Evaluating Procedural Tool-Calling in AI Agents for Lung Cancer Workflows

Flavio Stucchi[1], Matteo Tortora[2⋆], Valerio Guarrasi[4], Camillo Maria Caruso[1], and Paolo Soda[1,3]

[1] Research Unit of Artificial Intelligence and Computer Systems, Department of Engineering, Università Campus Bio-Medico di Roma, Rome, Italy
[2] Department of Naval, Electrical, Electronics and Telecommunications Engineering, University of Genoa, Italy. matteo.tortora@unige.it
[3] Department of Diagnostics and Intervention, Radiation Physics, Biomedical Engineering, Umeå University, Sweden
[4] UniCamillus - Saint Camillus International University of Health Sciences, Italy

Abstract. Agentic artificial intelligence (AI) systems are increasingly proposed to coordinate complex clinical workflows by integrating heterogeneous patient data and specialized tools. When used in high-risk clinical applications, these systems may fall within the scope of the EU AI Act, whose requirements for logging, transparency, human oversight, and risk management motivate evaluation beyond the final outputs. However, existing approaches, in both medical and general domains, remain largely outcome-oriented and provide limited deterministic assessments of how agents navigate clinical tasks. This work presents a deterministic framework for auditing tool-calling trajectories against protocol-informed clinical workflows. The framework assesses exact sequence agreement, positional correspondence, sequence deviation, omitted tools, and unnecessary tool invocations. We evaluate it in a simulated agentic system for lung cancer screening and diagnosis that uses mock tools with deterministic outputs and seven publicly accessible large language models (LLMs). Models with similar aggregate performance exhibited distinct procedural profiles, including omissions and unnecessary calls. These findings show that complementary trajectory-level metrics reveal failure modes obscured by aggregate scores and support the inclusion of process-level auditability in medical agent evaluation.

Keywords: LLMs · Evaluation Metrics · Auditability · EU AI Act

1 Introduction

Agentic AI frameworks are being explored in clinical settings to coordinate multi-step workflows, integrate heterogeneous patient data, and support clinical decision-making, with the aim of improving efficiency and enabling more personalized care [3,15].

⋆ Corresponding author

AI agents combine an LLM with an orchestration layer that manages planning, intermediate states, and access to external tools. This architecture suits clinical workflows, which require specialized analyses and sequential decisions. Task-specific medical models can achieve validated performance within a narrow scope but cannot coordinate end-to-end workflows. Standalone LLMs support reasoning and information synthesis but lack the grounding, reliability, and clinical validation required for autonomous decision-making. As medical AI agents move towards clinical deployment, systems used in high-risk applications may be subject to EU AI Act requirements on technical documentation, logging, transparency, human oversight, accuracy, robustness, and risk management [4]. These requirements motivate evaluation beyond final outputs. Process-level assessment should examine which tools are selected, when they are invoked, and whether required steps are omitted or unnecessary steps are introduced.

Recent medical agent frameworks demonstrate the potential of tool orchestration. For instance, MedRAX coordinates tools for chest X-ray classification, segmentation, grounding, visual question answering, and report generation through a ReAct-based system [5]. MedOrch extends tool orchestration to Alzheimer's disease diagnosis, chest X-ray interpretation, and medical visual question answering [7]. Their evaluations, however, remain centered on task-level performance and do not provide a general deterministic method for identifying incorrect tool selection, ordering errors, omissions, or unnecessary calls.

This limitation reflects a broader gap in agent evaluation [14]. ToolBench evaluates tool use over 16,464 real-world REST APIs through ToolEval, an LLM judge that reports pass rate and win rate based on task completion and coarse execution criteria [11]. Although ToolEval considers solution paths, it does not deterministically measure whether the required tools were selected, invoked in the expected position, omitted, or called unnecessarily. Its reliance on an LLM judge also limits reproducibility.

TRAJECT-Bench advances trajectory-aware evaluation through metrics for tool-list match, trajectory inclusion, tool usage, input correctness, and completion [6]. It targets general-purpose tool use across production-style APIs rather than adherence to protocol-informed clinical workflows. It also does not explicitly separate positional correspondence, sequence deviation, omissions, and unnecessary calls.

To address this gap, we present a deterministic framework for evaluating observable tool-calling trajectories, defined as the ordered sequence of tool invocations used to complete a clinical task. The contributions of this work are organized around two complementary components:

- **A clinically grounded evaluation suite for tool-calling trajectories.** We introduce deterministic metrics spanning positional correspondence, sequence deviation, and error decomposition (omissions and over-generation), to explicitly assess an agent's adherence to protocol-informed lung cancer screening and diagnosis workflows.
- **A simulated agentic framework for clinical orchestration.** We develop a controlled environment in which an LLM-based agent navigates lung

cancer screening and diagnostic workflows using mock tools with predefined outputs, thereby isolating its tool-selection and orchestration capabilities.

2 Methods

We introduce five deterministic metrics to assess the agreement between an agent's tool calling behavior and the reference clinical workflow. Let the reference tool sequence be denoted $G = (g_1, g_2, \ldots, g_{|G|})$ and let the sequence invoked by the agent be denoted $T = (t_1, t_2, \ldots, t_{|T|})$. Each metric is computed per case and macro-averaged across scenarios. As each tool can be invoked at most once within a scenario, the trajectories do not contain repeated tool calls.

Exact Match measures whether the agent trajectory exactly matches the reference trajectory in tool selection, order, and length. This definition follows TRAJECT-Bench [6]:

$$\mathrm{EM}(T, G) = \begin{cases} 1, & \text{if } T = G, \\ 0, & \text{otherwise.} \end{cases}$$

Correct Tool Position measures the proportion of reference positions at which the agent invokes the expected tool:

$$\mathrm{CTP}(T, G) = \begin{cases} \dfrac{1}{|G|} \displaystyle\sum_{j=1}^{\min(|T|,|G|)} \mathbb{I}[t_j = g_j], & |G| > 0, \\ 1, & |G| = 0 \text{ and } |T| = 0, \\ 0, & |G| = 0 \text{ and } |T| > 0. \end{cases}$$

Levenshtein Distance measures the minimum number of single-tool insertions, deletions, and substitutions required to transform the agent trajectory T into the reference trajectory G:

$$\mathrm{LD}(T, G) = d_{\mathrm{lev}}(T, G).$$

Missed Tools measures the proportion of required tools that the agent does not invoke. Let $\mathrm{TP}(T, G) = |\{\, t_i \in T : t_i \in G \,\}|$ denote the number of predicted calls that correspond to a tool in the reference. Then

$$\mathrm{MT}(T, G) = \frac{|G| - \mathrm{TP}(T, G)}{|G|}, \qquad |G| > 0.$$

Extra Tools counts the tool calls issued by the agent that do not correspond to any tool in the reference trajectory:

$$\mathrm{ET}(T, G) = |\{\, t_i \in T : t_i \notin G \,\}|.$$

Although these metrics are related, they capture distinct dimensions of procedural behavior. EM measures complete agreement with the reference workflow. CTP measures positional correspondence. LD quantifies deviation from the expected tool-calling sequence. MT captures the omission of required clinical steps, whereas ET identifies unnecessary calls.

2.1 Agentic Framework

The framework implements a stateful, graph-based agent equipped with simulated clinical tools. Following the ReAct paradigm [13], the LLM conditions each action on the system prompt, conversation history, available clinical evidence, accumulated state and tool descriptions. At each step, it either invokes a tool or returns a final response. When a tool is invoked, its deterministic output is appended to the agent state and informs the subsequent action. This action-observation loop continues until the LLM returns a response without requesting another tool. Each tool can be invoked at most once per patient scenario. Consequently, repeated-call errors are outside the scope of the present evaluation.

To isolate LLM orchestration from errors introduced by the underlying clinical tools, the framework uses mock tools with deterministic, predefined outputs. These tools emulate clinical functions including segmentation, classification, and report generation. This controlled design focuses the evaluation on tool selection, ordering, and workflow completion, independently of the predictive performance of individual tools.

3 Materials

To evaluate the agent's procedural orchestration, we simulated two lung cancer workflows: screening and diagnosis. Their sequential logic was derived from the Lung Cancer Screening and Diagnosis Pathway Maps published by Ontario Health (Cancer Care Ontario) [9,10], which describe the main steps, decision points, and dependencies of the corresponding patient pathways. The pathways were simplified and converted into textual procedural rules encoded in the agent's system prompt.

The prompt was structured according to the CO-STAR framework [8], with emphasis on the *role*, *instructions*, and *examples* components. The role defines the agent as an AI assistant supporting clinicians. The instructions encode the procedural rules required to navigate the workflows, while the examples provide three few-shot demonstrations of reference tool-calling sequences for representative clinical queries [1]. This prompt-based design approach enabled the agent to adapt to the clinical context without additional training or fine-tuning.

We constructed a synthetic dataset of patient scenarios. Each scenario comprises a case-specific set of available clinical evidence, an associated clinical label, and a reference trajectory specifying the ordered tool invocations defined by the corresponding workflow. The reference trajectory provides the procedural target against which the agent-generated sequence is evaluated.

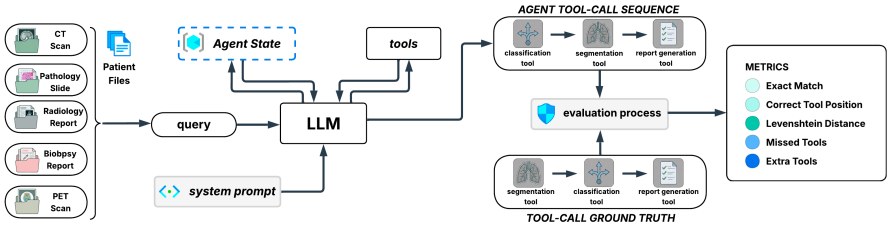

Fig. 1. Overview of the proposed evaluation framework. The orchestrating LLM receives a patient query, the available evidence indicators, the system prompt, and access to the simulated clinical tools. At each step, it updates the agent state and either invokes a tool or returns a final response. The resulting tool-call sequence is compared with the scenario-specific ground truth trajectory using the five deterministic metrics.

For the diagnostic pathway, we considered five medical evidence types: CT scans, pathology slides, radiology reports, biopsy reports, and PET scans. Each of the 48 diagnostic labels was paired with all $2^5 - 1$ non-empty combinations of these evidence types, corresponding to 31 combinations and yielding 1,488 scenarios. This combinatorial design provides systematic coverage of clinical labels and evidence availability, allowing us to test whether the agent adjusts its tool selection and invocation order when different information sources are present or absent. The screening dataset was generated using the same procedure and contains 122 scenarios. Its smaller size reflects the lower number of clinical states and shorter reference trajectories in the screening pathway. As illustrated in Figure 1, the agent receives the case query, the available evidence indicators, and the system prompt. Its state records previous tool calls, deterministic tool outputs, and the workflow progress. This information guides each subsequent action. The resulting tool-call sequence is compared with the scenario-specific ground-truth trajectory using the five deterministic metrics.

4 Experimental Setup

The experimental framework was implemented using LangChain, with LangGraph managing the cyclic, stateful execution [2].

We evaluated seven publicly accessible LLMs spanning dense and Mixture-of-Experts architectures: MedGemma (27B parameters), Nemotron-3-super (120B total, 12B active), Gpt-oss-120B (117B total, 5.1B active), Minimax-M2.7 (230B total, 10B active), Mistral-large-3 (675B total, 41B active), DeepSeek-v3.2 (671B total, 37B active), and Kimi-k2 (1T total, 32B active). The selected models differ in scale, architecture, domain specialization, and agent-oriented post-training, enabling comparison of their procedural tool-calling behavior. Since their memory requirements prevented local deployment, the models were accessed through cloud-based endpoints, with Ollama providing a common request interface.

Tool	Modalities	Pathways	Data flow
Risk Stratifier	EHR	Screening	EHR → patient risk
Detector Modules	Radiology Histopathology	Screening Diagnosis	CT/WSI → detected region
Segmentation Modules	Radiology Histopathology	Screening Diagnosis	Detected region → segmentation mask
Classifier Modules	Radiology Histopathology	Screening Diagnosis	Segmented region → predicted label
Mutation Predictor	Histopathology	Diagnosis	WSI → predicted mutation
Time Translator	Radiology	Diagnosis	CT → one-year CT
Domain Translator	Radiology	Diagnosis	CT/PET → PET/CT
Report Generators	Radiology Histopathology	Screening Diagnosis	WSI/CT → textual report

Table 1. Simulated clinical tools, supported data modalities, applicable clinical pathways, and corresponding data flows.

The agent interacts with mock clinical tools rather than executable predictive models. Each tool is implemented as a Python function that returns a deterministic, predefined output based on the scenario's label. The agent receives textual indicators of the available examinations rather than raw clinical files.

5 Results and Discussion

Table 2 reports the performance of each model across the diagnosis and screening pathways. Upward arrows indicate metrics for which higher values correspond to better performance, whereas downward arrows identify metrics for which lower values are preferable. Response time (RT) denotes the mean end-to-end response time per scenario, in seconds. DeepSeek-v3.2 and Kimi-k2 achieved the largest overall agreement between the agent-generated tool sequence and the case-specific reference workflow. In diagnosis, their EM scores were 0.688 and 0.672, respectively, increasing to 0.890 and 0.860 in screening. They also achieved the largest CTP scores and the lowest LDs, indicating that the expected tools were more frequently invoked at the correct positions and that the predicted tool sequences deviated less from the reference workflows. However, not all models were able to satisfy the framework's structured-output requirements. Despite its medical fine-tuning, MedGemma-27B failed to produce valid JSON tool calls and could therefore not be evaluated within the agentic framework. Consequently, no performance values were reported in the corresponding tables. This inability to

	Model	EM(↑)	CTP(↑)	LD(↓)	MT(↓)	ET(↓)	RT
Diagnosis	Nemotron-3-super	0.254	0.480	1.225	**0.033**	0.985	**26s**
	Gpt-oss	0.349	0.668	2.236	0.230	0.330	**17s**
	Minimax-M2.7	0.466	0.636	2.036	0.189	0.087	31s
	Mistral-large-3	0.496	0.668	1.845	0.103	0.248	32s
	DeepSeek-v3.2	**0.688**	**0.893**	**1.105**	0.036	**0.021**	146s
	Kimi-k2	**0.672**	**0.841**	**1.205**	**0.026**	**0.051**	30s
Screening	Nemotron-3-super	0.213	0.813	2.156	**0.005**	2.131	**20s**
	Gpt-oss	0.320	0.518	1.918	0.312	0.213	**16s**
	Minimax-M2.7	0.385	0.816	1.410	0.043	0.984	59s
	Mistral-large-3	0.508	0.742	1.033	0.094	0.672	72s
	DeepSeek-v3.2	**0.890**	**0.954**	**0.172**	0.023	**0.145**	53s
	Kimi-k2	**0.860**	**0.944**	**0.182**	**0.019**	**0.116**	33s

Table 2. Model performance in diagnosis and screening. Bold black indicates the best value, while bold blue indicates the second-best value.

generate structured outputs aligns with known performance degradation models experienced when forced to meet rigid structural constraints [12].

The proposed metrics revealed procedural differences that would not be captured by evaluating only the final output. Gpt-oss exhibited a conservative profile characterized by frequent omissions, which may result in decisions based on incomplete clinical evidence. Conversely, Nemotron-3-super rarely omitted required tools but produced the highest ET scores, reaching 0.985 in diagnosis and 2.131 in screening. This over-generation may increase computational cost, latency, and exposure to irrelevant outputs, which highlights the time and resources a hospital might lose. The causes of this over-generation cannot be determined from the present experiments and may involve instruction-following, decoding configuration, architecture, or post-training. Moreover, its high CTP score in screening, despite low EM, demonstrates that correctly placing individual tools does not necessarily produce a valid complete trajectory. The comparison between screening and diagnosis workflows further highlights the effect of procedural complexity. The former follows a relatively constrained sequence of eligibility checks and predefined decisions, whereas the latter involves more available tools, branching alternatives, and dependencies between intermediate findings. DeepSeek-v3.2 and Kimi-K2 achieved higher EM scores and lower sequence deviation in screening, a pattern consistent with the shorter and more constrained screening trajectories. However, this improvement was not observed uniformly across all models, suggesting that simpler workflows primarily benefit architectures capable of reliably interpreting and applying the encoded clinical rules.

Nominal model scale did not explain the observed procedural profiles. Models with comparable parameter counts produced different results, suggesting that architecture, instruction following, and agent-oriented post-training also influence tool orchestration. Kimi-k2 completed the diagnostic workflow in 30 seconds, compared with 146 seconds for DeepSeek-v3.2, despite being the larger model. Response times include uncontrolled variability introduced by cloud infrastructure and API communication. Therefore, they should be interpreted as indicative end-to-end latency measurements rather than direct estimates of model inference time.

6 Conclusions

The proposed framework evaluates medical agents by comparing observable tool-calling trajectories with case-specific reference workflows. It identifies selection, ordering, omission, and unnecessary-call errors that a single aggregate metric may obscure. Models with similar scores showed distinct procedural profiles: some omitted required steps, whereas others added unnecessary calls. Omissions may reduce evidential completeness, while unnecessary calls may increase latency, cost, and exposure to irrelevant outputs. A high CTP score did not ensure exact workflow completion, supporting complementary process-level metrics that may contribute to the traceability and risk-management requirements of high-risk AI systems under the EU AI Act [4]. Procedural reliability was not explained by nominal model scale alone and therefore requires direct trajectory evaluation. Future work should assess tool-call parameters, execution efficiency, interpretation of intermediate outputs, and error propagation. The simulated environment does not model tool uncertainty, erroneous outputs, raw clinical data, or clinician–agent interaction. It therefore evaluates controlled procedural orchestration rather than deployment readiness. Validation with real tools, clinical data, uncertainty, and clinician-reviewed trajectories is needed.

Acknowledgments. This work was partially funded by Project PNRR-MCNT2-2023-12377755 (CUP: C83C24000220007) - Rafforzamento e potenziamento della ricerca biomedica del SSN, finanziato dall'Unione europea – NextGenerationEU. Resources are provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) and the Swedish National Infrastructure for Computing (SNIC) at Alvis @ C3SE, partially funded by the Swedish Research Council through grant agreements no. 2022–06725 and no. 2018–05973

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

References

1. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
2. Chase, H.: Langchain (2022), https://github.com/langchain-ai/langchain
3. Collaco, B.G., Haider, S.A., Prabha, S., Gomez-Cabello, C.A., Genovese, A., Wood, N.G., Bagaria, S.P., Gopala, N., Tao, C., Forte, A.J.: The role of agentic artificial intelligence in healthcare: a scoping review. npj Digital Medicine (2026)
4. European Parliament and Council of the European Union: Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending certain Union legislative acts (Artificial Intelligence Act). Official Journal of the European Union, L 2024/1689, 12 July 2024 (2024), https://eur-lex.europa.eu/eli/reg/2024/1689/oj
5. Fallahpour, A., Ma, J., Munim, A., Lyu, H., Wang, B.: Medrax: Medical reasoning agent for chest x-ray. arXiv preprint arXiv:2502.02673 (2025)
6. He, P., Dai, Z., He, B., Liu, H., Tang, X., Lu, H., Li, J., Ding, J., Mukherjee, S., Wang, S., et al.: Traject-bench: A trajectory-aware benchmark for evaluating agentic tool use. arXiv preprint arXiv:2510.04550 (2025)
7. He, Y., et al.: Medorch: Medical diagnosis with tool-augmented reasoning agents for flexible extensibility. arXiv preprint arXiv:2506.00235 (2025)
8. Ohalete, N.C., Gittner, K.B., Matheny, L.M.: Costar-a: A prompting framework for enhancing large language model performance on point-of-view questions. arXiv preprint arXiv:2510.12637 (2025)
9. Ontario Health (Cancer Care Ontario): Lung cancer screening pathway map. https://www.cancercareontario.ca/sites/ccocancercare/files/assets/LungCancerSxPathwayMap.pdf (2025), version 2025.03
10. Ontario Health (Cancer Care Ontario): Lung cancer diagnosis pathway map. https://www.cancercareontario.ca/sites/ccocancercare/files/assets/LungCancerDiagnosisPathwayMap.pdf (2026), version 2026.04
11. Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., Lin, Y., Cong, X., Tang, X., Qian, B., et al.: Toolllm: Facilitating large language models to master 16000+ real-world apis. In: International Conference on Learning Representations. vol. 2024, pp. 9695–9717 (2024)
12. Tam, Z.R., Wu, C.K., Tsai, Y.L., Lin, C.Y., Lee, H.y., Chen, Y.N.: Let me speak freely? a study on the impact of format restrictions on large language model performance. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track. pp. 1218–1236 (2024)
13. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: React: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629 (2022)
14. Yehudai, A., Eden, L., Li, A., Uziel, G., Zhao, Y., Bar-Haim, R., Cohan, A., Shmueli-Scheuer, M.: A survey on evaluation of llm-based agents. In: Findings of the Association for Computational Linguistics: ACL 2026. pp. 26690–26714 (2026)
15. Zhao, L., Liu, S., Xin, T., Tan, J., Wang, X., Li, Y., Bian, Z., Chen, Y., Kong, F., Bian, J., et al.: Ai agent in healthcare: applications, evaluations, and future directions. npj Artificial Intelligence **2**(1), 31 (2026)