Toward Clinician-in-the-Loop CT Reporting: An OHIF-Integrated Modular Framework for Retrospective Workflow Backtesting

Anonymized Authors

Anonymized Affiliations
email@anonymized.com

Abstract. We propose an Open Health Imaging Foundation (OHIF)-integrated modular framework for pre-clinical evaluation of interactive Computed Tomography (CT) reporting workflows through retrospective backtesting before clinician-in-the-loop studies. Recent vision-language models show promise in CT report generation and visual question answering, but are typically assessed with offline benchmarks or one-shot generation, which reveals little about behavior during iterative clinical interactions where users ask follow-up questions, refine findings, and update reports over multiple turns. Direct radiologist evaluation of immature systems is also costly, hard to scale, and may expose users to unstable workflows. The framework we proposed provides standardized interfaces for agents, tools, interaction drivers, orchestration, and structured trace logging, supporting interchangeable workflow designs, automated pseudo-user backtesting, and detailed execution auditing. We demonstrate it with two workflows: QA-first, which accumulates visual question answering before report synthesis, and draft-first, which starts from an initial draft followed by question-driven refinement. On 200 retrospective 3D CT studies, both ran end-to-end within the same infrastructure with trace-level auditability: QA-first reached a turn-level success rate of 87.9% at 92.7 s latency, and draft-first 88.8% at 118.1 s. While these workflow comparisons hold only under the reference-guided backtesting condition, the framework itself enables systematic workflow comparison, failure analysis, and evaluation before involving clinicians.

Keywords: Interactive CT reporting · Modular VLM workflow · OHIF · Backtesting · Clinician-in-the-loop

1 Introduction

Recent radiology AI has moved from single-task Computed Tomography (CT) classification or detection [1] toward multimodal report generation, Visual Question Answering (VQA), and interactive assistance [25]. Medical Vision-Language Models (VLMs) have been applied to radiology report generation and medical VQA, including emerging work on 3D CT reporting [10,3,14,16]. Such capabilities are also beginning to appear in commercial radiology reporting assistants

(e.g., Nuance PowerScribe Copilot and Rad AI) [9,21]. Yet most medical VLMs are still evaluated outside the clinical viewing workflow, either as one-shot report generators or as standalone VQA systems. Such evaluation is useful for benchmarking model capability, but it does not capture the iterative nature of clinical reporting, where radiologists inspect images, ask follow-up questions, revise findings, and refine impressions over multiple turns.

This limitation matters because clinician-in-the-loop AI depends not only on model accuracy, but also on task design, user interaction, and workflow integration [8,20,22]. Directly moving from offline benchmarking to radiologist evaluation is premature when system behavior is still poorly characterized: clinician studies are costly, difficult to scale, often require institutional review, and may expose users to unstable or error-prone systems. We therefore study *backtesting*: retrospective simulation of interactive reporting against archived cases using an automated driver. Backtesting provides an intermediate evaluation layer for exercising workflows before involving clinicians, comparing candidate designs reproducibly, and tracing failures to specific modules or steps.

The key design principle is to separate reusable infrastructure from concrete workflow instantiations: agents, tools, and pseudo-users are replaceable implementations rather than framework-defining components. We demonstrate this principle with two example workflows, QA-first and draft-first, used only as test cases to show that the same OHIF-integrated core can execute, compare, and audit different interactive reporting designs before involving clinicians.

Our contributions are:

- We design a modular, OHIF-integrated framework for pre-clinical evaluation of interactive CT reporting inside the image-review environment, separating reusable infrastructure from workflow-specific implementations so that reporting, VQA, and synthesis agents, tools, and pseudo-users are replaceable without changing the core.
- We propose a retrospective backtesting methodology in which an automated pseudo-user exercises interactive reporting workflows before radiologist-in-the-loop evaluation, instrumented by a structured trace schema that records user turns, module selection, tool calls, intermediate outputs, latency, failures, module identifiers, and report versions for auditability and comparison.
- We demonstrate the framework through QA-first and draft-first instantiations, showing that different interactive reporting designs can be implemented and compared within the same core.

2 Related Work

Generic agent frameworks such as LangGraph [11], LlamaIndex Workflows [15], and AutoGen [23] orchestrate LLM and tool calls but ship no viewer integration, leaving the clinical image-review context to be built separately; conversely, OHIF-integrated platforms such as MONAI Label [7] embed AI in the viewer but target segmentation and annotation rather than interactive reporting. Neither addresses pre-clinical evaluation of a reporting workflow—exercising it with

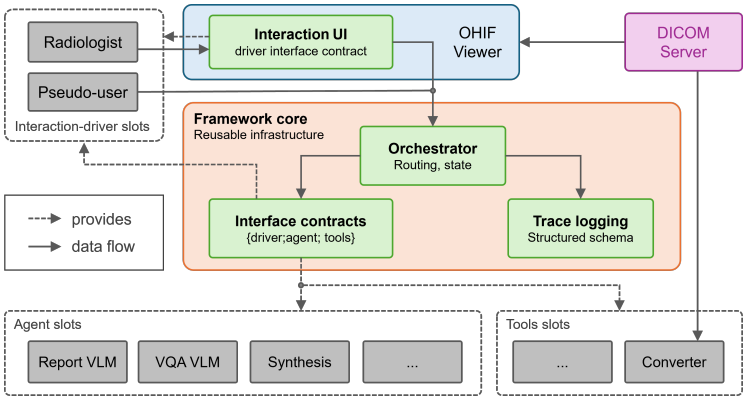

Fig. 1. Framework core and module slots. The core contains interface contracts, orchestration, interaction infrastructure, and structured trace logging. Pseudo-user, VLMs, and conversion components are replaceable; the orchestrator belongs to the core, with only its configuration adjustable.

a simulated user before radiologists are involved, and logging execution in units by which a reporting task, rather than a generic agent run, can be judged. Closest to our setting, OHIF-AI [4] embeds VLM-based report generation directly in the OHIF viewer, confirming the practicality of in-viewer reporting. It is, however, an operational tool for single-configuration, live report generation; it provides neither an automated pre-clinical evaluation layer nor a structured trace schema for per-turn quality trajectories and failure attribution, which are the focus of our framework.

3 Framework Design

3.1 Framework Core: Interfaces, Orchestration, and Logging

The framework separates *infrastructure* from *workflow instantiation*. The core does not prescribe a report agent, QA agent, synthesis agent, image-conversion tool, pseudo-user, or workflow order. It defines module interface contracts and the infrastructure that invokes and records their implementations (Fig. 1).

The *agent interface* is satisfied by any reasoning module that produces outputs based on the current workflow context and influences subsequent steps. The *tool interface* is satisfied by any module that performs a specific operation in response to input without determining workflow progression. The distinction between agents and tools lies in their role in the workflow rather than their input or output types. The *interaction-driver interface* is satisfied by any entity that supplies turns and consumes responses.

The *orchestrator* coordinates requests, routes calls to agents and tools, and manages workflow state. In this study, it is implemented as an LLM-based con-

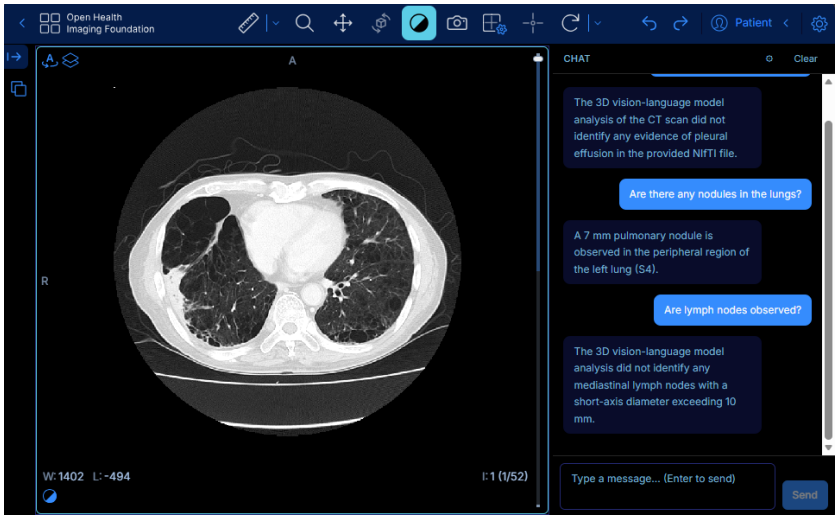

Fig. 2. OHIF chat extension beside the CT viewer. The same interaction endpoint can support retrospective backtesting or future radiologist-driven use.

troller whose model endpoint, prompt, and routing policy are configurable. All module invocations are coordinated centrally through the orchestrator.

Structured trace logging records workflow execution. Instead of free-text logs, we define a schema in which each trace links user turns, module selections, tool calls, intermediate outputs, latency, error categories, and report versions. Triggered at each module invocation through the orchestrator, it yields consistent, comparable traces across workflow instantiations, supporting auditability, failure analysis, and workflow comparison.

3.2 OHIF Interaction Layer

The user-facing layer is implemented as a custom chat panel embedded within the OHIF viewer [27] (Fig. 2). This *interaction UI* serves as an interface layer bridging real user input with the framework core, allowing users to submit natural language queries and receive textual responses through a text-based chat interface. During backtesting, an automated pseudo-user sends turns through the same interaction-driver contract; during future clinical evaluation, a radiologist supplies turns via this chat panel. Both are concrete drivers behind one contract: the pseudo-user stands in for a radiologist operating the UI.

4 Workflow Instantiation and Experimental Setup

4.1 Backtesting as a Pre-Clinical Evaluation Layer

We position retrospective workflow backtesting as an intermediate evaluation layer between offline model benchmarking and future clinician-in-the-loop eval-

uation. The present study focuses on the pseudo-user backtesting stage, where candidate workflows are executed, compared, and audited for report-quality trajectories, routing stability, tool failures, latency, and traceability before radiologists are involved. Future clinician-in-the-loop evaluation can replace the pseudo-user with a radiologist driver while reusing the rest of the system; thus, moving from simulation to clinician evaluation requires a driver swap rather than a redesign.

4.2 Retrospective Backtesting Setup

In this study, the pre-clinical backtesting layer is instantiated as a retrospective workflow-level evaluation over a set of CT cases. For each configured workflow, the backtesting procedure executes case-level interactions across the same retrospective dataset and records report trajectories, metric trajectories, structured workflow traces, system-level failures, and latency measurements.

We backtest the framework on 200 retrospective 3D CT studies paired with radiologist-written reference reports from an internal institutional dataset. These 200 studies form a held-out test set, partitioned at the patient level from the approximately 20,000 CT studies used for model development, so that no patient in the backtesting set appears in the training or validation data. This retrospective study was approved by the institutional review board of the participating institutions. All studies were de-identified before use.

4.3 Component Integration

The interaction driver is a prompt-based pseudo-user backed by Qwen3-14B [24], in line with recent LLM-based user simulators [5,26]. Given a reference report, it is prompted to generate case-specific questions while avoiding direct answer provision, correction of system outputs, or explicit injection of reference-report findings into the interaction. Its role is to drive controlled retrospective interactions, not to stand in for radiologist judgment. The pseudo-user communicates with the framework through the same interaction-driver contract that can later be used by alternative pseudo-users, reference-free drivers, scripted tests, or radiologists using OHIF.

For this instantiation, three agent components are connected to the framework through the standardized agent interface: a report-generation VLM, a VQA VLM, and a synthesis agent. Our report-generation model draws on SIGMA [17] but is re-built for a single-volume setting: it takes one volumetric input and produces a report, whereas the original SIGMA consumes two input volumes. The VQA agent follows the same design.

Final report synthesis is handled by Qwen3-14B [24] through a dedicated prompt that integrates intermediate workflow outputs into a final report. This component is included to instantiate a realistic multi-stage reporting workflow, rather than to claim an optimized report-generation pipeline or clinical validity of the generated reports.

One tool component is connected in this instantiation: a DICOM-to-NIfTI converter based on the `dcm2niix` [12]. It transforms each input DICOM study into a temporary volumetric `.nii.gz` file, which is then read by the report-generation and VQA agents as their volumetric input.

The orchestrator remains part of the framework core rather than a connected module. In the present instantiation, it is configured to use Qwen3-14B [24] as its LLM backend, with a specified model endpoint, prompt, and routing policy. These configuration choices can be changed without modifying the module interfaces, workflow trace schema, or OHIF interaction endpoint.

All agents were run on self-hosted endpoints, each using two NVIDIA H100 NVL GPUs, with components sharing the same underlying model served through a common host. Models were served at native precision without quantization, using automatic batching.

4.4 Example Workflows

We instantiate two workflows to demonstrate configurability rather than to prescribe a clinical interaction design. Each organizes user turns, model responses, and final report synthesis differently; in both, the pseudo-user described above operates under a fixed budget of five turns.

1. **QA-first**: the interaction begins with user questions. Each question is routed to the VQA agent, and the resulting answers are accumulated as workflow context. After the interaction budget, a final-report request triggers the synthesis agent to integrate the accumulated QA context into the final report.
2. **Draft-first**: the workflow first generates an initial report draft. The user then asks follow-up questions, which are routed to the VQA agent and added to the workflow context. After the interaction budget, the synthesis agent integrates the initial draft and accumulated QA context into the final report.

Both workflows run on the same framework core through configuration and component integration. This shared execution path, rather than either workflow itself, is the evidence of workflow interchangeability.

4.5 Evaluation Metrics and Validation Scope

Evaluation probes framework behaviour as well as output quality. Report quality is measured with GREEN [18], RadGraph-F1 [6], BLEU [19], METEOR [2], and ROUGE-L [13]. Feasibility is measured with latency and success/failure rate. Traceability is assessed by inspecting routing decisions, intermediate outputs, tool calls and system-level error categories. Content-level clinical error labels are not automatically produced; they are reserved for future expert annotation.

This setup cannot measure usability, trust, cognitive load, true reporting time, or real radiologist behaviour. Results should therefore be interpreted as pre-clinical workflow evidence, not evidence of clinical utility. The value of the proposed evaluation layer is that it can identify unstable workflows, routing failures, tool failures, latency bottlenecks, and report-quality degradation before radiologists are involved.

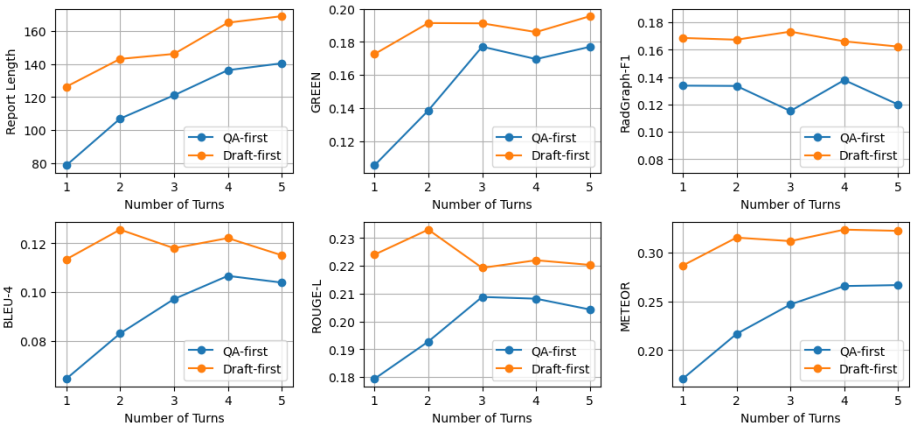

Fig. 3. Report-quality metrics as a function of conversation turn, measuring workflow trajectory rather than only a final report.

Table 1. Workflow-level reliability and latency decomposition during retrospective backtesting. Success and failure rates are reported at the turn level over all interaction turns. The other column aggregates the remaining traced runtime (e.g., communication, conversion, and preprocessing).

Workflow	Success (%)	Failure (%)		Latency (s)				
		Agent	Route	Draft	QA	Synthesis	Other	Total
QA-first	87.9	5.9	6.2	–	79.7	9.0	4.0	92.7
Draft-first	88.8	5.6	5.6	34.1	69.2	10.9	3.9	118.1

5 Results

Execution Feasibility and Traceability: The framework supports end-to-end viewer-integrated backtesting. The OHIF extension runs, the backend executes the concrete modules bound for this instantiation, both example workflows run on the same core, and complete workflow traces are exported under the structured schema. Trace records localise each step from user query to selected module, tool call, intermediate output, response, report version, latency, and system-level error category.

Report Quality Dynamics: Absolute report quality remains low, confirming that the tested VLMs are not suitable for autonomous CT reporting. The framework nevertheless exposes quality dynamics rather than only a final score: as shown in Fig. 3, metrics can be tracked across turns, initial-versus-final deltas can be measured, and cases can be separated into improved and worsened trajectories. This is the intended use of backtesting: not to validate clinical utility, but to reveal how an interactive workflow behaves before involving clinicians.

Execution Reliability and Latency: Table 1 summarizes workflow-level reliability and latency during retrospective backtesting. Both workflows completed most runs, with draft-first slightly more reliable than QA-first. However, draft-first also incurred higher latency, mainly due to the additional draft-generation step and longer synthesis stage. No timeout failures occurred. These results highlight a trade-off between report-quality gains and runtime cost to consider before clinician-in-the-loop evaluation.

6 Discussion

This distinction matters when interpreting the workflow comparison. Draft-first outperforming QA-first in simulation should be read as a backtest-generated hypothesis, not a clinical recommendation. It depends on the current VLMs, prompts, dataset and pseudo-user. Moreover, because the pseudo-user is reference-guided, reference information may leak into the generated questions and affect the two workflows unequally, so the draft-first-versus-QA-first comparison holds only under this reference-guided condition. By contrast, the framework claims—measurement, comparability, traceability, feasibility assessment, and pseudo-user-to-radiologist replacement—do not depend on the tested models.

The framework also opens a pathway to clinician-verified continual learning: once radiologists drive the same OHIF interface, captured traces of questions, edits, accepted or rejected suggestions, and signed reports could support failure mining and validated model updates. No such learning is performed here.

Limitations are substantial. No real radiologists are included, so the study cannot claim improved workflow, usability, trust, cognitive load, or reporting time. The pseudo-user is reference-guided and may overestimate real interaction benefits. Generated reports are analysis drafts, not clinically usable outputs. Structured trace logging captures system-level failures automatically, but content-level clinical failures such as omission, hallucination, wrong localisation, and severity mismatch require expert annotation or reference-based analysis. Future use of clinician traces for learning will require de-identification, access control, expert verification, validation, auditability, and rollback.

7 Conclusion

We presented an OHIF-integrated modular framework for interactive CT reporting workflows and used QA-first and draft-first as one backtesting instantiation. On 200 retrospective 3D CT studies, the same framework core executed both interaction designs, measured report-quality trajectories, recorded latency and system-level failures, and produced trace-level audit records. Although the tested models remain insufficient for autonomous reporting, the framework bridges retrospective simulation and future clinician-in-the-loop evaluation.

Next steps are clinician-in-the-loop evaluation with radiologists on the same OHIF endpoint, and extending the trace schema with expert-verified error labels to support a continual-learning loop with roll-back-capable model updates.

References

1. Ahmad, I.S., Dai, J., Xie, Y., Liang, X.: Deep learning models for CT image classification: a comprehensive literature review. Quantitative Imaging in Medicine and Surgery **15**(1) (2025)
2. Banerjee, S., Lavie, A.: METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. pp. 65–72 (2005)
3. Chen, H., Zhao, W., Li, Y., Zhong, T., Wang, Y., Shang, Y., Guo, L., Han, J., Liu, T., Liu, J., et al.: 3D-CT-GPT: Generating 3D radiology reports through integration of large vision-language models. arXiv preprint arXiv:2409.19330 (2024)
4. Cho, J., Rastogi, A., Liu, J., Schlamp, K., Vollmuth, P.: OHIF-SAM2: Accelerating radiology workflows with meta segment anything model 2. In: 2025 IEEE 22nd International Symposium on Biomedical Imaging (ISBI). pp. 1–5 (2025)
5. Davidson, S., Romeo, S., Shu, R., Gung, J., Gupta, A., Mansour, S., Zhang, Y.: User simulation with large language models for evaluating task-oriented dialogue. arXiv preprint arXiv:2309.13233 (2023)
6. Delbrouck, J.B., Chambon, P., Bluethgen, C., Tsai, E., Almusa, O., Langlotz, C.: Improving the factual correctness of radiology report generation with semantic rewards. In: Findings of the Association for Computational Linguistics: EMNLP 2022. pp. 4348–4360 (2022)
7. Diaz-Pinto, A., Alle, S., Nath, V., Tang, Y., Ihsani, A., Asad, M., et al.: MONAI label: A framework for AI-assisted interactive labeling of 3D medical images. Medical Image Analysis **95**, 103207 (2024)
8. Fogliato, R., Chappidi, S., Lungren, M., Fisher, P., Wilson, D., Fitzke, M., Parkinson, M., Horvitz, E., Inkpen, K., Nushi, B.: Who goes first? influences of human-ai workflow on decision making in clinical imaging. In: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency. pp. 1362–1374 (2022)
9. Friebe, M.: Ai in radiology and interventions: a structured narrative review of workflow automation, accuracy, and efficiency gains of today and what's coming. International Journal of Computer Assisted Radiology and Surgery **21**(1), 1–10 (2026)
10. Hamamci, I.E., Er, S., Menze, B.: CT2Rep: Automated radiology report generation for 3D medical imaging. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 476–486. Springer (2024)
11. LangChain: LangGraph. https://github.com/langchain-ai/langgraph, accessed: 2026-04-22
12. Li, X., Morgan, P.S., Ashburner, J., Smith, J., Rorden, C.: The first step for neuroimaging data analysis: DICOM to NIfTI conversion. Journal of neuroscience methods **264**, 47–56 (2016)
13. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text summarization branches out. pp. 74–81 (2004)
14. Liu, C., Wan, Z., Wang, Y., Shen, H., Wang, H., Zheng, K., Zhang, M., Arcucci, R.: Argus: benchmarking and enhancing vision-language models for 3D radiology report generation. In: Findings of the Association for Computational Linguistics: ACL 2025. pp. 16448–16460 (2025)
15. LlamaIndex: LlamaIndex workflows. https://docs.llamaindex.ai/, accessed: 2026-03-12

16. Mao, Y., Xu, W., Qin, Y., Gao, Y.: CT-agent: A multimodal-llm agent for 3D CT radiology question answering. Science China Information Sciences **69**(5), 150107 (2026)
17. Nguyen, K.C., Wang, C., Shi, Z.X., Heng, Y., Qi, C.Y., Oda, M., Mori, K.: SIGMA: Auto-regressive VLM for automated radiology report generation from longitudinal 3D CT volumes. In: International Workshop on Emerging LLM/LMM Applications in Medical Imaging (ELAMI). pp. 53–61 (2025)
18. Ostmeier, S., Xu, J., Chen, Z., Varma, M., Blankemeier, L., Bluethgen, C., Md, A.E.M., Moseley, M., Langlotz, C., Chaudhari, A.S., Delbrouck, J.B.: GREEN: Generative radiology report evaluation and error notation. In: Findings of the association for computational linguistics: EMNLP 2024. pp. 374–390 (2024)
19. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
20. Perivolaris, A., Adams-McGavin, C., Madan, Y., Kishibe, T., Antoniou, T., Mamdani, M., Jung, J.J.: Quality of interaction between clinicians and artificial intelligence systems: A systematic review. Future Healthcare Journal **11**(3), 100172 (2024)
21. Rad AI: Rad ai reporting. https://www.radai.com/, accessed: 2026-06-15
22. Tulk Jesso, S., Kelliher, A., Sanghavi, H., Martin, T., Henrickson Parker, S.: Inclusion of clinicians in the development and evaluation of clinical artificial intelligence tools: A systematic literature review. Frontiers in Psychology **13**, 830345 (2022)
23. Wu, Q., Bansal, G., Zhang, J., Wu, Y., Zhang, S., Zhu, E., Li, B., Jiang, L., Zhang, X., Wang, C.: AutoGen: Enabling next-gen LLM applications via multi-agent conversation framework. arXiv preprint arXiv:2308.08155 (2023)
24. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., Qiu, Z.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025)
25. Yi, Z., Xiao, T., Albert, M.V.: A survey on multimodal large language models in radiology for report generation and visual question answering. Information **16**(2), 136 (2025)
26. Yoon, S.e., He, Z., Echterhoff, J.M., McAuley, J.: Evaluating large language models as generative user simulators for conversational recommendation. arXiv preprint arXiv:2403.09738 (2024)
27. Ziegler, E., Urban, T., Brown, D., Petts, J., Pieper, S.D., Lewis, R., Hafey, C., Harris, G.J.: Open health imaging foundation viewer: an extensible open-source framework for building web-based imaging applications to support cancer research. JCO clinical cancer informatics **4**, 336–345 (2020)