QwenVLConnector: A Fast, Unified Medical VLM Chatbot for Fine-Grained Clinical Perception and Text Generation

Le Thien Phuc Nguyen[1*], Hoang-Thien Nguyen[2*], Thanh-Huy Nguyen[3], Gia Minh Hoang[4], Mai-Anh Vu[5], and Ulas Bagci[6]

[1] University of Wisconsin - Madison, USA
plnguyen6@wisc.edu
[2] Posts and Telecommunications Institute of Technology, Vietnam
[3] Carnegie Mellon University, USA
[4] Mayo Clinic, College of Medicine and Science, Scottsdale, USA
[5] University of Houston, USA
[6] Northwestern University, USA
ulas.bagci@northwestern.edu

Abstract. Most medical vision–language models (VLMs) excel at open-ended report generation and VQA but lack native support for *structured*, fine-grained perception (detection, counting, regression) within one interface. We present QwenVLConnector, a Qwen2.5-VL–based chatbot that unifies *classification*, *multi-label classification*, *textualized detection*, *counting*, *regression*, and free-form *report generation* under a single next-token objective via a lightweight dense multi-layer *Connector* that fuses multi-scale visual features without increasing sequence length. On FLARE-2D, QwenVLConnector improves detection F1 from 0.55 to 0.85 (+0.30), raises single-label classification from 0.37 to 0.51 (+0.15), and boosts report-generation GREEN by up to 18.3 points. Our code is available at https://github.com/plnguyen2908/QwenConnector.

Keywords: Multimodal large language model · Medical VLM · Dense connector · FLARE 2D

1 Introduction

Medical vision–language chatbots have progressed rapidly from task-specific tools to assistants that can describe, reason about, and converse over clinical imagery. Systems such as LLaVA-Med [6] and HuatuoGPT-Vision [3] already deliver strong open-ended report generation and VQA; however, most models remain optimized for text generation. They answer fluently but provide limited support for the *structured, fine-grained* perception clinicians routinely need—*detection*, *counting*, and *regression*—within a single conversational interface. The FLARE 2D challenge [9], spanning diverse modalities and task types, makes this gap especially salient and provides a concrete, clinically meaningful testbed.

* These authors contributed equally.

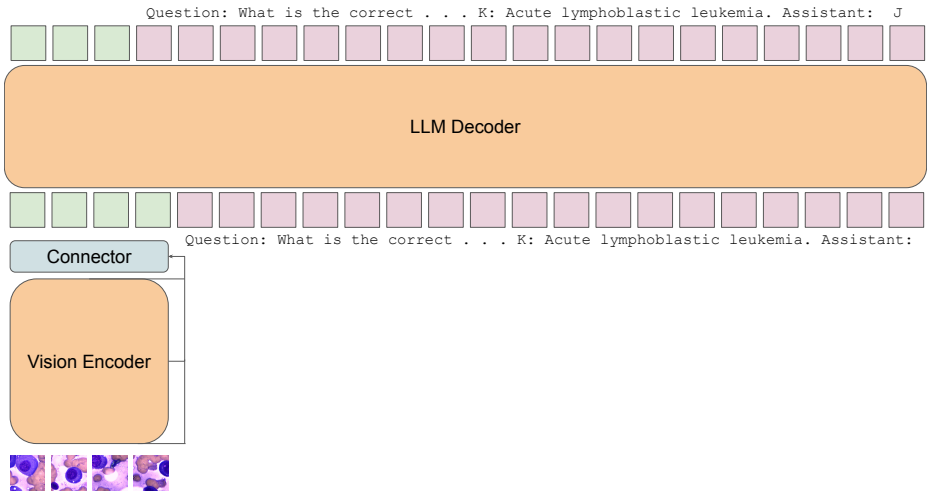

Fig. 1: **High-level architecture of QwenVLConnector.** A Vision Encoder extracts features from medical images; a lightweight dense multi-layer *Connector* fuses low/high-level cues and projects them to the LLM without increasing sequence length. The fused visual tokens (green) are prepended to the text tokens (pink), and a decoder-only LLM performs next-token generation for every task.

Modern VLMs typically couple a pretrained vision encoder to a decoder-only LLM via a lightweight connector and train the stack with next-token prediction over image/video–text corpora. Variants explore stronger backbones, smarter adapters, and refined alignment/instruction curricula—e.g., PandaGPT aligns multiple perceptual encoders via small projections for unified image/video QA [12]; Video-LLaMA extends to long-form video with temporal reasoning [16]; and Qwen2.5-VL enhances the vision stack with a ViT tower, a *Merger* to consolidate patch embeddings, and improved positional encoding [2]. In medicine, 2D MLLMs (LLaVA-Med, HuatuoGPT-Vision/PubMedVision) adapt this recipe through domain alignment and medical instruction tuning, while 3D frameworks pair volumetric encoders with LLMs for CT/MRI dialogue and localization [6,3,5,13]. Building on these insights, we introduce QwenVLConnector, a clinical VLM chatbot that unifies *classification*, *multi-label classification*, *object detection* (textualized outputs), *counting*, *regression*, and free-form *report generation* under a single next-token interface. The key design choice is a dense, multi-layer *Connector* that aggregates low- and high-level visual features *before* the LLM via channel-wise fusion, enriching tokens without increasing sequence length (Figure 1 and 2).

Contributions: (i) We create an unified medical VLM chatbot that executes both open-ended *text generation* and *structured* tasks (classification, multi-label classification, detection, counting, regression). (ii) We introduce QwenVLConnector,

a lightweight dense connector that enriches visual tokens with multi-layer cues, improving fine-grained performance while meeting the challenge's *fast and efficient* requirement. (iii) We introduce an in-context learning strategy to improve report generation's score.

2 QwenVLConnector

In this section, we first summarize preliminaries of our method and fix notation (Section 2.1), grounding our setup in prior work [2]. Next, we present the proposed *QwenVLConnector* architecture that fuses high- and low-level visual features (Section 2.2). We then detail the *training data* (Section 2.3). Finally, we describe our multimodal in-context learning strategies for prompting and adaptation (Section 2.4).

2.1 Preliminaries

Our model builds on the Qwen2.5-VL architecture, which employs a vision encoder f_θ (ViT with windowed attention) to extract hierarchical visual features from an image or video x_v. In Qwen2.5-VL, a *Merger* module aggregates patch-level embeddings into a sequence of high-level visual tokens. We introduce a *QwenVLConnector* module, a custom adapter, that fuses these high-level embeddings with selected low-level features from early encoder layers, enriching the visual representation with fine-grained spatial detail before integration with the language model. Let $\mathcal{V}$ denote the LLM vocabulary and $w_{1:n} = (w_1, \ldots, w_n) \in \mathcal{V}^n$ denote a length-n sequence of text tokens. The fused representation $\tilde{Z}$ is fed into a decoder-only LLM g_ϕ as an interleaved sequence $[\tilde{Z}, w_{1:n}]$ for autoregressive text generation.

2.2 QwenVLConnector Architecture

The vision encoder is a ViT with L=32 blocks. After windowed attention, the hidden state from block ℓ is $h^{(\ell)} \in \mathbb{R}^{N \times d_v}$ where N is number of visual tokens, and d_v is ViT width. The encoder's built-in *Merger* produces output tokens $Z_0 \in \mathbb{R}^{N \times d_m}$. We reuse that Merger as a mapping $m_\mu : \mathbb{R}^{N \times d_v} \to \mathbb{R}^{N \times d_m}$ to align intermediate features. P denotes the permutation induced by windowing; P^{-1} restores the original token order. The connector MLP is $c_\psi : \mathbb{R}^{N \times 3d_m} \to \mathbb{R}^{N \times d_t}$, where d_t matches the LLM embedding size.

Computation. We average features over two depth groups to obtain complementary summaries: G_1={1, . . . , 16} (edge/texture–heavy) and G_2={17, . . . , 32} (semantic-heavy),

$$A_1 = \tfrac{1}{|G_1|} \sum_{\ell \in G_1} h^{(\ell)}, \qquad A_2 = \tfrac{1}{|G_2|} \sum_{\ell \in G_2} h^{(\ell)} \in \mathbb{R}^{N \times d_v}.$$

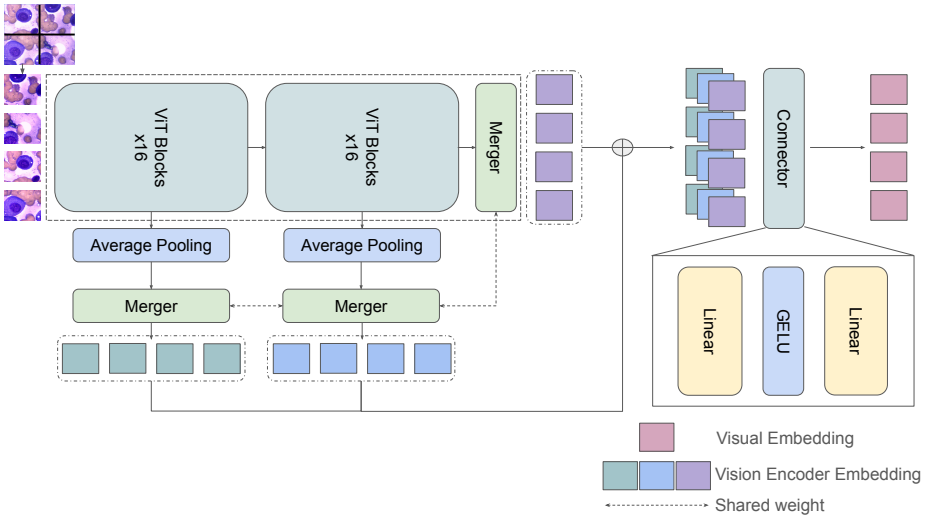

Fig. 2: **Dense multi-layer connector in QwenVLConnector.** Early and late ViT features are depth-averaged, aligned via the backbone *Merger*, concatenated channel-wise with the encoder output, and projected by a lightweight MLP to LLM visual tokens.

Each is aligned by the Merger and de-permuted, then concatenated channel-wise with the encoder output and projected:

$$\tilde{Z} = c_\psi\big([\,Z_0,\ P^{-1}m_\mu(A_1),\ P^{-1}m_\mu(A_2)\,]\big)\ ,$$

yielding visual tokens $\tilde{Z} \in \mathbb{R}^{N\times d_t}$ at the *same* token length N. These tokens are prepended to the text tokens $w_{1:n}$ and consumed by the decoder-only LLM for next-token generation as mentioned in 2.1. Detailed visualization of our method is provided in Figure 2.

2.3 Training Data

We use a two-stream mixture for both *alignment* and *instruction tuning*. For alignment, general LLaVA-style image–caption pairs (e.g., LAION [11,7]) are combined with the PubMedVision *alignment* split of denoised PubMed image–text pairs [3], yielding ∼1.2M examples that stabilize open-domain grounding while injecting clinical vocabulary and fine-grained cues. For instruction tuning, we merge LLaVA-Instruct with PubMedVision's medical instructions [7,3], formatting all samples in a unified chat-style template with interleaved image tokens and prompts (∼1.3M examples).

As a final stage, we fine-tune on FLARE-2D Task 5 [9], a multimodal medical VQA corpus spanning *8* imaging modalities and *7* task types. We use the official training split (45k question–answer pairs aggregated from 19 datasets) and apply the same chat-style formatting.

2.4 In-Context Learning

Motivated by prior findings that in-context learning (ICL) improves domain use of LLMs without extra training [1,10], we apply ICL to the report-generation task. Concretely, we form a support pool $\mathcal{S}$ of N distinct training examples and, at inference, uniformly sample k demonstrations ($k \ll N$). Each demonstration is an image–prompt–response triple formatted in our chat template and concatenated before the user query inside a fixed instruction wrapper (Table 1). No model weights are updated; the LLM conditions on the k demonstrations plus the query to generate the final report.

Table 1: Prompt template with ICL mechanism.

Prompt
Instruction: [Task Instruction]
User: [Demonstration 1]
Response: [Answer 1]
...
User: [Demonstration k]
Response: [Answer k]
User: [Question]
Response:

3 Experimental Results

In this section, we present: (i) *Hyperparameter setup* (Section 3.1); (ii) *Evaluation protocol* detailing splits, metrics, prompting templates, and evaluation procedure (Section 3.2); (iii) *Validation results* across classification, multi-label classification, detection, counting, and regression (Section 3.3); (iv) *In-context learning* analyses with zero-/few-shot prompting and ablations (Section 3.4); and (v) *Qualitative results* including case studies and error analyses (Section 3.5).

3.1 Implementation details

Training Protocols. We adopt a three-stage training pipeline: (i) *Vision–language alignment* using large-scale image-text corpora to map visual features into the LLM embedding space while stabilizing the language backbone; (ii) *Visual instruction tuning* on curated multimodal dialogue datasets to elicit robust instruction-following behavior; (iii) *Domain-specific finetuning* on the FLARE dataset to adapt the model for specialized clinical image understanding tasks.

Environmental settings. We use AdamW 8-bit [4], bfloat16 precision, a cosine learning-rate scheduler with warmup ratio = 0.03 [8], and max sequence length

= 2048. Also, we train our model under 8-bit quantization for memory efficient. Other hyperparameters follow Huggingface's trainer library defaults unless specified in Table 2. Other dependencies and code-related information are available in the codebase provided in the abstract.

Table 2: Key hyperparameters per training stage.

	Alignment	Instruction	FLARE tuning
Epochs	1	1	3
Learning rate	2e-3	2e-4	2e-4
Per-device batch	1	1	1
Number of devices	8	8	8
GPU type	A40	A40	A40
Training Time	3 days	7 days	1 day
Grad. accumulation	64	64	16
LoRA (rank, α)	–	(32, 32)	(32, 64)

3.2 Evaluation Protocol

We use the FLARE challenge's two-track validation [9]: *val-hidden* (Codabench) for classification, multi-label, detection, instance detection, regression; and *val-public* (local) for counting and report generation. Metrics are Balanced Accuracy (classification), micro-F1 (multi-label), F1 at IoU>0.5 (detection/instance), MAE (regression/counting), and GREEN score (report generation).

3.3 Validation Results

Table 3: **Classification evaluation.** The result is reported on validiation-hidden subset on Codabench. All metrics are reported as fractions in [0,1].

Model	Classification ↑	Multi-label classification ↑
4-bit QwenVL 2.5 7B [15]	0.37	0.57
8-bit QwenVL 2.5 7B [15]	0.36	0.56
16-bit QwenVL 2.5 7B [15]	0.35	0.55
4-bit QwenVLConnector 7B (Ours)	0.51	0.49
8-bit QwenVLConnector 7B (Ours)	0.46	0.53
16-bit QwenVLConnector 7B (Ours)	0.50	0.54

Table 4: **Detection evaluation.** The result is reported on validiation-hidden subset on Codabench. All metrics are reported as fractions in [0,1].

Model	Detection ↑	Instance Detection ↑
4-bit QwenVL 2.5 7B [15]	0.51	0
8-bit QwenVL 2.5 7B [15]	0.55	0
16-bit QwenVL 2.5 7B [15]	0.53	0
4-bit QwenVLConnector 7B (Ours)	0.76	0
8-bit QwenVLConnector 7B (Ours)	0.85	0
16-bit QwenVLConnector 7B (Ours)	0.81	0

Classification evaluation. On the Codabench validation-hidden split (Table 3), QwenVLConnector surpasses the Qwen2.5-VL baseline in *single-label classification* at all bit-widths: +0.14 at 4-bit quantization (from 0.37 to 0.51), +0.1 at 8-bit quantization (from 0.36 to 0.46), and +0.15 at 16-bit quantization (from 0.35 to 0.50). For *multi-label classification*, as the Micro-averaged F1 increases with precision - 0.49 (4-bit quantization), 0.53 (8-bit), and 0.54 (16-bit) - QwenVLConnector approaches the baseline (0.57), reducing the gap from −0.08 to −0.03. Overall, 8-bit quantization offers a strong efficiency–accuracy trade-off (large single-label gains with competitive multi-label scores), while 16-bit quantization maximizes multi-label performance.

Detection evaluation. Using F1 as the metric, QwenVLConnector markedly outperforms the Qwen2.5-VL baseline on detection in Table 4: 0.76 at 4-bit quantization (+0.25, from 0.51 to 0.76), 0.85 at 8-bit quantization (+0.3, from 0.55 to 0.85), and 0.81 at 16-bit quantization (+0.28, from 0.53 to 0.81). The best F1 is achieved at 8-bit quantization, suggesting an effective efficiency–accuracy sweet spot. *Instance detection* F1 is 0 across all settings, indicating that our current textualized detection output does not satisfy the instance-level scoring protocol; enabling structured instance outputs is left for future work.

Regression, counting, and report generation. From Table 5, *counting* decreases at all settings: from 287.30 to 284.44 (−2.86) at 4-bit quantization, from 276.77 to 275.81 (−0.96) at 8-bit quantization, and from 268.24 to 266.70 (−1.54) at 16-bit quantization, with the best MAE at 16-bit. *Report generation* increases at 4-bit quantization from 65.74 to 76.01 (+10.27), at 8-bit quantization from 55.77 to 74.05 (+18.28), and at 16-bit quantization from 74.78 to 74.93 (+0.15), peaking at 4-bit. In contrast, *regression* increases—at 4-bit quantization from 15.51 to 20.91 (+5.40), at 8-bit quantization from 15.19 to 21.13 (+5.94), and at 16-bit quantization from 15.43 to 19.04 (+3.61)—highlighting a remaining gap for purely numeric targets.

Table 5: **Regression, Counting, and Report Generation evaluation.** The result of Regression is reported on validation-hidden subset on Codabench. The result of Counting and Report Generation is reported on validation-public subset. Report Generation is reported as decimal number in [0, 100].

Model	Regression ↓	Counting ↓	Report Generation ↑
4-bit QwenVL 2.5 7B [15]	15.51	287.3	65.74
8-bit QwenVL 2.5 7B [15]	15.19	276.77	55.77
16-bit QwenVL 2.5 7B [15]	15.43	268.24	74.78
4-bit QwenVLConnector 7B (Ours)	20.91	284.44	76.01
8-bit QwenVLConnector 7B (Ours)	21.13	275.81	74.05
16-bit QwenVLConnector 7B (Ours)	19.04	266.7	74.93

3.4 In-Context Learning Results

Figure 3 summarizes the effect of adding k in-context demonstrations for report generation on the validation-public split. Across all four metrics—BLEU, GREEN Clinical Significance, GREEN Entity Matching, and overall GREEN Score—ICL consistently improves performance compared with prompts without demonstrations. These gains indicate that brief, in-domain exemplars help the model follow radiology style and terminology more faithfully, complementing our connector-based improvements without additional finetuning. Due to compute/memory limits on the competition testing server and Docker submission constraints, ICL was not enabled in our container; the results in Figure 3 are from offline runs and are shown as a potential future direction.

3.5 Qualitative Result

Qualitative results. Figure 4 showcases ultrasound images for two tasks. *Left (classification):* QwenVLConnector selects the correct option (A), whereas QwenVL 2.5 [15] predicts C. *Right (detection):* our textualized box (green) aligns more closely with the ground truth (blue), achieving IoU = 0.70, while QwenVL 2.5's [15] box (red) attains IoU = 0.25. These examples underscore QwenVLConnector's stronger recognition and localization on ultrasound imagery.

4 Limitation and future work

Despite unifying open-ended and structured tasks, the system has several gaps: instance-level prediction is not realized because textualized detection cannot satisfy instance scoring; numeric grounding is weak, with regression lagging and counting improvements modest; performance is sensitive to precision/quantization and training remains compute-intensive; and safety/consistency checks remain limited relative to clinical requirements. In addition, the in-context learning

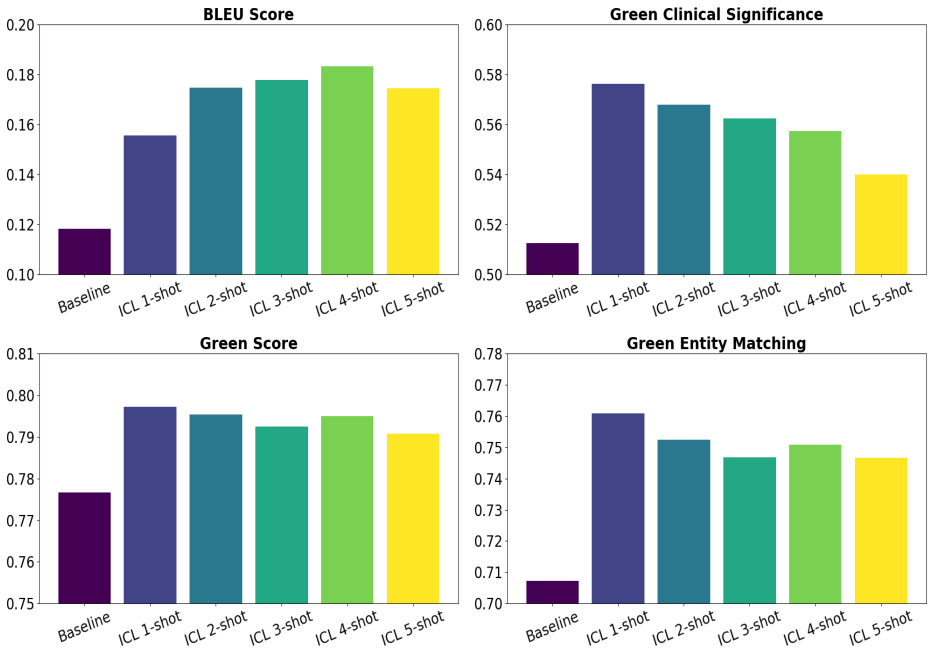

Fig. 3: Effect of in-context learning (ICL) on QwenVLConnector for the report-generation task: BLEU, GREEN Clinical Significance, GREEN Entity Matching, and overall GREEN Score on the validation-public split. Incorporating k demonstrations consistently improves performance over no-ICL prompts.

(ICL) strategy substantially increases prompt length and inference memory/latency. Therefore, an in-depth, task-specific study of demonstration budgeting is needed to balance accuracy, cost, and latency. Future work will need to explore stronger connector architectures that more effectively fuse multi-layer visual cues, improving structured prediction while preserving efficiency.

5 Conclusion

QwenVLConnector advances medical VLMs by unifying open-ended report generation with structured perception—classification, multi-label classification, detection, counting, and regression—within a single next-token interface. Built on Qwen2.5-VL with a lightweight dense multi-layer connector, the system improves fine-grained recognition while preserving efficiency and delivers consistent gains on FLARE 2D tasks, particularly for classification and detection. Nonetheless, gaps remain in instance-level prediction and numeric grounding. Future work will need to explore stronger connector architectures that fuse multi-layer visual cues more effectively to enhance structured prediction while maintaining speed and simplicity.

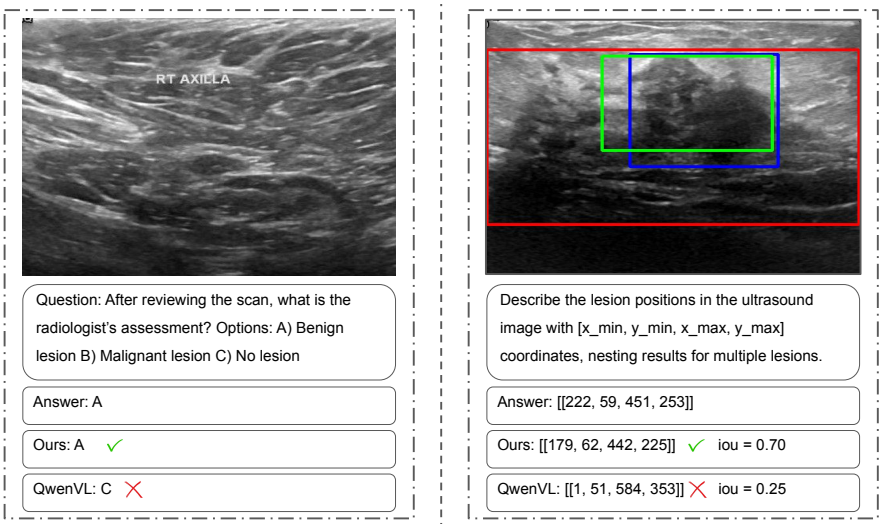

Fig. 4: **Qualitative examples on Ultrasound images.** Left: multiple-choice classification task. Right: lesion detection task where green is our detection, blueis ground truth, and red is QwenVL's [15] detection.

Acknowledgements This research is partially supported by the following NIH grants: R01-HL171376 and U01-CA268808. The authors of this paper declare that the proposed solution is fully automatic without any manual intervention. We thank all data owners and contributors for making the data publicly available and CodaLab [14] for hosting the challenge platform. We also thank AI VIETNAM for supporting the GPU resource for conducting the experiments.

Disclosure of Interests

The authors declare no competing interests.

References

1. Agarwal, R., Singh, A., Zhang, L.M., Bohnet, B., Rosias, L., Chan, S.C., Zhang, B., Faust, A., Larochelle, H.: Many-shot in-context learning. In: ICML 2024 Workshop on In-Context Learning (2024), https://openreview.net/forum?id=goi7DFHlqS 5
2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report (2025), https://arxiv.org/abs/2502.13923 2, 3

3. Chen, J., Gui, C., Ouyang, R., Gao, A., Chen, S., Chen, G.H., Wang, X., Zhang, R., Cai, Z., Ji, K., Yu, G., Wan, X., Wang, B.: Huatuogpt-vision, towards injecting medical visual knowledge into multimodal llms at scale (2024), https://arxiv.org/abs/2406.19280 1, 2, 4
4. Dettmers, T., Lewis, M., Shleifer, S., Zettlemoyer, L.: 8-bit optimizers via block-wise quantization. In: The Tenth International Conference on Learning Representations, ICLR (2022), https://openreview.net/forum?id=shpkpVXzo3h 5
5. Hamamci, I.E., Er, S., Almas, F., Simsek, A.G., Esirgun, S.N., Dogan, I., Dasdelen, M.F., Durugol, O.F., Wittmann, B., Amiranashvili, T., Simsar, E., Simsar, M., Erdemir, E.B., Alanbay, A., Sekuboyina, A., Lafci, B., Bluethgen, C., Ozdemir, M.K., Menze, B.: Developing generalist foundation models from a multimodal dataset for 3d computed tomography (2024), https://arxiv.org/abs/2403.17834 2
6. 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: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 28541–28564. Curran Associates, Inc. (2023) 1, 2
7. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS (2023) 4
8. Loshchilov, I., Hutter, F.: SGDR: stochastic gradient descent with warm restarts. In: 5th International Conference on Learning Representations, ICLR (2017), https://openreview.net/forum?id=Skq89Scxx 5
9. Ma, J., Gu, S., Wang, B.: Fast, low-resource, accurate, robust, and effectual medical image analysis (flare) 2025 (Mar 27, 2025), doi:10.5281/zenodo.15094799 1, 4, 6
10. Qin, L., Chen, Q., Fei, H., Chen, Z., Li, M., Che, W.: What factors affect multi-modal in-context learning? an in-depth exploration. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024), https://openreview.net/forum?id=REVdYKGcfb 5
11. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs (2021), https://arxiv.org/abs/2111.02114 4
12. Su, Y., Lan, T., Li, H., Xu, J., Wang, Y., Cai, D.: PandaGPT: One model to instruction-follow them all. In: Proceedings of the 1st Workshop on Taming Large Language Models: Controllability in the era of Interactive Assistants! pp. 11–23. Association for Computational Linguistics, Prague, Czech Republic (Sep 2023), https://aclanthology.org/2023.tllm-1.2/ 2
13. Xin, Y., Ates, G.C., Gong, K., Shao, W.: Med3dvlm: An efficient vision-language model for 3d medical image analysis. arXiv preprint arXiv:2503.20047 (2025) 2
14. Xu, Z., Escalera, S., Pavão, A., Richard, M., Tu, W.W., Yao, Q., Zhao, H., Guyon, I.: Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. Patterns **3**(7), 100543 (2022) 10
15. Yin, S.: Qwen2.5vl fine-tuned for flare 2025 medical image analysis (2025), https://huggingface.co/leoyinn/qwen2.5vl-flare2025 6, 7, 8, 10
16. Zhang, H., Li, X., Bing, L.: Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 543–553. Association for Computational Linguistics, Singapore (Dec 2023). https://doi.org/10.18653/v1/2023.emnlp-demo.49, https://aclanthology.org/2023.emnlp-demo.49/ 2

Table 6: Checklist Table. Please fill out this checklist table in the answer column.

Requirements	Answer
A meaningful title	Yes
The number of authors (≤6)	6
Author affiliations and ORCID	Yes
Corresponding author email is presented	Yes
Validation scores are presented in the abstract	Yes
Introduction includes at least three parts: background, related work, and motivation	Yes
A pipeline/network figure is provided	Figure 1, 2
Pre-processing (Training data)	Page 4
Strategies to improve model inference	Page 4
The dataset and evaluation metric section are presented	Page 5
Environment setting table is provided	Table 2
Training protocol table is provided	Page 5
Visualized example is provided	Figure 4
Limitation and future work are presented	Yes
Reference format is consistent.	Yes