SRH-ReliableAI at ImageCLEF 2026 Task on Multimodal Reasoning: Fine-Tuning Thinking Vision-Language Models for Multilingual Exam Question Answering

ImageCLEF at CLEF 2026

Pratik Priyanshu[1], Swati Chandna[1]

[1]SRH University Heidelberg, Heidelberg, Germany

Abstract

This paper describes our submission to the ImageCLEF 2026 Multimodal Reasoning task as team SRH-ReliableAI. The task requires answering multilingual exam questions in four tracks: Visual MCQ, Textual MCQ, Visual OpenQA, and Textual OpenQA. We fine-tune Qwen3-VL-8B-Thinking, an 8-billion parameter vision-language model with built-in chain-of-thought reasoning, using QLoRA on 16,300 EXAMS-V training samples. We also add subject-aware and language-matched prompting at no extra computational cost, and design a multi-stage answer extraction pipeline for handling the model's thinking tokens. Our system ranks **1st on Textual MCQ** (accuracy 0.7538) and **2nd on Textual OpenQA** (COMET 0.5285), while visual reasoning tracks prove more challenging (Visual MCQ: 0.5076, Visual OpenQA: 0.4286 COMET). A notable finding is the large gap between textual and visual performance: the model reasons well over text but struggles with diagrams and spatial content. We report ablation studies on prompt strategies, answer extraction, and per-language breakdowns.

Keywords

vision-language models, multimodal reasoning, QLoRA fine-tuning, chain-of-thought, multilingual VQA, ImageCLEF

1. Introduction

The ImageCLEF 2026 Multimodal Reasoning task [1], part of the broader ImageCLEF 2026 evaluation campaign [2], evaluates systems on their ability to answer exam-style questions that require understanding both visual content and natural language across multiple languages. The task features four tracks: Visual MCQ and Textual MCQ (multiple-choice, evaluated by accuracy) and Visual OpenQA and Textual OpenQA (open-ended, evaluated by COMET [3], BLEU, ROUGE-L, and METEOR). Questions span six languages (English, Bulgarian, Chinese, Croatian, Italian, Serbian) and cover diverse academic subjects, making the task both multimodal and multilingual.

A recent class of VLMs including Qwen3-VL [4] and DeepSeek-R1 [5] can now perform chain-of-thought (CoT) reasoning internally, emitting structured reasoning tokens (`<think>...</think>`) before producing a final answer. This makes them a natural fit for exam-style questions that require multi-step reasoning.

We apply one such model, Qwen3-VL-8B-Thinking (8B parameters, Normal $\geq$8B category), fine-tuned with QLoRA [6] on EXAMS-V [7]. Our leaderboard username is `pratikpriyanshu`. The main contributions of this work are:

1. We apply a thinking-mode VLM to multilingual exam question answering, ranking 1st on Textual MCQ.
2. We show that QLoRA fine-tuning on EXAMS-V clearly outperforms zero-shot inference.
3. We introduce subject-aware and language-matched prompting as zero-cost accuracy improvements.
4. We document a large textual vs. visual reasoning gap and investigate what drives it.

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

pratikpriyanshu12345@gmail.com (P. Priyanshu)

0009-0006-6192-3767 (P. Priyanshu); 0009-0003-8393-5337 (S. Chandna)

2. Related Work

Our system draws on three lines of prior work: multimodal reasoning with vision-language models, chain-of-thought reasoning in large language models, and parameter-efficient fine-tuning. We cover each below and note how our design choice relates to it.

Multimodal Reasoning with VLMs. Large closed VLMs such as GPT-4V and Gemini, together with the open-weight Qwen-VL family [8], have established strong baselines on multimodal benchmarks such as MMMU [9]. EXAMS-V [7] narrows this evaluation to multilingual exam questions with visual content, combining language understanding with diagram interpretation. Two distinct system-design patterns have emerged on exam-style multimodal tasks: (i) *OCR-then-VLM pipelines*, which first extract text from question images with a dedicated OCR component and then feed both the OCR output and the raw image into a VLM, and (ii) *end-to-end VLM approaches*, which rely entirely on the model's native visual encoder without an explicit OCR step. The 2025 ImageCLEF Multimodal Reasoning winner used an OCR-VLM ensemble pipeline reaching 81.4% accuracy [10]. Our submission takes the second route: no OCR module and no ensembling, relying on a single VLM to jointly interpret text and image content. The trade-off is that our system is simpler and cheaper to serve but loses the OCR-derived signal on text-in-image content, which we discuss in Section 6.

Chain-of-Thought and Thinking-Mode Models. Wei et al. [11] showed that step-by-step prompting substantially improves LLM performance on complex reasoning tasks, and Wang et al. [12] extended this with self-consistency by sampling multiple reasoning paths and majority-voting. Both techniques rely on explicit prompt-level intervention. A more recent design pattern moves reasoning inside the model itself: DeepSeek-R1 [5] and Qwen3 [4] emit explicit reasoning tokens wrapped in a dedicated `<think>...</think>` span before producing the final answer, making chain-of-thought a native model behaviour rather than a prompt-engineering step. Our system uses Qwen3-VL-8B-Thinking, a member of the second family. Compared to prompt-based CoT approaches used in earlier ImageCLEF submissions, this design shifts the engineering burden from prompt design to *output parsing*, since the emitted reasoning span must be excluded from the answer. Section 3.4 details the extraction pipeline this requires; Section 6.3 quantifies how much accuracy is at stake in that step.

Parameter-Efficient Fine-Tuning. LoRA [13] adapts large models by training small low-rank decomposition matrices, and QLoRA [6] combines LoRA with 4-bit weight quantization, making single-GPU fine-tuning of billion-parameter models tractable. For multimodal exam QA, prior systems have used either full fine-tuning on smaller VLMs or pure zero-shot inference on larger models. QLoRA occupies the middle ground, enabling us to adapt an 8B-parameter thinking VLM on task-relevant data with a single H200-class GPU and a training budget under 10 hours.

Positioning of Our System. Compared to the 2025 ImageCLEF Multimodal Reasoning winner [10], we drop the OCR component and the ensemble in favour of a single native-thinking VLM. Compared to prompt-based CoT approaches, we replace prompt-level intervention with native thinking-token generation and a dedicated extraction pipeline. Compared to zero-shot VLM baselines, we add lightweight task adaptation through QLoRA and two zero-cost prompt enhancements (subject-aware prefixes, language-matched answer instructions). The results in Section 5 show that this configuration is competitive on the textual tracks (1st on Textual MCQ, 2nd on Textual OpenQA) but does not close the gap on the visual tracks, where OCR-augmented pipelines and larger visual encoders still have the edge.

3. System Description

Figure 1 provides an overview of our system pipeline. We describe each component below.

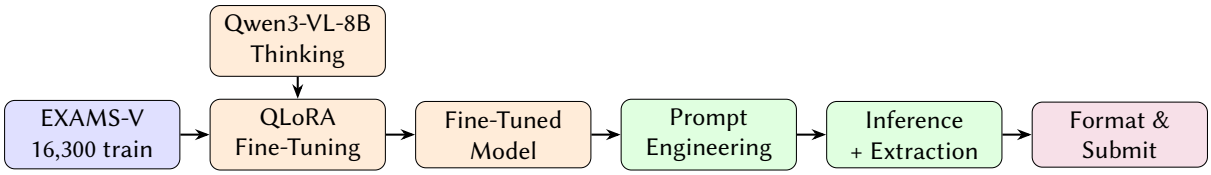

Figure 1: System pipeline overview. The base Qwen3-VL-8B-Thinking model is fine-tuned via QLoRA on EXAMS-V, then applied to competition test sets with subject-aware and language-matched prompting.

Table 1
QLoRA fine-tuning configuration.

Setting	Value
Quantization	NF4 (4-bit)
Compute dtype	bfloat16
Double quantization	Yes
LoRA rank (r)	32
LoRA alpha (α)	64
LoRA dropout	0.05
Target modules	All linear layers
Training data	16,300 EXAMS-V train
Epochs	3
Batch size	1 (gradient accumulation: 8)
Learning rate	2×10^{-4}
LR scheduler	Cosine with warmup (ratio 0.1)
Max sequence length	2048 tokens
Gradient checkpointing	Yes
Training hardware	1× NVIDIA H200 NVL (150 GB)
Training time	∼9 hours

3.1. Model Selection

We select **Qwen3-VL-8B-Thinking** [4] as our primary model for the Normal (≥8B) category. This model extends the Qwen3-VL architecture with a dedicated thinking mode that produces explicit reasoning tokens wrapped in `<think>...</think>` tags before generating the final answer. Key factors in our selection:

- **Reasoning benchmarks**: MMMU 74.1%, MathVista 81.4%, indicating strong multimodal reasoning.
- **Native CoT**: Internal thinking tokens eliminate the need for explicit CoT prompt engineering while maintaining reasoning depth.
- **Memory efficiency**: At approximately 18 GB VRAM for inference, the model comfortably fits on the competition's A40 GPU (40 GB) with room for KV cache.
- **Multilingual support**: Pre-trained on multilingual data, supporting all six task languages.

The model size is **8 billion parameters**, placing it in the Normal category as defined by the task organizers.

3.2. QLoRA Fine-Tuning

We fine-tune the model on the EXAMS-V [7] training set (16,300 MCQ samples with answer labels) using QLoRA [6]. Table 1 summarizes the configuration.

The training data is formatted as single-turn conversations: the user message contains the exam image and a reasoning prompt, and the assistant message contains only the correct answer letter (A–E). We apply the model's chat template to ensure consistency between fine-tuning and inference.

Importantly, the fine-tuning data is exclusively MCQ (multiple-choice); we do *not* fine-tune on OpenQA data, a limitation we discuss in Section 6.

3.3. Prompt Engineering

We evaluate three prompt strategies on a 500-sample stratified validation subset (balanced across languages and subjects):

Direct. A minimal instruction: "*Look at this exam question image. Select the correct answer. Reply with ONLY the letter (A, B, C, D, or E).*"

Thinking. A structured four-step instruction that encourages step-by-step reasoning: "*(1) Read the question and ALL answer options. (2) If there are diagrams, graphs, or visual elements, analyze them. (3) Think step by step about the correct answer. (4) Select exactly ONE answer.*"

Decomposed. An explicit three-phase instruction: "*First, describe what you see. Then, reason step by step. Finally, state your answer as a single letter.*"

In addition to these base prompts, we apply two zero-cost enhancements to all prompts:

Subject-Aware Prefix. The EXAMS-V metadata includes a subject field with 32 categories. We prepend "*You are an expert in {subject}.*" to every prompt, providing domain context at no computational cost.

Language-Matched Answer Instructions. Since Qwen3-VL is multilingual, we translate the answer instruction line into the question's language for six supported languages (English, Bulgarian, Chinese, Croatian, Italian, Serbian). For example, Chinese questions receive the answer instruction in Chinese rather than English.

3.4. Answer Extraction

Extracting the final answer from a thinking model's output is a non-trivial task. The response can contain hundreds of tokens of intermediate reasoning before the actual answer letter, and the reasoning span itself frequently contains *incidental* references to A–E options ("the reasoning suggests option A, but option D also fits..."), which naive parsers may mistake for the final answer.

Output structure. A canonical response has the form

```
<think>
The question asks about the acceleration of a
falling object. Option A suggests it stays
constant, which is wrong under gravity. Option B
gives $g = 9.8\,\mathrm{m/s^2}$, which matches
the free-fall constant. The answer is B.
</think>
B
```

but variants are common: the closing `</think>` may be missing under a small `max_new_tokens` budget, the post-think text may be empty (with the answer only appearing inside the reasoning span), or the answer may be phrased in the question's language (e.g., a Chinese "the answer is B" construction, or the Italian "la risposta è B").

Table 2
Split sizes. Train/val splits include answer labels; test splits do not.

Split	Use	Samples
EXAMS-V train	QLoRA fine-tuning	16,300
EXAMS-V val	Prompt ablation	4,650
stratified subset	Prompt engineering	500
MCQ-Visual test	Competition	1,117
MCQ-Textual test	Competition	1,653
OpenQA-Visual test	Competition	439
OpenQA-Textual test	Competition	424

Multi-stage pipeline. We apply four extraction stages in order, returning the first that succeeds:

1. **Post-think extraction.** If `</think>` appears in the response, we operate on the text after it. The first character that matches `[A-E]` (case-insensitive) is returned as the answer. Regex: `\</think\>\s*([A-E])`.
2. **Reasoning-tail extraction.** If stage 1 fails but a reasoning span exists, we search the last three lines of the reasoning span for explicit answer markers, using a set of language-specific patterns. Examples: English `answer\s+is\s+([A-E])` and `option\s+([A-E])`; equivalent forms are provided for Italian, Chinese, Bulgarian, Croatian, and Serbian.
3. **Generic extraction.** On the cleaned full response we apply a decreasing-specificity cascade of regex patterns: bracketed forms (`\[([A-E])\]`), boxed forms (`\boxed{([A-E])}` produced by LaTeX-style outputs), and single-letter-on-its-own-line matches.
4. **Last-letter fallback.** If all previous stages fail, we scan the response in reverse for the last occurrence of an A–E character, on the assumption that a stray letter near the end of the response is more likely to be the answer than one buried inside the reasoning. If no letter is found, the extractor defaults to `A`.

Token budget interaction. The `max_new_tokens` budget interacts strongly with this pipeline. At 50 tokens the model is truncated before emitting the closing `</think>` tag, so stage 1 fails on almost every instance and the pipeline degrades toward the fallback. At 4096 tokens the reasoning span completes on virtually all instances but inference latency roughly triples. We use 1024 tokens for MCQ, which provides sufficient budget for the thinking phase to complete on the majority of instances while keeping single-question latency below 30 seconds on our hardware. The full prompt strings and the exact extraction-regex list are provided in Appendix A.

3.5. OpenQA Inference

For the OpenQA tracks, we modify the pipeline to generate free-form answers rather than selecting from options. The prompt instructs the model to "*provide a concise, accurate answer*" and we use `max_new_tokens=512` to allow sufficient generation length. We decode with `skip_special_tokens=True` to remove thinking tokens from the output.

We note that the model was *not* fine-tuned on OpenQA data only on MCQ data from EXAMS-V. The OpenQA competition datasets provide separate training splits (528 Visual, ∼300 Textual) that we did not use for fine-tuning, which impacts OpenQA performance.

4. Experimental Setup

4.1. Datasets

We use EXAMS-V [7] for training and validation and the four ImageCLEF 2026 competition test sets for evaluation. Table 2 lists the split sizes.

Table 3
Official competition results across all four tracks. Ranks are drawn from the final task leaderboard; exact participant counts per track will be reported in the task overview paper [1].

Track	Metric	Our Score	Top Score	Our Rank
Visual MCQ	Accuracy	0.5076	0.8406	mid-table
Textual MCQ	Accuracy	**0.7538**	**0.7538**	**1st**
Visual OpenQA	COMET	0.4286	0.6488	mid-table
Textual OpenQA	COMET	**0.5285**	0.6563	**2nd**

Table 4
Visual MCQ accuracy by language.

EN	BG	ZH	HR	IT	SR	**Overall**
0.572	0.436	0.418	0.561	0.556	0.519	0.508

Table 5
Textual MCQ accuracy by language (best overall score in this track).

EN	BG	ZH	HR	IT	SR	**Overall**
0.838	0.676	0.648	0.803	0.850	0.807	0.754

4.2. Implementation Details

We use HuggingFace Transformers with PEFT [14] for QLoRA and TRL [15] for supervised fine-tuning. Quantization is handled by BitsAndBytes [16]. All training and inference runs on a single NVIDIA H200 NVL GPU (150 GB VRAM). Total GPU time is approximately 30 hours across all experiments.

For MCQ inference, we use greedy decoding (`temperature=0`, `do_sample=False`) with `max_new_tokens=1024`. For OpenQA, we use greedy decoding with `max_new_tokens=512`. Inference is checkpoint-based: predictions are saved every 50 samples to allow resumption.

5. Results

5.1. Official Competition Results

Table 3 presents our official results across all four tracks.

Our score matches the top score on Textual MCQ and is competitive on Textual OpenQA, while visual tracks remain challenging. The gap between our textual and visual performance is large, and we investigate it in Section 6.

5.2. Per-Language Breakdown

Tables 4 and 5 show the per-language accuracy for both MCQ tracks.

On Textual MCQ, Italian (0.850) and English (0.838) are our strongest languages, while Chinese (0.648) and Bulgarian (0.676) lag behind. This pattern holds consistently across both MCQ tracks. The Qwen3 technical report [4] does not publish a per-language breakdown of its pre-training mixture, so we cannot directly attribute the observed pattern to training-data composition. We suspect the ordering is shaped by both pre-training language coverage and the amount of downstream instruction-tuning data available per language, but confirming this properly is beyond what we can do here.

5.3. OpenQA Results

Table 6 presents the OpenQA results with additional metrics.

Table 6
OpenQA results across metrics.

Track	COMET	BLEU	ROUGE-L	METEOR
Visual OpenQA	0.429	0.038	0.099	0.071
Textual OpenQA	0.529	0.114	0.230	0.159

The same textual–visual pattern shows up here. The very low BLEU on Visual OpenQA (0.038) suggests our answers often use different words than the reference, though the COMET scores indicate the meaning is at least partially correct.

6. Analysis and Discussion

6.1. The Textual–Visual Performance Gap

The most notable pattern in our results is the gap between textual and visual tracks. On MCQ, we score 0.754 on textual questions but only 0.508 on visual ones – a difference of nearly 25 percentage points that holds across all languages.

We see three likely reasons for this:

(1) Thinking mode is inherently linguistic. The model reasons in natural language tokens. For textual questions, this works well the reasoning engages directly with the question content. But for visual questions with diagrams, graphs, or chemical structures, the model must first describe what it sees in words before it can reason about it. That extra translation step loses information.

(2) Submission timing. Visual MCQ was our first submission, made early in the development cycle. Later submissions benefited from iterative improvements most importantly, a fix to the answer extraction pipeline that ensured enough generation tokens for the model to complete its thinking. This fix may not have been in place for the initial Visual MCQ submission.

(3) Fine-tuning target. Although EXAMS-V contains both visual and textual questions, our fine-tuning only trains the model to predict the correct answer letter. It does not explicitly teach visual understanding, so textual reasoning likely benefits more from this setup.

6.2. OpenQA Limitations

Two design choices hurt our OpenQA scores:

No OpenQA-specific fine-tuning. We fine-tuned only on MCQ data, where the target output is a single letter (A–E). OpenQA needs free-form text, which is a fundamentally different task. The competition provides dedicated OpenQA training data (528 Visual, ∼300 Textual samples with answers), which we did not use for fine-tuning. Adding it would likely push both OpenQA scores up.

Thinking tokens eat into the answer budget. With `max_new_tokens=512`, the model's internal reasoning can consume most of the generation budget, leaving too few tokens for the actual answer. For MCQ this is not a problem since the answer is a single character, but OpenQA answers need real space.

6.3. Answer Extraction Ablation

Answer extraction has a large effect on measured accuracy for thinking-mode models. On the EXAMS-V validation set (zero-shot), we compared four extraction configurations:

Table 7
Post-hoc per-subject accuracy on MCQ test sets (subjects with ≥ 5 samples in both tracks).

Subject	Visual MCQ		Textual MCQ	
	Acc.	n	Acc.	n
Chemistry	0.386	197	0.716	176
Physics	0.330	103	0.634	41
Mathematics	0.298	47	0.651	189
Biology	0.563	87	0.860	136
Science (Chemistry)	0.631	65	0.818	55
Science (Biology)	0.638	94	0.846	26
Science (Physics)	0.578	64	0.889	36
Fine Arts	0.469	32	0.893	28
Fizika	0.524	21	0.902	51

- Naïve extraction (first A–E character found): ∼25%. The thinking tokens are full of incidental A–E characters that are not the answer.
- Post-`</think>` extraction with too few tokens (`max_new_tokens=50`): ∼25%. The reasoning gets cut off and the `</think>` tag never appears.
- Post-`</think>` extraction with 4096 tokens: ∼55%.
- Our full extraction pipeline with 1024 tokens: ∼60%.

The 35-point gap between naive and full-pipeline extraction is on the *same* model outputs. The model's reasoning does not change only how we read it. For thinking-mode models we therefore treat the parsing pipeline as part of the system, not a post-processing afterthought.

6.4. Prompt Engineering Observations

On a 500-sample stratified validation subset, the **thinking** prompt consistently outperformed the direct and decomposed variants. The direct prompt achieved comparable but slightly lower accuracy: the model's native thinking mode activates independently of the prompt formulation, and the structured prompt primarily helps by shaping the output format.

Subject-aware prefixes and language-matched instructions produced modest but consistent improvements at no additional compute cost. The language-matched instruction had the largest effect on non-English questions, where prompting in the question's language reduced instances of the model switching to English mid-response.

7. Error Analysis

Following the release of MCQ gold labels by the task organizers after the evaluation period, we conduct a post-hoc error analysis on both MCQ test sets. We report three views of the errors: per-subject accuracy, the confusion pattern on Visual MCQ, and the per-language pattern from Section 5.2 re-examined against the gold labels.

7.1. Per-Subject Accuracy

Table 7 lists per-subject accuracy for all subjects with at least five samples in both tracks.

The textual–visual gap holds for *every* subject in Table 7, with per-subject gaps between 18 and 38 percentage points. Physics (0.330 visual vs. 0.634 textual) and Chemistry (0.386 vs. 0.716) show the largest gaps, which fits the story that these subjects rely on diagram and graph interpretation in the visual track and lose that dependency in the textual track. Mathematics is the hardest subject on both tracks (0.298 visual, 0.651 textual), which suggests symbolic and quantitative reasoning remains a bottleneck regardless of modality.

7.2. Systematic Confusion: Option-E Bias on Visual MCQ

The confusion matrix on Visual MCQ reveals a systematic bias toward option E. The three most common error patterns are $C \to E$ (65 errors), $B \to E$ (51), and $D \to E$ (48). Notably, *no gold answers in the Visual MCQ test set have option E as the correct label*, so every E prediction is an error by construction. This is consistent with the extraction pipeline's fallback behaviour (Section 3.4, stage 4): when the reasoning span does not produce an unambiguous answer, the last-letter scan preferentially returns whichever A–E character appears latest in the response, and the option-E labels appearing near the end of many prompts (e.g., "E) None of the above") bias the fallback toward that letter. A properly calibrated system on this task should instead *abstain* in these cases; the current design forces a guess and therefore accumulates a systematic error.

7.3. Persistence of the Per-Language Pattern

The per-language pattern from Section 5.2 shows up again in the post-hoc analysis: English and Italian are strongest on both tracks, Chinese and Bulgarian weakest. As noted earlier, we suspect this ordering comes from both pre-training coverage and instruction-tuning data availability, but we cannot verify this without a published per-language breakdown of the Qwen3-VL training mixture.

7.4. Where the Model Fails

Two failure modes recur across our qualitative inspection of the Visual MCQ errors:

Diagram-anchored physics/chemistry questions. Questions that require reading a labelled circuit diagram, a reaction-mechanism scheme, or an annotated graph frequently fail even when the model correctly identifies the topic. In several inspected cases the reasoning span verbally describes the visible elements but assigns numeric values inconsistent with the diagram (e.g., inverts a graph axis, or reads a chemical structure as a related but distinct compound). This is the source of the largest per-subject gap in Table 7.

Long-option MCQs in Chinese or Bulgarian. Questions with multi-line answer options in Chinese or Bulgarian frequently produce output that switches to English partway through the reasoning span, at which point the model's option-comparison degrades. The language-matched answer instruction (Section 3.3) mitigates this but does not eliminate it; the model's cross-lingual maintenance of option identity remains fragile for the two lower-resource languages in the task.

7.5. Bottleneck Attribution

Taken together, the results point to *visual understanding at 8B scale* as the main bottleneck on the visual tracks, with a smaller contribution from the option-E fallback bias. Prompt engineering and answer extraction each move the measured accuracy a lot (Sections 6.3, 6.4), but neither closes the textual–visual gap: the gap is uniform across subjects and largest exactly where diagram interpretation matters.

8. Conclusion

We described the SRH-ReliableAI system for the ImageCLEF 2026 Multimodal Reasoning task: a QLoRA-fine-tuned Qwen3-VL-8B-Thinking model with subject-aware and language-matched prompting. The system ranks 1st on Textual MCQ and 2nd on Textual OpenQA, showing that thinking-mode VLMs are well suited for language-driven exam reasoning. At the same time, the gap between textual and visual performance makes it clear that 8B-scale thinking models still have room to grow on spatial and diagrammatic understanding.

Three directions seem promising for future work: (1) fine-tuning on the available OpenQA training data, which we left unused; (2) visual-specific training or auxiliary modules to improve diagram and graph comprehension; and (3) self-consistency voting, which we implemented but could not deploy because thinking-mode inference is slow (∼30s per question).

Reproducibility. Our code is publicly available.[1] The model uses the publicly available Qwen3-VL-8B-Thinking weights with a QLoRA adapter. All experiments were conducted on a single NVIDIA H200 GPU.

Acknowledgments

Compute resources were provided by the University DataLab at SRH University Heidelberg.

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.-C. Stanciu, A. Radu, R.-G. Bolborici, M. Negru, A.-F. Ene, V.-M. Vasilescu, A.-A. Nicolae, L.-D. Ştefan, M.-G. Constantin, M. Dogariu, A.-G. 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. Eryılmaz, 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. wai 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.-N. 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), Lecture Notes in Computer Science, Springer, Jena, Germany, 2026.

[3] R. Rei, C. Stewart, A. C. Farinha, A. Lavie, COMET: A neural framework for MT evaluation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020.

[4] Qwen Team, Qwen3 technical report, arXiv preprint arXiv:2505.09388 (2025).

[5] DeepSeek-AI, DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, arXiv preprint arXiv:2501.12948 (2025).

[6] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, QLoRA: Efficient finetuning of quantized language models, in: Advances in Neural Information Processing Systems, volume 36, 2023.

[7] 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), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 7768–7791. doi:10.18653/v1/2024.acl-long.420.

[8] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, et al., Qwen2.5-VL technical report, arXiv preprint arXiv:2502.13923 (2025).

[9] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun, et al., MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024.

[10] D. Dimitrov, M. S. Hee, M. Ahsan, S. Ahmad, D. Zlatkova, P. Nakov, I. Koychev, Overview of the ImageCLEF 2025 task on multimodal reasoning, in: CLEF 2025 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, 2025.

[1] https://github.com/Pratik25priyanshu20/imageclef-multimodal-reasoning-2026-

[11] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: Advances in Neural Information Processing Systems, volume 35, 2022.
[12] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, in: International Conference on Learning Representations, 2023.
[13] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: International Conference on Learning Representations, 2022.
[14] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, B. Bossan, PEFT: State-of-the-art parameter-efficient fine-tuning methods, in: GitHub Repository, 2022. https://github.com/huggingface/peft.
[15] L. von Werra, Y. Belkada, L. Tunstall, E. Beeching, T. Thrush, N. Lambert, S. Huang, TRL: Transformer reinforcement learning, in: GitHub Repository, 2020. https://github.com/huggingface/trl.
[16] T. Dettmers, M. Lewis, Y. Belkada, L. Zettlemoyer, LLM.int8(): 8-bit matrix multiplication for transformers at scale, Advances in Neural Information Processing Systems 35 (2022).

A. Prompt Strings and Extraction Regex

For reproducibility we list the exact prompt strings used at inference time and the extraction-regex cascade applied to model outputs.

Prompt Strings

The three MCQ prompt strategies evaluated in Section 3.3, shown in their English form (language-matched variants translate only the final answer-format instruction):

```
Look at this exam question image. Select the
correct answer. Reply with ONLY the letter
(A, B, C, D, or E).
```

```
(1) Read the question and ALL answer options.
(2) If there are diagrams, graphs, or visual
    elements, analyze them.
(3) Think step by step about the correct answer.
(4) Select exactly ONE answer.
Reply with ONLY the letter (A, B, C, D, or E).
```

```
First, describe what you see. Then, reason step
by step. Finally, state your answer as a single
letter (A, B, C, D, or E).
```

```
You are an expert in {subject}.
```

```
Look at this exam question and provide a
concise, accurate answer.
```

Extraction Regex Cascade

Patterns are applied in the order listed. All matches use case-insensitive matching. RESP denotes the model's raw response string; THINK denotes the reasoning span between <think> and </think> (if any); POST denotes the text after </think>.

```
# Stage 1: Post-think extraction.
r'</think>\s*([A-E])'               # match on RESP

# Stage 2: Reasoning-tail extraction (last 3 lines of THINK).
r'\banswer\s+is\s+([A-E])\b'
r'\boption\s+([A-E])\b'
r'\bchoice\s+([A-E])\b'
r'\banswer[:\s]+([A-E])\b'
# Italian:
r'\brisposta\s+[eE\'e\'E]\s+([A-E])\b'
# Bulgarian / Serbian / Croatian / Chinese variants
# defined analogously per language.

# Stage 3: Generic extraction on cleaned RESP.
r'\\boxed\{\s*([A-E])\s*\}'          # LaTeX-style boxed answers
r'\[\s*([A-E])\s*\]'                # bracketed answers
r'^\s*([A-E])\s*$'                  # single-letter line

# Stage 4: Last-letter fallback (reverse scan).
last = None
for ch in reversed(RESP):
    if ch.upper() in 'ABCDE':
        last = ch.upper(); break
return last if last is not None else 'A'
```