbian (0.5167), and lowest in Bulgarian (0.4798), with the remaining three languages clustered close together. The BLEU and ROUGE L scores, at 0.0735 and 0.2013 respectively, are low in absolute terms, which is consistent with the open ended nature of the task and the strict requirement that the system produce a short exact answer.

7. Discussion

7.1. Why Letter-Logit Scoring Helped on the MCQ Track

The MCQ submission directly reads the answer letter from the next token logits rather than generating it freely. This has three practical advantages. First, it cannot emit an invalid letter, which removes a class of trivial errors that pure generation can produce when the model hallucinates extra commentary before the letter. Second, it is several times faster than generation, since only a single forward pass is required per question. Third, it is robust to surface form variation, because we aggregate across the bare letter, the leading space and the leading newline token identifiers. These properties matter at the 1,653 question test scale, where small per item costs add up. We attribute part of our placement to this design choice rather than to the LoRA training alone, although we did not run an ablation that disentangles the two contributions before submission.

7.2. The Bulgarian Gap

Our Bulgarian performance was clearly the weakest both on the held out EXAMS V evaluation (0.3582) and on the leaderboard (0.4346). One likely contributing factor is OCR quality on Cyrillic exam pages, especially when the text is printed in a serif typeface at moderate resolution. Although Qwen2.5 VL supports Cyrillic, the visual token budget for these images may not be enough to resolve the per character detail required for short technical terms. Increasing the maximum pixel budget for the vision processor would be a natural next step. Another factor is the imbalance in the EXAMS V training distribution, where Bulgarian is the second largest language but still well below Croatian, and where Bulgarian exam questions cover a broader subject mix than the other languages.

7.3. Self-Consistency and Small Model Noise

For the Visual OpenQA submission, we used Qwen3 VL 8B rather than the 72 billion parameter variant, in order to fit comfortably on a single A100 and to keep the test run under fifteen minutes with self consistency. The trade off is that each individual chain of thought is noisier than at the 72B scale, which is exactly the regime in which self consistency tends to help most [12]. We observed qualitatively that self consistency reduced the rate of degenerate outputs in which the model emits only a fragment of reasoning before being cut off, especially in Bulgarian and Serbian. We did not, however, run a

Table 4

Common error categories observed for the Textual MCQ system.

Error Type	Typical Symptom	Potential Improvement
OCR failure	Option or question text is mis-read	Increase image resolution, improve OCR preprocessing, or allocate more visual tokens.
Reasoning failure	Mathematical formulas or scientific concepts are misunderstood	Apply subject-specific fine-tuning and instruction tuning on exam-style reasoning tasks.
Language issue	Confusion over Cyrillic or Chinese terminology	Improve language-balanced training and multilingual data augmentation.
Option-letter bias	Prediction favors frequently occurring answer letters	Apply confidence calibration or answer prior correction.
Visual layout issue	Tables, diagrams, or answer options are parsed incorrectly	Use image tiling, adaptive cropping, or higher-resolution visual encoding.

quantitative comparison between the three decoding strategies, because the development split was distributed with masked gold answers. A more rigorous ablation would require local labels, which is a useful direction for future participation.

7.4. Subject and Language Coverage

The Visual OpenQA training pool is dominated by Biology, while the development split is dominated by Physics and Mathematics. Our system does not exploit this distribution shift and applies the same multilingual prompt to all subjects, modulo a one line subject hint. A stronger system would either select a few shot prompt set per subject from the training pool or fine tune Qwen3 VL on the available 528 training examples. We did not pursue this direction in 2026 because the small size of the training pool relative to the parameter count makes naive LoRA fine tuning risky, and because the gain from prompt engineering at the 8B scale is competitive with light fine tuning in many of our earlier experiments.

7.5. Error Analysis

To better understand the limitations of our systems, we manually inspected representative failure cases from both the Textual MCQ and Visual OpenQA submissions. Although we did not perform a large-scale annotation study, several recurring error patterns were consistently observed. Tables 4 and 5 summarize the major categories, their typical symptoms, and potential directions for improvement.

For the Textual MCQ system (see Table 4, the most common failures were related to OCR quality on dense multilingual exam pages, particularly for Cyrillic text, subject-specific reasoning errors involving mathematical formulas and scientific concepts, and occasional failures to correctly interpret complex page layouts containing tables or densely packed answer options. We also observed that language-specific vocabulary and subtle answer-letter biases occasionally affected prediction quality.

For the Visual OpenQA system (see Table 5, the primary sources of error arose from the open-ended nature of answer generation. The model sometimes produced unnecessarily verbose responses despite constrained prompting, generated correct numerical values with incorrect units, or produced semantically correct answers whose wording differed from the reference. Additional failures were caused by OCR errors on complex exam pages and occasional hallucinated reasoning, where the generated explanation appeared plausible but led to an incorrect final answer.

7.6. Future Work

These observations suggest that future improvements should focus on enhancing multilingual OCR robustness, incorporating stronger subject-specific adaptation, improving answer normalization and semantic matching, and introducing verification or reranking mechanisms for open-ended generation.

Table 5

Common error categories observed for the Visual OpenQA system.

Error Type	Typical Symptom	Potential Improvement
Overly verbose answer	Explanation is included instead of only the final answer	Use stronger constrained decoding and answer extraction rules.
Wrong unit	Numeric value is correct but the unit is incorrect or omitted	Apply unit normalization and answer canonicalization.
Synonym mismatch	Prediction is semantically correct but differs from the reference wording	Use semantic canonicalization or embedding-based answer matching.
OCR failure	Question text or diagram is incorrectly interpreted	Increase image resolution, adaptive cropping, or OCR-aware preprocessing.
Hallucinated reasoning	Reasoning appears plausible but produces an incorrect final answer	Introduce answer verification or reranking using a secondary model.

Future work will focus on three directions. First, raising the maximum pixel budget of the vision processor for Cyrillic items, which we expect to recover part of the Bulgarian gap. Second, adding a light LoRA fine tuning pass on the 528 Visual OpenQA training items, with careful regularization to avoid overfitting. Third, exploring multi sample re ranking based on chain agreement, which would generalize self consistency to OpenQA settings where strict majority voting is too brittle.

8. Conclusion

We presented two submissions of the MarsadLab team to the ImageCLEF 2026 Multimodal Reasoning task. For Textual MCQ, we showed that LoRA fine tuning of Qwen2.5 VL 7B Instruct on a language balanced subset of EXAMS V, combined with letter logit scoring at inference time, reaches an overall accuracy of 0.6152 across six languages and a third place finish on the official leaderboard. For Visual OpenQA, we showed that Qwen3 VL 8B Instruct with native language prompts and self consistency decoding obtains an overall COMET score of 0.5143 and a sixth place finish. Both systems rely entirely on open weight models and run on a single A100 GPU, which we believe is a practical sweet spot for academic groups working on multilingual multimodal reasoning. The most pressing weakness of both systems is performance on Bulgarian, which we attribute to a combination of Cyrillic OCR limits at the chosen image resolution and training distribution imbalance. We plan to address this in future participations through higher resolution vision inputs and language aware data augmentation.

Acknowledgment

This work was partially supported by the Qatar Research, Development and Innovation Council (QRDI) under grant NPRP14C-0916-210015. We also acknowledge the use of Claude (Anthropic) and GPT (OpenAI) as AI writing assistants for drafting, revising, and improving the manuscript content.

Reproducibility

To support reproducibility, we release the full implementation of both submissions, including data preparation, LoRA fine-tuning, letter-logit inference for the Textual MCQ subtask, and multilingual prompting, self-consistency decoding, and answer normalization for the Visual OpenQA subtask, at: <https://github.com/rafiulbiswas/MultimodalReasoning>

Declaration on Generative AI

The authors used ANTHROPIC CLAUDE OPUS 4.8 and OPENAI CHATGPT (GPT-5.5 AND GPT-5.4) for LaTeX formatting, grammar checking, paraphrasing, rewording, and language polishing. All AI-assisted content was critically reviewed and verified by the authors, who take full responsibility for the final manuscript.

References

- [1] D. Dimitrov, M. S. Hee, M. Ahsan, S. Ahmad, D. Zlatkova, G. Pachov, Z. Xie, P. Nakov, I. Koychev, Overview of the ImageCLEF 2026 Task on Multimodal Reasoning, in: CLEF 2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Jena, Germany, 2026.
- [2] B. Ionescu, H. Müller, D. Stanciu, A. Radu, R. Bolborici, M. Negru, A. Ene, V. Vasilescu, A.-A. Nicolae, L. Ștefan, M. Constantin, M. Dogariu, A. Andrei, H. Damm, T. M. G. Pakull, A. Ben Abacha, A. García Seco de Herrera, C. M. Friedrich, R. Brüngel, L. Reinartz, H. Schäfer, C. S. Schmidt, B. Bracke, P. Nath, B. Eryilmaz, M. Hjuler, D. Fabre, C. Lemaire, B. Lecouteux, D. Schwab, D. Dimitrov, M. S. Hee, M. Ahsan, S. Ahmad, D. Zlatkova, G. Pachov, Z. Xie, P. Nakov, I. Koychev, J. E. Heras Rivera, D. K. Low, W. Yim, J. Ruzevick, D. Child, M. Kurt, Z. Sun, F. Xia, M. Yetisgen, A. Radzhabov, Y. Prokopchuk, V. Kovalev, D. Karpenka, S. A. Hicks, S. Gautam, M. A. Riegler, V. Thambawita, P. Halvorsen, M. El Sakka, J. Mothe, A. Băicoianu, C. Florea, M. Ivanovici, Overview of imageclef 2026: Multimodal challenges in medicine, science, agritech, and security, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Springer Lecture Notes in Computer Science LNCS, Jena, Germany, 2026.
- [3] Q. Team, Qwen2.5-vl, 2025. URL: <https://qwenlm.github.io/blog/qwen2.5-vl/>.
- [4] R. Das, S. Hristov, H. Li, D. Dimitrov, I. Koychev, P. Nakov, Exams-v: A multi-discipline multilingual multimodal exam benchmark for evaluating vision language models, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 7768–7791.
- [5] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, et al., Qwen3-vl technical report, arXiv preprint arXiv:2511.21631 (2025).
- [6] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, I. Stoica, Efficient memory management for large language model serving with pagedattention, in: Proceedings of the 29th symposium on operating systems principles, 2023, pp. 611–626.
- [7] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, et al., Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2024, pp. 24185–24198.
- [8] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al., Lora: Low-rank adaptation of large language models., Iclr 1 (2022) 3.
- [9] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms, Advances in neural information processing systems 36 (2023) 10088–10115.
- [10] T. Dettmers, M. Lewis, Y. Belkada, L. Zettlemoyer, Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale, Advances in neural information processing systems 35 (2022) 30318–30332.
- [11] T. Dettmers, M. Lewis, S. Shleifer, L. Zettlemoyer, 8-bit optimizers via block-wise quantization, arXiv preprint arXiv:2110.02861 (2021).
- [12] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, arXiv preprint arXiv:2203.11171 (2022).