How Well Can Modern LLMs Act as Agent Cores in Radiology Environments?

Anonymous ACL submission

Abstract

Large language models (LLMs) hold promise for building accurate and interpretable agentic systems in complex domains like radiology. To evaluate whether modern LLMs can serve as agent cores in radiology settings, we introduce **RadA-BenchPlat**, a comprehensive platform built on 2,200 patient records spanning 6 anatomical regions, 5 imaging modalities, and 2,200 diseases. The dataset includes 24,200 QA pairs and 10 tool categories for radiology task-solving. Our benchmarking of 7 leading LLMs reveals significant gaps: while models such as Claude-3.7-Sonnet achieve 67.1% task completion in routine scenarios, they struggle with complex reasoning and tool coordination. We then apply prompt engineering strategies, yielding an overall 48.2% performance gain ($p < 0.001$) on complex tasks-with **prompt backpropagation** and **multi-agent collaboration** contributing 16.8% ($p < 0.01$) and 30.7% ($p < 0.001$) improvements, respectively. We further enhance robustness via automated tool building, reaching 65.4% success. Our work provides critical benchmarks and actionable strategies for developing reliable radiology AI agents, moving closer to fully automated clinical applications. Code and data are prepared and will be available upon publication.

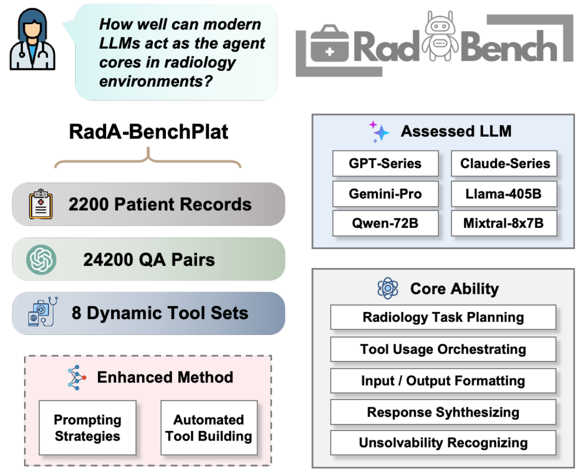

Figure 1: **Benchmark Overview.** The motivation and dataset statistics of the benchmark & Architectural components are integrated into the evaluation platform.

1 Introduction

Recent advancements in Large Language Models (LLMs) have revolutionized domains ranging from natural language processing to computer vision (Wu et al., 2023; Cui et al., 2024a; Liu et al., 2024). Particularly noteworthy is the emergence of LLM-powered agent systems (Tang et al.; Jin et al., 2024; Qin et al., 2023; Cui et al., 2024b), which orchestrate complex task planning and dynamically invoke specialized tools to deliver robust analyses (Mehandru et al., 2024). By positioning the LLM as a central reasoning core capable of integrating external resources, these systems find a compelling yet challenging testbed in radiology, a field demanding accurate interpretation of complex medical images and detailed textual reporting (Rao et al., 2025; Zhao et al., 2023; Zhou et al., 2025; Lei et al., 2024). While recent efforts have introduced “radiology generalist” models designed to handle diverse imaging tasks within a single framework (Wu et al., 2023; Li et al., 2023; Liu et al., 2023; Zhang et al., 2024, 2023), these monolithic models often struggle with the full spectrum of radiological complexity, which spans imaging modalities, anatomical regions, and evolving diagnostic protocols.

Motivated by the necessity for radiological agents to collaborate with external environments, this paper investigates the feasibility of developing an agent-based system capable of handling varied imaging tasks. Specifically, we explore how effectively current LLMs, serving as an agentic core, can understand professional medical contexts, translate diverse clinical queries into executable plans, and sequentially invoke appropriate tools to complete complex radiological workflows.

To systematically explore this question, we introduce **RadA-BenchPlat**, a comprehensive resource for evaluating LLM-based agents in radiology (Figure 1). The platform is built on three key components: **First**, we construct a dataset of 2,200 synthetic patient records, spanning 6 anatomical regions and 5 imaging modalities, all verified as clinically reasonable by a senior radiologist. **Second**, for each record, we generate 11 diverse radiology tasks using GPT-4, resulting in a total of 24,200 QA pairs. **Third**, we design a dynamic toolset simulation strategy to mimic real-world cross-center variations. This involves dynamically combining or excluding 10 core tool categories to create 8 distinct evaluation environments, such as redundant or insufficient tool sets.

Based on the platform, we evaluate 8 leading LLMs as agentic cores, including the GPT-series (OpenAI, Accessed by October 12, 2024), Claude-series (Anthropic, Accessed by October 12, 2024), Gemini Pro (Team et al., 2023), LLaMA (Touvron et al., 2023), Mixtral (Jiang et al., 2024), and Qwen (Bai et al., 2023). We focus on 5 key competencies: *planning tasks*, *orchestrating tool usage*, *formatting inputs/outputs*, *synthesizing responses*, and *recognizing unsolvable cases*. Our results reveal a significant gap between current LLM capabilities and real-world demands. While models perform reasonably on basic tasks like diagnosis, they struggle with integrative tasks; for instance, Claude 3.7 achieves a 68.3% completion rate in diagnosis but only 30.1% in comprehensive report generation.

Building on these findings, we briefly investigate four prompting strategies—*back-propagation*, *self-reflection*, *few-shot learning*, and *multi-agent collaboration*—which collectively drive a 19.8% overall improvement in task completion and a substantial 48.2% gain ($p < 0.001$) in complex scenarios. Beyond these prompt-level optimizations, we further explore the potential for self-evolution in radiology agents, inspired by recent advances in automated tool building (AutoTB) (Feng et al., 2025; Wölflein et al., 2025; Cai et al., 2023). We design an enhancement pipeline that integrates the agentic system with external AutoTB modules, enabling the dynamic creation of tools when critical components are missing. This combined system achieves an additional 65.4% success rate in previously unsolvable tasks, underscoring the vast potential of fully autonomous, self-evolving agent-based healthcare systems.

2 Methods

2.1 Building RadA-BenchPlat

Patient Records. A "patient record" here refers to a radiology-centric medical record that includes patient demographics, medical history, annotated imaging data (with anatomical and pathological details), and extensive clinical findings. We categorize patient records by **anatomy, modality, and disease**. This taxonomy covers 22 common anatomy-modality combinations derived from 6 anatomical regions and 5 imaging modalities:

Head & Neck: {*X-ray, CT, MRI, US*},
Chest: {*X-ray, CT, MRI, US*},
Limb: {*X-ray, CT, MRI, US*},
Abd. & Pelv.: {*X-ray, CT, MRI, US*},
Spine: {*X-ray, CT, MRI*},
Breast: {*Mammo., MRI, US*}.

We select 100 common diseases for each of the 22 anatomymodality combinations, resulting in a total of 2,200 patient records (22 $\times$ 100). Detailed procedures for record synthesis, expert verification, and a sample patient record are provided in the Appendix. A radiologist with over 10 years of experience manually reviewed the records, confirming that 97.32% met information validity standards and 96.73% were clinically consistent. Figure 2a present the demographic distributions across sex. While the dataset is generally sex-balanced, breast-related cases are predominantly female, with 99% of such records corresponding to women. Analysis of these cases also displays an age concentration between 45 and 65 years, aligning with the broader middle-aged range of 35 to 70 years. Normal distributions in height and weight further support the datasets authenticity and diversity, establishing a robust foundation for evaluating LLM agents in complex radiology workflows.

Task-related QA pairs. To comprehensively evaluate LLM-based agents in radiology, we propose a taxonomy of 11 distinct tasks: (a) organ segmentation, (b) anomaly detection, (c) standard end-to-end diagnosis, (d) organ and anomaly joint grounding, (e) diagnosis with grounding clues, (f) organ-wise biomarker calculation, (g) anomaly-wise biomarker calculation, (h) standard report generation, (i) report generation focused on specific biomarkers, (j) report generation focused on both biomarkers and indicators[1], and (k) detailed treatment planning.

[1] refers to a specific quantification of health status or disorder grading, e.g. CURB-65, tumor grading

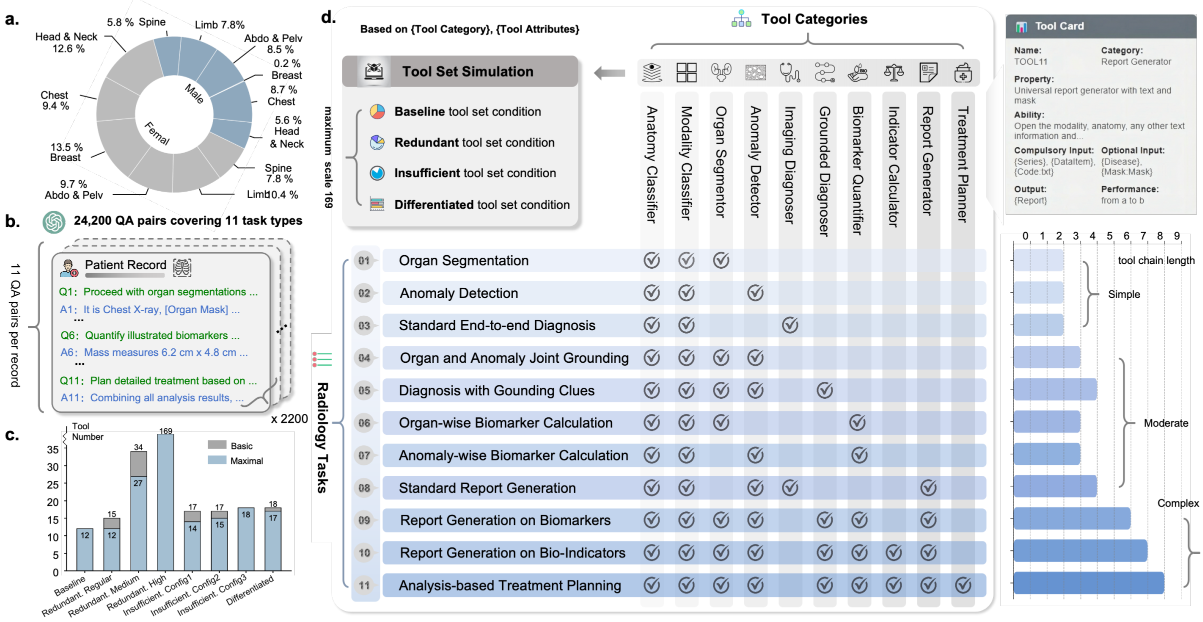

Figure 2: **Statistics in RadA-BenchPlat. a.** Distribution of patient sex across different anatomical regions. **b.** Ten categories of tools and eleven types of radiology tasks, with checkmarks indicating tools required for task completion. Tool sets are simulated through various tool attribute combinations. **c.** 24,200 generated QA-pairs based on radiology task types using GPT-4. **d.** Number of tools across eight simulated tool set conditions. **f.** Maximum token length of the open-source and closed-source LLMs used in this study.

As illustrated in Figure 2d, we categorize these tasks into three complexity levels‘Simple’, ‘Moderate’, and ‘Complex’based on the number of reasoning steps: fewer than 4, between 4 and 6, and 6 or more, respectively. For instance, organ segmentation is relatively simple, involving anatomy and modality classification followed by segmentation. In contrast, detailed treatment planning is complex, requiring multi-step reasoning including diagnosis, quantification, report synthesis, *etc.* We further simulate each of the 11 tasks across 2,200 patient records (Figure 2b), generating 24,200 QA pairs (2,200 $\times$ 11) using a prompt template. For example, a task may prompt “What disease can be inferred?” or “Please write a radiologic report for the image”. The QA pairs are evenly distributed across tasks to ensure balanced and diverse representation, and all undergo manual verification.

Radiology Tool Sets. To simulate the real-world clinical settings, we define 10 high-level radiology tool categories (detailed in the supplementary): Anatomy Classifier (AC), Modality Classifier (MC), Organ Segmentor (OS), Anomaly Detector (AD), Imaging Diagnoser (ID), Grounded Diagnoser (GD), Biomarker Quantifier (BQ), Indicator Calculator (IC), Report Generator (RG), and Treatment Planner (TP). A specific tool is denoted by a “tool card” (Figure 2d), detailing its category, properties, capabilities, required/optional inputs, and performance, with values sampled from clinically plausible ranges. For example, a tool may be described as an “anatomy classifier” with a performance score. This design allows for infinite tool variations, enabling diverse evaluation scenarios.

To reflect varied clinical environments, we define four evaluation tool set conditions: **“Baseline”**, **“Redundant”**, **“Insufficient”**, and **“Differentiated”** (Figure 2d). For each patient record and task query, the **“Baseline”** condition provides a minimal, solvable tool set of 12 tools across 10 categories, ensuring both biomarker quantifier and indicator evaluator categories always include two tools (covering organ evaluation and anomaly assessment). The **“Redundant”** condition introduces extra, potentially irrelevant tools at three redundancy levels: regular (1215 tools, with at most one extra per category), medium (2734 tools, adding 23 extras per category), and high (a comprehensive fixed set of 169 tools), testing the agents ability to filter relevant tools in noisy settings. For **“Insufficient”** conditions, the tool set is unsolvable due to missing essentials in three con-

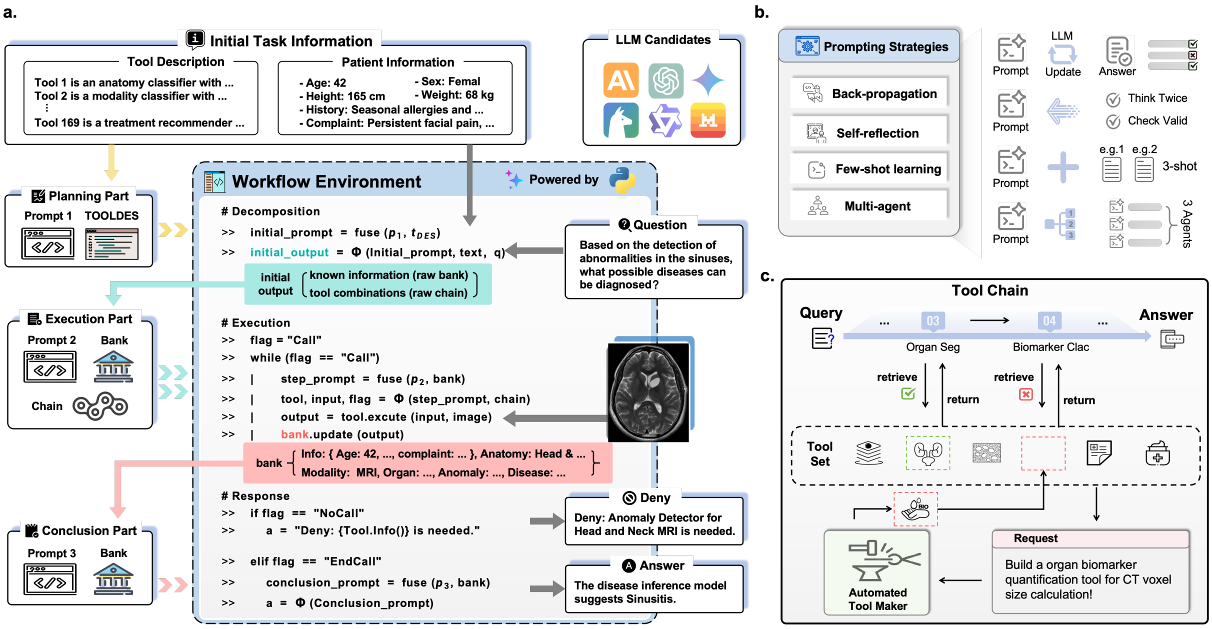

Figure 3: **Detailed agentic workflow and extended methods. a.** Three-phase architecture consisting of: initialization phase, circulation phase and conclusion phase. **b.** Four extended prompt engineering methods for performance improvement. **c.** Automated tool building process implemented when no suitable tool is available.

figurations: config1 omits an entire tool category (1417 tools), config2 has modality or anatomy mismatches (1517 tools), and config3 lacks necessary capabilities (fixed at 18 tools), with increasing recognition difficulty for mismatched tools. Finally, **"Differentiated"** provides overlapping tools (1718 tools) with distinct performance levels, requiring agent to choose the most accurate tool.

2.2 Benchmarking LLMs as Agent Core

Main Workflow Formulation. The LLM-based agent system consists of three main components (Figure 3a): an LLM agent core (Φ), a set of specialized tools ($\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$), and a memory bank ($\mathcal{B}$). Given a clinical query $\mathcal{Q}$ (e.g., *"Please make a diagnosis based on the patient's information"*) and patient history, the workflow proceeds as follows: The agent core (Φ) first parses $\mathcal{Q}$ and decomposes it into sub-tasks based on the available tools in $\mathcal{T}$, initializing $\mathcal{B}$ with relevant patient data. For each sub-task, the core selects and executes the appropriate tool from $\mathcal{T}$, storing outputs in $\mathcal{B}$ to iteratively refine the process. Finally, the agent core synthesizes all information in $\mathcal{B}$ to generate a coherent response that addresses the clinical query.

2.3 Leveraging Enhanced Methods

To address the limited performance of LLMs on complex tasks (completion rate: 27.4%), we introduce four prompting enhancement strategies (Figure 3b) to boost automation and reliability. In this work, we employ **Prompt Back-propagation** by iteratively refining prompts through three rounds of automated feedback from Claude-3.7-Sonnet combined with manual adjustments based on 220 QA pairs under "Redundant" and "Insufficient" tool set conditions, after which the refined prompt is applied across the full benchmark. In parallel, **Self-reflection** prompts the agent to verify input correctness before each tool invocation, a process supported by an automated verifier to reduce execution errors. Additionally, **Few-shot Learning** is used to provide exemplar workflows and tool templates that guide planning and standardize execution, thereby minimizing misunderstandings and formatting issues. Moreover, **Multi-agent Collaboration** restructures the single-agent setup into a three-agent systemcomprising a planner for task decomposition, an executor for tool usage and I/O management, and a summarizer for final response synthesisall leveraging Claude-3.7-Sonnet.

In "Insufficient" tool set conditions, tasks become unsolvable due to missing essential tools. Leveraging advances in automated machine learning, our systembased on the M^{3}Builder (Feng et al., 2025) framework and enhanced with prompting strategiescan now detect missing tools and generate instructions for dynamic tool con-

Table 1: Target Hit Rates across models under different conditions. Note: Llama-3.1 refers to the 405B version, Mixtral to 8x7B, and Qwen-2.5 to 72B. (**Claude-3.7-sonnet** achieves peak performance in Regular condition).

Condition	**GPT-4-Turbo**	**GPT-4o**	**Gemini-1.5**	**Claude-3.5**	**Claude-3.7**	**Llama-3.1**	**Mixtral**	**Qwen-2.5**
Regular	0.286	0.481	0.403	0.519	**0.671**	0.428	0.056	0.198
Medium	0.217	0.429	0.390	**0.450**	0.428	0.387	0.000	0.162
High	0.124	0.176	0.085	0.231	**0.292**	0.093	0.000	0.020

struction (Figure 3c). For instance, if a biomarker quantification tool is absent during lung nodule analysis on chest X-rays, the agent can request a model for biomarker size calculation, triggering automated tool building.

3 Results

We perform a quantitative analysis that includes performance on task completion and agentic abilities for RadAgent in radiology tasks.

3.1 Main Observations

Observation 1: Decreasing Task Completion Performance from Simple to Complex Tasks

While evaluating LLMs for decomposing radiological tasks and effectively utilize medical tools (Table. 1), task completion rate is used as primary metric, which is defined as successfully solving a query by appropriately utilizing available tools without incurring any intermediate errors during response synthesis. Claude-3.7-Sonnet outperforms GPT-3.5 and GPT-4o, achieving a 67.1% success rate on routine tasks. However, performance drops to 42.8% on advanced tasks and plunges to 29.2% on complex ones, highlighting the increasing difficulty of tasks requiring multi-step reasoning and coordination. A detailed evaluation of the LLMs' specific capabilities as agent cores is provided in the subsequent sections.

Observation 2: Acceptable Planning Ability while Frequent Offset During Execution

To evaluate task planning ability and consistency, we assess tool chain generation under "Redundant" conditions using three metrics: Levenshtein Distance (LD, edit distance between generated and ground-truth tool chains), False Discovery Rate (FDR, proportion of incorrectly included tools), and Tool Matching Accuracy (TMA, step-wise alignment with ground truth). As shown in Table. 2, our findings are: **(1)** closed-source models generally outperform open-source models (except Llama-3.1-405B), achieving LD < 1.4 despite an average chain length of 5; **(2)** differences between planned and executed chains emerge during multi-iteration execution, with Claude-3.5 and Llama-3.1 showing improved convergence, while others show stable or increased deviations; **(3)** GPT-4o achieves the lowest FDR across all "Redundant" tool set conditions, indicating lower tool selection redundancy; **(4)** GPT-4o and Claude-3.5 yield higher TMA, reflecting better step-wise alignment with ground truth; **(5)** as tool complexity increases, all LLMs exhibit significant performance declines, especially open-source models.

Table 2: Performance metrics under Redundant Regular, Medium, and High conditions. (↓: Lower is better, ↑: Higher is better).

Model	**Exec. ↓ Dist.**	**Cons. ↓ Dist.**	**FD Rate ↓**	**TM Acc. ↑**	**EC Rate ↑**	**PFS % ↑**
Redundant. Regular Condition						
GPT-4-Turbo	2.02	1.32	0.212	0.640	0.731	0.442
GPT-4o	1.13	0.60	0.106	0.733	0.712	0.667
Gemini-1.5-pro	1.36	0.78	0.154	0.695	0.695	0.519
Claude-3.7-Sonnet	1.31	0.69	0.102	0.763	0.795	0.780
Llama-3.1-405B	1.60	1.43	0.226	0.498	0.682	0.356
Mixtral-8x7B	4.31	3.76	0.117	0.063	0.112	0.093
Qwen-2.5-72B	2.50	2.21	0.313	0.633	0.669	0.669
Redundant. Medium Condition						
GPT-4-Turbo	2.15	1.68	0.209	0.608	0.613	0.413
GPT-4o	1.24	0.65	0.113	0.735	0.679	0.506
Gemini-1.5-pro	1.50	0.86	0.157	0.677	0.713	0.487
Claude-3.7-Sonnet	1.34	0.73	0.159	0.685	0.737	0.512
Llama-3.1-405B	2.28	1.64	0.190	0.475	0.631	0.361
Mixtral-8x7B	0.00	0.00	0.000	0.000	0.000	0.000
Qwen-2.5-72B	2.76	2.41	0.371	0.456	0.560	0.542
Redundant. High Condition						
GPT-4-Turbo	2.23	2.18	0.285	0.519	0.323	0.388
GPT-4o	1.34	0.72	0.121	0.696	0.327	0.266
Gemini-1.5-pro	1.58	3.05	0.417	0.458	0.144	0.231
Claude-3.7-Sonnet	1.35	0.93	0.184	0.599	0.425	0.332
Llama-3.1-405B	2.41	2.72	0.395	0.324	0.158	0.235
Mixtral-8x7B	0.00	0.00	0.000	0.000	0.000	0.000
Qwen-2.5-72B	3.13	3.49	0.331	0.421	0.082	0.284

Observation 3: Consistent Optimal Tool Selection amidst Tool Diversification

To assess how well agent cores select the best tools from multiple candidates, we use the Optimal Tool Score (OTS), which shows how closely the chosen tools rank in performance relative to available alternatives. Evaluation is conducted under the "Differentiated" tool set condition. As shown in Figure 4a, most LLMs demonstrate strong performance in selecting high-quality tools

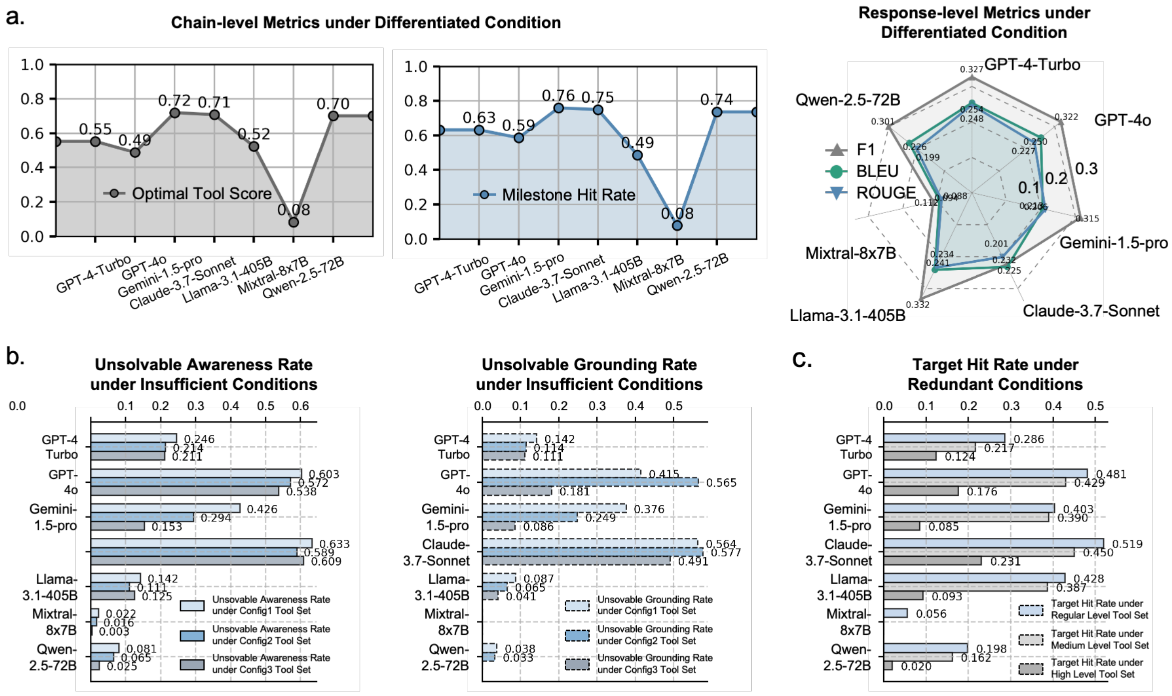

Figure 4: **Evaluation results under "Differentiated" and "Redundant" tool set conditions. a.** Comparative results across leading LLMs under "Differentiated" condition, showing both tool chain-level performance and free text final response-level results. **b.** Performance results under "Insufficient" conditions across LLMs. **c.** Performance results under "Redundant" conditions for leading LLMs.

and completing key task steps. Gemini-1.5-Pro achieves the highest score at 0.720, closely followed by the open-source Qwen-2.5 at 0.701, highlighting their robust comparative reasoning and tool selection capabilities in diverse settings.

Observation 4: Limited I/O Management at Large Tool Set Scale

For evaluating data flow management between sequential tools, we use two metrics under "Redundant" tool set conditions: Execution Completion Rate (ECR, the percentage of tool chains completed without I/O errors) and Pre-Failure Success Percentage (PFSP, the proportion of completed chains before failure in unsuccessful cases). Our findings are: **(1)** as shown in Table. 1, Claude-3.7-sonnet achieves the highest ECR across all "Redundant" conditions (0.795 and 0.737 for regular and medium levels), demonstrating superior execution reliability over other LLMs; **(2)** in medium and high-level cases, about half of the LLMs reach PFSP > 0.5, indicating that even if the full task isn't completed, they can execute a substantial part of the tool chain before failure; **(3)** as tool set complexity increases, all LLMs experience sharp drops in both ECR and PFSP, especially in high-level settings (Table. 1), highlighting major challenges in managing data flow under complex conditions; **(4)** Mixtral consistently shows poor performance on both metrics, reflecting its limited ability to accurately organize and manage I/O relationships between tools.

Observation 5: Inadequate Response Synthesis due to Target Missing

To assess the generation of correct and coherent responses, we evaluate Target Hit Rate (THR, whether the final tool aligns with the required answer format), Milestone Hit Rate (MHR, success in producing key intermediate outputs), and text similarity metrics (BLEU, ROUGE, F1, measuring alignment with reference responses) under both "Redundant" and "Differentiated" tool set conditions. Our analysis shows: **(1)** Claude-3.5-Sonnet achieves the highest THR in Redundant" conditions (0.519, 0.450, and 0.231 for regular, medium, and high-level cases, respectively; Figure 4a); **(2)** even so, the best model only succeeds in 51.9% of tasks under regular "Redundant" settings, mainly due to deviations from the correct tool path and I/O errors; **(3)** for MHR (Figure 4a), Claude-3.5, GPT-4o, and Qwen-2.5 perform similarly well ($\approx$ 0.8) in the "Differentiated" condition, demonstrating strong ability to accomplish key task steps; **(4)** all models exhibit low BLEU, ROUGE, and F1 scores (rarely exceeding 0.35), indicating persistent challenges in synthesizing intermediate outputs into coherent final responses, largely due to planning and execution errors.

Observation 6: Impressive Unsolvability Awareness but Limited at Grounding

To evaluate agent core performance on unsolvable tasks, we use Unsolvability Awareness Rate (UAR, the ability to recognize when a task is impossible) and Unsolvability Grounding Rate (UGR, the ability to identify what is missing) under increasingly challenging "Insufficient" conditions: config1 (missing tool category), config2 (missing modality-specific tools), and config3 (tools present but insufficient). As shown in Figure 4c, Claude-3.5-Sonnet and GPT-4o achieve the highest UARs (around 0.6), indicating strong capability to detect unsolvable cases. However, accurate grounding of failure causes is more challenging; while these two models maintain relatively high UGR across all levels, most otherssuch as Gemini-1.5-Proshow notable drops, especially under subtle "Deficient" settings. Success requires not only recognizing infeasibility but also specifying what is missing (e.g., absent tool categories in config1, missing modality-region combinations in config2, or minimum tool capabilities in config3). Many models, including GPT-4-Turbo, LLaMA-3.1-405B, Mixtral, and Qwen-2.5, frequently attempt to solve tasks with inappropriate tools, reflecting limited awareness of tool constraints and difficulty in accurately judging task solvability.

3.2 Performance Enhancement

Integrated Prompting Strategies Yield Significant Performance Gains

Integrating four prompting strategies (detailed in supp.) significantly boosts task completion rates (Figure. 5). In simple and moderate radiology taskswhere Claude-3.7 and 3.5 already perform wellthe strategies yield extra gains of 0.095 and 0.058, leading to total improvements of 14.4% and 13.9% over baseline. For complex tasks, the impact is more pronounced with improvements up to 48.2%. Each strategy contributes independently, but their combined effect is not strictly additive due to diminishing returns. The results highlight that prompt refinement, self-reflection, and few-shot learning are essential for effective reasoning and guidance, while multi-agent collaboration is critical for managing long-context tasks. However, the added prompt complexity can occasionally lead to performance inconsistencies.

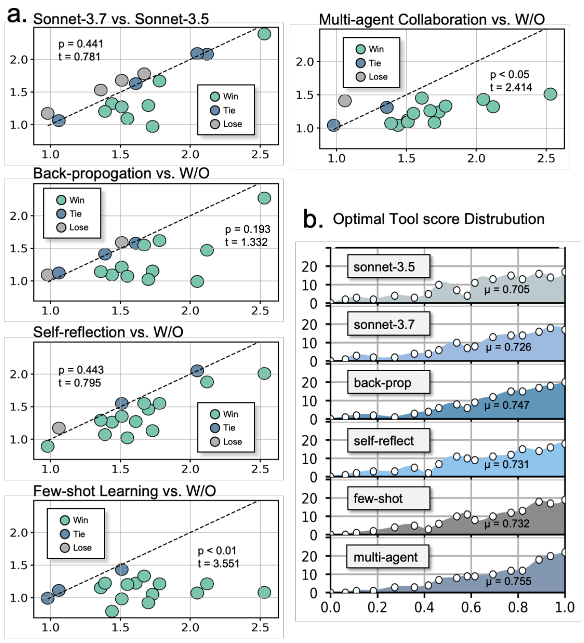

Figure 5: **Performance improvement comparison among prompt engineering strategies.** a. Significance comparison between strategy groups, where "W/O" represents pure Sonnet-3.7 without engineering, and "Win" indicates the performance method before "vs." outperforming the latter. b. Performance distribution across six prompting strategies under "Differentiated" tool set condition.

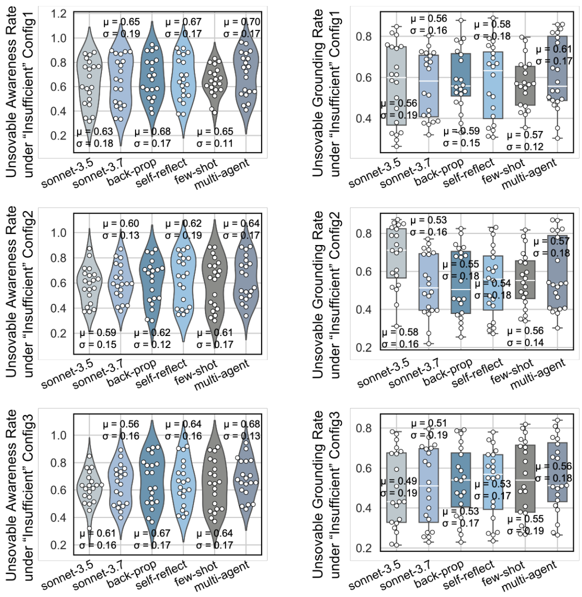

Figure 6: Performance distribution across six prompting strategies under several "Insufficient" tool set conditions.

Multi-agent and Back-propagation Strategies Drive the Most Impact

In our main evaluation (Figure 5a), we compared Claude-3.5, Claude-3.7, and Claude-3.7 with different prompting strategies on 220 QA pairs from 20 patient records over five trials, using average Levenshtein distance per patient and defining a win as outperforming the baseline. Self-reflection and back-propagation showed some improvement, but without statistical significance ($p > 0.05$); in contrast, few-shot learning and multi-agent collaboration achieved statistically significant gains ($p < 0.05$), with multi-agent collaboration leading in most comparisons. Further tests under Insufficient" and Differentiated" conditions (Figure 6) show that multi-agent collaboration improved unsolvability awareness and grounding rates by about 5%, and back-propagation by about 3%, while occasional declines were observed with self-reflection and back-propagation alone. In the "Differentiated" setting, among 139 QA pairs and six methods, prompting strategies consistently raised optimal tool scores, with back-propagation and multi-agent contributing the largest improvements (score increases of 0.021 and 0.029, Figure 5b).

Relaxing Tool Constraints through Automated Tool Building

Evaluated on 220 QA pairs, the system achieves a 65.4% completion rate for routine tasks under Insufficient" conditions, dropping to 44.4% for advanced and 26.2% for complex tasks. The main limitation is the difficulty in accurately identifying missing tool capabilities, especially under "Insufficient config3", leading to erroneous tool building requests. Nonetheless, the system shows strong potential by autonomously building tools and executing end-to-end tasks without human intervention, representing a significant step toward fully automated radiology workflows. Further details are provided in the supplementary material.

4 Related Works

LLM-based Agents in Medicine. The evolution of medical AI has transitioned from static classifiers to dynamic LLM-based agents capable of autonomous reasoning. To enhance clinical validity, recent frameworks have adopted collaborative multi-agent strategies (Tang et al.) or integrated external tools for evidence-based planning (Jin et al., 2024; Wang et al., 2025), thereby addressing the hallucination and precision limitations of generalist models. In the imaging domain, architectures such as ChatCAD (Wang et al., 2024) utilize LLMs as controllers to orchestrate specialized vision models. However, existing benchmarks (Jiang et al., 2025) predominantly focus on textual EHR tasks or isolated classification, lacking a unified evaluation of the multi-modal, tool-integrated workflows essential for radiological agency.

Synthetic Data Generation and Validation. Parallel to agent development, addressing the data scarcity-privacy paradox has driven a shift from traditional de-identification to ab-initio synthesis. While early methods relied on masking real clinical notes (Ren et al., 2025), contemporary approaches utilize persona-based prompting to generate high-fidelity patient cohorts and dialogues from scratch (Haider et al., 2025; Yilmaz et al., 2025). To ensure reliability, validation frameworks have evolved beyond simple n-gram overlap, adopting rigorous metrics such as the Quality-Privacy Score (QPS) to quantify the trade-off between semantic realism and re-identification risk (Sella et al., 2025; Adams et al., 2025). Despite these advancements in general clinical text, frameworks specifically designed to synthesize the structured, multi-modal reporting required for complex radiological error detection remain underexplored (Sun et al., 2025).

5 Conclusion

We introduced RadA-BenchPlat to evaluate the feasibility of Modern LLMs acting as agent cores within radiology workflows. Our extensive benchmarking reveals that while current SOTA modelsparticularly the Claude-3 seriesdemonstrate potential in simplified scenarios, they currently lack the robustness required for autonomous execution of complex clinical tasks due to planning deviations and tool orchestration failures. However, we demonstrate that integrating adaptive prompting strategies, such as multi-agent collaboration and prompt back-propagation, alongside automated tool building, significantly mitigates these limitations. These findings highlight a promising trajectory for future research: moving beyond standalone LLMs toward engineered, self-correcting agentic systems capable of meeting the rigorous demands of real-world radiology.

6 Limitations

Our study is intentionally scoped to establish foundational benchmarks for LLM-based agents in radiology. Consequently, we focus on a controlled synthetic dataset to isolate and evaluate reasoning capabilities without the confounding noise of real-world data; future work will involve validating our framework in clinical settings. Furthermore, we benchmark general-purpose LLMs due to their superior tool-use and planning abilities, which are prerequisites for our agentic framework. Extending this evaluation to medical-specific models and integrating multi-modal (imaging) inputs are critical next steps as these technologies mature.

References

Tim Adams, Colin Birkenbihl, Karen Otte, Hwei Geok Ng, Jonas Adrian Rieling, Anatol-Fiete Näher, Ulrich Sax, Fabian Prasser, and Holger Fröhlich. 2025. On the fidelity versus privacy and utility trade-off of synthetic patient data. *iScience*, 28(5).

Anthropic. Accessed by October 12, 2024. [link].

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Tianle Cai, Xuezhi Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. 2023. Large language models as tool makers. *arXiv preprint arXiv:2305.17126*.

Hejie Cui, Lingjun Mao, Xin Liang, Jieyu Zhang, Hui Ren, Quanzheng Li, Xiang Li, and Carl Yang. 2024a. Biomedical visual instruction tuning with clinician preference alignment. *arXiv preprint arXiv:2406.13173*.

Hejie Cui, Zhuocheng Shen, Jieyu Zhang, Hui Shao, Lianhui Qin, Joyce C Ho, and Carl Yang. 2024b. Llms-based few-shot disease predictions using ehr: A novel approach combining predictive agent reasoning and critical agent instruction. *arXiv preprint arXiv:2403.15464*.

Jinghao Feng, Qiaoyu Zheng, Chaoyi Wu, Ziheng Zhao, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2025. Mˆ 3builder: A multi-agent system for automated machine learning in medical imaging. *arXiv preprint arXiv:2502.20301*.

Syed Ali Haider, Srinivasagam Prabha, Cesar Abraham Gomez-Cabello, Sahar Borna, Ariana Genovese, Maissa Trabilsy, Bernardo G Collaco, Nadia G Wood, Sanjay Bagaria, Cui Tao, and 1 others. 2025. Synthetic patient–physician conversations simulated by large language models: A multidimensional evaluation. *Sensors*, 25(14):4305.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Yixing Jiang, Kameron C Black, Gloria Geng, Danny Park, James Zou, Andrew Y Ng, and Jonathan H Chen. 2025. Medagentbench: A realistic virtual ehr environment to benchmark medical llm agents. *arXiv preprint arXiv:2501.14654*.

Qiao Jin, Zhizheng Wang, Yifan Yang, Qingqing Zhu, Donald Wright, Thomas Huang, W John Wilbur, Zhe He, Andrew Taylor, Qingyu Chen, and 1 others. 2024. Agentmd: Empowering language agents for risk prediction with large-scale clinical tool learning. *arXiv preprint arXiv:2402.13225*.

Jiayu Lei, Xiaoman Zhang, Chaoyi Wu, Lisong Dai, Ya Zhang, Yanyong Zhang, Yanfeng Wang, Weidi Xie, and Yuehua Li. 2024. Autorg-brain: Grounded report generation for brain mri. *arXiv preprint arXiv:2407.16684*.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2023. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *arXiv preprint arXiv:2306.00890*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Zhengliang Liu, Yiwei Li, Peng Shu, Aoxiao Zhong, Longtao Yang, Chao Ju, Zihao Wu, Chong Ma, Jie Luo, Cheng Chen, and 1 others. 2023. Radiology-llama2: Best-in-class large language model for radiology. *arXiv e-prints*, pages arXiv–2309.

Nikita Mehandru, Brenda Y Miao, Eduardo Rodriguez Almaraz, Madhumita Sushil, Atul J Butte, and Ahmed Alaa. 2024. Evaluating large language models as agents in the clinic. *NPJ digital medicine*, 7(1):84.

OpenAI. Accessed by October 12, 2024. [link].

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Vishwanatha M Rao, Michael Hla, Michael Moor, Subathra Adithan, Stephen Kwak, Eric J Topol, and Pranav Rajpurkar. 2025. Multimodal generative ai for medical image interpretation. *Nature*, 639(8056):888–896.

Libo Ren, Samuel Belkadi, Lifeng Han, Warren Del-Pinto, and Goran Nenadic. 2025. Synthetic4health: generating annotated synthetic clinical letters. *Frontiers in Digital Health*, 7:1497130.

Nadir Sella, Florent Guinot, Nikita Lagrange, Laurent-Philippe Albou, Jonathan Desponds, and Hervé Isambert. 2025. Preserving information while respecting privacy through an information theoretic framework for synthetic health data generation. *npj Digital Medicine*, 8(1):49.

Cong Sun, Kurt Teichman, Yiliang Zhou, Brian Critelli, David Nauheim, Graham Keir, Xindi Wang, Judy Zhong, Adam E Flanders, George Shih, and 1 others. 2025. Generative large language models trained for detecting errors in radiology reports. *Radiology*, 315(2):e242575.

Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. Medagents: Large language models as collaborators for zero-shot medical reasoning. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Sheng Wang, Zihao Zhao, Xi Ouyang, Tianming Liu, Qian Wang, and Dinggang Shen. 2024. Interactive computer-aided diagnosis on medical image using large language models. *Communications Engineering*, 3(1):133.

Ziyue Wang, Junde Wu, Linghan Cai, Chang Han Low, Xihong Yang, Qiaxuan Li, and Yueming Jin. 2025. Medagent-pro: Towards evidence-based multi-modal medical diagnosis via reasoning agentic workflow. *arXiv preprint arXiv:2503.18968*.

Georg Wölflein, Dyke Ferber, Daniel Truhn, Ognjen Arandjelović, and Jakob Nikolas Kather. 2025. Llm agents making agent tools. *arXiv preprint arXiv:2502.11705*.

Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Towards generalist foundation model for radiology. *arXiv preprint arXiv:2308.02463*.

Abdurrahim Yilmaz, Furkan Yuceyalcin, Rahmetullah Varol, Ece Gokyayla, Ozan Erdem, Donghee Choi, Ali Anil Demircali, Gulsum Gencoglan, Joram M Posma, and Burak Temelkuran. 2025. A synthetic data generation framework for scalable and resource-efficient medical ai assistants. *medRxiv*, pages 2025–05.

Kai Zhang, Rong Zhou, Eashan Adhikarla, Zhiling Yan, Yixin Liu, Jun Yu, Zhengliang Liu, Xun Chen, Brian D Davison, Hui Ren, and 1 others. 2024. A generalist vision–language foundation model for diverse biomedical tasks. *Nature Medicine*, pages 1–13.

Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-vqa: Visual instruction tuning for medical visual question answering. *arXiv preprint arXiv:2305.10415*.

Ziheng Zhao, Yao Zhang, Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. One model to rule them all: Towards universal segmentation for medical images with text prompts. *arXiv preprint arXiv:2312.17183*.

Xiao Zhou, Xiaoman Zhang, Chaoyi Wu, Ya Zhang, Weidi Xie, and Yanfeng Wang. 2025. Knowledge-enhanced visual-language pretraining for computational pathology. In *Computer Vision – ECCV 2024*, pages 345–362, Cham. Springer Nature Switzerland.

A Appendix

A.1 Terminologies Explanation

Key terminologies used throughout our work are defined in Extended Figure. 7.

A.2 Diseases/abnormalities Names for Patient Records

We consider 100 common conditions for each anatomy-modality combination in our synthetic patient records. The considered diseases/abnormalities names are listed in the CSV file which will be made available upon publication.

A.3 Detailed Explanations for Tool Categories

We explain the 10 tool categories here:

- **Anatomy Classifier (AC)** is a classifier to predict what anatomy region the input image is shot on.
- **Modality Classifier (MC)** is a classifier to predict what imaging modality the input image is based on.
- **Organ Segmentor (OS)** is a tool to predict the dense segmentation masks for a certain organ on the input image.
- **Anomaly Detector (AD)** is a detection tool to predict the anomaly region masks or box coordinates for a certain anomaly type based on the input image.

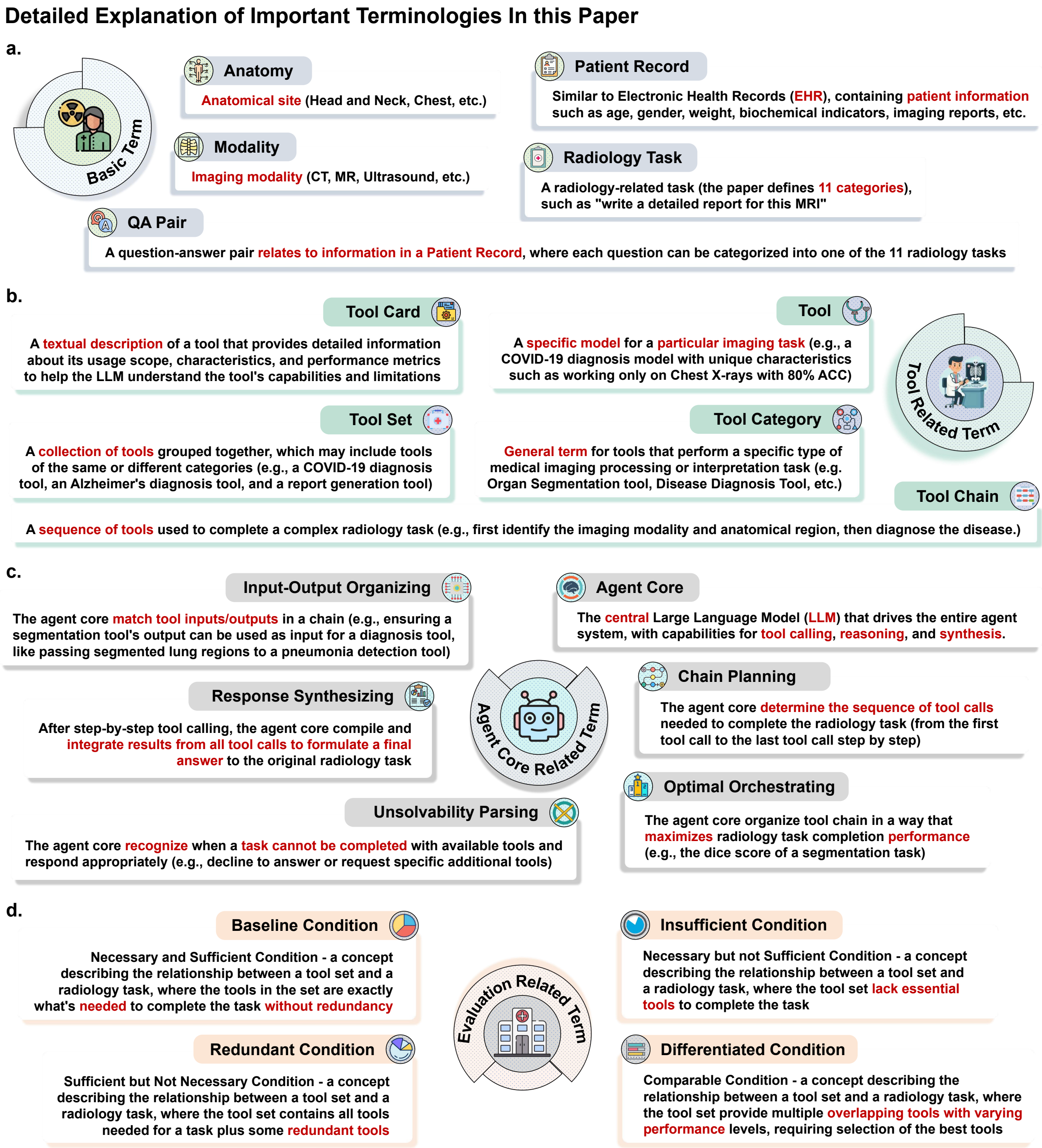

Figure 7: **A detailed explanation of 20 important terminologies used in this paper. a.** Basic terms commonly used throughout the paper. **b.** Sophisticated definitions of tool-related concepts. **c.** Agent-related terms including five key abilities. **d.** Four radiology environments featuring various tool combinations for different evaluation purposes.

- **Imaging Diagnoser (ID)** is a diagnosis tool to predict related diseases solely based on the input image.
- **Synthetic Dignoser (SD)** is a diagnosis tool to predict related diseases synthetically based on the input image and text information.
- **Biomarker Quantifier (BQ)** is a biomarker calculation tool to exactly calculate the radiology biomarker based on the input image and organ-or-anomaly-wise dense masks.
- **Indicator Evaluator (IE)** is an indicator (like tumor grading) calculation tool to exactly calculate or predict some medical indicators based on the input image and organ-or-anomaly-wise dense masks.
- **Report Generator (RG)** is a text generation tool to predict radiology reports.
- **Treatment Recommendation (TR)** is a treatment recommendation tool (or system) to provide treatment recommendation based on current clinical findings.

A.4 Detailed Explanations for Task Types

We explain the 11 radiology tasks here with their detailed tool category chains:

- **Organ mask annotation** is a task targeting predicting a dense mask for a certain organ. **Chain:** *anatomy classification modality classification organ segmentation*
- **Anomaly region annotation** is a task targeting predicting a regional mask for a certain anomaly type. **Chain:** *anatomy classification modality classification anomaly detection*
- **Organ-wise biomarker calculation** is a task to calculate or measure some organ-wise biomarkers. **Chain:** *anatomy classification modality classification organ segmentation biomarker quantification*
- **Anomaly-wise biomarker calculation** is a task to calculate or measure some anomaly-wise biomarkers. **Chain:** *anatomy classification modality classification anomaly detection anomaly quantification*
- **Organ-and-anomaly-wise image interpretation** is a task to interpret the image and answer the user query on organs or anomaly regions or both. **Chain:** *anatomy classification modality classification [organ segmentation, anomaly detection]*
- **Disease diagnosis without visual clues** is a task to end-to-end make a diagnosis based on a radiology-central patient record. **Chain:** *anatomy classification modality classification disease diagnosis*
- **Disease diagnosis with visual clues** is a task to make a diagnosis based on not only original images but also relevant segmentation annotations (organs or anomalies). **Chain:** *anatomy classification modality classification [organ segmentation, anomaly detection] disease inference*
- **Common report generation** is a basic radiology routine task to interpret a certain radiology image into free-text descriptions. **Chain:** *anatomy classification modality classification anomaly detection disease diagnosis report generation*
- **Report generation focusing on specific biomarkers** is a report generation task variant emphasizing reflecting certain biomarker conditions in the final report. **Chain:** *anatomy classification modality classification [organ segmentation, anomaly detection] [organ biomarker quantification, anomaly biomarker quantification] report generation*
- **Report generation focusing on specific biomarkers and indicators** is a report generation task variant emphasizing reflecting certain biomarkers and indicator conditions in the final report. **Chain:** *anatomy classification modality classification [organ segmentation, anomaly detection] disease diagnosis [organ biomarker quantification, anomaly quantification] indicator evaluation report generation*
- **Treatment planning** is a task to provide a treatment plan for a patient. **Chain:** *anatomy classification modality classification [organ segmentation, anomaly detection] disease diagnosis [organ biomarker quantification, anomaly quantification] indicator evaluation report generation treatment recommendation*

A.5 Feature Distributions in Patient Record

Extended Figure 8e visualizes anomalies, diseases, biomarkers, and indicators within patient records. Here, "biomarkers" refer to imaging features (*e.g.*, dimensions, textures), while "indicators" correspond to clinical classifications (*e.g.*, cancer staging) or scoring systems (*e.g.*, CURB-65). We embed these attributes using BioLORD and MedCPT. The t-SNE plot reveals a uniform distribution and confirming that our dataset covers a wide range of clinical scenarios.

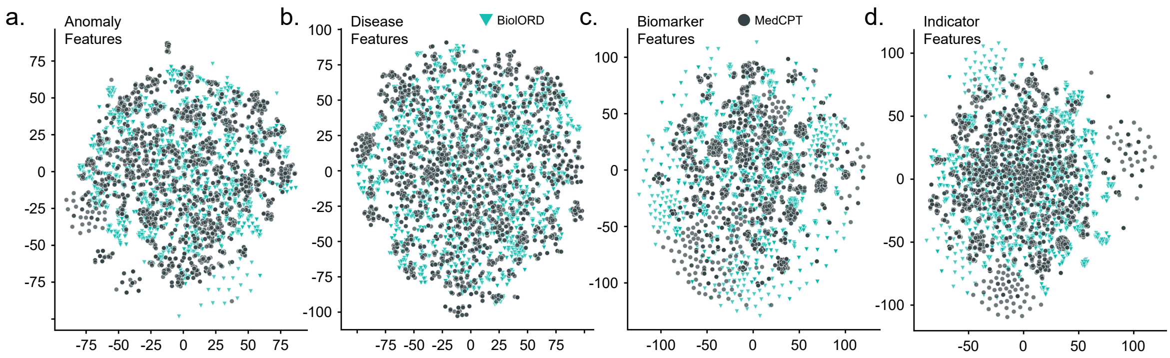

Figure 8: Features distribution on anomaly/disease/biomarker/indicator extracted by BioLORD and MedCPT.

A.6 Context Token Consumption Analysis

As shown in Extended Figure 9, we present the multi-turn response token lengths generated by two LLMs, GPT-4o and Llama-3.1, representing closed-source and open-source models, respectively. In most conditions, the total token lengths for both models are comparable, ranging from 3,000 to 30,000 tokens. However, in the Insufficient tool set conditions, GPT-4o exhibits significantly longer context lengths than Llama, primarily due to Llama's frequent failure to execute proper denial responses, resulting in premature terminations.

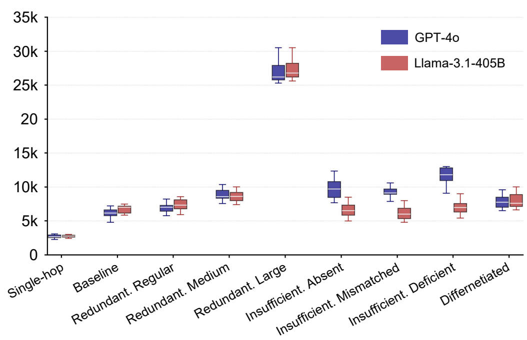

Figure 9: Context token length in 8 simulated conditions using GPT-4o/Llama-3.1-405B as the agent core.

A.7 Performance on Baseline Tool Set Condition

1) closed-source models generally outperform open-source models (except Llama-3.1-405B) across both "Baseline" tool set conditions, with closed-source models achieving LD between planning and ground truth less than 1.5 (average 1.5 edits needed to match ground truth chains) as shown in Extended Figure. 10 - this represents relatively good performance given the average chain length of 5 step; **2)** as shown in Extended Figure 10, differences emerge between planned and executed chains during multi-iteration execution, where models can adjust tool selections - Claude-3.5 and LLama-3.1 show convergence toward ground truth chains, while others maintain or increase deviations.

A.8 A Patient Record Example

We, here, demonstrate a concrete synthetic patient record in our RadABench-Data:

Listing 1: Patient Record Example

```
"Information": {
    "Age": "42",
    "Sex": "Female",
    "Height": "165",
    "Weight": "68",
    "History": "Patient has a history of
         seasonal allergies and recurrent upper
    respiratory infections",
    "Complaint": "Persistent facial pain, nasal
         congestion, and headache for the
    past 2 weeks"
},
"Anatomy": "Head and Neck",
"Modality": "X-ray",
"Anomaly": {
    "Part": "Maxillary sinuses",
    "Symptom": "Opacification"
},
"Disease": "Sinusitis",
"OrganBiomarker": {
    "OrganObject": "Maxillary sinus",
    "OrganDim": "density",
    "OrganQuant": "+40 Hounsfield Units"
},
"AnomalyBiomarker": {
    "AnomalyObject": "Opacification",
    "AnomalyDim": "intensity",
    "AnomalyQuant": "80% increase compared to
         normal airspace"
},
```

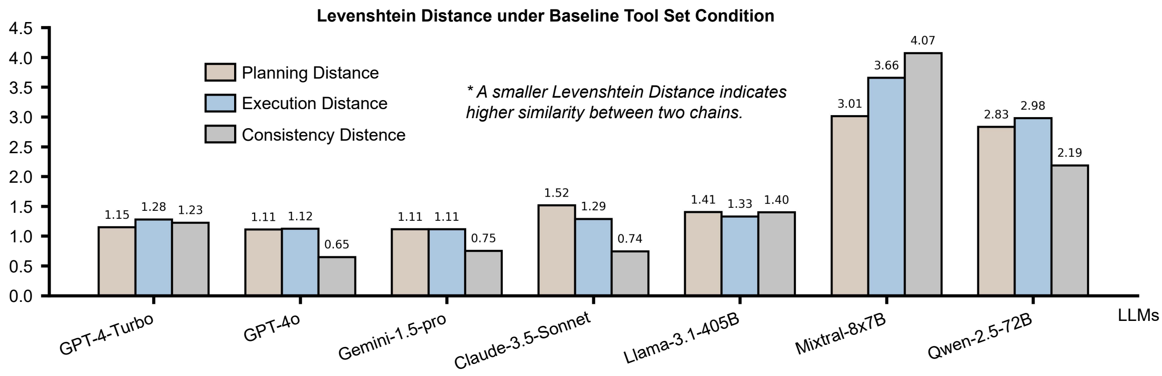

Figure 10: Three Levenshtein Distance between pairs of decision, execution, and ground truth tool chains under NS. Condition among 7 LLMs.

```
"Indicator": {
    "Name": "Lund-Mackay Score",
    "Value": "8 (Moderate sinusitis)"
},
"Report": {
    "Finding": "X-ray of the paranasal sinuses
         demonstrates bilateral maxillary
    sinus opacification. The right maxillary
         sinus shows complete opacification,
    while the left maxillary sinus demonstrates
         air-fluid levels. Frontal and ethmoid
    sinuses appear clear. No evidence of bone
         erosion or destruction. Nasal septum
    appears midline. Soft tissues of the face
         and neck are unremarkable.",
    "Impression": "Findings consistent with
         bilateral maxillary sinusitis, more
    pronounced on the right side. No evidence of
         complications such as orbital or
    intracranial involvement."
},
"Treatment": "Given the patient's symptoms and
    radiographic findings, a diagnosis
of acute bacterial sinusitis is likely. Initial
    treatment should include a 10-14 day
course of broad-spectrum antibiotics such as
    amoxicillin-clavulanate or, in case of
penicillin allergy, a respiratory
    fluoroquinolone. Adjunctive treatments
    include nasal
saline irrigation, intranasal corticosteroids,
    and oral decongestants for symptom
relief. The patient should be advised to stay
    well-hydrated and use over-the-counter
pain relievers as needed. If symptoms persist
    or worsen after 72 hours of antibiotic
therapy, reassessment is warranted. A follow-up
    appointment should be scheduled in 2-3
weeks to ensure resolution of symptoms. If
    recurrent episodes occur, further evaluation
with CT imaging and potential referral to an
    ENT specialist for consideration of
endoscopic sinus surgery may be necessary."
```

which is generated by prompting GPT-4 leveraging the following prompts:

Listing 2: Patient Record Generation Prompt

```
You are an experienced clinical radiologist.
    Your task is to generate a detailed
medical case based on a hypothetical 256x256
    medical image. I will provide you with
the Anatomy and Modality of the image, as well
    as an overall Disease. Using this
information, you should create a comprehensive
    case report including patient
information, specific anomalies, biomarkers,
    indicators, a radiology report, and
treatment recommendations.

Please structure your response using the
    following template:

<Case>
  <Information>
    <Age> [Number without units] </Age>
    <Sex> [Male / Female] </Sex>
    <Height> [Number in cm] </Height>
    <Weight> [Number in kg] </Weight>
    <History> [Brief descriptive text]
        </History>
    <Complaint> [Brief descriptive text]
        </Complaint>
  </Information>
  <Anatomy> [Head and Neck / Chest / Breast /
     Abdomen and Pelvis / Limb / Spine]
  </Anatomy>
  <Modality> [CT / MRI / X-ray / Ultrasound /
     Mammography] </Modality>
  <Anomaly>
    <Part> [Specific location of anomaly
        (e.g., right upper lobe of lung)]
        </Part>
    <Symptom> [Type of anomaly (e.g.,
        nodule)] </Symptom>
  </Anomaly>
  <Disease> [Corresponding disease name]
     </Disease>
  <OrganBiomarker>
    <OrganObject> [A specific organ serving
        as biomarker] </OrganObject>
    <OrganDim> [number / length / size /
        volume / angle / density / intensity
```

```
            /
        texture] </OrganDim>
        <OrganQuant> [Specific quantitative
             value] </OrganQuant>
    </OrganBiomarker>
    <AnomalyBiomarker>
        <AnomalyObject> [The same Anomaly
             described in the Anomaly Symptom
             serving
        as biomarker] </AnomalyObject>
        <AnomalyDim> [number / length / size /
             volume / angle / density / intensity
             /
        texture] </AnomalyDim>
        <AnomalyQuant> [Specific quantitative
             value] </AnomalyQuant>
    </AnomalyBiomarker>
    <Indicator>
        <Name> [Name of the indicator (e.g.,
             Lung Cancer TNM Staging Score)]
             </Name>
        <Value> [Specific value or grade (e.g.,
             cT2aN0M0 (Stage IB))] </Value>
    </Indicator>
    <Report>
        <Finding> [Findings section in the style
             of a MIMIC-CXR report] </Finding>
        <Impression> [Impression section in the
             style of a MIMIC-CXR report]
             </Impression>
    </Report>
    <Treatment> [A paragraph including
         diagnostic procedures, medication
    recommendations, and follow-up suggestions]
         </Treatment>
</Case>

Guidelines for generating the case report:

Information: Provide realistic patient
     demographics, medical history, and chief
complaint.
Anomaly: Describe a specific anomaly consistent
     with the given anatomy, modality,
and disease.
Biomarkers: Choose a relevant organ biomarker
     and the anomaly biomarker mentioned
before that can be observed in the image and
     provide plausible quantitative value.
Indicator: Calculate an appropriate indicator
     based on the patient information and
biomarker value. Provide a specific, medically
     accurate score or grade.
Report: Generate a concise radiological report
     in the style of MIMIC-CXR, with
separate Findings and Impression sections.
Treatment: Offer a comprehensive treatment plan
     including diagnostic procedures,
medication recommendations, and follow-up care.
Ensure that all parts of the case are medically
     accurate and consistent with each
other. Use your expertise as a radiologist to
     provide realistic and detailed
information throughout the case.

Given the following parameters:
- Anatomy: {ANATOMY}
- Imaging Modality: {MODALITY}
- Disease: {DISEASE}
Please generate this patient record.
```

A.9 A Tool Card Example

Here is an example of a tool card of a report generation tool focused on Head and Neck X-ray images:

Listing 3: Tool Card Example

```
=== Tool Description for TOOL17 ===
Name: TOOL17
Category: Report Generator
Ability: Given the Head and Neck X-ray Image,
     any other text information and
organ/anomaly masks and labels, generate a
     radiology report.
Property: Report Generator only suitable for
     Head and Neck X-ray image with
Text and Mask
Compulsory Input: ['$Image$']
Optional Input: ['$Information$',
     '$OrganObject$', '$AnomalyObject$',
     '$Disease$',
'$OrganDim$', '$OrganQuant$', '$AnomalyDim$',
     '$AnomalyQuant$', '$IndicatorName$',
'$ValueName$', '$OrganMask$', '$AnomalyMask$']
Output: ['$Report$']
Performance: Score from 0.4 to 0.88, increases
     with optional inputs
```

A.10 A Task Simulation Example

Here is an example of QA pair generation from a chest X-ray record of a pneumonia patient:

Listing 4: QA Pair Example

```
    <Q1> Identify and segment the lung fields in
         this chest X-ray. </Q1>
    <A1> The lung fields are segmented [{\em
         organ mask}: left lung, right lung].
    Both lungs show clear boundaries. </A1>

    ...

    <Q11> Based on the imaging findings and
         clinical indicators, what treatment plan
    would you recommend? </Q11>
    <A11> Given the moderate pneumonia severity
         (CURB-65 score: 2) and [anomaly mask],
    recommend oral antibiotics and follow-up
         chest X-ray in 2 weeks. </A11>
```

which is generated by the following prompt, leveraging GPT-4:

Listing 5: QA Pair Generation Prompt

```
Assume a clinical medical imaging scenario
     where you, as the Agent core, play the role
of a doctor. Given a patient's radiological
     image, you want to complete different tasks
by calling various tools. There are ten tools
     in total (numbered 0 to 9):

TOOLKIT
0. |Modality Classifier|
   Property: A classification model
   Ability: Determine the modality of the Image.
   Input: [Image]
```

```
    Output: [Modality]

1. |Anatomy Classifier|
    Property: A classification model,
    Ability: Determine the anatomy of the Image.
    Input: [Image]
    Output: [Anatomy]

2. |Organ Segmentation Model|
    Property: A segmentation model
    Ability: Given the modality and anatomy,
        segment all organs in the Image
    (can not segment any lesion or abnormality).
    Input: [Image] & [Modality] & [Anatomy]
    Output: [Organ Mask] & [Organ Label]

3. |Anomaly Detection Model|
    Property: A detection model
    Ability: Given the modality and anatomy,
        determine the location and type of
    abnormality.
    Input: [Image] & [Modality] & [Anatomy]
    Output: [Anomaly Mask] & [Anomaly Label]

4. |Disease Diagnosis Model|
    Property: A classification model
    Ability: Given the modality and anatomy,
        diagnose diseases directly from the
        input
    image.
    Input: [Image] & [Modality] & [Anatomy]
    Output: [Diseases]

5. |Disease Inference Model|
    Property: A Inference model
    Ability: Infer disease based on organ
        segmentation and anomaly detection
        results.
    Input: [Image] & [Organ Mask] & [Organ
        Label] & [Anomaly Mask] & [Anomaly
        Label]
    Output: [Diseases]

6. |Biomarker Quantification Model|
    Property: A quantification model
    Ability: Given the organ region or anomaly
        region and the biomarker of interest,
    estimate its property. The biomarker can be
        either organ or anomaly. The dimension
    can be one of the number, length, size,
        volume, angle, density, intensity or
    texture of the organ or anomaly.
    Input: [Image] & [Object] & [Dim] &
        [Biomarker Mask] & [Biomarker Label]
    Output: [Quant]

7. |Indicator Evaluation Model|
    Property: A calculation model
    Ability: Use prior patient information and
        several biomarkers values to calculate
    the indicator, the indicator can be a score
        or a grading.
    Input: [Priors] & [Indicator] & [Biomarkers]
        & [Quants]
    Output: [Value]

8. |Report Generation Model|
    Property: A multimodal model
    Ability: Generate a medical report by
        integrating results processed by former
        tools.
    Input: [Image] & [Modality] & [Anatomy]
    Optional Input: ([Organ Mask] & [Organ
        Label]), ([Anomaly Mask] & [Anomaly
        Label]),
    [Diseases], ([Object] & [Dim] & [Quant]),
        ([Indicators] & [Values])
    Output: [Report]

9. |Treatment Recommendation Model|
    Property: a language model
    Ability: Recommend personalized treatment
        plans based on all results processed by
    former tools and the patient's information.
    Input: [Image] & [Priors] & [Modality] &
        [Anatomy] & [Diseases]
    Optional Input: ([Organ Mask] & [Organ
        Label]),([Anomaly Mask] & [Anomaly
        Label]),
    ([Object] & [Dim] & [Quant]), ([Indicators]
        & [Values])
    Output: [Treatment]

There are 11 different tasks and the chain of
    tools each task hopes to break down into:

(1) Basic Image Analysis and Organ Segmentation
    ToolUse: 012
    Description: Perform basic image analysis
        and segment organs within the medical
    image.

(2) Basic Image Analysis and Anomaly Detection
    ToolUse: 013
    Description: Perform basic image analysis
        and detect anomalies within the medical
    image.

(3) Image-based Direct Disease Diagnosis
    ToolUse: 014
    Description: Diagnose disease directly from
        the medical image without intermediate
    steps.

(4) Organ segmentation and anomaly localization
    ToolUse: 0123
    Description: Segment organs and locate
        anomalies within the medical image.

(5) Anomaly-based Disease Diagnosis
    ToolUse: 01235
    Description: Diagnose disease based on
        disease inference by finding
        abnormalities
    and the organ in which they occur.

(6) Organ Biomarker Quantification
    ToolUse: 0126
    Description: Quantify specific biomarkers
        related to organs in the medical image.

(7) Anomaly Biomarker Quantification
    ToolUse: 0136
    Description: Quantify specific biomarkers
        related to anomalies in the medical
        image.

(8) Disease and Anomaly Based Report Generation
    ToolUse: 01348
    Description: Generate a medical report based
```

```
        on detected diseases and anomalies.

(9) Disease and Biomarker Based Report
     Generation
    ToolUse: 0123568
    Description: Generate a comprehensive report
        incorporating anomaly detection,
    disease diagnosis and biomarker
        quantification.

(10) Comprehensive Evaluation Report Generation
    ToolUse: 01235678
    Description: Generate a detailed evaluation
        report including all aspects of the
    medical image analysis.

(11) Comprehensive Report Generation and
     Treatment Recommendations
    ToolUse: 012356789
    Description: Generate a comprehensive report
        including all analysis results and
    treatment recommendations.

Please generate 11 mutually independent
    question-answer pairs corresponding to tasks
above, based on the different task natures and
    the content of the case. Specifically,
strictly avoid including information
in the questions that should be determined by
    the tools (such as imaging modality,
specific anatomy, or precise abnormality types).
Pay attention: create task-specific
    question-answer pairs that highlight the
    unique
tool usage patterns for different tasks. The
    questions should be:

1. Naturally aligned with the task description
2. Representative of real-world scenarios for
    that task type
3. Questions must not reveal information about
    modality or anatomy that should be
determined by Tools 0 and 1 or other tools. The
    questions should be phrased in a way
that necessitates the use of these basic
    identification tools.

Provide answers in the most concise free-text
    form possible. If visual results such as
masks are involved, please embed them in the
    text in the form of [Organ Mask] or
[Anomaly Mask]. (eg. The organ segmentation
    result is shown as [Organ Mask].) The 11
generated question-answer pairs should follow
    this template:

<Q1> ... </Q1>
<A1> ... </A1>
<Q2> ... </Q2>
<A2> ... </A2>
...
<Q11> ... </Q11>
<A11> ... </A11>

The answer only needs to provide a simple final
    result based on the information already
available in the Case, without showing the
    thought process. I will now provide you with
a Case containing all the information.
The patient record is {Patient Record}, please
    generate the corresponding QA pairs.
```

A.11 Detailed Workflow Prompts in RadABench-EvalPlat

In this part we detail the prompt template used in the main workflow of our RadABench-EvalPlat.

Task Decomposition.

Listing 6: Task Decomposition Prompts

```
# Medical Image Analysis Assistant

## Task Overview
You are a radiological agent analyzing medical
    images.
For each query, you will receive:
    1. A medical imaging examination (Image) of
        a patient (assume already provided)
    2. Known patient Information including
        demographics, medical history,
    and main complaints.

Your task involves three sequential parts:

1. Problem Decomposition (Part 1)
- Identify available information
- Break down the question into sequential steps

2. Sequential Tool Application (Part 2)
- Execute one tool at a time
- Record each tool's output
- Continue until sufficient information is
    gathered

3. Solution Synthesis (Part 3)
- Integrate all results
- Generate final answer

## Available Information Categories
The following categories must be used exactly
    as written:

['$Information$', '$Anatomy$', '$Modality$',
    '$Disease$', '$OrganObject$',
'$OrganDim$', '$OrganQuant$',
    '$AnomalyObject$', '$AnomalyDim$',
'$AnomalyQuant$', '$IndicatorName$',
    '$IndicatorValue$', '$Report$',
    '$Treatment$']

Where:
- $Information$: Patient demographics
(e.g., "45-year-old male", "BMI: 24", "history
    of diabetes")
- $Anatomy$: Body part (e.g., "chest",
    "abdomen", "brain")
- $Modality$: Imaging technique (e.g., "X-ray",
    "CT", "MRI")
- $Disease$: Medical condition (e.g.,
    "pneumonia", "cancer", "fracture")
- $OrganObject$: Organ to measure (e.g.,
    "liver", "heart")
- $OrganDim$: Organ measurement type (e.g.,
    "number", "length", "size",
  "volume", "angle", "density", "intensity",
      "texture")
```

```
- $OrganQuant$: Organ measurement value (e.g.,
     "5cm", "120ml")
- $AnomalyObject$: Abnormality to measure
     (e.g., "tumor", "fracture")
- $AnomalyDim$: Abnormality measurement type
     (e.g., "number", "length", "size",
  "volume", "angle", "density", "intensity",
       "texture")
- $AnomalyQuant$: Abnormality measurement value
     (e.g., "2cm", "5ml")
- $IndicatorName$: Clinical indicator name
- $IndicatorValue$: Clinical indicator value
- $Report$: Medical report content
- $Treatment$: Treatment recommendations

## Available Tool Categories
Tool categories must be used exactly as written:

[*Anatomy Classification Tool*, *Modality
     Classification Tool*,
*Organ Segmentation Tool*, *Anomaly Detection
     Tool*,
*Disease Diagnosis Tool*, *Disease Inference
     Tool*,
*Organ Biomarker Quantification Tool*,
*Anomaly Biomarker Quantification Tool*,
     *Indicator Evaluation Tool*,
*Report Generation Tool*, *Treatment
     Recommendation Tool*]

## Response Format for Part 1
For each query, respond ONLY with:

Known Info: [list any categories explicitly
     mentioned in the query]
Tool Chain: [list required tools connected by
     ->]

## Examples

Query 1: "For a straightforward approach to
     diagnosing the patient's condition
based on her symptoms and the image, what
     diseases can be directly identified?"
Response:
Known Info: []
Tool Chain: [*Anatomy Classification Tool* ->
     *Modality Classification Tool*
            -> *Disease Diagnosis Tool*]

Query 2: "This 45-year-old male's chest CT
     shows a 2cm nodule in the right lung.
Can you give a report?"
Response:
Known Info: ['$Information$', '$Anatomy$',
     '$Modality$', '$AnomalyObject$',
            '$AnomalyDim$', '$AnomalyQuant$']
Tool Chain: [*Organ Segmentation Tool* ->
     *Anomaly Detection Tool*
            -> *Disease Inference Tool* ->
                *Report Generation Tool*]
            (because some information is
                provided, so
            *Anatomy Classification Tool*,
                *Modality Classification Tool*,
            *Anomaly Biomarker Quantification
                Tool* are optimized.)

## Important Rules
1. Assume the medical image is already provided
2. Use exact item category names with $$ as
     listed (e.g., '$Anatomy$')
3. Use exact tool category names with ** as
     listed
   (e.g., '*Anatomy Classification Tool*')
4. Only respond with Part 1 analysis - Parts 2
     & 3 will be addressed
   in subsequent interactions
5. Include only the categories explicitly
     mentioned in the query
6. Connect tools using -> symbol

Please wait for my query.
When provided, analyze it following the format
     shown in the examples above.

{Patient Record}

{Query}
```

Tool Selection & Execution.

Listing 7: Tool Selection & Execution Prompts

```
# Next Step Planning

## Current Status
Current results dictionary: {value_dict}

## Planning Guidelines
1. Reference your high-level tool chain from
     Part 1 decomposition
2. Consider current results to refine specific
     tool selection
3. Maintain sequential progression according to
     planned workflow
4. Adjust tool selection if needed based on
     intermediate results
5. Check if the tool category is missing when
     this category of tools is required
6. Check if the tool is suitable for the
     detected Anatomy and Modality in reserved
value dictionary based on the Tool description
     Ability and Property
7. Check if the result in reserved value
     dictionary can be derived from each tool
     used
in former steps based on the limited label list
     described in Tool Ability
8. If no suitable tool exists, identify which
     type of denial applies:
   - Missing tool category
   - Missing specific modality-anatomy tool
   - Insufficient tool capability

## Input Requirements
1. Required inputs: Must include all mandatory
     inputs specified in tool description
2. Optional inputs: Include if available and
     beneficial to tool performance
3. Do not include variables that are not
     relevant to the tool's function
4. All variables must exist in the current
     results dictionary
5. Use proper $$ notation for all variables

## Response Format
For ongoing analysis (if not final step):
<Call>
   <Purpose>Brief, clear statement of this
```

```
        step's goal in context of overall
    analysis</Purpose>
    <Tool>TOOL[number] - must match available
        specific tools</Tool>
    <Input>['$variable1$', '$variable2$', ...] -
        use only existing variables from
    results</Input>
</Call>

For final step only:
<EndCall>
    <Purpose>Brief, clear statement of this
        final step's goal</Purpose>
    <Tool>TOOL[number] - must match available
        specific tools</Tool>
    <Input>['$variable1$', '$variable2$', ...] -
        use only existing variables from
    results</Input>
</EndCall>

When a tool category is completely missing:
<NoCall>
    <Purpose>The purpose requiring a missing
        tool category</Purpose>
    <Category>The missing category from
        ['Anatomy Classifier', 'Modality
        Classifier',
    'Organ Segmentor', 'Anomaly Detector',
        'Disease Diagnoser', 'Disease
        Inferencer',
    'Biomarker Quantifier', 'Indicator
        Evaluator', 'Report Generator',
        'Treatment
    Recommender']</Category>
    <Anatomy>Universal</Anatomy>
    <Modality>Universal</Modality>
    <Ability>CategoryMissing</Ability>
</NoCall>

When specific tools for a modality-anatomy
     combination are missing:
<NoCall>
    <Purpose>The purpose requiring a specific
        modality-anatomy tool</Purpose>
    <Category>The required tool
        category</Category>
    <Anatomy>The specific anatomy from
        ['Universal', 'Head and Neck', 'Chest',
    'Breast', 'Abdomen and Pelvis', 'Limb',
        'Spine']</Anatomy>
    <Modality>The specific modality from
        ['Universal', 'X-ray', 'CT', 'MRI',
    'Ultrasound']</Modality>
    <Ability>SpecificToolMissing</Ability>
</NoCall>

When existing tools lack required capabilities:
<NoCall>
    <Purpose>The purpose requiring advanced
        capabilities</Purpose>
    <Category>The category of existing but
        insufficient tools</Category>
    <Anatomy>The relevant anatomy</Anatomy>
    <Modality>The relevant modality</Modality>
    <Ability>InsufficientCapability</Ability>
</NoCall>

## Format Requirements
1. Maintain proper XML structure
2. Use exact tool numbers as specified in tool
     descriptions
3. Mark all variables with $$ notation
4. Include only existing variables from results
     dictionary
5. Keep purpose statements clear and concise
6. Brief response only includes one Call,
     EndCall, or NoCall XML part without
additional explanations
7. For NoCall responses, use the appropriate
     format based on denial type

## Decision Making Process
1. Review planned tool chain from Part 1
2. Check current results in value dictionary
3. Check if tool category is missing when this
     category of tools is required
4. Check tool Ability and Property in detail to
     judge its suitability for detected
Anatomy and Modality in the value dictionary
5. Check if the result in reserved value
     dictionary can be derived from each tool
     used
in former steps based on the limited label list
     described in Tool Ability
6. Evaluate tool availability and capability:
   - Is the required tool category available?
   - Are specific tools available for the
       needed modality-anatomy combination?
   - Do available tools have sufficient
       capabilities?
7. If tools are insufficient, use appropriate
     NoCall format
8. If tools are available, select and format
     appropriate Call/EndCall
9. Use <EndCall> only if this will be the final
     step

Please provide your next step based on:
- Original tool chain plan
- Current results
- Available specific tools
- Remaining analysis needs
- Tool availability and capability assessment
```

Response Generation.

Listing 8: Response Generation Prompt

```
Based on your Part 1 analysis plan, Part 2 tool
     execution sequence, and the final
results dictionary {value_dict}, provide:

    1. A concise answer to the initial question
    2. Key supporting evidence from your results
    3. How your findings align with the planned
         analysis

Keep your response brief and focused on
     directly answer the initial question.
```

A.12 An In-depth Case Study

Here we select a representative case from the agent core evaluation workflow to demonstrate the process, where we test the agent cores' performance using a QA pair based on the task type

"Anomaly-wise Biomarker Calculation" and a patient record (a 62-year-old female with hypertension, osteoarthritis of the knees, and chronic neck pain with stiffness radiating to right shoulder) under the NR. Denyl2 tool set condition, using Claude-3.5-Sonnet as the agent core, documenting each decision it makes. We can observe both the strengths and limitations reflected by this example.

Initial Input. The agent core receives the patient information along with a virtual image. For our evaluation task "Anomaly-wise Biomarker Calculation", we generate a corresponding QA pair based on the patient's complete medical record. The initial input is provided below:

Listing 9: Initial Input

```
$Information$: {
    "Age": "62",
    "Sex": "Female",
    "Height": "165",
    "Weight": "72",
    "History": "Hypertension, osteoarthritis of
        the knees",
    "Complaint": "Chronic neck pain and
        stiffness, radiating pain to right
        shoulder"
},

$Query$: From an anomaly perspective in a
    specific medical image, after identifying
the type and area, could you quantify specific
    biomarker characteristics?

$Image$: ‘PLACEHOLDER_IMAGE’
```

Task Decomposition. The agent core processes these inputs by analyzing the patient information and identifying key data for extraction. It determines the appropriate high-level tool chain required for task completion. The agent core then stores all relevant information in a memory bank and establishes a tool category chain to guide subsequent execution. The process is illustrated below:

Listing 10: Stage 1: Task Decomposition

```
Initial Output: Known Info: []
Tool Chain: [*Anatomy Classification Tool* ->
    *Modality Classification Tool* ->
*Anomaly Detection Tool* -> *Anomaly Biomarker
    Quantification Tool*]
Initial Value Dict: {'$Image$':
    'PLACEHOLDER_IMAGE', '$Information$':
    'PLACEHOLDER_
INFORMATION'}
Initial Score Dict: {'$Image$': 1.0,
    '$Information$': 1.0}
Initial Fixed Dict:
    frozendict.frozendict({'$Image$': 1.0,
    '$Information$': 1.0})
High-level Tool chain: Anatomy Classification
    Tool -> Modality Classification Tool ->
Anomaly Detection Tool -> Anomaly Biomarker
    Quantification Tool

Memory bank: {'$Image$': 'PLACEHOLDER_IMAGE',
    '$Information$': 'PLACEHOLDER_
INFORMATION'}
Score bank: {'$Image$': 1.0, '$Information$':
    1.0}
Fixed bank: frozendict.frozendict({'$Image$':
    1.0, '$Information$': 1.0})
```

Tool Selection & Execution: The first phase of tool selection and execution commences. At each step, the agent core integrates the planned high-level tool chain with stored information from the memory bank to generate a tool API call, specifying the purpose, tool name, and input parameters. The Platform then executes the API call by activating the selected tool and processing the input. Upon successful execution, the tool's output is stored in the memory bank. We configure the tool set according to the NR. Denyl2 condition, where there is a mismatch between the anomaly detection tool and the provided $Image$ characteristics (Head and Neck X-ray) in terms of anatomy and modality. The established tool set is as follows:

Listing 11: Tool Set Description

```
"TOOL1": {
    "Name": "TOOL1",
    "Category": "Anatomy Classifier",
    "Ability": "Determine the anatomy of the
        Image.",
    "Property": "Universal Anatomy Classifier",
    "Compulsory Input": [
        "$Image$"
    ],
    "Optional Input": [],
    "Output": [
        "$Anatomy$"
    ],
    "Performance": "Score from 0.95 to 0.95,
        increases with optional inputs"
},
"TOOL2": {
    "Name": "TOOL2",
    "Category": "Modality Classifier",
    "Ability": "Determine the modality of the
        Image.",
    "Property": "Universal Modality Classifier",
    "Compulsory Input": [
        "$Image$"
    ],
    "Optional Input": [],
    "Output": [
        "$Modality$"
    ],
    "Performance": "Score from 0.95 to 0.95,
        increases with optional inputs"
},
```

```
...

"TOOL5": {
    "Name": "TOOL5",
    "Category": "Anomaly Detector",
    "Ability": "Given the Limb Ultrasound Image,
         determine the location and
    type of abnormality.",
    "Property": "Anomaly Detector only suitable
         for Limb Ultrasound image",
    "Compulsory Input": [
        "$Image$"
    ],
    "Optional Input": [],
    "Output": [
        "$AnomalyMask$",
        "$AnomalyObject$"
    ],
    "Performance": "Score from 0.8 to 0.8,
         increases with optional inputs"
},
"TOOL6": {
    "Name": "TOOL6",
    "Category": "Anomaly Detector",
    "Ability": "Given the Breast MRI Image,
         determine the location and type
    of abnormality.",
    "Property": "Anomaly Detector only suitable
         for Breast MRI image",
    "Compulsory Input": [
        "$Image$"
    ],
    "Optional Input": [],
    "Output": [
        "$AnomalyMask$",
        "$AnomalyObject$"
    ],
    "Performance": "Score from 0.8 to 0.8,
         increases with optional inputs"
},

...

"TOOL8": {
    "Name": "TOOL8",
    "Category": "Disease Diagnoser",
    "Ability": "Given the Head and Neck X-ray
         Image, diagnose the disease.",
    "Property": "Disease Diagnoser only suitable
         for Head and Neck X-ray image",
    "Compulsory Input": [
        "$Image$"
    ],
    "Optional Input": [],
    "Output": [
        "$Disease$"
    ],
    "Performance": "Score from 0.75 to 0.75,
         increases with optional inputs"
},

...

"TOOL10": {
    "Name": "TOOL10",
    "Category": "Biomarker Quantifier",
    "Ability": "Measure the anomaly biomarker of
         the Image.",
    "Property": "Universal Anomaly Biomarker
         Quantifier",
    "Compulsory Input": [
        "$Image$",
        "$AnomalyObject$",
        "$AnomalyMask$"
    ],
    "Optional Input": [
        "$AnomalyDim$"
    ],
    "Output": [
        "$AnomalyDim$",
        "$AnomalyQuant$"
    ],
    "Performance": "Score from 0.75 to 0.8,
         increases with optional inputs"
},

...
```

In this initial step, the agent core selects the anatomy classification tool. We observe that the anatomical information is successfully stored in the memory bank after the tool execution:

Listing 12: Tool Selection & Execution Step 1

```
Step 1 starts!
Stepoutput: Based on the tool chain from Part 1
     (Anatomy Classification Tool ->
Modality Classification Tool -> Anomaly
     Detection Tool -> Anomaly Biomarker
Quantification Tool) and current results
     dictionary, I'll proceed with the first
     step:

<Call>
    <Purpose>Determine the anatomy region of the
         medical image</Purpose>
    <Tool>TOOL1</Tool>
    <Input>['$Image$']</Input>
</Call>
Call Dict: {'Purpose': 'Determine the anatomy
     region of the medical image', 'Tool':
'TOOL1', 'Input': ['$Image$']}
Memory bank: {'$Image$': 'PLACEHOLDER_IMAGE',
     '$Information$': 'PLACEHOLDER_
INFORMATION', '$Anatomy$': 'Head and Neck'}
Score bank: {'$Image$': 1.0, '$Information$':
     1.0, '$Anatomy$': 0.95}
Fixed bank: frozendict.frozendict({'$Image$':
     1.0, '$Information$': 1.0})
Step 1 completes!
```

In the second step, the modality classification tool is selected and executed. The successful execution is demonstrated below. As we can observe that the modality information is added to memory bank after tool execution.

Listing 13: Tool Selection & Execution Step 2

```
Step 2 starts!
Stepoutput: Based on the tool chain and current
     results showing Head and Neck anatomy,
the next step is to determine the image
     modality:
```

```
<Call>
    <Purpose>Determine the imaging modality of
        the medical image</Purpose>
    <Tool>TOOL2</Tool>
    <Input>['$Image$']</Input>
</Call>
Call Dict: {'Purpose': 'Determine the imaging
     modality of the medical image', 'Tool':
'TOOL2', 'Input': ['$Image$']}
Memory bank: {'$Image$': 'PLACEHOLDER_IMAGE',
     '$Information$': 'PLACEHOLDER_
INFORMATION', '$Anatomy$': 'Head and Neck',
     '$Modality$': 'X-ray'}
Score bank: {'$Image$': 1.0, '$Information$':
     1.0, '$Anatomy$': 0.95, '$Modality$':
0.95}
Fixed bank: frozendict.frozendict({'$Image$':
     1.0, '$Information$': 1.0})
Step 2 completes!
```

In the third step, contrary to the expected utilization of the organ segmentation tool, the disease diagnosis tool is invoked instead. While this tool selection is irrelevant to the current task, the execution completes without errors. The tool call is detailed below:

Listing 14: Tool Selection & Execution Step 3

```
Step 3 starts!
Stepoutput: Given the current results showing
     Head and Neck anatomy and X-ray modality,
the next step is to detect any anomalies in the
     medical image:

<Call>
    <Purpose>Detect abnormalities in the Head
        and Neck X-ray image</Purpose>
    <Tool>TOOL8</Tool>
    <Input>['$Image$']</Input>
</Call>
Call Dict: {'Purpose': 'Detect abnormalities in
     the Head and Neck X-ray image', 'Tool':
'TOOL8', 'Input': ['$Image$']}
Memory bank: {'$Image$': 'PLACEHOLDER_IMAGE',
     '$Information$': 'PLACEHOLDER_
INFORMATION', '$Anatomy$': 'Head and Neck',
     '$Modality$': 'X-ray', '$Disease$':
'Cervical spine degenerative changes'}
Score bank: {'$Image$': 1.0, '$Information$':
     1.0, '$Anatomy$': 0.95, '$Modality$':
0.95, '$Disease$': 0.75}
Fixed bank: frozendict.frozendict({'$Image$':
     1.0, '$Information$': 1.0})
Step 3 completes!
```

During the fourth step, when selecting the anomaly detection tool, the agent core correctly identifies that the available tools in the set lack the capability to detect and localize anomalies in Head and Neck X-ray images. As demonstrated below, the agent core appropriately rejects this invalid operation and successfully recognizes the tools' capability gap.

Listing 15: Tool Selection & Execution Step 4

```
Step 4 starts!
Stepoutput: Before we can quantify biomarkers
     (per the original tool chain), we need
the anomaly detection results first. Looking at
     the available tools for Head and Neck
X-ray anomaly detection:

<NoCall>
    <Purpose>Detect specific anomalies in Head
        and Neck X-ray for biomarker
    quantification</Purpose>
    <Category>Anomaly Detector</Category>
    <Anatomy>Head and Neck</Anatomy>
    <Modality>X-ray</Modality>
    <Ability>SpecificToolMissing</Ability>
</NoCall>
Deny Call Dict: {'Purpose': 'Detect specific
     anomalies in Head and Neck X-ray for
biomarker quantification', 'Category': 'Anomaly
     Detector', 'Anatomy': 'Head and Neck',
'Modality': 'X-ray', 'Ability':
     'SpecificToolMissing'}Agent denies the call
     because no
suitable tool is found.
```

Overall, while the agent core demonstrates correct initialization and tool chain planning, it exhibits a deviation during the tool selection & execution stage. Despite having access to tool chain guidance in each round, the agent core fails to follow these directives, resulting in an unnecessary disease diagnosis tool call. Nevertheless, the execution proceeds without any IO errors, and the agent core successfully identifies the tools' capability limitations, appropriately concluding with a denial of task execution due to the missing functionality.