MEDVISTAGYM: A Scalable Training Environment for *Thinking with Medical Images* via Tool-Integrated Reinforcement Learning

Anonymous ACL submission

Abstract

Vision-language models (VLMs) perform well on general image understanding but struggle to *think with medical images*, where reliable answers often require iterative visual inspection, verification, and evidence refinement. Existing medical VLMs typically rely on static visual embeddings and single-pass inference, while open-source VLMs lack the training infrastructure to learn effective medical tool selection, invocation, and coordination. We introduce MEDVISTAGYM, a scalable interactive environment for tool-integrated medical visual reasoning. MEDVISTAGYM provides a unified executable interface where VLM agents learn when and which tools to invoke, localize task-relevant regions, and integrate single or multiple visual evidence sources through multi-turn reasoning. Using MEDVISTAGYM, we train MEDVISTA-R1 with trajectory sampling and end-to-end reinforcement learning to interleave tool use with agentic reasoning. Across six medical VQA benchmarks, MEDVISTA-R1-8B outperforms comparably sized backbones with naive tool access by **19.71%** to **24.21%**, showing that structured agentic training, rather than tool access alone, is key to effective tool-integrated medical image reasoning.

1 Introduction

Vision language models (VLMs) have achieved remarkable progress on medical image understanding, demonstrating strong performance across visual question answering (VQA) (Chen et al., 2024b), disease diagnosis (Liu and Song, 2025), diagnostic report generation (Goswami et al., 2025; Xia et al., 2025), and agentic medical visual analysis (Guo et al., 2025b). Much of this progress can be attributed to text-based reasoning paradigms such as Chain-of-Thought (CoT) (Wei et al., 2022) and reinforcement learning (RL) (Guo et al., 2025a), which decompose complex problems into intermediate reasoning steps and improve reasoning quality through outcome-based optimization (Liu et al., 2025; Huang et al., 2025; Zhang et al., 2025). Together, these advances have moved medical VLMs beyond direct prediction toward more structured, step-by-step clinical reasoning.

Despite these advances, current medical VLMs remain limited in interactive visual reasoning, i.e., the ability to iteratively acquire, verify, and refine visual evidence during multi-step inference. Existing approaches largely rely on static visual embeddings and shallow cross-modal alignment, causing models to attend to irrelevant anatomy while overlooking fine-grained diagnostic cues such as blurred lesion boundaries, low-contrast abnormalities, and subtle tissue textures (Zheng et al., 2025; Fan et al., 2025). These limitations stem not from model capacity but from static, single-pass reasoning frameworks that lack mechanisms for continuous image interaction. As perception and decision-making occur in one forward pass, models cannot re-observe, verify, or refine visual evidence during reasoning, preventing dynamic visual exploration. Consequently, existing approaches fall short of true "*thinking-with-images*" (Su et al., 2025), where reasoning is tightly coupled with iterative visual perception in complex clinical scenarios.

To bridge this gap, recent work has explored *tool-integrated reasoning* (TIR) (Guo et al., 2025c; Zheng et al., 2025; Lu et al., 2025; Xu et al., 2025b), which augments VLMs with external tools to support fine-grained visual reasoning. However, current open-source VLMs still struggle to leverage these tools effectively for medical reasoning tasks, largely due to the absence of a training environment that allows agents to learn dynamic tool selection and coordinated multi-step interaction through experience. Although some studies explore tool use in specific medical settings (Li et al., 2024a; Fathi et al., 2025), systematic simulation and training of tool-integrated thinking for medical visual reasoning remain largely unexplored.

Motivated by these challenges, we introduce

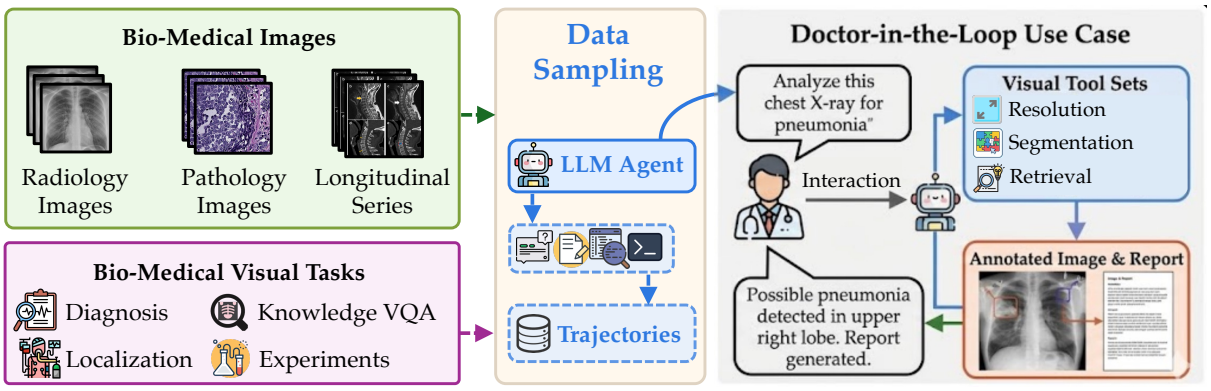


Figure 1: Overview of MEDVISTAGYM, which contains a comprehensive suite of reasoning-intensive medical image analysis tasks and tools in an interactive execution environment, scaling visual-centric tool-integrated agentic reinforcement learning for VLM agents.

MEDVISTAGYM (**Med**ical **Vis**ual-centric **T**ool-integrated **A**gentic training environment), a scalable and interactive training environment designed to operationalize tool-integrated thinking for medical visual reasoning. MEDVISTAGYM encapsulates diverse visual tool operations and provides structured textual feedback, enabling closed-loop interaction between reasoning and perception (Figure 1). It supports (i) multimodal medical reasoning across *six* public datasets, (ii) a unified and extensible interface to *fifteen* visual and medical knowledge tools, and (iii) scalable infrastructure for efficient agent training and evaluation. Building upon MEDVISTAGYM, we develop MEDVISTA-R1, a VLM-based agent for robust tool-augmented medical image reasoning. Across three in-domain and three out-of-domain medical VQA benchmarks, MEDVISTA-R1-8B delivers consistent gains over comparably sized open-source baselines under both tool-enabled and tool-free settings, validating MEDVISTAGYM as a scalable environment for training agentic VLMs.

2 MEDVISTAGYM

2.1 Problem Formulation

We formulate tool-integrated medical image analysis as a partially observable Markov decision process (POMDP). Given a medical question and image(s), an agent receives partial observations $\mathcal{O}$ from latent states $\mathcal{S}$ and selects executable tool actions $a_t \in \mathcal{A}$. The environment applies a deterministic transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ and returns tool outputs as new observations. At each step, the agent produces a reasoning step g_t, invokes a tool action a_t, and updates its decision based on the returned evidence. The interaction forms a trajectory $U = (g_0, a_0, \ldots, g_T, \hat{y})$, where $\hat{y}$ is the final answer after T turns.

2.2 Environment Design

MEDVISTAGYM instantiates this formulation as a scalable, executable environment for multi-turn medical visual reasoning. It defines the tasks, tools, action interface, observation space, and execution backend independently of any specific training algorithm. Unlike scripted tool-use pipelines, MEDVISTAGYM supports online adaptive interaction: each tool call returns a real observation that conditions the next reasoning step and action.

2.3 Tasks and Tool Sets

Medical VQA Tasks. MEDVISTAGYM includes verifiable medical VQA tasks that require grounded, multi-step reasoning over images and intermediate evidence. The training tasks cover radiology VQA, including VQA-RAD (Lau et al., 2018) and SLAKE (Liu et al., 2021), and pathology VQA, including PathVQA (He et al., 2020). These datasets span anatomy recognition, finding identification, lesion localization, subtle abnormality detection, and tissue-level reasoning. More dataset details are provided in Appendix A.2.

Visual-Centric Tools Sets. MEDVISTAGYM exposes standardized executable tools for perception, localization, segmentation, region refinement, and knowledge retrieval. Tools are grouped into four families: (1) *Resolution and Region Refinement*, such as `agent4k` and `zoom-in`; (2) *Medical Localization and Segmentation*, such as `groundingdino` and `medsam2`; (3) *Medical Visual Understanding and Parsing*, such as `biomedclip` and `biomedparse`; and (4) *Biomedical Knowledge*

Retrieval, such as `GoogleSearch`, `DrugBank`, and `PubMed`. Tool outputs serve as intermediate evidence for subsequent reasoning. More details are provided in Appendix F.

2.4 Training Infrastructure

Agent-Env Interface. MEDVISTAGYM follows a Gym-style protocol with explicit reset and step interfaces. It defines a verifiable action space in which each action corresponds to an executable tool invocation, and a structured observation space containing tool outputs, execution feedback, and visual or textual evidence. This design enables reproducible multi-turn interaction for evidence-grounded medical reasoning. Details are provided in Appendix I.1.

Scalable Execution Infrastructure. To support high-frequency rollouts, MEDVISTAGYM exposes all medical tools as independent services under a unified execution interface. The backend supports asynchronous and batched tool invocation across episodes, enabling efficient multi-turn training while preserving real tool execution. A unified `BaseTool` abstraction further allows new perception or knowledge tools to be plugged in with minimal engineering overhead. Implementation details are provided in Appendix I.2.

3 MEDVISTA-R1

3.1 Training Data Collection

Data Collection and QA Generation. We curate training data from PathVQA, SLAKE, and VQA-RAD following three criteria: diverse medical tasks and imaging modalities, measurable benefit from tool use, and the need for multi-step or multi-tool reasoning. To identify tool-beneficial instances, `GPT-5` (OpenAI, 2025) performs tool-augmented reasoning under the MEDVISTAGYM tool constraints. We discard samples with ambiguous, incorrect, or unverifiable answers.

Reasoning Trajectory Generation and Data Selection. We initialize the policy with behavioral cloning on tool-augmented reasoning trajectories that interleave thoughts, tool calls, observations, and final answers. Candidate trajectories are generated by `GPT-5` under the same task and tool constraints as training. We retain trajectories whose final answers match the ground truth, then use `GPT-5` as a trajectory judge to score reasoning-guided tool use on a 4-point scale. These scores are used to filter and weight demonstrations, favoring coherent, medically grounded tool orchestration. More details are provided in Appendix A.1.

3.2 Training Framework

We train the agent in two stages. First, cold-start SFT teaches basic multi-turn tool invocation, visual interaction, and tool-output verification. Second, online RL trains the agent inside MEDVISTAGYM, where real environmental feedback refines tool orchestration and evidence-grounded reasoning.

Stage I: Cold-Start Supervised Bootstrapping. Vanilla VLMs are unreliable at precise tool execution and long-horizon clinical reasoning. We therefore initialize the agent with supervised fine-tuning on a curated cold-start dataset D_{cold} of multi-turn trajectories that interleave explicit thoughts and executable tool calls. Behavioral cloning over reasoning and tool-call tokens teaches valid tool syntax, appropriate tool selection and ordering, and coherent multi-step reasoning.

Stage II: Agentic Tool-Integrated Online RL. The agent is further optimized through online reinforcement learning in the executable environment. At each turn, the policy generates a reasoning segment and a tool invocation; the environment executes the tool and returns structured observations, which are appended to the context. This closed-loop interaction enables the agent to improve tool orchestration beyond static demonstrations.

Rollout Formulation. MEDVISTAGYM treats tool outputs as external environment observations rather than as model-generated tokens in text-only RL. Given a query Q and image I, a trajectory up to step k is $R_k = (r_1, t_1, o_1; \ldots; r_k, t_k, o_k)$, where r_i, t_i, and o_i denote reasoning text, tool invocation, and tool observation. The policy generates the next reasoning-tool pair as $(r_{k+1}, t_{k+1}) \sim \pi_\theta(\cdot \mid I, Q, R_k)$. Rollouts continue until the model produces a final answer or reaches the tool-call limit, with early termination for repeated tool calls. Observation tokens are appended to context but excluded from loss computation. Prompt and output-format details are provided in Appendix H.

Reward Design. We use a rule-based rollout reward with three components: format validity, answer accuracy, and answer-conditioned tool use. The format reward R_{format} checks whether required special tokens appear in the correct order. The answer reward $R_{\text{acc}}(U) = \mathbb{I}\{\hat{y} = y\}$

evaluates the final prediction $\hat{y}$ extracted from the <answer>...</answer> span and is assigned only to well-formed, non-repetitive outputs. The tool reward R_{tool} is given only when a trajectory both invokes at least one external tool and produces the correct final answer, discouraging arbitrary tool calls. The final reward is $R(U) = R_{\text{format}}(U) + R_{\text{acc}}(U) + R_{\text{tool}}(U)$, encouraging valid *think→tool_call→answer* trajectories with task-relevant tool use. More formulations and details are provided in Appendix D.

Optimization. We optimize the policy with Group Relative Policy Optimization (GRPO) (Guo et al., 2025a). GRPO normalizes advantages across groups of sampled trajectories, stabilizing policy updates and promoting higher-quality reasoning paths. More details are provided in Appendix E.

4 Experiments

4.1 Experiment Setups

Datasets. We evaluate on representative medical VQA benchmarks, including in-distribution datasets PathVQA, SLAKE, and VQA-RAD, and out-of-distribution datasets PMC-VQA (Zhang et al., 2024), MMMU (H&M) (Yue et al., 2024), and MicroVQA (Burgess et al., 2025). MicroVQA further evaluates *Expert Visual Understanding (V)*, *Hypothesis Generation (H)*, and *Experiment Proposal (E)*. We also test open-ended medical VQA to assess free-form generation.

Evaluation Metrics. To support reliable evaluation and semantic rewards, we use biomedical encoders: PubMedBERT for BERTScore and BioBERT-mnli-snli-scinli-scitail-mednli-stsb for cosine similarity. We report accuracy (ACC) for multiple-choice questions, and BLEU-1, ROUGE-1, BERTScore, and CosSim for open-ended answers. More details are provided in Appendix B.4

Implementation Details. We implement MEDVISTA-R1 on InternVL3 (Zhu et al., 2025), Qwen3VL (Bai et al., 2025a) with 2B and 8B variants, and train the model using Verl-Tool (Jiang et al., 2025) on NVIDIA A100 GPUs. During SFT, we train for 5 epochs with a learning rate of 1×10^{-5} and a total batch size of 256. For RL, we use a batch size of 256, sample 8 candidate reasoning trajectories per question with up to 6 tool calls, and adopt a constant learning rate of 1×10^{-6} with a maximum context length of 2.6K tokens. During inference, we deploy external tools as FastAPI (Ramírez and Contributors, 2018) services to accelerate tool invocation. More implementation details are provided in Appendix B.

Baselines. We compare MEDVISTA-R1 with three categories of baselines: (1) general-purpose VLMs with visual reasoning, including API-based VLMs such as GPT-5 (OpenAI, 2025), GPT-5-mini (OpenAI, 2025), GPT-o4-mini (OpenAI, 2024), Gemini-2.5-Pro (Google, 2025b), Gemini-2.5-Flash (Google, 2025a), Claude-4.5-Sonnet (Anthropic, 2025a), and Claude-4.5-Haiku (Anthropic, 2025b); (2) open-source and medical-specific VLMs, including InternVL3, Qwen2.5-VL (Bai et al., 2025b), Qwen3-VL (Bai et al., 2025a), LLaVA-Med (Li et al., 2023), HuatuoGPT-Vision (Chen et al., 2024a), Chiron-o1 (Sun et al., 2025), and Lingshu (Team et al., 2025); and (3) medical agent frameworks, including MMedAgent (Li et al., 2024a), VILA-M3 (Nath et al., 2025), MMedAgent-RL (Li et al., 2024b), and MedAgent-Pro. All proprietary baselines are evaluated *without tool access* unless otherwise specified.

4.2 Main Results

Table 1 reports results on three in-distribution and three out-of-distribution benchmarks. To test generality across model families, we evaluate MEDVISTA-R1 with Qwen3-VL-8B in addition to InternVL3. MEDVISTA-R1 improves Qwen3-VL-8B from 62.94% to 73.53% on in-distribution benchmarks and from 55.12% to 63.69% overall, yielding +10.59% and +8.57% absolute gains. It also outperforms naive tool-access baselines by +19.71% and +24.21% overall, consistent with the InternVL3 results and supporting the model-agnostic nature of MEDVISTAGYM. Table 2 further compares MEDVISTA-R1 with other RL-based medical agent baselines.

Open-ended Medical Generation. To assess free-form answer generation, we evaluate on open-ended subsets of SLAKE, VQA-RAD, PathVQA, and VQA-MED. As shown in Table 3, MEDVISTA-R1 (Qwen3-VL-8B) improves over Qwen3-VL-8B from 50.65% to 63.45%, with consistent gains across all four datasets. This indicates that environment-based tool-integrated training benefits both closed-form selection and free-form generation in medical VQA. More evaluation details are provided in Appendix C.4.

Models(↓) / Datasets(→)	In-Distribution				Out-of-Distribution							All
	SLAKE	VQA-RAD	PathVQA	Avg.	PMC-VQA	MMMU (H&M)	MicroVQA				Avg.	Avg.
	Overall	Overall	Overall		Overall	Overall	Overall	V	H	E		
Commercial VLMs (vanilla prompt)												
GPT-5	70.28	67.53	64.40	67.40	62.00	77.86	60.80	62.00	61.20	58.30	66.89	67.15
GPT-5-mini	64.59	73.31	68.65	68.85	60.00	74.28	57.10	58.70	57.61	53.51	63.79	66.32
GPT-o4-mini	76.27	63.21	66.00	68.49	56.00	78.75	65.00	68.75	67.78	54.35	66.52	67.51
Claude-4.5-haiku	77.00	62.50	60.00	66.50	59.00	72.14	53.14	52.81	51.21	57.00	61.43	63.96
Claude-4.5-sonnet	75.00	63.39	63.00	67.13	59.00	76.73	56.33	55.60	57.91	54.83	64.02	65.58
Gemini-2.5-pro	85.00	71.43	71.00	75.81	62.00	71.29	60.90	61.20	62.60	57.40	64.73	70.27
Gemini-2.5-flash	72.00	62.50	36.08	56.86	35.00	46.43	47.00	50.00	44.50	46.50	42.81	49.84
Base Size: 7–13B parameters												
Qwen2.5-VL-7B	42.11	64.14	62.40	56.22	49.00	46.43	33.00	34.38	31.11	34.78	42.81	49.51
LLaVA-Med-7B	61.97	56.60	59.00	59.19	27.00	37.71	16.51	12.20	17.60	21.70	27.07	41.13
Qwen3-VL-8B family												
Qwen3-VL-8B	52.63	73.11	63.08	62.94	52.50	52.86	35.90	38.80	31.90	38.30	47.09	55.12
MEDVISTAGYM (Qwen3-VL-8B)	27.00	51.89	56.40	45.10	49.50	42.86	36.28	40.62	35.61	32.63	42.88	43.98
SFT only w/o Tools	39.81	71.78	68.55	60.05	47.13	52.95	45.07	48.31	48.06	38.83	48.38	54.28
SFT only w/ Tools	62.79	67.68	64.40	66.04	50.96	47.49	39.60	41.71	40.95	40.57	50.15	56.03
SFT + GRPO w/o Tools	75.34	71.56	66.80	71.23	50.50	45.94	43.94	42.75	46.42	42.64	46.84	59.04
GRPO w/ Tools from Base	33.50	71.01	64.80	56.43	45.75	42.88	43.75	44.17	41.32	43.07	44.13	50.28
SFT + GRPO w/ Tools (MEDVISTA-R1)	79.33	**73.31**	67.94	73.53	53.70	55.36	**52.50**	**55.10**	**52.21**	50.20	**53.85**	**63.69**
InternVL3-8B family												
InternVL3-8B	43.06	72.91	68.10	61.36	56.50	55.89	31.50	31.90	26.70	39.60	47.96	54.66
MEDVISTAGYM (InternVL3-8B)	28.50	31.70	46.00	35.40	43.50	48.57	35.00	39.06	32.20	34.80	42.36	38.88
SFT w/o Tools	35.00	70.75	**72.00**	59.25	51.00	55.00	42.50	43.75	43.33	39.10	49.50	54.38
SFT w/ Tools	61.00	66.04	66.00	64.35	56.00	52.14	39.00	39.10	37.80	41.30	49.05	56.70
SFT + GRPO w/o Tools	35.00	70.75	70.50	58.75	51.50	45.30	43.00	40.63	44.44	43.48	46.60	52.68
GRPO w/ Tools from Base	30.00	72.64	68.00	56.88	49.00	43.80	41.00	43.75	38.89	41.30	44.61	50.74
SFT + GRPO w/ Tools (MEDVISTA-R1)	**81.36**	70.75	69.00	**73.70**	**58.00**	**56.43**	43.00	42.20	37.80	**54.41**	52.48	63.09
Base Size: < 7B parameters												
InternVL3-2B	42.25	41.04	41.80	41.71	44.50	43.57	30.48	30.00	28.57	34.86	39.52	40.61
MEDVISTAGYM (InternVL3-2B)	19.00	18.87	31.11	22.99	40.00	36.43	31.00	34.40	25.60	37.00	35.81	29.40
SFT w/o Tools	31.00	59.43	61.00	50.48	46.50	44.29	35.50	39.10	41.11	34.78	42.10	46.29
SFT w/ Tools	45.00	52.83	57.00	51.61	47.50	45.71	37.00	35.90	38.90	34.80	43.40	47.51
SFT + GRPO w/o Tools	29.00	58.49	63.00	50.16	48.50	37.50	37.00	34.38	37.78	39.13	41.00	45.58
GRPO w/ Tools from Base	25.00	64.15	66.00	51.72	47.00	42.00	38.50	35.90	36.67	43.48	42.50	47.11
SFT + GRPO w/ Tools (MEDVISTA-R1)	60.99	58.49	55.00	58.16	49.50	47.43	39.50	37.50	37.78	45.65	45.48	51.82

Table 1: Main results (Acc.%) on in-distribution and out-of-distribution medical VQA benchmarks. MEDVISTAGYM *(Backbone)* denotes direct tool access without training. *SFT only* removes GRPO; *w/o Tools* removes tool access; *GRPO w/ Tools from Base* removes SFT warm-up.

Models	Tool	PathVQA	Slake	VQARAD	Avg.
MLLMs can think with image					
DeepEyes-7B (Zheng et al., 2025)	✓	52.9	68.2	65.9	62.33
Mini-o3-7B-vl (Lai et al., 2025a)	✓	53.4	67.8	65.7	62.30
PixelReasoner-RL-vl-7B (Wang et al., 2025a)	✓	52.6	67.3	66.0	61.97
Opensource SOTA					
Llava-Next-13B (Li et al., 2024c)	✗	39.8	57.1	54.8	50.57
Qwen2.5vl-32B (Bai et al., 2025b)	✗	47.4	70.1	71.7	63.07
Medical MLLMs					
LlaVa-med-7B (Xu et al., 2025a)	✗	44.6	47.7	52.5	48.27
HuatuoGPT-Vision-34B (Chen et al., 2024a)	✗	50.7	68.3	61.7	60.23
Medical MLLMs with CoT Reasoning					
MedVLM-R1-2B (Pan et al., 2025)	✗	38.3	54.3	45.2	45.93
Med-R1-2B (Lai et al., 2025b)	✗	19.2	52.1	36.5	35.93
Lingshu-7B (Team et al., 2025)	✗	68.4	77.8	66.4	70.87
Chiron-01-8B (Sun et al., 2025)	✗	68.8	77.4	**72.5**	72.90
Multimodal medical agents					
MMedAgent-7B (Li et al., 2024b)	✓	59.47	68.7	64.0	64.06
AURA (Fathi et al., 2025)	✓	59.8	68.4	64.5	64.23
SMR-Agents (Wang et al., 2026)	✓	38.2	53.5	46.9	46.20
MedAgent-Pro (Wang et al., 2025d)	✓	58.5	69.4	63.3	63.73
MMedAgentRL-7B (Li et al., 2024a)	✗	58.5	67.9	66.1	64.17
VILA-M3-40B (Nath et al., 2025)	✓	66.4	71.4	65.7	67.83
MEDVISTA-R1 (Qwen3-VL-8B)	✓	67.94	79.33	**73.31**	73.53
MEDVISTA-R1 (InternVL3-8B)	✓	**69.00**	**81.36**	70.75	**73.70**

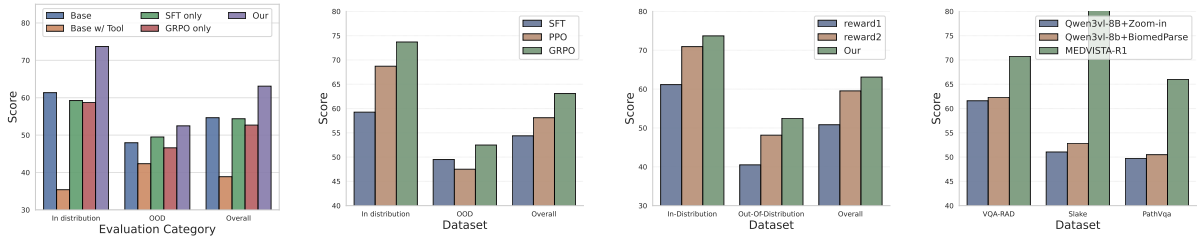
Table 2: Comparison of MEDVISTA-R1 and RL-based multimodal medical agent baselines on medical VQA benchmarks.

Model	SLAKE	VQA-RAD	PathVQA	VQA-MED	Avg.
Qwen3-VL-8B	54.84	54.67	44.66	48.42	50.65
Qwen2.5-VL-7B	56.03	44.87	28.61	26.22	38.93
LLaVA-Med-7B	37.91	34.33	12.34	25.11	27.42
InternVL3-8B	63.31	45.02	29.73	29.67	41.93
MEDVISTA-R1 (Qwen3-VL-8B)	**78.88**	**69.20**	**52.47**	**53.23**	**63.45**

Table 3: MEDVISTA-R1-8B performance on VQA open-ended generation tasks.

4.3 Ablation Studies

Effect of Tool-Integrated Reasoning. To assess the impact of MEDVISTAGYM, we isolate tool use, SFT warm-up, and reinforcement learning through controlled variants in Table 4. We observe that naive tool access is not inherently beneficial: InternVL3-8B drops from 54.66% to 38.88% when directly placed in MEDVISTAGYM without training. SFT alone provides limited gains, with SFT w/ Tools reaching 56.70%, while SFT + GRPO w/o Tools remains below the vanilla model at 52.68%. GRPO w/ Tools from Base also underperforms, indicating that RL exploration is unstable without SFT warm-up. The full SFT + GRPO w/ Tools setting achieves the best performance at 63.09%, demonstrating that effective tool-integrated reasoning requires the combination of tool access, supervised warm-up, and online RL.



(a) Effect of training stages (b) Effect of RL algorithms (c) Effect of reward design (d) Effect of multi-agents

Figure 2: Ablation study on training configurations.

Variant	Tools	Reasoning	RL	ID Avg.	OOD Avg.	Overall	Δ
GPT-5 backbone							
GPT-5 (vanilla)	✗	✗	✗	67.40	66.89	67.15	–
+ Direct Tool Access	✓	✗	✗	65.81	62.36	64.08	−3.07
+ MEDVISTAGYM (tool access)	✓	✗	✗	76.33	66.91	71.62	+4.47
InternVL3-8B backbone							
InternVL3-8B (vanilla)	✗	✗	✗	61.36	47.96	54.66	–
+ MEDVISTAGYM (tool access)	✓	✗	✗	35.40	42.36	38.88	−15.78
+ SFT w/o Tools	✗	✓	✗	59.25	49.50	54.38	−0.28
+ SFT w/ Tools	✓	✗	✗	64.35	49.05	56.70	+2.04
+ SFT + GRPO w/o Tools	✗	✓	✓	58.75	46.60	52.68	−1.98
+ GRPO w/ Tools from Base	✓	✓	✓	56.88	44.61	50.74	−3.92
+ SFT + GRPO w/ Tools (MEDVISTA-R1)	✓	✓	✓	73.70	52.48	63.09	+8.43

Table 4: Effect of tool-integrated reasoning. Δ denotes the overall accuracy change relative to the corresponding vanilla backbone (GPT-5 = 67.15; InternVL3-8B = 54.66).

Effect of Training Stages. Figure 2(a) shows that SFT provides a crucial warm-up by aligning the model with tool-use formats and syntax, while RL builds on this foundation to deliver larger gains. Together, SFT establishes reliable tool-use priors, and RL enables deeper interleaved reasoning for solving complex tasks.

Effect of RL Algorithm. We evaluate PPO and DAPO as alternatives to GRPO. Figure 2(b) shows that GRPO yields more robust performance. To provide the full comparison, Table 5 reports SFT, DAPO, and GRPO results across all benchmarks. GRPO substantially outperforms DAPO by 9.24% overall, suggesting that group-normalized comparison among multiple rollouts is better suited to sparse, multi-turn tool-interaction rewards.

Effect of Reward Design. Figure 2(c) compares three reward designs: (1) reward-1, a base reward $R_{\text{sparse}} = R_{\text{format}} \cdot R_{\text{correct}}$ that evaluates only format compliance and answer correctness; (2) reward-2, an extra tool use reward added to the base reward; and (3) ours, a conditional tool use reward that is granted only when tool use leads to a correct answer. The results show that removing the tool reward leads to a substantial performance drop, highlighting its importance. Among all settings, the answer-conditioned tool-use reward achieves the highest accuracy.

Exploration of Multiagent Workflow Variants. To evaluate whether performance gains arise solely from tool usage, we compare MEDVISTA-R1 with tool-based and multi-agent workflow baselines of comparable model sizes (Figure 2d). These baselines isolate the effect of tool invocation from reasoning-driven interaction. Across these variants, simply incorporating tools yields limited gains. In contrast, MEDVISTA-R1 dynamically coordinates tool invocation with reasoning, enabling selective and task-dependent use of visual evidence.

4.4 Generalization Across Backbones and Frameworks

Backbone Generalization. We evaluate MEDVISTAGYM across three VLM families: InternVL3-8B, Qwen3-VL-8B (Table 1), and Qwen2.5-VL-7B (Appendix Table 11). Across all backbones, naive tool access hurts performance, SFT restores baseline behavior, and GRPO brings substantial gains. This consistency shows that MEDVISTAGYM's benefits come from the training environment and protocol, not backbone-specific effects. More analysis is provided in Appendix C.3.

Comparison with Concurrent Tool-Reasoning Frameworks. Table 6 summarizes the distinction between MEDVISTAGYM and closely related tool-reasoning frameworks. Unlike VTool-R1 and AgentThink, MEDVISTAGYM provides a reusable interactive environment with closed-loop tool execution, scalable infrastructure, unseen-tool generalization evaluation, and controlled analysis, showing that naive tool access can be harmful.

4.5 Efficiency and Resource Analysis

Training and Inference Efficiency. Table 7 summarizes both training and inference costs. The full training process costs approximately 219 A100 GPU-hours, including 3 GPU-hours for cold-start SFT and 216 GPU-hours for GRPO. At inference

Method	SLAKE	VQA-RAD	PathVQA	PMC-VQA	MMMU	MicroVQA	ID Avg.	OOD Avg.	Overall
SFT	35.00	70.75	**72.00**	51.00	55.00	42.50	59.25	49.50	54.38
DAPO	69.30	**71.00**	63.10	43.15	36.33	40.20	67.80	39.89	53.85
GRPO	**81.36**	70.75	69.00	**58.00**	**56.43**	**43.00**	**73.70**	**52.48**	**63.09**

Table 5: Effect of post-training algorithms. GRPO achieves the best overall performance under sparse, multi-turn tool-interaction rewards.

Dimension	VTool-R1	AgentThink	MEDVISTAGYM (Ours)
Domain	Chart/Table VQA	Autonomous driving	Medical image analysis
Tool interaction	Single-turn (max 1 call)	Multi-step static trajectories	Multi-turn closed-loop (up to 6 calls)
Interactive environment	✗	✗	✓ Gym-style reset/step API
Number of tools	6 visual editing tools	~10 driving-specific tools	15 tools in 4 families
Async/scalable infra	✗	✗	✓ Ray + FastAPI microservices
Tool generalization eval	✗	✗	✓ unseen-tool transfer
Naive tool access analysis	✗	✗	✓ tool access alone can be harmful

Table 6: Comparison with concurrent tool-integrated reasoning frameworks.

time, MEDVISTA-R1 requires more tokens and tool calls than the vanilla backbone, reflecting deeper multi-turn reasoning rather than simple answer selection.

Training Cost				
Stage	**GPU**	**Duration**	**GPU-Hours**	**Data**
Cold-start SFT	2×A100	~1.5h	3	3.3K
GRPO RL	6×A100	~36h	216	15.5K
Total	–	~37.5h	**219**	–
Inference Cost				
Method	**Time/Sample**	**Tokens**	**Tool Calls**	**Notes**
InternVL3-8B	2.62s	17	0	Vanilla
MEDVISTAGYM (InternVL3-8B)	16.37s	203	0.97	Direct tool access
MEDVISTA-R1 (InternVL3-8B)	28.53s	746	2.57	Multi-turn reasoning
MEDVISTAGYM (GPT-5)	–	–	1.80	Teacher trajectories

Table 7: Efficiency analysis of MEDVISTA-R1, including training cost and inference-time latency, token length, and tool usage.

Performance under Tool-Call Budgets. To evaluate the trade-off between tool usage and performance, we vary the maximum number of allowed interaction turns at inference time. Table 8 shows that performance remains stable across different budgets, while the average *end round*, defined as the number of tool calls actually used before the agent emits a final answer, remains around 2.3–2.6. This indicates that the model learns to terminate after gathering sufficient evidence rather than exhausting the maximum budget.

4.6 Reasoning Quality and Grounding

To assess evidence-grounded reasoning, we conduct three evaluations summarized in Table 9. First, 92% of 150 localization/segmentation tool outputs are correct or partially correct. Second, among 200 correctly answered trajectories, 82% are fully supported and 12% partially supported by tool outputs, suggesting that answers rely on tool evidence rather than shortcuts. Third, MEDVISTA-R1 achieves higher accuracy when tools succeed but still retains fallback reasoning when tools fail or return invalid outputs. More details are provided in Appendix J.

Max Turns	SLAKE	VQA-RAD	PathVQA	PMC-VQA	MMMU	Avg.	End Round
T=1	79.66	68.87	68.50	57.00	55.00	65.81	–
T=3	**84.75**	69.09	**69.40**	56.50	**57.29**	**67.41**	2.30
T=6	81.36	**70.75**	69.00	**58.00**	56.43	67.11	2.57
T=9	82.07	70.50	69.00	57.50	56.80	67.17	2.38

Table 8: Performance and average end round under different maximum tool-call budgets T.

Category	Proportion	Count
(a) Tool Output Quality (150 trajectories)		
Correct	77.33%	116
Partially correct	14.67%	22
Incorrect	8.00%	12
(b) Evidence-Answer Consistency (200 trajectories)		
Fully supported by tool outputs	82%	164
Partially supported	12%	24
Unrelated to tool outputs	6%	12
(c) Robustness under Tool Execution		
Tool calls succeeded	65%	130
Tool calls failed/returned errors	48%	96

Table 9: Human evaluation of reasoning quality and evidence grounding: (a) tool output correctness, (b) evidence-answer consistency on correct trajectories, and (c) accuracy under tool success vs. failure.

4.7 Error Analysis and Case Study

Error Analysis. To understand why vanilla VLMs struggle with tool use (Table 4), we man-

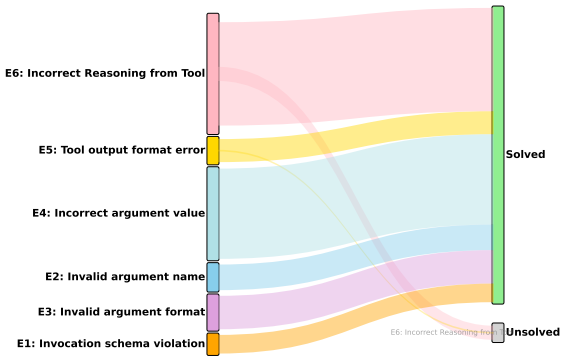


Figure 3: Error analysis.

ually analyze 100 error cases per benchmark for GPT-5 and InternVL3-8B. Common failures include mistimed or inappropriate tool calls, incorrect tool selection, malformed arguments, and faulty reasoning over tool outputs, indicating a lack of interaction priors for multi-step tool use. Re-evaluating the same samples with MEDVISTA-R1-8B substantially reduces these errors (Figure 3), showing that MEDVISTAGYM improves both tool-use discipline and reasoning over tool evidence.

Case Study. We summarize the capabilities acquired through agentic training in MEDVISTAGYM into four categories, illustrated with representative case studies: (1) schema- and argument-correct tool invocation (Appendix Figure 10); (2) effective post-tool reasoning grounded in evidence from tool outputs (Appendix Figure 11); (3) coordinated multi-tool usage for complex problem solving (Appendix Figure 12); and (4) robustness to imperfect tool outputs through multi-turn reasoning (Figure 13). Additionally, we analyze representative *hard cases* that reveal the boundaries of tool-integrated medical reasoning, where failures occur despite correct tool usage and coherent reasoning (Table 15).

5 Related Work

Medical Vision-Language Agents. Recent medical VLMs have been extended toward agent-like reasoning for tasks such as medical VQA, diagnosis, and radiology report generation. MedRAX (Fallahpour et al., 2025), CXR-Agent (Sharma, 2024), and CheXagent (Chen et al., 2024c) develop agentic or foundation-model frameworks for chest X-ray understanding and structured clinical reasoning. These systems show the promise of medical VLM agents but mainly target task-specific inference rather than scalable environment-based training for interactive tool use.

Tool-Integrated Medical Reasoning. External tools can enhance medical VLMs with specialized perception, retrieval, and quantitative analysis capabilities. Existing methods mainly fall into two directions: instruction-tuned tool-use systems, such as MMedAgent (Li et al., 2024a), VILA-M3 (Nath et al., 2025), and AURA (Fathi et al., 2025); and multi-step medical reasoning frameworks, such as Med-VRAgent (Guo et al., 2025b), TriageAgent (Lu et al., 2024), and MedAgent-Pro (Wang et al., 2025d). However, most rely on fixed tools, scripted workflows, or offline trajectories, limiting agent-environment training.

Agentic Training Environments. Agentic environments provide standardized interfaces for multi-step decision-making and reinforcement learning. General-purpose environments such as AgentGym-RL (Xi et al., 2025), Collaborative Gym (Shao et al., 2024), and RAGEN (Wang et al., 2025c) primarily target text-based agents, while multimodal environments such as VAGEN (Wang et al., 2025b) are designed for non-medical domains. In contrast, medical visual reasoning requires grounded perception, adaptive tool selection, and iterative verification over domain-specific images. MEDVISTAGYM fills this gap by providing a scalable Gym-style environment for tool-integrated medical VLM training, enabling closed-loop interaction with executable medical tools.

6 Conclusion

In this work, we introduce MEDVISTAGYM, a scalable agentic training environment for tool-integrated medical image analysis in VLMs, along with MEDVISTA-R1, an agent trained to interleave multi-turn reasoning with structured tool use. Our findings indicate that effective tool-integrated medical reasoning does not arise from tool access alone, but instead requires learning disciplined interaction policies. In particular, successful training relies on (i) explicitly interleaving reasoning with tool invocation, (ii) a two-stage paradigm that combines supervised warm-up with online reinforcement learning to refine multi-turn tool reasoning, and (iii) broad action-space coverage through diverse medical imaging tasks and tools to promote generalizable interaction behaviors. By providing unified interfaces, executable feedback, and efficient trajectory logging, MEDVISTAGYM serves as a learnable training substrate that supports systematic progress in medical image analysis.

Limitations

Although MEDVISTA-R1 achieve substantial improvements in tool-integrated medical visual reasoning using MEDVISTAGYM, several limitations remain. Despite optimization, multi-turn agentic training with frequent tool invocation is computationally expensive, and rollout depth and action-space size impose practical constraints on exhaustive exploration of all possible reasoning paths. The current environment and evaluation focus primarily on medical VQA tasks, and extending MEDVISTAGYM to other clinical reasoning paradigms or non-medical domains may require additional task and tool adaptation. Furthermore, tool effectiveness can be limited when tool outputs are noisy or when images exhibit low quality, subtle abnormalities, or highly complex visual patterns, which may still lead to incorrect reasoning.

Ethical Considerations

Ethical considerations are integral to this work. All experiments are conducted using publicly available datasets, and we employ open-source or widely adopted models without accessing any private patient data. Our study focuses on methodological advances in tool-integrated medical visual reasoning rather than clinical deployment, and model outputs should not be interpreted as medical advice. We emphasize transparency throughout the research process and advocate responsible use of these techniques with appropriate human oversight to ensure safe and beneficial application.

References

Anthropic. 2025a. Claude sonnet 4.5. https://www.anthropic.com/claude/sonnet. Accessed: 2026-01-04.

Anthropic. 2025b. Introducing claude haiku 4.5. https://www.anthropic.com/news/claude-haiku-4-5. Accessed: 2026-01-04.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

James Burgess, Jeffrey J Nirschl, Laura Bravo-Sánchez, Alejandro Lozano, Sanket Rajan Gupte, Jesus G. Galaz-Montoya, Yuhui Zhang, Yuchang Su, Disha Bhowmik, Zachary Coman, Sarina M. Hasan, Alexandra Johannesson, William D. Leineweber, Malvika G Nair, Ridhi Yarlagadda, Connor Zuraski, Wah Chiu, Sarah Cohen, Jan N. Hansen, and 4 others. 2025. Microvqa: A multimodal reasoning benchmark for microscopy-based scientific research. *Preprint*, arXiv:2503.13399.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. 2024a. Huatuogpt-o1: Towards medical complex reasoning with large language models. *arXiv preprint arXiv:2412.18925*.

Junying Chen, Chi Gui, Ruyi Ouyang, Anningzhe Gao, Shunian Chen, Guiming Hardy Chen, Xidong Wang, Ruifei Zhang, Zhenyang Cai, Ke Ji, Guangjun Yu, Xiang Wan, and Benyou Wang. 2024b. Huatuogpt-vision, towards injecting medical visual knowledge into multimodal llms at scale. *Preprint*, arXiv:2406.19280.

Zhihong Chen, Maya Varma, Justin Xu, Magdalini Paschali, Dave Van Veen, Andrew Johnston, Alaa Youssef, Louis Blankemeier, Christian Bluethgen, Stephan Altmayer, Jeya Maria Jose Valanarasu, Mohamed Siddig Eltayeb Muneer, Eduardo Pontes Reis, Joseph Paul Cohen, Cameron Olsen, Tanishq Mathew Abraham, Emily B. Tsai, Christopher F. Beaulieu, Jenia Jitsev, and 4 others. 2024c. A vision-language foundation model to enhance efficiency of chest x-ray interpretation. *Preprint*, arXiv:2401.12208.

Adibvafa Fallahpour, Jun Ma, Alif Munim, Hongwei Lyu, and Bo Wang. 2025. Medrax: Medical reasoning agent for chest x-ray. *Preprint*, arXiv:2502.02673.

Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Sravana Jyothi Narayanaraju, Xinze Guan, and Xin Eric Wang. 2025. Grit: Teaching mllms to think with images. *arXiv preprint arXiv:2505.15879*.

Nima Fathi, Amar Kumar, and Tal Arbel. 2025. Aura: A multi-modal medical agent for understanding, reasoning & annotation. *Preprint*, arXiv:2507.16940.

Google. 2025a. Gemini-2.5-flash-image model — google ai studio. https://aistudio.google.com/models/gemini-2-5-flash-image. Accessed: 2026-01-04.

Google. 2025b. Gemini 2.5 pro. https://aistudio.google.com/app/prompts/new_chat?model=gemini-2.5-pro. Accessed: 2026-01-04.

Debanjan Goswami, Ronast Subedi, and Shayok Chakraborty. 2025. MediVLM: A vision language model for radiology report generation from medical images. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10287–10304, Suzhou, China. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025a. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Guangfu Guo, Xiaoqian Lu, and Yue Feng. 2025b. Medvragent: A framework for medical visual reasoning-enhanced agents. *Preprint*, arXiv:2510.18424.

Xingang Guo, Utkarsh Tyagi, Advait Gosai, Paula Vergara, Ernesto Gabriel Hernández Montoya, Chen Bo Calvin Zhang, Bin Hu, Yunzhong He, Bing Liu, and Rakshith Sharma Srinivasa. 2025c. Beyond seeing: Evaluating multimodal llms on tool-enabled image perception, transformation, and reasoning. *arXiv preprint arXiv:2510.12712*.

Xuehai He, Yichen Zhang, Luntian Mou, Eric Xing, and Pengtao Xie. 2020. Pathvqa: 30000+ questions for medical visual question answering. *Preprint*, arXiv:2003.10286.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, Tianyu Pang, and Wenhu Chen. 2025. Verltool: Towards holistic agentic reinforcement learning with tool use. *Preprint*, arXiv:2509.01055.

Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. 2025a. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. *Preprint*, arXiv:2509.07969.

Yuxiang Lai, Jike Zhong, Ming Li, Shitian Zhao, Yuheng Li, Konstantinos Psounis, and Xiaofeng Yang. 2025b. Med-r1: Reinforcement learning for generalizable medical reasoning in vision-language models. *Preprint*, arXiv:2503.13939.

James Lau, Sam Gayen, Asma Ben Abacha, and 1 others. 2018. A dataset of clinically generated visual questions and answers about radiology images. *Scientific Data*, 5:180251.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Binxu Li, Tiankai Yan, Yuanting Pan, Jie Luo, Ruiyang Ji, Jiayuan Ding, Zhe Xu, Shilong Liu, Haoyu Dong, Zihao Lin, and Yixin Wang. 2024a. Mmedagent: Learning to use medical tools with multi-modal agent. *Preprint*, arXiv:2407.02483.

Binxu Li, Tiankai Yan, Yuanting Pan, Jie Luo, Ruiyang Ji, Jiayuan Ding, Zhe Xu, Shilong Liu, Haoyu Dong, Zihao Lin, and Yixin Wang. 2024b. MMedAgent: Learning to use medical tools with multi-modal agent. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8745–8760, Miami, Florida, USA. Association for Computational Linguistics.

Bo Li, Hao Zhang, Kaichen Zhang, Dong Guo, Yuanhan Zhang, Renrui Zhang, Feng Li, Ziwei Liu, and Chunyuan Li. 2024c. Llava-next: What else influences visual instruction tuning beyond data?

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. *Preprint*, arXiv:2306.00890.

Bo Liu, Li-Ming Zhan, Li Xu, Lin Ma, Yan Yang, and Xiao-Ming Wu. 2021. Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering. *Preprint*, arXiv:2102.09542.

Xinyao Liu and Diping Song. 2025. Constructing ophthalmic mllm for positioning-diagnosis collaboration through clinical cognitive chain reasoning. *Preprint*, arXiv:2507.17539.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. 2025. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*.

Meng Lu, Brandon Ho, Dennis Ren, and Xuan Wang. 2024. TriageAgent: Towards better multi-agents collaborations for large language model-based clinical triage. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5747–5764, Miami, Florida, USA. Association for Computational Linguistics.

Meng Lu, Ran Xu, Yi Fang, Wenxuan Zhang, Yue Yu, Gaurav Srivastava, Yuchen Zhuang, Mohamed Elhoseiny, Charles Fleming, Carl Yang, Zhengzhong Tu, Yang Xie, Guanghua Xiao, Hanrui Wang, Di Jin, Wenqi Shi, and Xuan Wang. 2025. Scaling agentic reinforcement learning for tool-integrated reasoning in vlms. *Preprint*, arXiv:2511.19773.

Vishwesh Nath, Wenqi Li, Dong Yang, Andriy Myronenko, Mingxin Zheng, Yao Lu, Zhijian Liu, Hongxu

Yin, Yucheng Tang, Pengfei Guo, Can Zhao, Ziyue Xu, Yufan He, Greg Heinrich, Yee Man Law, Benjamin Simon, Stephanie Harmon, Stephen Aylward, Marc Edgar, and 6 others. 2025. Vila-m3: Enhancing vision-language models with medical expert knowledge. *Preprint*, arXiv:2411.12915.

OpenAI. 2024. Gpt-4o mini: Advancing cost-efficient intelligence. https://openai.com/zh-Hans-CN/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. Accessed: 2026-01-04.

OpenAI. 2025. Gpt-5. https://openai.com/zh-Hans-CN/gpt-5/. 2025-12-23.

Jiazhen Pan, Che Liu, Junde Wu, Fenglin Liu, Jiayuan Zhu, Hongwei Bran Li, Chen Chen, Cheng Ouyang, and Daniel Rueckert. 2025. Medvlm-r1: Incentivizing medical reasoning capability of vision-language models (vlms) via reinforcement learning. *Preprint*, arXiv:2502.19634.

Sebastián Ramírez and FastAPI Contributors. 2018. Fastapi: High-performance web framework for building apis with python. https://github.com/fastapi/fastapi. Accessed: 2025-09-17.

Yijia Shao, Vinay Samuel, Yucheng Jiang, John Yang, and Diyi Yang. 2024. Collaborative gym: A framework for enabling and evaluating human-agent collaboration. *arXiv preprint arXiv:2412.15701*.

Naman Sharma. 2024. Cxr-agent: Vision-language models for chest x-ray interpretation with uncertainty aware radiology reporting. *Preprint*, arXiv:2407.08811.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, and 1 others. 2025. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*.

Haoran Sun, Yankai Jiang, Wenjie Lou, Yujie Zhang, Wenjie Li, Lilong Wang, Mianxin Liu, Lei Liu, and Xiaosong Wang. 2025. Chiron-o1: Igniting multimodal large language models towards generalizable medical reasoning via mentor-intern collaborative search. *Preprint*, arXiv:2506.16962.

LASA Team, Weiwen Xu, Hou Pong Chan, Long Li, Mahani Aljunied, Ruifeng Yuan, Jianyu Wang, Chenghao Xiao, Guizhen Chen, Chaoqun Liu, Zhaodonghui Li, Yu Sun, Junao Shen, Chaojun Wang, Jie Tan, Deli Zhao, Tingyang Xu, Hao Zhang, and Yu Rong. 2025. Lingshu: A generalist foundation model for unified multimodal medical understanding and reasoning. *Preprint*, arXiv:2506.07044.

Dujuan Wang, Tao Cheng, Sutong Wang, Youhua (Frank) Chen, and Yunqiang Yin. 2026. Smr-agents: Synergistic medical reasoning agents for zero-shot medical visual question answering with mllms. *Inf. Process. Manage.*, 63(1).

Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. 2025a. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *Preprint*, arXiv:2505.15966.

Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Chi Wan, Yiping Lu, Zhengyuan Yang, and 1 others. 2025b. Vagen: Reinforcing world model reasoning for multi-turn vlm agents. *arXiv preprint arXiv:2510.16907*.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, and 1 others. 2025c. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, and 1 others. 2025. Agentgym-rl: Training llm agents for long-horizon decision making through multi-turn reinforcement learning. *arXiv preprint arXiv:2509.08755*.

Peng Xia, Kangyu Zhu, Haoran Li, Tianze Wang, Weijia Shi, Sheng Wang, Linjun Zhang, James Zou, and Huaxiu Yao. 2025. Mmed-rag: Versatile multimodal rag system for medical vision language models. *Preprint*, arXiv:2410.13085.

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. 2025a. Llava-cot: Let vision language models reason step-by-step. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2087–2098.

Ran Xu, Yuchen Zhuang, Yishan Zhong, Yue Yu, Xiangru Tang, Hang Wu, May D Wang, Peifeng Ruan, Donghan Yang, Tao Wang, and 1 others. 2025b. Medagentgym: Training llm agents for code-based medical reasoning at scale. *arXiv preprint arXiv:2506.04405*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *Preprint*, arXiv:2311.16502.

Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2024. Pmc-vqa: Visual instruction tuning for medical visual question answering. *Preprint*, arXiv:2305.10415.

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and 1 others. 2025. Chain-of-focus: Adaptive visual search and zooming for multimodal reasoning via rl. *arXiv preprint arXiv:2505.15436*.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. 2025. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Dataset Details

A.1 Details of Curated Datasets

We curate our data following three core principles. (1) Diverse tasks and imaging distributions: we incorporate heterogeneous datasets to enhance model generalization. (2) Tool effectiveness: we prioritize scenarios where tool usage leads to measurable accuracy improvements. In generating reasoning trajectories, unlike prior work that first plans tool usage and then executes multiple tools in a predetermined order, we allow GPT-5 to operate directly within the MEDVISTAGYM environment and generate interactive reasoning trajectories through multiple rounds of real tool invocations. We retain only trajectories that produce correct final answers and further apply GPT-5 for self-verification. We believe this approach more faithfully captures the practical reasoning process involved in multi-tool collaboration. These trajectories are filtered through format validation and answer-correctness checks. The prompt template for constructing agentic reasoning trajectories and the prompt for trajectory verification are provided in Figure 5 and Figure 6, respectively. To further validate data quality, we manually inspected 150 randomly sampled trajectories from the final training set: 89.33% exhibit appropriate tool selection, coherent reasoning, and accurate medical terminology, while the remaining ∼10% involve only minor issues such as suboptimal tool ordering rather than factual errors or hallucinations.

Datasets	VQA-RAD	SLAKE	PathVQA	PMC-VQA	MMMU (H&M)	MicroVQA
Train	3500	3500	3500	0	0	0
Test	400	400	400	400	200	300

Table 10: Data statistics. The three in-distribution datasets (VQA-RAD, SLAKE, PathVQA) follow their official train/test splits. The three out-of-distribution datasets (PMC-VQA, MMMU H&M, MicroVQA) are used only for OOD evaluation.

A.2 Dataset Details

We conduct experiments on six representative datasets. SLAKE (Liu et al., 2021),PathVQA (He et al., 2020) and VQA-RAD (Lau et al., 2018) are widely used benchmarks in medical VQA research. For higher-level medical reasoning, we further evaluate PMCVQA (Zhang et al., 2024), a generative medical visual question answering benchmark that requires models to perform fine-grained visual understanding and domain-specific reasoning beyond fixed answer classification, MMMU(H&M) (Yue et al., 2024) which focuses on expert-level multimodal understanding and reasoning over complex medical imagery, and MicroVQA (Burgess et al., 2025), which emphasizes fine-grained, region-level visual reasoning. The statistics of the datasets are shown in Figure 10. For evaluation metrics,we use answer accuracy for multiple-choice VQA benchmarks. MMMU(H&M) (Yue et al., 2024) is a subset extracted from the multimodal reasoning benchmark MMMU (Yue et al., 2024).

B Implementation Details

We provide detailed training and inference configurations to implementation.

B.1 Training Configuration

Backbones. We instantiate MEDVISTA-R1 on three open-source VLM backbones to evaluate cross-family generalization: InternVL3-2B, InternVL3-8B (Zhu et al., 2025), Qwen2.5-VL-7B (Bai et al., 2025b), and Qwen3-VL-8B (Bai et al., 2025a). All backbones are loaded from their official HuggingFace checkpoints in bfloat16 precision.

Stage I: Cold-Start SFT. We perform full-parameter supervised fine-tuning on 3.3K curated tool-augmented reasoning trajectories. We train for 5 epochs with AdamW (learning rate 1×10^{-5}, weight decay 0.01, cosine schedule with 3% warmup), a global batch size of 256, and a maximum sequence length of 2.6K tokens. Loss is computed only on reasoning and tool-call tokens;

tool-executed observation tokens are masked out. The SFT stage takes approximately 1.5 hours on 2 NVIDIA A100 80GB GPUs.

Stage II: GRPO Online RL. We use Verl-Tool (Jiang et al., 2025) for multi-turn rollouts with executable tool calls. For each question, we sample $G = 8$ candidate trajectories, with up to 6 tool invocations per trajectory. We adopt a constant learning rate of 1×10^{-6}, a global batch size of 256, and a maximum context length of 2.6K tokens. The clipping coefficient is $\epsilon = 0.2$. Observation tokens returned by tools are excluded from gradient computation. Training runs on 6 NVIDIA A100 80GB GPUs for approximately 36 hours.

Reward Hyperparameters. The final rollout reward is $R = R_{\text{format}} + R_{\text{acc}} + R_{\text{tool}}$, where $R_{\text{format}} \in \{0, 1\}$, $R_{\text{acc}} \in \{0, 1\}$, and $R_{\text{tool}} \in \{0, 2\}$ (granted only when both tool invocation occurs and the final answer is correct). Full reward formulations are provided in Appendix D.

Early Termination. A rollout terminates when (i) the agent emits an `<answer>` block, (ii) the maximum tool-call budget (6) is reached, or (iii) a tool invocation repeats a previously executed action with identical arguments, to discourage degenerate looping.

B.2 Inference Configuration

Decoding. At inference time we use greedy decoding (temperature 0) for all evaluation runs to ensure deterministic comparison across baselines and ablations.

Tool Serving. External tools are deployed as independent FastAPI (Ramírez and Contributors, 2018) microservices, with each tool exposed via an asynchronous RESTful endpoint. Compute-intensive tools (GroundingDINO, BiomedParse, MedSAM2, BiomedCLIP, Agent4K) are pinned to dedicated GPU workers, while lightweight knowledge-retrieval tools (DrugBank, PubMed) run on shared CPU resources. Tool invocations are batched and orchestrated through Ray actors with automatic failure recovery. Full infrastructure details are provided in Appendix I.2.

B.3 Data Splits and Leakage Prevention

Train/Test Splits. Table 10 summarizes the dataset statistics. For the three in-distribution benchmarks (VQA-RAD, SLAKE, PathVQA), we follow the official train/test splits and use 3,500 training samples per dataset. The three out-of-distribution benchmarks (PMC-VQA, MMMU H&M, MicroVQA) are held out entirely and are never seen during either SFT or GRPO training.We follow the official train/test splits of VQA-RAD, SLAKE, and PathVQA for in-distribution training and evaluation.

Leakage Prevention in Trajectory Curation. When generating SFT trajectories with GPT-5 in the MEDVISTAGYM environment, we strictly use only training-split questions and images. Test-set samples are never exposed to the trajectory generator or to any tool during training. Knowledge retrieval tools (DrugBank, PubMed, GoogleSearch) are disabled during evaluation on all six benchmarks to avoid any possibility of answer leakage via external retrieval; only visual perception and segmentation tools remain available at inference time.

B.4 Open-Ended VQA Scoring

For the open-ended evaluation in Table 3, we report the average of two complementary semantic-similarity metrics computed between each model prediction $\hat{y}$ and the ground-truth answer y. We use (i) **BERTScore F1** ($s_{\text{BS}} \in [0, 1]$) computed with PubMedBERT (Gu et al., 2021), and (ii) **cosine similarity** ($s_{\text{Bio}} \in [-1, 1]$) between mean-pooled BioBERT-NLI embeddings (Lee et al., 2019). Both predictions and ground-truth answers are lower-cased and stripped of trailing punctuation before scoring; no other normalization is applied. The reported open-ended accuracy is the unweighted mean of the two scores scaled by 100:

$$\text{Acc}_{\text{open}}(\hat{y}, y) = \frac{s_{\text{BS}}(\hat{y}, y) + s_{\text{Bio}}(\hat{y}, y)}{2} \times 100. \tag{1}$$

We use this composite metric rather than BLEU/ROUGE because lexical-overlap metrics are well known to penalize clinically equivalent paraphrases (e.g., "CXR" vs. "chest X-ray") in medical text.

B.5 Human Evaluation Protocol

The three human evaluations summarized in Table 9 were conducted by two annotators with biomedical research backgrounds (one with a graduate degree in biomedical sciences and one with clinical radiology experience). For each evaluation, trajectories were uniformly sampled across in-distribution benchmarks, and annotators were

blinded to the system identity (i.e., they did not know whether the trajectory came from a baseline or from MEDVISTA-R1). Annotators independently labeled each sample using a written guideline, and disagreements were resolved through discussion. Cohen's κ inter-annotator agreement was 0.81 for tool-output quality, 0.78 for evidence-answer consistency, and 0.85 for the success/failure classification, all in the substantial-agreement range. Full annotation guidelines will be released alongside the code.

C Additional Experimental Results Analysis

C.1 Effect of Tool-Integrated Reasoning

Table 1 further disentangles the contributions of reasoning and tool use by comparing the following controlled variants on InternVL3-8B (Overall accuracy): (1) *SFT + GRPO w/o Tools* (52.68%), which retains multi-step reasoning through reinforcement learning while disabling all external tool access during both training and inference; (2) *SFT w/ Tools* (56.70%), which exposes the model to external tools using a fixed, predefined invocation format learned during cold-start supervision, but removes the reinforcement learning stage that induces adaptive reasoning over tool usage; (3) *SFT w/o Tools* (54.38%), which performs only supervised fine-tuning without tool access or reinforcement learning; (4) *GRPO w/ Tools from Base* (50.74%), which initializes reinforcement learning directly from the base backbone without cold-start supervision, allowing tool access but without prior exposure to tool syntax or invocation patterns, leading to unstable exploration in the large action space. All four variants underperform the full MEDVISTA-R1 configuration (*SFT + GRPO w/ Tools*, 63.09%), confirming that effective tool-integrated medical reasoning requires the joint combination of tool access, supervised warm-up, and online reinforcement learning, rather than any single component in isolation.

C.2 Additional Baselines

As shown in Table 2, we obtain the following observations:

Compared with recent SOTA general-purpose VLMs, MEDVISTA-R1 outperforms DeepEyes-7B (Zheng et al., 2025) with tools (e.g., zoom-in), Mini-o3-7B (Lai et al., 2025a), and PixelReasoner (Wang et al., 2025a) by an average of 11.50%, demonstrating superior performance and highlighting the effectiveness of environment-driven tool interactions.

Compared with SOTA medical-specific reasoning VLMs, MEDVISTA-R1 outperforms LLaVa-med-7B (Li et al., 2023) 25.43%, HuatuoGTP-vision-34B (Chen et al., 2024a) 13.47%, Chiron-o1-8B (Sun et al., 2025) 0.80%, and Lingshu-7B (Team et al., 2025) 2.83%, demonstrating that integrating external tools enables MLLMs to acquire and utilize fine-grained visual cues for medical image reasoning more effectively.

Compared with prior medical agent frameworks, including MMedAgent-7B (Li et al., 2024a), VILA-M3-40B (Nath et al., 2025) and MMedAgent-RL-7B (Li et al., 2024b), MEDVISTA-R1 outperform by average of 8.35%, which demonstrates a substantial margin of improvement. Unlike approaches that treat tool usage as isolated function calls, our agent–environment framework enables structured, multi-turn interaction with external medical tools, producing verifiable reasoning trajectories that support coherent multi-tool composition and more faithful visual grounding.

C.3 Effect of Backbone Models

To further verify that the gains from MEDVISTAGYM are not tied to a specific VLM family, we apply the same two-stage training pipeline (cold-start SFT followed by GRPO with tools) to an additional backbone, Qwen2.5-VL-7B (Bai et al., 2025b). We evaluate on the same three in-distribution benchmarks (SLAKE, VQA-RAD, PathVQA) and three out-of-distribution benchmarks (PMC-VQA, MMMU (H&M), MicroVQA) used in the main experiments. Results are reported in Table 11.

We observe a consistent pattern with the InternVL3 and Qwen3-VL results in Table 1. First, **naive tool access without training is harmful**: directly placing Qwen2.5-VL-7B in MEDVISTAGYM without any training drops accuracy substantially across all six benchmarks (e.g., 42.11% $\rightarrow$ 15.67% on SLAKE and 64.14% $\rightarrow$ 26.44% on VQA-RAD), confirming that vanilla VLMs lack the interaction priors needed to invoke and reason over medical tools. Second, **cold-start SFT recovers and slightly exceeds baseline performance** (e.g., 64.14% $\rightarrow$ 66.13% on VQA-RAD, 49.00% $\rightarrow$ 50.35% on PMC-VQA), indicating that supervised warm-up teaches valid tool syntax and selection but yields only modest gains on

Method	SLAKE	VQA-RAD	PathVQA	PMC-VQA	MMMU	MicroVQA
Qwen2.5-VL-7B (Vanilla)	42.11	64.14	62.40	49.00	46.43	34.38
MEDVISTAGYM (Qwen2.5-VL-7B)	15.67	26.44	41.37	31.06	37.49	24.15
Qwen2.5-VL-7B SFT + Tool	43.48	66.13	63.15	50.35	48.38	37.63
Qwen2.5-VL-7B GRPO + Tool	**70.50**	**68.52**	**64.50**	**51.50**	**54.29**	**43.86**

Table 11: Backbone generalization on Qwen2.5-VL-7B. The same training pipeline improves performance across a different VLM family.

its own. Third, **adding GRPO on top of SFT delivers the largest improvements**, lifting SLAKE from 42.11% to 70.50% (+28.39), VQA-RAD from 64.14% to 68.52% (+4.38), PathVQA from 62.40% to 64.50% (+2.10), PMC-VQA from 49.00% to 51.50% (+2.50), MMMU from 46.43% to 54.29% (+7.86), and MicroVQA from 34.38% to 43.86% (+9.48). The full SFT + GRPO + Tools setting achieves the best score on every benchmark.

These results mirror the trends observed on InternVL3-8B and Qwen3-VL-8B and reinforce three conclusions: (i) tool access alone is insufficient and can actively degrade performance without structured training, (ii) cold-start SFT is necessary but not sufficient, and (iii) online RL inside MEDVISTAGYM is the key driver of effective tool-integrated medical reasoning. The consistency of this pattern across three independent VLM families (InternVL3, Qwen3-VL, and Qwen2.5-VL) supports the claim that MEDVISTAGYM's benefits stem from the training environment and protocol rather than from backbone-specific effects, validating its role as a general-purpose, model-agnostic training substrate for agentic medical VLMs.

C.4 Evaluation Details

Open-ended VQA evaluation For open-ended subsets of SLAKE, VQA-RAD, PathVQA, and VQA-MED, we measure the semantic similarity between each predicted answer and the ground truth using two medical-domain embedding models: BERTScore F1 computed with PubMedBERT (microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract) and cosine similarity between mean-pooled BioBERT-NLI embeddings (pritamdeka/BioBERT-mnli-snli-scinli-scitail-mednli-stsb). We report the mean of these two scores (×100) as the open-ended accuracy. This setting avoids the well-known sensitivity of BLEU/ROUGE to clinically equivalent paraphrases (e.g., "CXR" vs. "chest X-ray") that plagues lexical metrics in medical text.

D Reward Details

We design the reward function to provide rich and fine-grained feedback signals. The individual reward components are described below.

The reasoning-action format reward The format reward S_{format} evaluates the structural validity of R by verifying that the model output contains all required special tokens in the prescribed order: <think></think><tool_call></tool_call> <think></think><answer></answer>. Specifically, before invoking a tool, the model must enclose its tool-selection reasoning within <think></think> tags and place the tool-call json between <tool_call></tool_call> tags. After tool execution, instead of directly producing the final answer, the model is required to include an additional reasoning step enclosed by <think></think> and then output the final answer within <answer></answer>. Outputs that strictly follow this structured reasoning–action format receive a positive reward.

The final-answer accuracy reward The final-answer accuracy reward S_{ans} evaluates whether the predicted answer matches the ground-truth answer.

The answer-conditioned tool-use reward To align with the interaction paradigm in MEDVISTAGYM, where correct reasoning trajectories are generated through multiple rounds of tool invocation, we introduce a conditional reward mechanism that grants an additional reward only when the model both appropriately invokes external tools during the trajectory and produces a correct final answer. This design encourages meaningful tool usage when tools substantively contribute to successful task completion, rather than arbitrary or redundant tool calls.

E Optimization Details

We optimize the agent in Stage II (Section 3.2) using Group Relative Policy Optimization (GRPO) (Guo et al., 2025a), which performs policy updates based on relative performance among a group of sampled rollouts.

Multi-turn Rollouts Given an input (I, Q), we sample a group of G multi-turn interaction trajectories $\{\tau_1, \ldots, \tau_G\}$ from the current policy $\pi_{\theta_{\text{old}}}$. Each trajectory $\tau_i = (u_{i,1}, \ldots, u_{i,T_i})$ consists of interleaved reasoning tokens, tool-call tokens, and the final answer, following the rollout formulation defined in Section 3.2. Each trajectory is assigned a scalar rollout-level reward $R(\tau_i)$ according to the reward design in Section 3.2.

Group-Normalized Advantage To emphasize relative quality among sampled reasoning paths, we compute a group-normalized advantage for each trajectory:

$$A_i = \frac{R(\tau_i) - \text{mean}(\{R(\tau_1), \ldots, R(\tau_G)\})}{\text{std}(\{R(\tau_1), \ldots, R(\tau_G)\})}. \quad (2)$$

This normalization encourages the policy to prefer trajectories that perform better than others within the same rollout group, rather than relying on absolute reward magnitudes.

Token-Level GRPO Objective Policy optimization is performed at the token level. For the k-th token in trajectory τ_i, we define the importance ratio:

$$r_{i,k}(\theta) = \frac{\pi_\theta(\tau_{i,k} \mid \tau_{i,<k})}{\pi_{\theta_{\text{old}}}(\tau_{i,k} \mid \tau_{i,<k})}. \quad (3)$$

The GRPO objective is then defined as:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \frac{1}{G} \sum_{i=1}^{G} \frac{1}{|\tau_i|} \sum_{k=1}^{|\tau_i|} \min \Big(r_{i,k}(\theta) \cdot A_i, \; \text{clip}(r_{i,k}(\theta), 1-\epsilon, 1+\epsilon) \cdot A_i \Big). \quad (4)$$

where $|\tau_i|$ denotes the number of trainable tokens in trajectory τ_i, excluding tool-executed observations.

This objective updates the policy to increase the likelihood of tokens belonging to higher-quality reasoning trajectories while maintaining stable updates through clipping.

F Tools Details

In MEDVISTAGYM, we enable access to 15 tools organized into four complementary families.

Resolution and Region Refinement This family enables focused inspection and quality enhancement of image regions by improving visual fidelity and local details.
4KAgent is an agentic image enhancement system that supports super-resolution upscaling with configurable scale factors (2×/4×/8×/16×) using models such as HAT-PSNR, DiffBIR, and OSEDiff. It further provides dehazing (DehazeFormer, RIDCP, MAXIM), denoising (NAFNet, Restormer, SwinIR), and image brightening (CLAHE, Gamma Correction, FourierDiff), allowing agents to recover fine-grained visual cues from degraded or low-quality medical images.

Medical Localization and Segmentation This family supports the detection and delineation of anatomical and pathological regions, providing region-level evidence for downstream reasoning.
GroundingDINO performs open-set, text-conditioned object detection, localizing anatomical structures or pathological regions described by natural-language queries.
SAM2 is a promptable foundation model for image and video segmentation that enables zero-shot segmentation using points, bounding boxes, or masks.
BiomedParse is a unified biomedical image parsing model that segments organs and pathological regions across diverse medical imaging modalities, supporting organ-level, lesion-level, comprehensive, and text-prompted segmentation.
MedSAM2 is a medical-domain adaptation of the Segment Anything Model, optimized for high-precision anatomical segmentation with improved boundary accuracy using bounding-box prompts and organ names.

Medical Visual Understanding and Parsing This family provides higher-level semantic interpretation of medical images.
BiomedCLIP is a vision-language foundation model pre-trained on large-scale biomedical image–text pairs. It enables zero-shot medical image classification across multiple label types (e.g., abnormality, modality, organ) by computing vision–language similarity scores, producing structured semantic signals for reasoning.

External Biomedical Knowledge Retrieval In MEDVISTAGYM, agents may retrieve external medical knowledge from resources such as *DrugBank* and *PubMed*, allowing integration of visual findings with domain-specific medical knowledge when required.

G Additional Ablation Analysis

In this section, we provide additional ablation studies to further analyze the behavior and generalization properties of our framework. Specifically,

Category	Main Function	Specific Tool	Key Capability
Image Enhancement	4KAgent	`super_resolution`	Super-resolution upscaling (2×–16×)
		`dehazing`	Haze and fog removal
		`denoising`	Noise artifact elimination
		`brightening`	Low-light enhancement
Perception	GroundingDINO	`groundingdino`	Open-set object detection
	SAM2	`sam2`	Universal promptable segmentation
Medical Analysis	BiomedCLIP	`biomedclip`	Zero-shot medical classification
	BiomedParse	`biomedparse_organ`	Anatomical structure segmentation
		`biomedparse_lesion`	Pathological region detection
		`biomedparse_all`	Comprehensive target segmentation
		`biomedparse_text`	Custom text-prompted segmentation
	MedSAM2	`medsam2`	Prompt-based medical segmentation
Knowledge Retrieval	GoogleSearch	`google_search`	Real-time web search
	DrugBank	`drugbank`	Pharmaceutical knowledge lookup
	LongDocRAG	`longdocrag`	Long document QA

Table 12: Summary of tools in MEDVISTAGYM. The **Main Function** column lists the underlying model or service; the **Specific Tool** column gives the executable tool identifier exposed to the agent.

we examine: (1) whether the model can reliably invoke tools with correct formatting and accurate tool selection, and whether training effectively eliminates unstructured or erroneous tool calls (§G.1); (2) how the quantity and type of tools influence agent performance, including the scaling effects of tool composition (§G.2, §G.3); (3) whether the learned tool-use policy generalizes to unseen tools without additional training (§G.2); and (4) how increasing model size further amplifies the benefits of tool-integrated reasoning (§G.4).

G.1 The Improvement of Tool Invocation

Figure 3 further measures the reliability of tool invocation by jointly checking format adherence and correct tool calling, offering a complementary perspective to the error analysis discussed earlier. The vanilla model without tool-use training exhibits poor reliability, achieving only a 24.2% accuracy rate across evaluation benchmarks. In contrast, our trained model achieves near-ceiling performance with a 98.96% accuracy rate, demonstrating consistently precise and reliable tool-use behavior. The absolute improvement of approximately 75% highlights the effectiveness of our training paradigm. These results confirm that the model learns to reliably and proactively invoke appropriate tools when solving problems. This further corroborates the importance of our carefully designed training strategy for eliciting robust tool-calling capabilities—intelligent behaviors that mere prompting of a base model cannot achieve.

G.2 Impact of Tool Quantity and Type

We investigate the ability of our model to utilize out-of-domain (OOD) tools in a training-free setting, evaluated on both in-domain datasets and an OOD benchmark. For fair comparison with current SOTA baselines, we evaluate each agent under a single-tool setting. As shown in Table 13, when only the zoom-in tool is available, MEDVISTA-R1 maintains a clear performance advantage on the SLAKE dataset. This demonstrates that MEDVISTAGYM supports deeply tool-integrated reasoning even under severely constrained tool access. When provided with an unseen tool, SAM2, which is functionally similar to MedSAM, MEDVISTA-R1 achieves performance on SLAKE that is comparable to using seen tools, highlighting strong generalization across similar tool types.

G.3 Scaling of Tools Yields Consistent Gains

As shown in Table 13, overall performance improves steadily as the model gains access to a larger set of tools within the environment. Under single-tool settings, BiomedParse yields the best performance (73.74%), clearly surpassing zoom-in only (69.50%) and MedSAM only (72.51%). Moreover, augmenting BiomedParse with MedSAM results in a further (+)1.78% performance gain, demonstrating that MEDVISTA-R1, after training in MEDVISTAGYM, is able to effectively coordinate multiple tool invocations and perform tool-integrated reasoning.

G.4 Scaling Model Sizes Yields Consistent Gains

We analyze the effect of model size on performance within MEDVISTAGYM. As shown in Table 1, scaling the backbone from 2B to 8B leads to consistent performance gains across benchmarks. In particular, MEDVISTA-R1-8B outperforms its 2B counterpart in average accuracy, with larger improvements on reasoning-intensive datasets that require multi-step visual evidence aggregation and tool-conditioned decision making. This indicates that larger models can more effectively support long-horizon, tool-integrated reasoning within MEDVISTAGYM.

Notably, despite its smaller parameter count, MEDVISTA-R1-2B matches or even surpasses several 8B baselines. This highlights the strong parameter efficiency of our framework and suggests that the observed gains stem not merely from increased model size, but from the synergy between agentic reasoning, structured tool use, and reinforcement learning. Scaling the backbone further amplifies these advantages, yielding reliable and consistent improvements rather than qualitatively different behaviors.

Table 13: Ablation on Tool Quantity and Type. Tools are grouped into Seen (seen during training) and Unseen (held out). A checkmark (✓) indicates a used tool. Tools # denotes the total number of tools actually used in this setting. Avg. is the mean over available metrics per row.

Method	Tools #	Tools (used in this setting)						Benchmarks		
		Seen					*Unseen*	*VQA tasks*		Avg.
		Zoom-in	MedSAM	BioMed Parse	Biomed CLIP	4KAgent	SAM2	VQA-RAD	SLAKE	
DeepEyes-7B	1	✓						65.9	68.2	67.1
Mini-o3-7B-v1	1	✓						65.7	67.8	66.8
PixelReasoner-RL-v1-7B	1	✓						66.0	67.3	66.7
MMedAgent-MedSAM	1		✓					64.0	68.7	66.4
MEDVISTA-R1	1	✓						65.1	69.5	67.3
	1						✓	64.5	72.8	68.7
	1		✓					65.1	72.5	68.8
	1			✓				66.9	73.5	70.2
	2		✓	✓				67.1	75.2	71.2
	2	✓	✓					67.6	76.0	71.8
	2	✓		✓				68.4	77.3	72.9
MEDVISTA-R1	**5**	✓	✓	✓	✓	✓		**70.8**	**79.7**	**75.3**

Overall, these results demonstrate that MEDVISTAGYM serves as a scalable training ground for medical VLM agents. Model scaling functions as a complementary factor that enhances agentic, tool-integrated reasoning, rather than acting as the primary driver of performance gains.

G.5 Effect of Thinking Trajectory Length

We analyze the relationship between thinking trajectory length and downstream performance under the vanilla with tool setting. As shown in Figure 4, we report the average reasoning trajectory length of GPT-5, Qwen3VL-8B, and InternVL3-8B across the In-Distribution, Out-of-Distribution, and Overall settings.

We observe that GPT-5 consistently produces substantially longer reasoning trajectories across all three settings, whereas Qwen3VL-8B and InternVL3-8B generate much shorter and comparable trajectories. Consistent with this trend, GPT-5 achieves markedly higher performance in all evaluation scenarios, with the most pronounced gains in the In-Distribution and Overall settings. In contrast, although Qwen3VL-8B and InternVL3-8B are also capable of invoking external tools, their shorter reasoning trajectories are associated with more limited performance improvements.

These results suggest that, in the absence of agentic reinforcement learning, tool availability alone is insufficient to guarantee performance gains. Instead, the ability to generate sufficient and well-structured intermediate reasoning trajectories plays a critical role in effectively leveraging tools for complex problem solving. This observation further validates our design choice of constructing explicit thinking trajectories using GPT-5 with tools in the MEDVISTAGYM as supervision for SFT, providing high-quality reasoning traces that are difficult to elicit from vanilla open-source models.

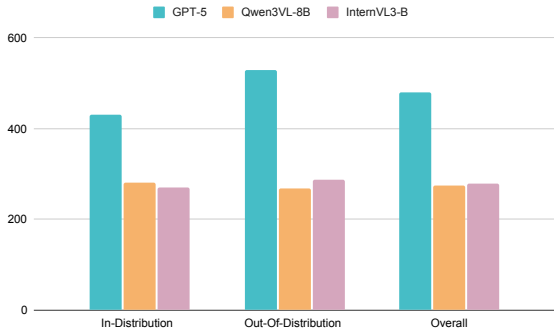


Figure 4: Effect of thinking trajectory length.

H Prompt Details

H.1 Prompt For Reasoning Trajectory Construction and Filtering

In this section, we introduce the prompt design for constructing reliable agentic reasoning trajectories over our curated VQA data. Unlike generic approaches that directly sample from a teacher model, and unlike methods that predefine a static tool set for GPT-5 or other large models, we provide GPT-5 with ground-truth clinical metadata and executable tool-calling interfaces during prompting, enabling

System Prompt for Trajectory Evaluation:
You are an expert evaluator for tool-augmented medical VQA reasoning trajectories.

Context:
- The final answer is ALREADY CORRECT and FIXED. Do NOT evaluate answer correctness.
- All tool executions completed successfully (though outputs may or may not be useful).
- Your job is to evaluate the QUALITY OF REASONING in the generated think segments.

Evaluation Dimensions:
1. Pre-tool Reasoning Quality (for each tool call): Does it clearly explain WHY this specific tool is being called? Is the reasoning for tool selection appropriate? Does it avoid "hindsight bias"? Score: 1 (poor) - 5 (excellent)
2. Tool Selection Appropriateness: Is the chosen tool appropriate for the stated purpose? Are the tool arguments reasonable for the medical context? Score: 1 (poor) - 5 (excellent)
3. Post-observation Reasoning Quality: Does it correctly interpret what the tool returned? If the tool output was unhelpful, does the reasoning acknowledge this failure? Score: 1 (poor) - 5 (excellent)
4. Final Reasoning to Answer: Does it properly synthesize all gathered evidence? Is the logical chain from observations to conclusion clear? Score: 1 (poor) - 5 (excellent)
5. Overall Coherence: Is the reasoning flow natural and logical throughout? Is the medical terminology used correctly? Score: 1 (poor) - 5 (excellent)

Critical Issues to Flag: `hindsight_bias`, `hallucination`, `tool_failure_ignored`, `logic_gap`, `inconsistent_reasoning`

Output Format (strict JSON):
```
{"evaluation": {"pre_tool_reasoning": {"score": 1-5, "comments": "..."}, "tool_selection": {...}, "post_observation_reasoning": {...}, "final_reasoning": {...}, "overall_coherence": {...}}, "issues_found": [...], "overall_score": 1-5, "quality_tier": "excellent/good/acceptable/needs_improvement/poor", "summary": "..."}
```

IMPORTANT: Output ONLY valid JSON. No markdown, no code blocks, no extra text.

Figure 5: Prompt template for trajectory quality evaluation.

real tool interaction within the MEDVISTAGYM environment. At each step, the model triggers a tool call based on the current action and determines the subsequent action according to the returned observation, with tool outputs fed back to the model immediately after each interaction.

Throughout the entire environment interaction and reasoning process, the agent is explicitly constrained to treat the final answer as unknown, which is revealed only during the verification stage. Combined with verified medical annotations, this prompt design enables multi-turn rollouts with environment interaction and tool use to construct step-by-step reasoning trajectories. Figure 9 illustrates the generation template, while Figure 5 presents the trajectory quality evaluation and filtering where GPT-5 assigns fine-grained scores to different aspects of the reasoning process, including tool-use justification, effective integration of observations, and exact answer matching.

H.2 System Prompt and User Prompt

We present the two prompt templates that we used in our experiments. Figure 7 shows the System Prompt template. Figure 8 shows the User Prompt template

I Interface and Infrastructure Details

I.1 Agent-Environment Interface Details

MEDVISTAGYM is an executable training environment built upon a Gym-style interaction protocol and specifically designed for medical image analysis. It provides a flexible *API interface*, an explicit and verifiable *action space*, and a structured *observation space*, enabling agents to perform multi-step reasoning through continuous interaction with the environment in medical settings.

Executable Medical Interface. MEDVISTAGYM provides two core interface functions, `reset()` and `step()`. Calling `reset()` initializes a new interaction episode and returns the initial observation o_0, which contains the current medical question along with the associated medical images. Each episode corresponds to a complete and independent agent–environment interaction instance.

Executable Medical Action Space The action space $\mathcal{A}$ is strictly restricted to the set of executable medical tools defined in MEDVISTAGYM. Each action $a_t \in \mathcal{A}$ is formalized as a typed tuple that explicitly specifies the selected tool identifier and its corresponding arguments, which are passed to the appropriate tool interface for execution. This design ensures that all agent actions are explicitly defined, executable, and verifiable.

Medical Evidence Observation Space After an

action a_t is executed, the environment returns an observation $o_t \in \mathcal{O}$ containing structured tool outputs—such as region localizations, segmentation masks, quantitative measurements, or retrieved medical facts—as well as potential execution-time error messages. These observations serve as external evidence that supports subsequent medical reasoning and decision-making.

I.2 Scalable Execution Infrastructure Details

To enhance fine-grained medical visual perception and domain grounding, we incorporate compute-intensive medical foundation models as interactive tools, including high-resolution visual encoders and medical segmentation models.

To support large-scale, multi-turn medical visual reasoning, we design a scalable execution infrastructure that encapsulates computationally intensive medical foundation models as interactive tools. This design enhances fine-grained medical visual perception and domain grounding by enabling on-demand invocation of high-resolution visual encoders and medical image segmentation models, which provide reliable intermediate visual evidence during reasoning. All tools are deployed as independent services to accommodate the high-frequency invocation required by multi-turn agent–environment interaction. The system adopts a highly concurrent microservice architecture, where each tool is encapsulated as an HTTP service and organized into three functional layers: (1) a **FastAPI**-based interface layer that exposes asynchronous and batched RESTful endpoints; (2) a **Tool logic layer** that parses agent-issued tool-call instructions, retrieves the corresponding medical images from episode or trajectory metadata, and formats tool outputs into structured medical observations; and (3) a **Ray Actor execution layer** that keeps model weights resident in GPU memory after initialization, thereby avoiding repeated model loading under high-frequency tool invocations and significantly improving execution efficiency.

Asynchronous Tool-augmented Training To sustain high throughput during RL rollouts, we employ Ray to coordinate asynchronous execution between agents and tools. At each decision step, the policy first generates an explicit reasoning segment(between <think>...</think>), followed by a tool invocation (<tool_call>. . . </tool_call>). When the model emits the tool-call termination token </tool_call>, decoding is temporarily paused, and the framework aggregates tool requests—containing trajectory identifiers and image paths—into batched HTTP calls. Ray manages request queues and performs load balancing across different tool services. To improve resource efficiency, compute-intensive medical vision tools are pinned to dedicated GPUs, while lightweight utilities and knowledge-retrieval tools (e.g., `DrugBank` and `PubMed`) are multiplexed on shared CPU resources.

Extensible Tool Infrastructure To facilitate rapid extension beyond the tools used in our experiments, MEDVISTAGYM provides a unified `BaseTool` abstraction that enables *plug-and-play* integration of new medical perception or knowledge tools with minimal engineering overhead. This design significantly reduces the complexity of tool expansion and maintenance, supporting the continuous evolution of the environment's capabilities. System robustness is ensured through standardized health-check and monitoring endpoints (`/health`, `/metrics`), together with Ray's automated failure recovery mechanism. In the event of tool or execution failures, Ray transparently restarts affected actors without disrupting ongoing training, thereby maintaining training continuity and reliability.

J Reasoning Quality and Evidence Grounding Details

To verify that MEDVISTA-R1's performance gains stem from genuine evidence-grounded reasoning rather than answer shortcuts or spurious correlations, we conduct three complementary human evaluations targeting different aspects of the reasoning pipeline: tool output quality, evidence-answer consistency, and robustness under tool failure. Results are summarized in Table 9. All annotations were performed by two trained annotators with biomedical backgrounds; disagreements were resolved through discussion.

Tool Output Quality. Since the medical VQA benchmarks used in our evaluation do not provide pixel-level segmentation ground truth, standard quantitative metrics such as Intersection-over-Union (IoU) cannot be directly computed. To assess whether tool outputs are clinically meaningful, we manually inspect 150 trajectories that invoke localization or segmentation tools (e.g., GroundingDINO, BiomedParse, MedSAM2), sampled uniformly across in-distribution benchmarks. Each tool output is categorized into one of three labels: *correct* (the tool correctly localizes or segments

the target anatomical or pathological region), *partially correct* (the output covers part of the target but includes spurious regions or omits portions of the target), or *incorrect* (the output is entirely unrelated to the target). As shown in Table 9(a), 77.33% of outputs are fully correct and 14.67% are partially correct, yielding 92% of outputs that provide at least partially usable visual evidence for downstream reasoning. This confirms that the visual tools integrated into MEDVISTAGYM are sufficiently reliable to serve as grounded perception modules for medical reasoning, even in the absence of pixel-level supervision.

Evidence-Answer Consistency. Reliable tool outputs alone do not guarantee that the agent's final answers are genuinely derived from tool evidence—the model could still arrive at correct answers through dataset priors or memorization while ignoring tool outputs. To rule out this shortcut behavior, we randomly sample 200 trajectories that produced correct final answers and manually examine whether the answer is causally supported by the tool outputs. Each trajectory is labeled as *fully supported* (the tool output contains decisive visual or textual evidence that directly justifies the answer), *partially supported* (the tool output is relevant but does not fully justify the answer), or *unrelated* (the answer is correct but the tool output does not contribute to the reasoning). As reported in Table 9(b), 82% of correct answers are fully supported by tool outputs and an additional 12% are partially supported, leaving only 6% where correctness appears unrelated to tool evidence. This indicates that MEDVISTA-R1 learns to genuinely ground its answers in tool-acquired evidence rather than relying on spurious shortcuts, validating that the observed accuracy improvements reflect actual evidence-based reasoning.

Robustness under Tool Failure. A robust tool-integrated agent should remain functional even when tools occasionally fail—for example, when segmentation models produce noisy masks, when detection models return false negatives, or when knowledge retrieval returns irrelevant content. To quantify this property, we partition all evaluation trajectories into two groups based on tool execution outcomes: those in which all tool calls returned valid, non-degenerate outputs (*tool calls succeeded*), and those in which at least one tool call failed or returned invalid output (*tool calls failed/returned errors*). Table 9(c) reports the accuracy within each group. MEDVISTA-R1 achieves ∼65% accuracy when tools succeed and retains ∼48% accuracy when tools fail. This ∼17-point degradation is substantially smaller than what would be expected if the model relied entirely on tool outputs (which would collapse to near-random performance on tool failures), demonstrating that the agent has acquired robust fallback reasoning ability and can recover from imperfect tool outputs through multi-turn reasoning over the partial evidence available.

Summary. Together, these three evaluations show that MEDVISTA-R1 (i) operates on reliable tool outputs (92% usable), (ii) genuinely grounds its answers in tool evidence rather than shortcuts (94% supported), and (iii) remains robust when tool execution fails. Furthermore, the gains reported in Section 4 cannot be attributed to distribution matching with the GPT-5 teacher: cold-start SFT alone (which directly mimics GPT-5's trajectories) reaches only 54.38% Overall, while adding GRPO improves performance to 63.09% (+8.71%). This gain comes entirely from online interaction with MEDVISTAGYM rather than teacher distillation, confirming that the model learns genuine evidence-grounded reasoning rather than imitating the teacher's distribution. This combination establishes that the gains reported in Section 4 are driven by evidence-grounded multi-turn reasoning rather than memorization or spurious correlation.

K Error Analysis Details

K.1 Error Type Definitions

We follow previous paper (Lu et al., 2025) that categorize tool-related failures into six error types based on inspection of model outputs and tool execution traces. Note that a single sample may exhibit multiple error types.

- **E1: Invocation schema violation.** The model produces malformed function calls that violate the expected tool invocation schema (e.g., missing required fields or incorrect call structure).
- **E2: Argument name error.** The model specifies incorrect or non-existent parameter names when invoking tools.
- **E3: Argument format error.** The model provides arguments with invalid formats or data types (e.g., malformed bounding boxes or invalid coordinate values).

ID	Error Type	GPT5	InternVL3-8B
E1	Invocation schema violation (malformed function call)	2.8	11.7
E2	Argument name error (incorrect parameter name)	0	16.1
E3	Argument format error (invalid value format)	19.2	21
E4	Argument content error (semantically incorrect value)	22.1	56.7
E5	Tool output format error (malformed tool response)	31.9	15.5
E6	Tool-induced reasoning error (incorrect reasoning after tool execution)	11.6	73.8

Table 14: Error pattern identification and distribution from 100 error samples(%). Note that one case may contain multiple error types.

- **E4: Argument content error.** The model supplies semantically incorrect argument values despite using valid formats (e.g., selecting irrelevant regions or incorrect anatomical targets).
- **E5: Tool output format error.** The model fails to correctly parse or utilize tool outputs due to malformed responses or misinterpretation of returned results.
- **E6: Tool-induced reasoning error.** The model invokes tools correctly but performs incorrect reasoning after tool execution, leading to erroneous conclusions.

K.2 Errors in Vanilla Open-Source VLMs

Table 14 reports the distribution of error types observed in 100 error samples for GPT-5 and InternVL3-8B. For the vanilla InternVL3-8B model, the majority of failures stem from higher-level interaction and reasoning issues rather than low-level invocation syntax.

In particular, E4 (argument content errors) and E6 (tool-induced reasoning errors) dominate the error distribution, accounting for 56.7% and 73.8% of the inspected cases, respectively. These errors indicate that the model often selects inappropriate tool arguments or fails to reason correctly over tool outputs, even when tool invocation is syntactically valid. In contrast, lower-level schema and argument format errors (E1–E3) occur less frequently, suggesting that the model can partially learn tool syntax from prompting or supervised data alone.

L Case Study Details

L.1 Successful Cases

Case-1: Targeted Visual Evidence Seeking for Modality Identification When a single global inspection of the image is insufficient to resolve the task, the model learns to actively search for diagnostic visual cues by invoking spatially targeted tools. As shown in Figure 10, the agent identifies that determining the imaging modality requires reading projection-specific markers that are not reliably visible at full scale. Rather than reasoning solely from global appearance, it deliberately zooms into a corner region where modality indicators (e.g., side markers and projection labels) are typically located. The retrieved local evidence directly grounds the final answer. This behavior demonstrates an emergent ability to translate abstract task requirements into concrete visual search strategies, selecting both the appropriate tool and the relevant region to inspect.

Case-2: Post-Tool Reflection and Corrective Evidence Alignment Beyond executing tools, the model exhibits an ability to critically evaluate whether tool outputs truly support the question being asked. In Figure 11, the agent initially relies on a detection tool to localize brown-stained cells. However, upon inspecting the tool output, it recognizes a semantic mismatch: scattered detections do not align with the question's emphasis on tubular lining. This triggers a corrective action, prompting the agent to zoom into the tubular regions instead. The subsequent observation reveals continuous luminal staining, which resolves the ambiguity and leads to the correct conclusion. This pattern reflects post-tool self-reflection, where the agent does not passively accept tool outputs but instead assesses their relevance and consistency with the problem semantics.

Case-3: Coordinated Multi-Tool Reasoning under Structural Ambiguity For complex cases, the model learns to orchestrate multiple complementary tools, assigning each a distinct epistemic role and synthesizing their outputs into a unified judgment. As illustrated in Figure 12, the agent sequentially employs a global classifier (BiomedCLIP), a segmentation model (BiomedParse), and a detector (GroundingDINO) to assess whether cortical gyri are abnormal. Rather than treating each tool independently or relying on any single result, the agent integrates signals across tools—global priors, structural segmentation, and localized detection confidence. The final decision is supported by convergent evidence rather than majority voting or isolated observations. This behavior demonstrates an emergent capacity for tool-role differentiation and coordinated reasoning, enabling robust decision-making in ambiguous scenarios.

Case-4: Robust Reasoning under Imperfect Tool Outputs Importantly, agentic training also

Aspect	Case-1	Case-2
Failure Cause	Insufficient Visual Evidence	Knowledge Gap
Tool Usage	✓ Correct (zoom-in)	✓ Correct (all 3 tools)
Reasoning Process	✓ Sound	✓ Sound
Tool Output	Normal but limited	Correct and informative
Root Problem	Ultra-low visual signal	Incorrect medical knowledge
Implication	Tools cannot overcome visual limits	Tools cannot fix knowledge deficits

Table 15: Comparison of two representative failure cases. Both cases exhibit correct tool usage and sound reasoning, but fail due to fundamentally different limitations: perceptual boundaries (Case-1, Figure 14) versus knowledge gaps (Case-2, Figure 15).

equips the model with robustness to tool failures and unreliable outputs. In Figure 13, both segmentation and detection tools fail to localize the target subcellular structures due to inherent limitations. Rather than propagating these errors, the agent explicitly recognizes the mismatch between tool outputs and the biological target. It then falls back on direct visual inspection and domain knowledge, reasoning from the complete absence of PMP70 staining in knockdown cells to infer impaired peroxisomal membrane assembly. This pattern highlights an emergent ability to estimate tool reliability and selectively override tool guidance, ensuring that reasoning remains grounded even when tool outputs are noisy or misleading.

Together, these patterns reveal a progression from correct tool invocation, to reflective tool interpretation, to coordinated multi-tool reasoning, and finally to robust reasoning beyond tool limitations. Rather than merely augmenting perception, tools in MEDVISTAGYM become integrated components of a higher-level reasoning process, enabling the model to actively seek evidence, critique intermediate results, and adapt its strategy across multiple interaction turns.

L.2 Failure Cases

Case-1: Insufficient Visual Evidence We first analyze a representative failure case to examine the limits of the proposed agentic framework under extremely subtle visual conditions. As shown in Figure 14, the task is to determine whether the patient's left lung exhibits abnormal findings. The image contains only weak and ambiguous visual cues, making a reliable diagnosis challenging even after localized inspection.

The agent first correctly interprets the diagnostic goal and recognizes that a global inspection of the radiograph is insufficient. To reduce uncertainty, it deliberately invokes the zoom-in tool to focus on the left lung field, selecting an anatomically appropriate region of interest. This targeted action reflects an explicit attempt to acquire localized evidence relevant to pulmonary abnormality detection.

After inspecting the magnified region, the agent systematically evaluates common radiographic indicators of abnormality, including focal opacity, parenchymal consolidation, and costophrenic angle blunting. Based on the absence of clearly discernible pathological signs in the zoomed view, it concludes that the left lung is normal and outputs a negative result. This prediction is incorrect, as the ground truth indicates a subtle abnormality that remains visually indistinguishable at the examined scale.

Although the final conclusion is incorrect, this failure does not arise from improper tool usage or unreflective reasoning. Instead, it highlights an inherent limitation imposed by ultra-low-signal visual evidence. The trajectory demonstrates that the agent (i) localizes uncertainty, (ii) selects and applies an appropriate verification tool, and (iii) grounds its decision in explicit visual criteria. This case thus illustrates a *principled failure mode*, where the reasoning process remains coherent and introspective, yet the available visual evidence is insufficient to support a correct diagnosis.

Case-2: Knowledge Gap Despite Successful Tool Execution We present a second failure case that reveals a complementary limitation of current tool-integrated medical reasoning systems, and more importantly, highlights a promising direction for extending medical tool collaboration within MEDVISTAGYM.

As shown in Figure 15, the task requires identifying the eponymous fracture type depicted in a wrist radiograph, with candidate options including Monteggia, Bennett, Jones, and Smith fractures. Throughout its reasoning trajectory, the agent exhibits systematic and appropriate tool orchestration, demonstrating strong perceptual competence.

In Turn 1, the agent invokes BiomedParse to segment the fracture region, successfully isolating the relevant anatomical structures at the distal radius. Recognizing that subtle visual cues may be diagnostically important, it proceeds in Turn 2 to apply the Agent4K super-resolution tool, enhancing the image by a factor of $4\times$ to better resolve cortical bone features and fracture morphology. In Turn 3, the agent further employs GroundingDINO to precisely localize the fracture site, obtaining bounding box coordinates that correctly identify the distal radius region.

Despite these methodologically sound and successful visual operations, the agent arrives at an incorrect diagnosis, predicting a Jones fracture instead of the correct Smith fracture. Importantly, this error does not stem from tool malfunction or insufficient visual evidence. Rather, it arises from an incorrect mapping between visual findings and medical concepts: the agent incorrectly associates the observed distal radius fracture with the definition of a Jones fracture, which in fact refers to a fracture at the base of the fifth metatarsal, whereas a Smith fracture denotes a volar-displaced distal radius fracture.

This case illustrates that while visual enhancement tools such as segmentation, super-resolution, and localization substantially improve perceptual evidence acquisition, accurate medical diagnosis in certain scenarios further requires access to specialized domain-specific medical knowledge. Visual tools address how to perceive and localize relevant evidence, but the interpretation of that evidence relies on precise medical definitions and conceptual grounding.

Crucially, this observation does not diminish the value of visual tools or the MEDVISTAGYM environment. Instead, it highlights their role as a necessary foundation for perception-centric reasoning, while motivating the integration of complementary medical knowledge tools, such as structured diagnostic knowledge bases or task-specific medical reasoning modules. We view this case as a representative example of how increased tool diversity and richer collaboration between visual and medical knowledge tools could further enhance VLM-based medical image analysis, and leave the expansion of the MEDVISTAGYM tool set in this direction as an important avenue for future work.

System Prompt: You are a medical VQA reasoning generator. Your task is to generate natural, step-by-step thinking processes for a tool-augmented reasoning trajectory.

Context: You are given a COMPLETED trajectory with:
- Question: the medical image question
- Ground Truth Answer: the verified correct answer
- Tool Execution Sequence: the actual sequence of tool_call → observation pairs
- Image: the medical image (if available)

Your Task: Generate realistic `<think>` content for EACH step, simulating how an agent would reason through the problem step-by-step.

Key Principle - Sequential Reasoning:
Each think must simulate forward-looking reasoning as if you DON'T know what tools will be called next:
1. *Before Tool 1*: Explain why you're calling THIS tool (not mentioning future tools)
2. *After Tool 1 Output*: Interpret the result, then decide what to do next
3. *Before Tool 2*: Based on Tool 1's output, explain why Tool 2 is now needed
4. *Final Verification*: Before the answer, summarize evidence and verify reasoning

Important Rules for Each Think:
Pre-tool thinking (before each tool_call):
- Explain the reasoning for calling THIS specific tool NOW
- Do NOT mention tools you haven't called yet
- Do NOT say "I will call tool1, then tool2, then tool3"

Post-observation thinking (after receiving tool output):
- Interpret what the tool returned
- If failed/irrelevant: explicitly acknowledge the failure
- Based on current evidence, decide the next action

Final verification thinking (before answer):
- Summarize all evidence gathered; Verify the reasoning chain

Rules:
- COPY tool_call, observation, and answer content EXACTLY
- ONLY generate the "think" content
- Each think should be 2-5 sentences, natural and concise
- Use first person ("I need to...", "Based on the result...")
- Handle tool failures gracefully

User Prompt Input:
- Question: {question}; Options: {options}
- Ground Truth Answer: {answer}
- Tool Execution Sequence: {tool_sequence}
- Image: {image}

Output Format (strict JSON, no markdown):
```
{"generated_trajectory": [{"step": 1, "type": "think", "content": "..."}, {"step": 1, "type": "tool_call", "tool": "...", "arguments": {...}}, {"step": 1, "type": "observation", "tool": "...", "content": "..."}, ..., {"step": N, "type": "answer", "content": "..."}]}
```

Figure 6: Prompt Template for Reasoning Trajectory Generation.

System Prompt:

You are an expert medical visual agent equipped with specialized medical imaging tools. Your task is to answer medical visual questions by **using tools first, then reasoning based on tool observations**. The medical image alone is NOT reliable enough. Do NOT attempt to answer without validating information via tools. You must strictly adhere to the following protocol for every interaction:
1. ALWAYS call appropriate tools before giving any final answer;
2. You must choose and call tools to obtain observations, and ONLY THEN provide the final answer;
3. Answering without tool usage will be considered medically unsafe and incorrect.

{tools_description}

Instructions:
1. First, carefully analyze the image and question;
2. Decide which tool will provide the most useful information NEXT (you may use multiple tools across multiple steps);
3. Call ONE tool at a time, observe its output, and update your reasoning;
4. After each observation, either call another tool or, if you have enough information, provide the final answer;
5. Prefer using tools whenever they can improve reliability, localization, or clinical correctness.

Output Format:
You MUST use this EXACT format:
`<think>` Your reasoning and analysis here. Explain WHY you need a specific tool. `</think>`
Then EITHER call a tool:
`<tool_call>{"name": "<tool_name>", "arguments": {"param": "value"}}</tool_call>`
OR provide final answer:
`<answer>` Your final answer here `</answer>`

Important Rules:
- For the FIRST response to a new question, you MUST output a `<tool_call>` and MUST NOT output `<answer>`;
- Before producing `<answer>`, you must have successfully called AT LEAST ONE tool, unless ALL tools are clearly irrelevant or repeatedly fail;
- You can use MULTIPLE tools across multiple steps, but in EACH response you must choose EXACTLY ONE action: either a single `<tool_call>` OR a single `<answer>`;
- Never include both `<tool_call>` and `<answer>` in the same response;
- Maximum {max_tool_calls} tool calls allowed per question;
- Base your final answer on all tool observations and your step-by-step reasoning.

Figure 7: System Prompt.

Initial User Prompt:

Question: {question}
You MUST use one or more tools before answering. Start by thinking step by step and decide which tool to call first.

Continuation Prompt:

`<observation>` {observation} `</observation>`
`<think>` Interpret the observation and update your understanding of the medical question. Explain what the observation confirms, rules out, or suggests. Then decide whether the information is sufficient to answer the question. If not sufficient, determine which tool will provide the most useful next evidence and WHY — based strictly on this observation. `</think>`

If another tool is needed:
`<tool_call>{"name": "<tool_name>", "arguments": {"param": "value"}}</tool_call>`
If information is now sufficient to answer:
`<answer>` your final answer `</answer>`

Force Answer Prompt:

You have reached the maximum number of tool calls. Based on ALL tool observations gathered so far, you must now provide your final answer.
Output ONLY: `<answer>` your answer `</answer>`

Figure 8: User Prompt.

System Prompt:

You are an expert medical visual agent equipped with specialized medical imaging tools. Your task is to answer medical visual questions by **using tools first, then reasoning based on tool observations**. The medical image alone is NOT reliable enough. Do NOT attempt to answer without validating information via tools. You must strictly adhere to the following protocol for every interaction:
1. ALWAYS call appropriate tools before giving any final answer;
2. You must choose and call tools to obtain observations, and ONLY THEN provide the final answer;
3. Answering without tool usage will be considered medically unsafe and incorrect.

{tools_description}

Instructions:
1. First, carefully analyze the image and question;
2. Decide which tool will provide the most useful information NEXT (you may use multiple tools across multiple steps);
3. Call ONE tool at a time, observe its output, and update your reasoning;
4. After each observation, either call another tool or, if you have enough information, provide the final answer;
5. Prefer using tools whenever they can improve reliability, localization, or clinical correctness.

Output Format:
You MUST use this EXACT format:
`<think>` Your reasoning and analysis here. Explain WHY you need a specific tool. `</think>`
Then EITHER call a tool:
`<tool_call>{"name": "tool_name", "arguments": {"param": "value"}}</tool_call>`
OR provide final answer:
`<answer>` Your final answer here `</answer>`

Important Rules:
- For the FIRST response to a new question, you MUST output a `<tool_call>` and MUST NOT output `<answer>`;
- Before producing `<answer>`, you must have successfully called AT LEAST ONE tool, unless ALL tools are clearly irrelevant or repeatedly fail; in that case, explain this in `<think>` and then answer;
- You can use MULTIPLE tools across multiple steps, but in EACH response you must choose EXACTLY ONE action: either a single `<tool_call>` OR a single `<answer>`;
- Never include both `<tool_call>` and `<answer>` in the same response;
- Maximum {max_tool_calls} tool calls allowed per question;
- Base your final answer on all tool observations and your step-by-step reasoning.

Figure 9: GPT5 Data Curation.

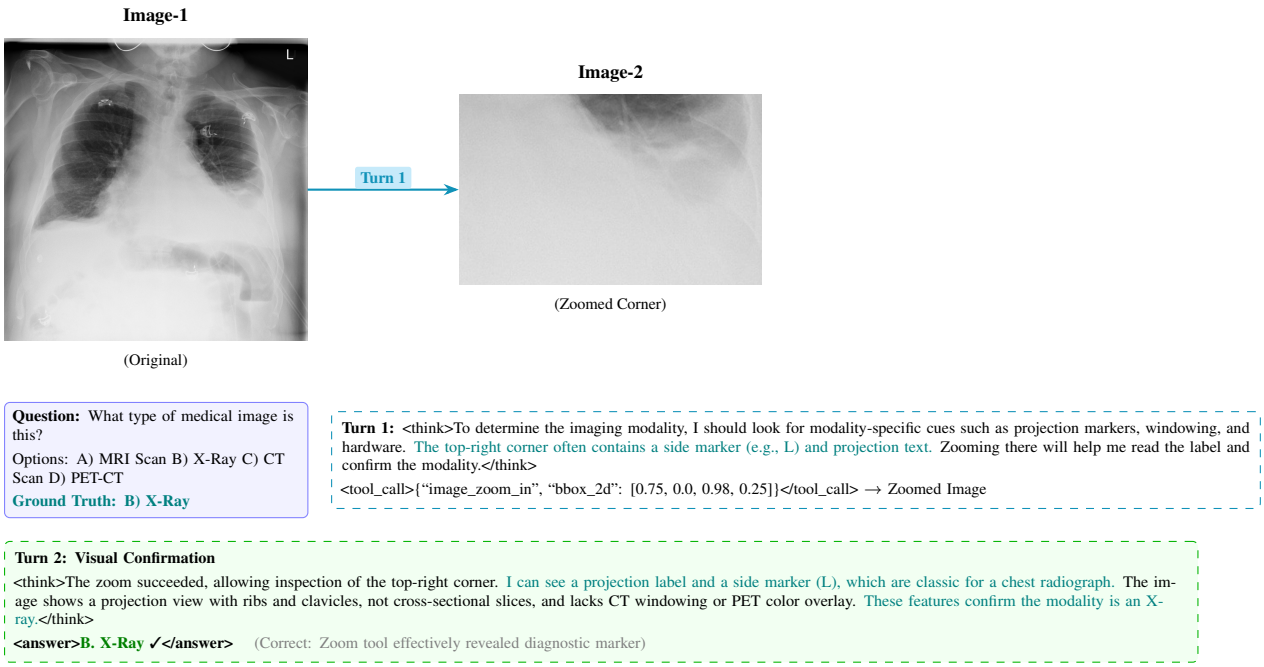
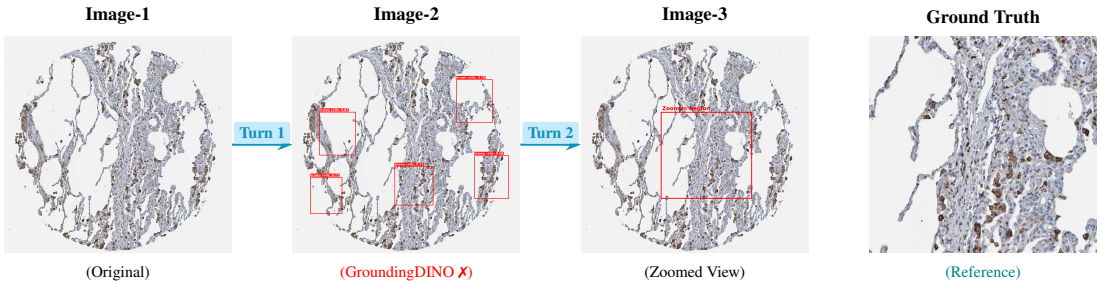


Figure 10: **Effective Tool Usage.** The agent identifies the imaging modality by strategically zooming into the corner region (Turn 1) to examine the projection marker "L", which is characteristic of chest X-ray images. The tool provides decisive evidence for the correct answer.



Question: In hepatic tissue stained with hematoxylin (blue) and DAB (brown), the antibody binds to components of tubular lining. Which cell type is the protein principally expressed in?

Options: A) Macrophages B) Fibroblasts C) Endothelial cells D) Epithelial cells E) Pericytes

Ground Truth: C) Endothelial cells

Turn 1: <think>The question asks which cell type the protein is expressed in. The DAB staining marks the target protein. I will use GroundingDINO to detect and localize the brown-stained cell clusters.</think> <tool_call>{"GroundingDINO", "query": "brown stained cells"}</tool_call> → Detected scattered brown clusters

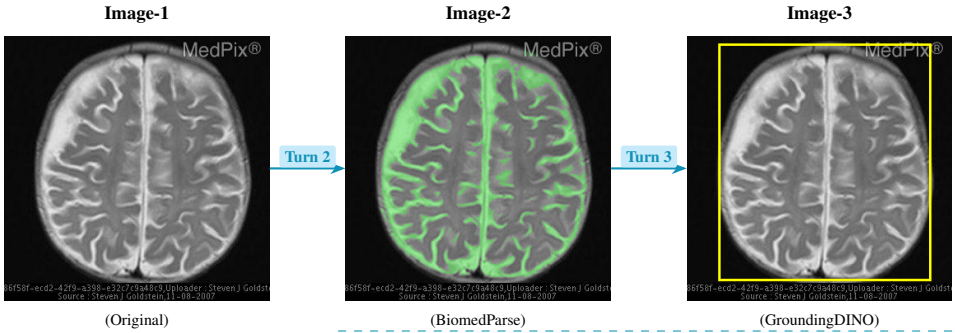
Turn 2: <think>GroundingDINO detected scattered brown clusters. However, the question states the antibody binds to tubular lining components. Scattered cells do not line tubular structures. This suggests a misclassification. I need to zoom into the tubular region to verify.</think> <tool_call>{"image_zoom_in", "bbox_2d": [0.35, 0.35, 0.75, 0.75]}</tool_call>

Turn 3: Self-Reflection & Correction

<think>In the zoomed view, the brown DAB signal is concentrated along the luminal surface of tubular structures, forming a continuous lining pattern rather than scattered clusters. This is inconsistent with macrophages. The continuous luminal lining is characteristic of endothelial cells. Therefore, GroundingDINO's detection was misleading, and the protein is principally expressed in endothelial cells.</think>

<answer>C. Endothelial cells ✓</answer> (Correct: Agent identified tool misclassification through self-reflection)

Figure 11: **Self-Reflection and Correction.** The agent initially uses GroundingDINO to detect scattered brown-stained cells (Turn 1), but identifies a mismatch with the question's description of "tubular lining" (Turn 2). Through self-correction, it zooms into the tubular region and confirms the continuous luminal staining pattern characteristic of endothelial cells.



Question: Is the anatomy of the brain gyri affected?

Options: A) Yes B) No

Ground Truth: B) No

Turn 1: <think>I will use a zero-shot classifier to screen for global cortical abnormalities.</think> <tool_call>{"BiomedCLIP"}</tool_call> → normal gyri (56%), abnormal gyri (44%). Moderate confidence warrants further investigation.

Turn 2: <think>I will segment cortical structures to reveal localized abnormalities.</think> <tool_call>{"BiomedParse", "prompt": "cortex"}</tool_call> → Detected cortical structures, no abnormal region highlighted.

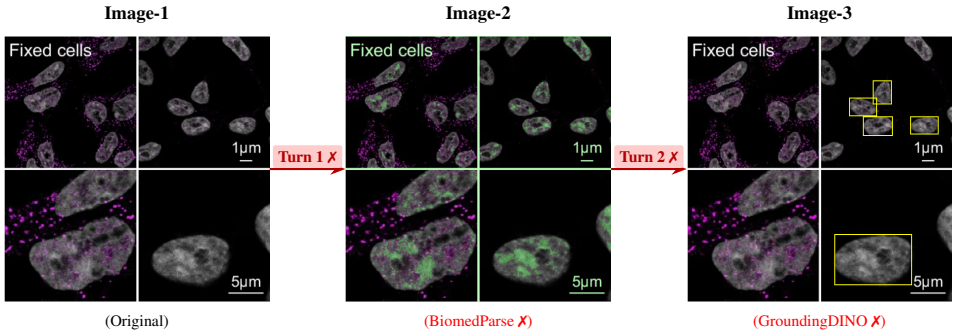
Turn 3: <think>I will use detection to localize any abnormal gyral patterns.</think> <tool_call>{"GroundingDINO", "query": "abnormal gyri"}</tool_call> → 1 detection, confidence 21.7% (very low). No convincing evidence.

Turn 4: Multi-tool Synthesis

<think>After comprehensive multi-tool analysis: (1) classifier favors normal gyri, (2) segmentation revealed no abnormal cortical regions, (3) detector found no high-confidence abnormality. The convergent evidence from all three tools supports the conclusion that the gyral anatomy is not affected.</think>

<answer>B. No ✓</answer> (Correct: Multi-tool convergent evidence supports normal gyri anatomy)

Figure 12: **Multi-tool Reasoning for Comprehensive Assessment.** The agent uses three complementary tools: BiomedCLIP for classification (Turn 1), BiomedParse for segmentation (Turn 2), and GroundingDINO for detection (Turn 3). The convergent evidence from all tools supports the correct conclusion that gyral anatomy is not affected.



Question: PEX19 knockdown cells lack PMP70 staining (magenta) while wildtype shows it. What is the most likely explanation?

Options: A) Misfolding B) Transport disruption C) Synthesis affected D) Lipid raft attachment E) Peroxisomal membrane assembly impaired

Ground Truth: E

Turn 1: <think>I will segment peroxisomal structures to assess if they are present.</think> <tool_call>{"BiomedParse", "prompt": "peroxisome"}</tool_call>

Result: Segmented nuclei (green) instead of peroxisomes. Tool limitation: not trained for subcellular organelles.

Turn 2: <think>Let me try to localize peroxisomes using text-guided detection.</think> <tool_call>{"GroundingDINO", "query": "peroxisome"}</tool_call>

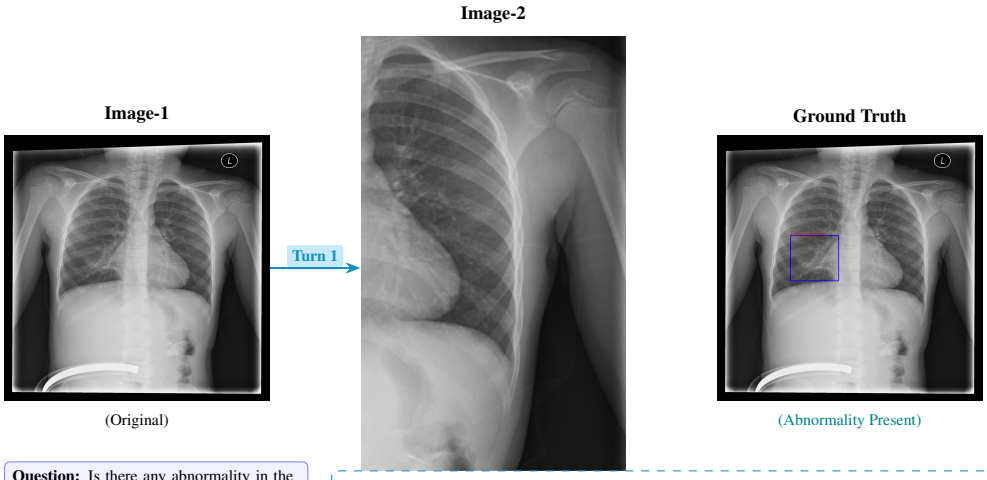
Result: Detected cells/nuclei (yellow boxes), not peroxisomes. Detection failed: boxes on wrong structures.

Turn 3: Visual Observation + Domain Knowledge

<think>Both tools failed to detect peroxisomes correctly. I should rely on direct visual observation and biological knowledge.
Visual observation: Left panels (WT) show magenta PMP70 punctate staining; right panels (KD) completely lack this staining.
Biological reasoning: PEX19 is essential for peroxisomal membrane assembly. Without PEX19, peroxisomes cannot form.</think>

<answer>E. Peroxisomal membrane assembly impaired ✓</answer> (Correct: Model recovered from tool failures using domain knowledge)

Figure 13: **Robustness to Tool Failure.** BiomedParse incorrectly segments nuclei instead of peroxisomes (Turn 1); GroundingDINO detects cells rather than subcellular organelles (Turn 2). Despite both tool failures, the agent correctly answers by combining visual observation (WT shows punctate PMP70 staining while KD shows complete absence) with domain knowledge (PEX19 is essential for peroxisomal membrane biogenesis).



Question: Is there any abnormality in the patient's left lung?

Options: A) Yes B) No

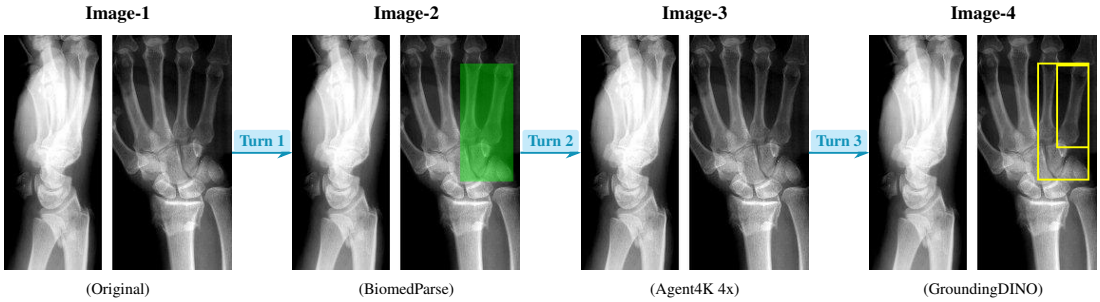
Ground Truth: A) Yes

Turn 1: <think>To decide if the left lung is abnormal, I need a closer look at the left hemithorax for any focal opacity, consolidation, or pleural effusion. A targeted zoom on the left lung field will help me assess the parenchyma and pleural space more precisely.</think> <tool_call>{"image_zoom_in", "bbox_2d": [0.55, 0.15, 0.95, 0.85]}</tool_call> → Zoomed Image

Turn 2: <think>With the left lung magnified, I can evaluate for any focal opacity, air bronchograms, or a meniscus sign that would suggest effusion. The parenchyma appears clear without a discrete mass or consolidation, and the costophrenic angle looks sharp without blunting. Given these findings, I can conclude the left lung is not abnormal.</think>

<answer>B. No ×</answer> (Correct: A. Yes — Abnormality present in left lung field)

Figure 14: **Failure Case: Incorrect Visual Interpretation Despite Tool Usage.** The agent correctly uses zoom (Turn 1) to examine the left lung field, but fails to identify the abnormality present (compare with Ground Truth). This illustrates that tool usage alone does not guarantee correct diagnosis—accurate visual interpretation remains critical.



Question: What person's name is associated with the fracture shown below?

Options: A) Monteggia B) Bennett C) Jones D) Smith

Ground Truth: D) Smith

Turn 1: <think>To identify the fracture type, I will use a segmentation tool to highlight the fracture region.</think> <tool_call>{"BiomedParse", "prompt": "fracture"}</tool_call>

Turn 2: <think>Segmentation highlights fracture at distal radius. Enhancing resolution for better visualization.</think> <tool_call>{"Agent4K", "scale": 4}</tool_call>

Turn 3: <think>Using object detection to localize fracture location.</think> <tool_call>{"GroundingDINO", "query": "fracture"}</tool_call> → Detected at [177,29,231,152]

Turn 4: <think>Fractures located at distal radius. Jones fracture involves distal radius with capitate dislocation; Smith fracture involves distal radius with capitate fracture. Most likely fracture type is Jones.</think>

<answer>C. Jones ×**</answer>** (Correct: D. Smith — Smith fracture is a volar-displaced distal radius fracture)

Figure 15: **Hard Case: Tool Success with Knowledge Gap.** All three tools (BiomedParse, Agent4K, GroundingDINO) executed successfully and provided correct visual evidence. However, the model's incorrect medical knowledge about fracture type definitions led to the wrong final answer.