Two-Stage LoRA Adaptation of Qwen3-VL for Image-Grounded Medical Visual Question Answering

Mahnaz Mohammed Salih[1] and Ziyang Wang[1]

Aston University, Birmingham, UK
m.mohammedsalih19@aston.ac.uk, z.wang47@aston.ac.uk

Abstract. Medical visual question answering requires both selecting a clinically correct answer and remaining faithful to image evidence. We describe the method used in our final MedReason submission: a single-model system based on Qwen3-VL-32B-Instruct, adapted with Low-Rank Adaptation (LoRA) on the official train split only. Closed-ended items are answered with a single option label. Open-ended items return a mandatory image-grounded `reasoning_trace` together with a short definitive `answer`, matching the challenge contract and the organizer-side Ground-Truth Correctness (GT) and Visual Accuracy (VA) metrics. Adaptation proceeds in two stages. Stage 1 applies LoRA to all 17,722 labelled training cases. Stage 2 continues from that adapter with open-ended upsampling (open×3) and a lower learning rate, and is paired with stricter open-ended prompts and lightweight answer post-processing. The submitted container loads the frozen base checkpoint and the Stage-2 adapter separately and runs offline, single-pass, deterministic decoding. On the hidden public pre-evaluation split, the Stage-1 system obtained 97.14% MCQ accuracy, open-ended GT 1.638/4, and open-ended VA 2.487/4. Local Stage-2 checks retained high in-distribution MCQ accuracy while producing complete, reference-length open answers. Following the challenge camera-ready policy, we do not report forthcoming final leaderboard metrics here.

Keywords: Medical visual question answering · Vision-language models · LoRA · Image-grounded reasoning · MedReason

1 Introduction

Multimodal large language models (MLLMs) are increasingly applied to medical images, but clinically useful visual question answering (VQA) is not only a matter of picking the right option. A plausible free-text answer can still rest on hallucinated findings, while an image-faithful explanation can fail to match a concise reference statement. MedReason [11] makes this distinction explicit. It evaluates one medical VQA task across closed-ended multiple-choice questions (MCQ) and open-ended questions, and it scores open-ended outputs with separate Ground-Truth Correctness (GT) and Visual Accuracy (VA) metrics. The

public benchmark backbone is Med-CMR [4], with additional out-of-distribution testing on low-dose X-ray and 7T brain MRI.

This paper describes the method used in our *final* challenge submission. We do not introduce a new architecture. Instead, we show that a strong general-purpose vision-language model, Qwen3-VL-32B-Instruct [1], can be specialized to MedReason with parameter-efficient LoRA [6], task-specific JSON prompting, and output post-processing that separates visual reasoning from the scored answer string. Two design choices follow directly from the evaluation protocol. First, MCQ performance is an exact-match label problem, so the model is trained and decoded to emit a single option letter. Second, open-ended VA depends on both the answer and the reasoning trace, with a threshold gate that caps VA when the trace is unfaithful. Prompts therefore keep evidence and uncertainty in `reasoning_trace` and keep `answer` short and definitive.

A Stage-1 LoRA adapter trained on the full official train split already transferred well to hidden MCQ cases in organizer-side pre-evaluation (97.14% accuracy). Open-ended GT lagged VA, which motivated a second continued-training stage with open-ended upsampling and tighter answer normalization. The final Docker package ships that Stage-2 adapter unmerged, so inference remains a single frozen 32B checkpoint plus a small PEFT module [10].

Our contributions are:

- A complete, containerized MedReason system: Qwen3-VL-32B-Instruct with one LoRA adapter, task-specific prompts, and schema-valid JSON outputs.
- A practical two-stage LoRA recipe aligned with MedReason's asymmetric MCQ versus open-ended GT/VA protocol: full-mix Stage 1, then open-ended upsampling in Stage 2 without extra private data.
- An analysis of local development monitors versus organizer-side pre-evaluation, highlighting the MCQ/open-ended gap and the limits of train-set accuracy as a generalization estimate.

2 Related Work

Medical VQA datasets such as VQA-RAD [7], SLAKE [9], and PathVQA [5] established image–question–answer evaluation, typically with short answers or closed options. Recent medical MLLMs, including LLaVA-Med [8] and Med-Flamingo [12], adapt general vision-language models to biomedical instruction following. Med-CMR [4] stresses fine-grained visual evidence and multi-step clinical logic; reported zero-shot MCQ accuracy for strong general models remains far below typical in-domain fine-tuned scores, which is consistent with our own zero-shot baseline on MedReason train MCQ cases.

Parameter-efficient fine-tuning, especially LoRA [6], is the practical route to adapting 32B-scale models on a single high-memory GPU. Qwen3-VL [1] provides a native vision tower and chat template that we use without architectural modification. MedReason differs from earlier VQA leaderboards by scoring open-ended visual faithfulness with a vision-language judge and by requiring a

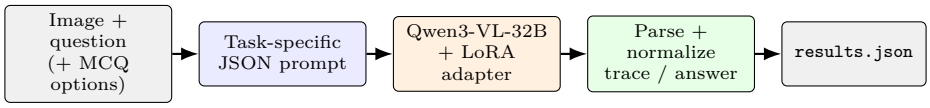

Fig. 1. Submitted inference pipeline. A single Qwen3-VL-32B-Instruct checkpoint and one unmerged LoRA adapter produce JSON with `reasoning_trace` and `answer`. Post-processing enforces the official option labels for MCQ and short reference-style answers for open-ended cases.

reasoning trace in the official output schema [11]. Our method is therefore built around that contract rather than around retrieval, tool use, or model ensembling.

3 Method

Figure 1 summarizes the submitted inference pipeline. Each case is processed independently in one generation pass. There is no agent loop, no retrieval, and no external API at evaluation time.

3.1 Backbone and Parameter-Efficient Adaptation

The backbone is Qwen3-VL-32B-Instruct [1], loaded in bfloat16 with its native processor and vision tower. Images are consumed as RGB PNG files referenced by `cases.json`; we apply no extra normalization beyond the model processor. Full fine-tuning of a 32B multimodal model is unnecessary for this dataset size and would not fit comfortably in a single-GPU training budget. We therefore freeze the pretrained weights and train LoRA adapters [6, 10] on the attention and MLP projections `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`. Rank r=16, α=32, and dropout 0.05 are held fixed. The adapter is *not* merged into the base weights for submission: the container sets `MEDREASON_MODEL_PATH` and `MEDREASON_LORA_PATH` and loads the PEFT module at runtime. This keeps the packaged adapter small, reversible, and consistent with the trained checkpoint.

3.2 Task-Specific Prompting

A shared system prompt instructs the model to use image evidence, to avoid inventing unsupported findings, and to return only JSON with keys `reasoning_trace` and `answer`. User prompts then split by task type.

For MCQ cases, the prompt lists every official option and requires the `answer` field to contain only the selected label (for example, `A`). The trace is a short image-grounded rationale. This matches exact-match MCQ scoring.

For open-ended cases, the prompt requires image-grounded evidence and step-by-step reasoning *only* in `reasoning_trace`, including uncertainty and evidence limits. The `answer` must be a short definitive clinical statement (typically

Table 1. LoRA adaptation used in the final submission. Stage 2 resumes from the Stage 1 adapter. Record counts include open-ended upsampling.

Setting	Stage 1	Stage 2 (submitted)
Train cases / records	17,722 / 17,722	17,722 / 24,370
Open-ended repeat	×1	×3
Epochs	1	1 (continue)
Learning rate	1×10^{-4}	5×10^{-5}
Warmup / weight decay	0.03 / 0.01	0.03 / 0.01
Effective batch size	16	16
Approx. wall time (1× GH200)	∼6 h	∼7–8 h
Final reported train loss	–	∼0.37

under about 25 words), without hedging prefixes such as "Based on the image" or "The answer is". The motivation is the official open-ended protocol [11]: GT is a text-only comparison of the submitted answer with the reference, whereas VA uses the image together with the answer and, through a threshold gate, the faithfulness of the reasoning trace. Mixing hedging into the answer string can hurt GT even when the visual reading is conservative; omitting a grounded trace can cap VA even when the answer is correct.

Training targets follow the same JSON structure. For MCQ, the target answer is the official label and the target trace is a short template that names the chosen option. For open-ended items, the target answer is the released reference string and the target trace is the released visual description when available, otherwise a generic image-grounded placeholder. Supervising the trace with visual description encourages the model to verbalize findings that VA judges can check against the image.

3.3 Two-Stage LoRA Training

Table 1 lists the two training stages. Both use AdamW, a cosine schedule, gradient checkpointing, per-device batch size 1, and 16 accumulation steps (effective batch 16). Training ran on a single NVIDIA GH200 GPU (∼96 GB). No private, proprietary, or extra public datasets were used.

Stage 1 exposes the adapter to the entire official mix (14,398 MCQ and 3,324 open-ended cases). This stage is the system that was sent to organizer-side pre-evaluation. After that run, open-ended GT was clearly weaker than MCQ accuracy and weaker than VA, which may partly reflect the smaller proportion of open-ended examples in the training mix (18.8% of the train split); GT also rewards concise reference-style wording rather than lengthy explanation.

Stage 2 continues from the Stage 1 adapter rather than training from scratch. Each open-ended record is repeated three times (open×3), yielding 24,370 records, and the learning rate is halved to 5×10^{-5} for one additional epoch. The open×3 factor is a pragmatic engineering choice rather than the result of a search over upsampling rates. Continuation preserves the already-strong MCQ behaviour

while increasing the gradient share of open-ended JSON targets. Stage 2 is the adapter shipped in the final Docker image.

3.4 Decoding, Parsing, and Post-Processing

Inference uses greedy decoding: temperature 0, top-$p = 1$, and `max_new_tokens` = 512. Deterministic decoding is appropriate for a single official run per Docker package.

The raw completion is parsed as JSON, with fallbacks for fenced code blocks and for "Answer:" markers if the model leaves the JSON schema. Empty traces are replaced with a non-empty placeholder so that open-ended schema validation cannot fail on a missing field. MCQ answers are mapped onto the official option set by exact label, case-insensitive label, or option-text match; if all heuristics fail, the first official label is used so that the container still writes a valid file under time pressure.

Open-ended answers are normalized toward short reference-style statements: leading hedges are stripped, wrapping quotes are removed, and overlong outputs are cut to the first sentence or a 220-character budget. The reasoning trace is left intact, because VA and reasoning visual faithfulness depend on it. We do not ensemble models or apply a second-pass verifier.

3.5 Containerized Inference

The official evaluator mounts `/input` and `/output` and disables internet access. Our container follows that contract: it reads `/input/cases.json` and the referenced images, writes `/output/results.json`, and uses only baked-in weights. The implementation is Python 3.11 with PyTorch [13] and Hugging Face Transformers [14]. Each case is a single-pass generation with the Qwen chat template and `qwen-vl-utils` vision preprocessing.

4 Experiments

4.1 Data and Splits

We use only the official MedReason participant packages [11]. The train split contains 17,722 labelled cases (14,398 MCQ and 3,324 open-ended). The participant-facing validation split contains 2,532 cases without public answers and is used solely to check input/output format and end-to-end runtime. Hidden public and private leaderboard splits are never accessed locally; official scores come only from organizer-side execution of the submitted container.

4.2 Evaluation Protocol

MCQ accuracy is exact match of the submitted option against the reference option. Open-ended scoring is organizer-side [11]. Ground-Truth Correctness (GT)

Table 2. Results. Train-subset MCQ figures are in-distribution monitors. Pre-evaluation is organizer-side on the hidden public split and corresponds to the Stage 1 adapter. Stage 2 is the final submitted system. Per challenge camera-ready guidance, forthcoming official final metrics are omitted here.

System	Split / protocol	MCQ Acc.	Open GT	Open VA
Zero-shot 32B	Train MCQ 1–500 (local)	42.6%	–	–
Stage 2 LoRA	Train MCQ 1–500 (local)	99.8%	–	–
Stage 2 LoRA	Train MCQ 501–1000 (local)	99.8%	–	–
Stage 1 LoRA	Hidden public pre-eval	97.14%	1.638/4	2.487/4

is assigned by Llama-3.1-70B-Instruct [3] in a text-only setting from the question, reference answer, and submitted answer. Visual Accuracy uses Qwen2.5-VL [2] on the image, question, and generated text. Case-level rubric scores lie in $\{0, 1, 2, 3, 4\}$. Final VA is threshold-gated: if reasoning visual faithfulness is at most 1, VA is capped at 1; if it equals 2, VA is capped at 3. Leaderboard GT and VA are means over open-ended cases. We cannot reproduce the hidden-test judges locally, so local open-ended analysis is limited to schema completeness, empty-field rates, answer length, and (on train) rough string agreement.

4.3 Local Development Checks

Because validation labels are withheld, we used labelled train subsets as *development monitors*, not as held-out estimates of hidden-test performance. A zero-shot Qwen3-VL-32B-Instruct baseline, with the same prompting stack but no LoRA, reached 42.6% MCQ accuracy on train MCQ cases 1–500. After Stage 2, the same slice and the next slice (cases 501–1000) both reached 99.8% MCQ accuracy. These near-ceiling numbers show that the adapter fits the training distribution; they do not imply 99.8% on hidden data. The Stage-1 hidden public MCQ score of 97.14% is the better indicator of closed-ended transfer.

An open-ended smoke set of 100 train open cases with the Stage-2 adapter produced valid schema for every item, with zero empty answers and zero empty traces. Median answer length was about 79 characters, in the range of the released reference answers rather than long unconstrained essays.

Full validation inference (2,532 cases) was run with the Stage-1 system to confirm that the Docker I/O path scaled to the official case count.

5 Results

Table 2 collects the measurements we can report under the challenge publication policy. LoRA adaptation produces a large MCQ gain over the same 32B backbone without fine-tuning (42.6% $\rightarrow$ 99.8% on the first 500 train MCQ cases). Organizer-side pre-evaluation confirms that this gain is not confined to the training IDs: Stage 1 reached 97.14% MCQ accuracy on hidden public cases.

Open-ended pre-evaluation is more mixed. VA (2.487/4) is substantially higher than GT (1.638/4). Under the official rubric, that pattern suggests that answers and traces are often visually supported, but the final answer string is only partly aligned with the reference. Possible causes include wording mismatch (definitive vs. hedging, granularity, laterality phrasing), the smaller open-ended fraction of the train mix, and the fact that Stage 1 traces for MCQ used a generic template rather than rich supervision. This gap is what Stage 2 was designed to address: open×3 upsampling, a lower continued learning rate, stricter open prompts, and answer normalization that strips hedges while leaving the trace untouched.

Evidence and limitations of Stage 2. Stage 2 is the adapter used in the final Docker submission, but its benefit on hidden open-ended metrics is *not* directly validated. Each team receives only one organizer-side pre-evaluation run; we used that slot for the Stage 1 package, so no Stage 1 versus Stage 2 ablation is available on the hidden public split. The positive evidence we can report is therefore local and indirect: (i) Stage 2 retained near-ceiling MCQ accuracy on two disjoint train slices (cases 1–500 and 501–1000), reducing the risk that continued training collapsed closed-ended behaviour; and (ii) an open-ended smoke set of 100 train cases produced complete schema with non-empty traces and reference-like answer lengths. These checks support packaging Stage 2 as a low-risk final system, but they do not substitute for organizer-side GT/VA. We therefore treat open×3 upsampling, the lower learning rate, and the stricter open prompts as motivated engineering choices rather than as statistically confirmed hidden-split improvements. Additional hidden ablations would strengthen the claim, but new experiments are outside the scope of this camera-ready revision.

6 Discussion

The main empirical finding is that closed-ended MedReason performance can be pushed into a high-accuracy regime with a single 32B VLM and one epoch of LoRA, whereas open-ended GT remains the harder objective even when VA is reasonable. This matches the challenge design. MCQ is a constrained classification problem with an official option set. Open-ended GT is a semantic agreement problem against a short reference, judged by a strong text-only LLM [3], while VA additionally penalizes unfaithful traces [2, 11]. Optimizing only a long chain-of-thought can help the visual judge and still miss the reference wording; optimizing only a terse answer can help GT and starve the trace.

Separating the two fields in both the prompt and the post-processor is our practical response. It is a simple inductive bias, not a guarantee. Residual GT errors likely include clinically close but non-matching phrasing, missed subtle findings, and out-of-distribution imaging (low-dose X-ray, 7T MRI) that the train split may under-represent. We did not add extra public datasets or test-time computation, so any OOD robustness must come from the pretrained Qwen3-VL prior plus MedReason LoRA.

Limitations are explicit. Train-subset MCQ accuracy overestimates generalization. We lack local open-ended judges, so we cannot quantify Stage 2 GT/VA locally, and we lack a Stage 2 hidden pre-evaluation because the single organizer-side slot was used for Stage 1. We did not search LoRA rank, learning rate, or upsampling factor beyond the two-stage recipe. The MCQ fallback to the first option is a validity-preserving last resort, not a calibrated uncertainty mechanism. Finally, this paper reports a team method for the workshop track; it is not a cross-team challenge analysis.

7 Conclusion

We submitted a single-model MedReason system: Qwen3-VL-32B-Instruct with an unmerged LoRA adapter, task-specific JSON prompting, and light post-processing aligned to MCQ exact match and open-ended GT/VA. Two-stage adaptation first covers the full official train split, then continues with open-ended upsampling. Hidden public pre-evaluation of Stage 1 showed strong MCQ transfer (97.14%) with a remaining open-ended GT gap (1.638/4) relative to VA (2.487/4). Stage 2 is the final packaged method; it is motivated by that open-ended gap and by local schema/MCQ safety checks, while its benefit on hidden GT/VA remains an acknowledged limitation rather than a measured claim. Aggregate final challenge outcomes are left to the joint MedReason challenge manuscript.

Acknowledgments. Compute for LoRA training and local evaluation was provided by the Isambard-AI national AI Research Resource. We thank the MedReason organizers for the challenge infrastructure and feedback.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-VL technical report. arXiv preprint arXiv:2511.21631 (2025)
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2.5-VL technical report. arXiv preprint arXiv:2502.13923 (2025)
3. Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al.: The Llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
4. Gong, H., Ji, X., Liu, Y., Wu, W., Yan, X., Liu, J., Wu, K., Pan, J., Jian, B., Zhang, J., Hu, X., Li, H.: Med-CMR: A fine-grained benchmark integrating visual evidence and clinical logic for medical complex multimodal reasoning. arXiv preprint arXiv:2512.00818 (2025)

5. He, X., Zhang, Y., Mou, L., Xing, E., Xie, P.: PathVQA: 30000+ questions for medical visual question answering. In: Medical Imaging with Deep Learning (2020)
6. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022)
7. Lau, J.J., Gayen, S., Ben Abacha, A., Demner-Fushman, D.: A dataset of clinically generated visual questions and answers about radiology images. Scientific Data **5**, 180251 (2018)
8. Li, C., Wong, C., Zhang, S., Usuyama, N., Liu, H., Yang, J., Naumann, T., Poon, H., Gao, J.: LLaVA-Med: Training a large language-and-vision assistant for biomedicine in one day. In: Advances in Neural Information Processing Systems (2023)
9. Liu, B., Zhan, L.M., Xu, L., Ma, L., Yang, Y., Wu, X.M.: SLAKE: A semantically-labeled knowledge-enhanced dataset for medical visual question answering. In: IEEE International Symposium on Biomedical Imaging (ISBI). pp. 1650–1654 (2021)
10. Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S.: PEFT: Parameter-efficient fine-tuning of billion-scale models on low-resource hardware. https://github.com/huggingface/peft (2022)
11. MedReason challenge. https://medreason26.github.io/challenge.html (2026), mICCAI 2026 Satellite Event, last accessed 2026/08/16
12. Moor, M., Huang, Q., Wu, S., Yasunaga, M., Zakka, C., Dalmia, Y., Reis, E.P., Rajpurkar, P., Leskovec, J.: Med-Flamingo: A multimodal medical few-shot learner. arXiv preprint arXiv:2307.15189 (2023)
13. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems **32** (2019)
14. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45 (2020)