LOOK BEFORE YOU LEAP: PLANNING WITH RISK AND COST AWARENESS FOR A CLINICAL DIAGNOSIS AGENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Clinical diagnosis unfolds as a sequence of consequential decisions: every test reveals information, consumes resources, and changes what should be done next. Yet diagnostic agents typically decide one step at a time, without explicitly considering how an action may shape the future diagnostic trajectory. We ask whether an agent can make better decisions by *looking ahead before acting*. To study this question in realistic clinical settings, we construct **FISC**, a Fast healthcare interoperability resources-based Interactive Sequential Clinical case dataset derived from 156 longitudinal real-world patient records, enabling standardized and interactive diagnostic trajectories. We then introduce **SPARC**, a Sequential Planning with Anticipated Risk and Cost agent, which formulates diagnosis as resource- and risk-aware planning under partial observability. Before executing a test, SPARC imagines its plausible outcomes and their downstream diagnostic trajectories, and uses stochastic AND–OR search to choose actions by jointly considering information gain, cumulative cost, diagnostic error, and the risk of missing critical disease. Following receding-horizon planning, SPARC executes only the selected next action and replans after observing its real outcome. Across real-world clinical cases, SPARC achieves diagnostic performance comparable to alternative sequential strategies while using fewer tests, and its behavior across planning depths reveals a trade-off between lookahead horizon, resource use, and diagnostic performance. These results suggest that diagnostic actions can be usefully evaluated not only by their immediate utility, but also by their anticipated downstream consequences.

1 INTRODUCTION

Clinical diagnosis unfolds as a sequential decision process under incomplete information (Detsky, 2022). Clinicians begin with an initial patient presentation, acquire additional evidence through history taking, examination, and diagnostic tests, update their diagnostic beliefs, and determine whether further information is needed before reaching a diagnosis. Each information-acquisition action consumes resources, including monetary expense, time, clinical workload, and potential patient burden (Müskens et al., 2022). An effective diagnostic strategy therefore needs to determine *which* information to acquire, *in what order*, and *when to stop*, while maintaining reliable diagnostic decisions. Active feature acquisition formalizes a closely related problem by allowing a learner to sequentially acquire informative features while balancing predictive performance against acquisition cost (Shim et al., 2018; Li & Oliva, 2021). More recent work further shows that jointly informative acquisitions can make non-greedy decision making valuable, because the utility of an observation may depend on observations acquired later (Valancius et al., 2024). Clinical diagnosis provides a particularly consequential setting for this problem: the underlying disease state is partially observed, evidence arrives sequentially, and each acquired observation can reshape subsequent diagnostic decisions (Gheorghe et al., 1976). Diagnostic testing can therefore be viewed as *sequential resource allocation under partial observability*.

Recent advances in large language models (LLMs) have increasingly brought automated clinical diagnosis into this sequential setting. Clinical agents can start from an incomplete patient presentation, actively request additional clinical information, and progressively refine their diagnoses (Chen

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

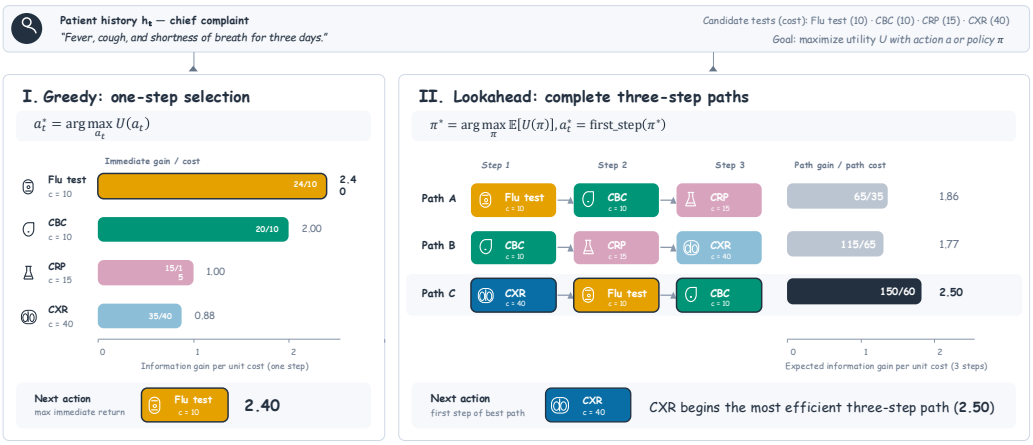

Figure 1: From next-test selection to diagnostic-pathway planning. Active test selection evaluates candidate information acquisitions from the current diagnostic state (Ruhrberg Estévez et al., 2025; Valancius et al., 2024). While one-step utility favors the influenza test in this illustrative example, lookahead identifies the chest X-ray as the first action of a more efficient three-step pathway. SPARC builds on this insight by evaluating diagnostic actions through their anticipated downstream resource and risk consequences.

et al., 2026; Xu et al., 2026). This capability has motivated more explicit approaches to active test selection. ACTMED uses Bayesian experimental design to select tests according to their expected reduction in diagnostic uncertainty (Ruhrberg Estévez et al., 2025). Resource use has also become part of the decision objective: GraphDx incorporates action costs into sequential diagnostic reasoning (Tan et al., 2026), while recent work on differential diagnosis studies cost-aware feature acquisition under realistic clinical availability constraints (Bingham & Arussy, 2026). Together, these developments move clinical AI toward actively deciding *what evidence to acquire next*. They also motivate a further planning question: *what downstream diagnostic process will a candidate action initiate?* The immediate utility of a test captures its near-term value, whereas its overall value also depends on the future observations, decisions, and resource expenditures enabled by its possible outcomes. Consequently, **the best next test may differ from the first test of the best diagnostic pathway**.

Figure 1 illustrates this distinction. A one-step strategy based on immediate information gain per unit cost selects the influenza test, which provides the highest immediate utility among the available actions. Looking several steps ahead reveals a different decision: beginning with the chest X-ray leads to a downstream sequence with higher overall efficiency. Related work in active feature acquisition has similarly demonstrated the benefit of reasoning beyond greedy, one-step acquisitions when features are jointly informative (Valancius et al., 2024). More broadly, sequential value-of-information formulations evaluate an observation through how it changes subsequent decisions, providing a decision-theoretic foundation for planning information acquisition over time (Memarzadeh & Pozzi, 2016). The value of a diagnostic action therefore reflects both the information it provides immediately and the future decisions that it enables. This observation motivates *diagnostic-pathway planning*, in which an agent anticipates plausible future outcomes before committing resources to the next action. Clinical diagnosis adds an important safety dimension because diagnostic errors carry asymmetric consequences: delaying evidence relevant to a time-critical condition can have substantially greater consequences than delaying a lower-risk alternative (National Academies of Sciences, Engineering, and Medicine, 2015). Sequential diagnostic testing has been formulated using value-of-information and partially observable decision models that jointly consider belief updating, further testing, and stopping decisions (Allenbrand, 2026). Building on this perspective, we ask: *how can a clinical agent anticipate the downstream consequences of candidate actions and plan an efficient diagnostic trajectory while maintaining diagnostic safety?*

We address this question with **SPARC** (**S**equential **P**lanning with **A**nticipated **R**isk and **C**ost), a resource- and risk-aware planning framework for clinical diagnostic agents. SPARC evaluates can-

didate tests through their anticipated downstream consequences. For each candidate action, it generates plausible test outcomes and updates diagnostic beliefs along hypothetical future branches. A finite-depth stochastic AND–OR search evaluates these branches according to their expected downstream loss, jointly accounting for cumulative testing costs, diagnostic error, and additional penalties associated with missed time-critical conditions. A safety-constrained stopping rule governs when sufficient evidence has been collected. Following a receding-horizon strategy, SPARC executes the first action of the selected plan, incorporates the returned result, and replans from the updated diagnostic state. Diagnostic actions are represented through an Health Level Seven International (HL7) Fast Healthcare Interoperability Resources (FHIR) –encoded test catalog, providing a standardized interface between planning decisions and clinical test requests (Mandel et al., 2016; Lehne et al., 2019).

Our contributions are as follows:

- We introduce FISC, a FHIR-based interactive sequential clinical case dataset derived from 156 longitudinal real-world inpatient records. It transforms real-world diagnostic trajectories into standardized, interactive clinical environments, enabling clinical diagnosis to be studied as a sequential decision-making task in which agents iteratively acquire information and make diagnostic decisions under partial observability.
- We introduce SPARC, an agent with receding-horizon diagnostic planner that combines counterfactual test-outcome prediction, diagnostic belief updating, stochastic AND–OR lookahead, cumulative testing costs, and safety-constrained stopping to evaluate future diagnostic trajectories before action execution.
- We evaluate SPARC on FISC and systematically characterize diagnostic performance, resource utilization, planning depth, stopping behavior, and safety for time-critical conditions. Our results characterize how trajectory-level planning trades off diagnostic performance, resource use, and safety across different lookahead horizons.

2 Related Work

2.1 Active Feature Acquisition and Sequential Information Acquisition

Active feature acquisition (AFA) formulates prediction as *cost-sensitive sequential information acquisition* (Shim et al., 2018; Li & Oliva, 2021), where a policy selectively acquires missing observations before making a prediction. Generative models have been used to estimate the utility of unobserved features (Li & Oliva, 2021), while recent methods explicitly address *non-greedy acquisition* when features become informative through their interaction with future observations (Valancius et al., 2024). Stochastic representations further support acquisition decisions over possible unobserved feature realizations, providing an alternative to myopic information-gain policies (Norcliffe et al., 2025). Related work extends cost-aware acquisition to clinical decision making, including personalized treatment selection where tests incur monetary cost and delay (Piskorz et al., 2025). SPARC builds on this sequential acquisition perspective in clinical diagnosis, where each observation additionally changes diagnostic beliefs, future test choices, stopping decisions, and clinical risk.

2.2 LLM Agents for Sequential Clinical Diagnosis

Medical LLMs are increasingly moving from reasoning over complete cases toward *interactive evidence gathering* (Chen et al., 2026) and *full-process diagnostic agents* (Xu et al., 2026), allowing models to request clinical information and progressively update diagnostic decisions. Agentic benchmarks further evaluate multimodal tool use in simulated clinical environments (Schmidgall et al., 2026) and multi-turn diagnostic decision making (Hsu et al., 2026). Within this sequential setting, ACTMED introduces *uncertainty-driven active test selection* through Bayesian experimental design (Ruhrberg Estévez et al., 2025), whereas GraphDx incorporates *diagnostic relevance and action cost* into sequential diagnostic planning (Tan et al., 2026). These studies establish active evidence acquisition as an important capability of clinical agents. SPARC advances this direction toward *explicit trajectory-level lookahead*: candidate tests are evaluated through plausible future

outcomes and subsequent actions, allowing the planner to anticipate cumulative resource use and time-critical risk before execution.

2.3 Planning and Safety in Clinical Reasoning

Decision-theoretic approaches provide a complementary foundation for sequential diagnosis through *value-of-information reasoning* (Memarzadeh & Pozzi, 2016) and *belief-state planning under partial observability* (Allenbrand, 2026), where decisions depend on both current uncertainty and the expected value of future observations. For clinical agents, reliability additionally depends on whether accumulated evidence is sufficient to support an autonomous decision (Zhang et al., 2026). Recent disease-centric verification further formalizes *required evidence and contradictions*, *discriminative tests*, and *justified diagnostic closure* as explicit obligations of diagnostic reasoning (Cao & Chen, 2026). SPARC addresses the complementary planning problem: given unresolved diagnostic uncertainty, it determines which evidence to acquire next by anticipating downstream diagnostic loss, cumulative resource expenditure, and the risk of missing time-critical conditions.

3 Problem Formulation

We formulate clinical diagnosis as a sequential decision problem under partial observability. Starting from an initial patient presentation, the agent repeatedly selects an available diagnostic test, observes its result, updates its diagnostic belief, and decides whether to acquire more evidence or stop. At decision step t, the planning state is $S_t = (H_t, b_t)$, where H_t is the visible clinical history and b_t is the belief over a fixed candidate disease set. Legal actions comprise unused tests from a fixed FHIR-compliant catalog and an explicit stopping action. The notation used throughout this paper is summarized in Table 1 in Appendix A ,and the complete problem formulation is provided in Appendix B.

The objective is to balance resource use, diagnostic accuracy, and safety by minimizing cumulative testing cost, general diagnostic error, and the additional loss caused by missing time-critical conditions (TCCs):

$$\pi^* = \arg \min_{\pi \in \Pi_{\text{adm}}} \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} C(a_t) + L_{\text{terminal}}\big(S_T, \hat{d}(S_T)\big) \right],$$

where T is the number of executed tests and

$$L_{\text{terminal}}(S_T, \hat{d}(S_T)) = L_{\text{diag}}(S_T, \hat{d}(S_T)) + L_{\text{TCC}}^{\text{extra}}(S_T, \hat{d}(S_T)).$$

SPARC addresses this problem with finite-horizon lookahead. Given the current diagnostic state and the associated testing costs and diagnostic losses (Appendix B), it predicts plausible outcomes for candidate tests and evaluates their downstream consequences through finite-horizon stochastic AND–OR search, where OR nodes represent diagnostic actions and AND nodes represent their possible test outcomes (Appendix C).A safety-constrained stopping rule determines when the agent can issue a diagnosis (Appendix C.6). Otherwise, SPARC executes only the first test of the selected plan, incorporates the returned real or simulated result (Appendix D), and replans from the updated state. Algorithm E.1 summarizes the full procedure.

4 Method

4.1 System Overview

As shown in Figure 2, the system consists of the SPARC decision system, a patient database, and a FHIR-based test-result environment. At each decision step, SPARC performs three processes. State updates the diagnostic state using currently available patient information and returned test results. Dreamer anticipates possible outcomes of candidate tests and the decisions that may follow them. Explore selects the next test based on these projected pathways and incorporates the result returned by the environment. Once the stopping criteria are met, SPARC submits a final diagnosis. The test-result environment provides feedback after each selected test, as described in Section 4.2.

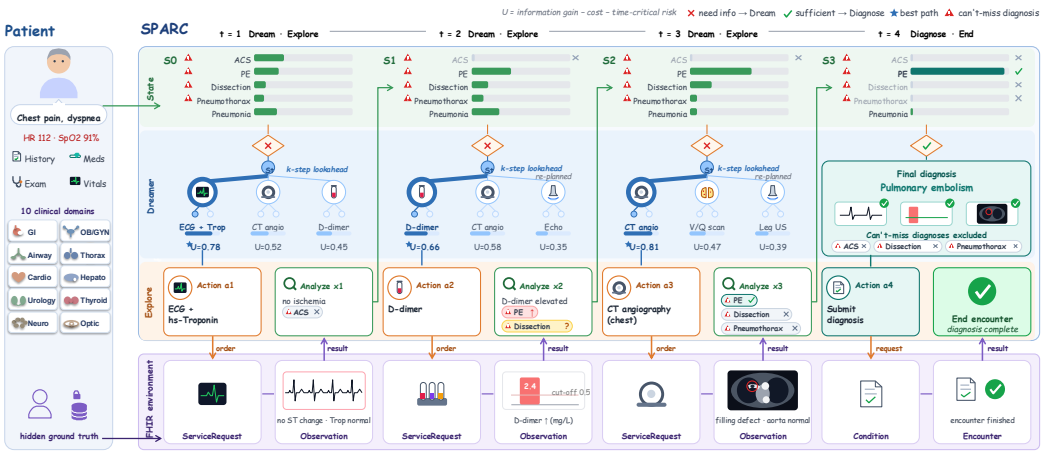

Figure 2: SPARC plans before acting in a FHIR-based interactive diagnostic environment. State tracks the observed patient history and current diagnostic beliefs. Dreamer simulates multiple possible test sequences without executing them, comparing their anticipated diagnostic benefit, cumulative cost, and clinical risk. Explore executes the selected next action through a FHIR test request. SPARC then incorporates the returned result, updates its state, and repeats this process until it can safely stop and issue a diagnosis.

4.2 FHIR-Based Test-Result Environment

The test-result environment converts diagnostic actions selected by SPARC into test requests and returns results for subsequent decisions. Candidate actions are drawn from a standardized FHIR test catalog; each test is associated with a name, code, cost, and corresponding result resource. When SPARC selects a test, the environment creates a `ServiceRequest` and represents the returned result as an `Observation` or `DiagnosticReport`. The selected test and its result are added to the current case record for subsequent state updates. A concrete example is provided in Appendix H.4.

We consider two result environments. Both first query the original patient record and return the recorded result if the requested test was performed. When no recorded result is available, the real-result environment (R) returns "no result," whereas the real-plus-simulated environment (R+S) generates a simulated result after the test has been selected. Dreamer predicts possible test outcomes during planning using only the currently observable state; the test-result simulator generates feedback only after a test has been selected and does not participate in planning over candidate actions. The simulator's execution rules and prompt constraints are provided in Appendices H.6 and H.8.

4.3 State: Diagnostic State Updates

At step t, State maintains the planning state $S_t = (H_t, b_t)$, as defined in Section 3. Here, H_t contains the initial patient information and the tests selected so far with their returned results, while b_t is the diagnostic belief computed from H_t. Only information available up to step t is used; the reference diagnosis and unobserved test results remain hidden from SPARC.

After selecting test a_t and receiving result x_{t+1}, State appends (a_t, x_{t+1}) to the visible history and recomputes the diagnostic belief:

$$H_{t+1} = H_t \cup \{(a_t, x_{t+1})\}, \qquad b_{t+1} = P_\omega(\cdot \mid H_{t+1}).$$

Thus, $S_{t+1} = (H_{t+1}, b_{t+1})$. If the environment returns "no result," the attempted test is recorded, but no clinical finding is inferred from the missing result.

4.4 EXPLORE: RISK- AND COST-AWARE ACTION SELECTION

Explore compares the candidate pathways anticipated by Dreamer and selects either the next test or the action to stop and diagnose. For a test action a, its evaluation accounts for the cost of that test, the costs of subsequent tests, and the risk of ultimately missing a time-critical condition:

$$Q(S_t, a) = C(a) + \sum_x P_\theta(x \mid S_t, a) V(T(S_t, a, x)) \,,$$

where $C(a)$ is the test cost and V is the estimated remaining loss from the successor state. Thus, Explore evaluates a test through its effect on the subsequent diagnostic pathway rather than its immediate information gain alone.

To stop, the system assesses whether the current diagnostic belief, residual risk of missing time-critical conditions, and expected benefit of further testing satisfy the safe-stopping criteria. If they do, Explore outputs the final diagnosis. Otherwise, it executes the highest-valued feasible test and awaits feedback from the environment. When the maximum number of decision steps is reached or no feasible tests remain, the system ends the decision process and records the reason for termination, allowing forced termination to be distinguished from safe stopping during evaluation.

5 EXPERIMENTAL SETUP

5.1 DATASET

We introduce FISC, an institutional clinical dataset comprising 156 cases obtained from the Xiamen Medical Big Data Center and re-encoded as HL7 FHIR JSON files. Each case contains structured patient information, a reference diagnosis, recorded test results, and standardized diagnostic tests with associated codes and costs. The cases span ten clinical domains. Hepatobiliary and pancreatic diseases account for the largest share (26.9%), followed by pulmonary and thoracic diseases (17.3%) and gastrointestinal and acute abdominal emergencies (16.7%). Figure 6 shows the distribution across domains, and Table 3 lists their definitions and representative diagnoses.

Diagnostic tests are mapped to FHIR resources, allowing SPARC to select tests from a standardized catalog and receive their results. The resource mapping and catalog construction process are described in Appendix H.3. FISC also identifies time-critical conditions (TCCs), including acute myocardial infarction, pulmonary embolism, acute cerebrovascular conditions, and ovarian torsion. These conditions require timely recognition and appropriate intervention; delays in diagnosis or treatment may increase the risk of severe outcomes, including death, permanent neurological injury, or loss of organ function. The TCC diagnoses and corresponding case identifiers are listed in Table 4.

5.2 BASELINES AND EVALUATION

We compare SPARC with Greedy, ACTMED, `self_depending`, and `primary_info`, as detailed in Appendix H.1.All methods receive the same initial patient information and use the same candidate test catalog, test-cost table, and test-result environment. At each step, a method selects an unused test or stops and returns a ranked list of diagnoses. Selected test results are added to the visible patient history; reference diagnoses are used only for evaluation. The comparison methods and shared execution protocol are detailed in Appendix H.1.

We evaluate SPARC at planning depths $D_{\text{search}} \in \{0, 1, 2, 3, 4, 5\}$ to examine the effect of lookahead, where depth 0 denotes no lookahead. We also compare two test-result environments: R returns recorded results when available and "no result" otherwise, whereas R+S returns recorded results when available and simulated results otherwise. The simulator is invoked only after a test has been selected. Further details of the result environments and simulator are provided in Appendix H.6.

The compared methods retain their original action-selection objectives; therefore, the cross-method comparison evaluates complete sequential diagnostic strategies rather than isolating the causal effect of lookahead. To isolate planning horizon within SPARC, we additionally vary D_{search} while holding the belief model, candidate generation, objective, stopping rule, action space, and execution environment fixed.

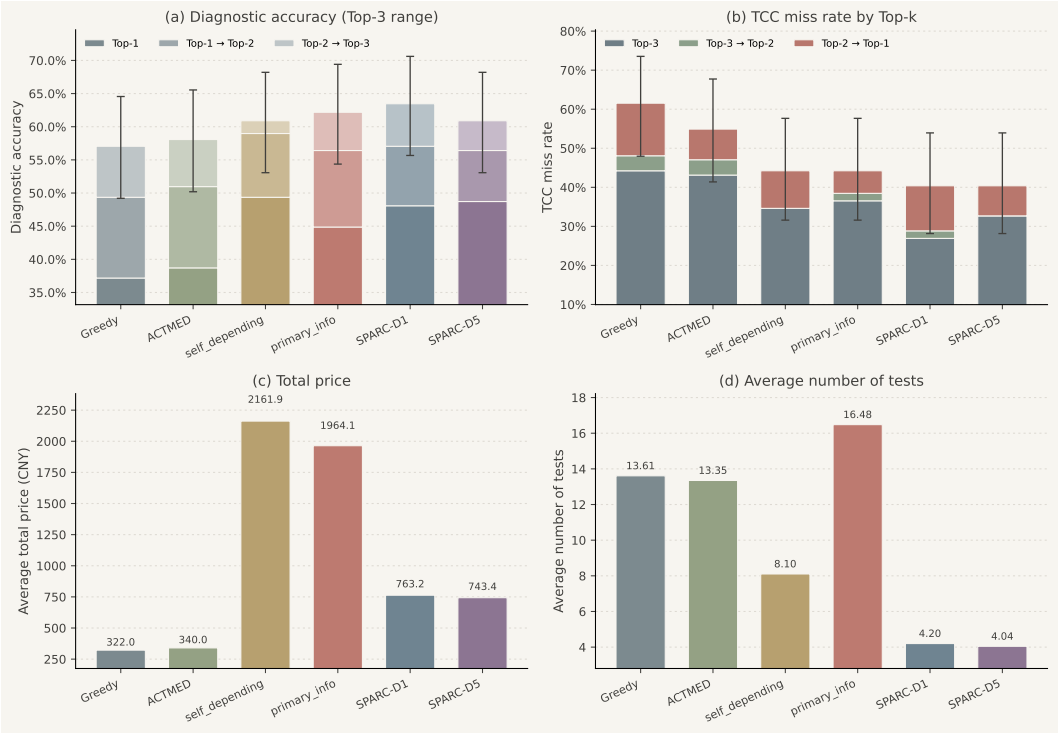

Figure 3: Diagnostic performance and resource use across Greedy, ACTMED, `self_depending`, `primary_info`, and SPARC at search depths 1 (D1) and 5 (D5). (a) Top-k diagnostic accuracy; stacked segments show Top-1 accuracy and the additional cases covered at Top-2 and Top-3. (b) Top-k TCC miss rate; stacked segments show the Top-3 miss rate and the additional cases missed at Top-2 and Top-1. (c) Average total testing cost per case in CNY. (d) Average number of tests per case. Error bars in (a) and (b) denote 95% Wilson score confidence intervals.

5.3 Evaluation Metrics

We evaluate diagnostic quality using Top-k accuracy for $k \in \{1, 2, 3\}$, defined as the proportion of cases whose reference diagnosis is covered by the first k predicted diagnoses. The diagnostic matching rule is specified in Appendix F.

To assess safety, we report the Top-k TCC miss rate among cases with a reference TCC: a case is counted as missed when its reference TCC is not covered by the first k predictions. Resource use is measured by the mean cumulative test cost in CNY and the mean number of selected tests per case; test-cost accounting is specified in Appendix H. For the reported accuracy and miss-rate estimates, error bars denote 95% Wilson score confidence intervals.

6 Results

6.1 RQ1: Can SPARC Improve the Recognition of Time-Critical Conditions with Fewer Tests?

As shown in Figure 3, SPARC uses the fewest tests on average while achieving relatively high diagnostic accuracy and low TCC miss rates. Compared with `primary_info`, SPARC reduces average testing cost by approximately 61%. Greedy and ACTMED incur lower costs, but have higher TCC miss rates, suggesting that prioritizing low testing cost may come at the expense of recognizing time-critical conditions. Overall, SPARC achieves a more favorable observed trade-off between testing cost and TCC recognition while reducing the number of tests.

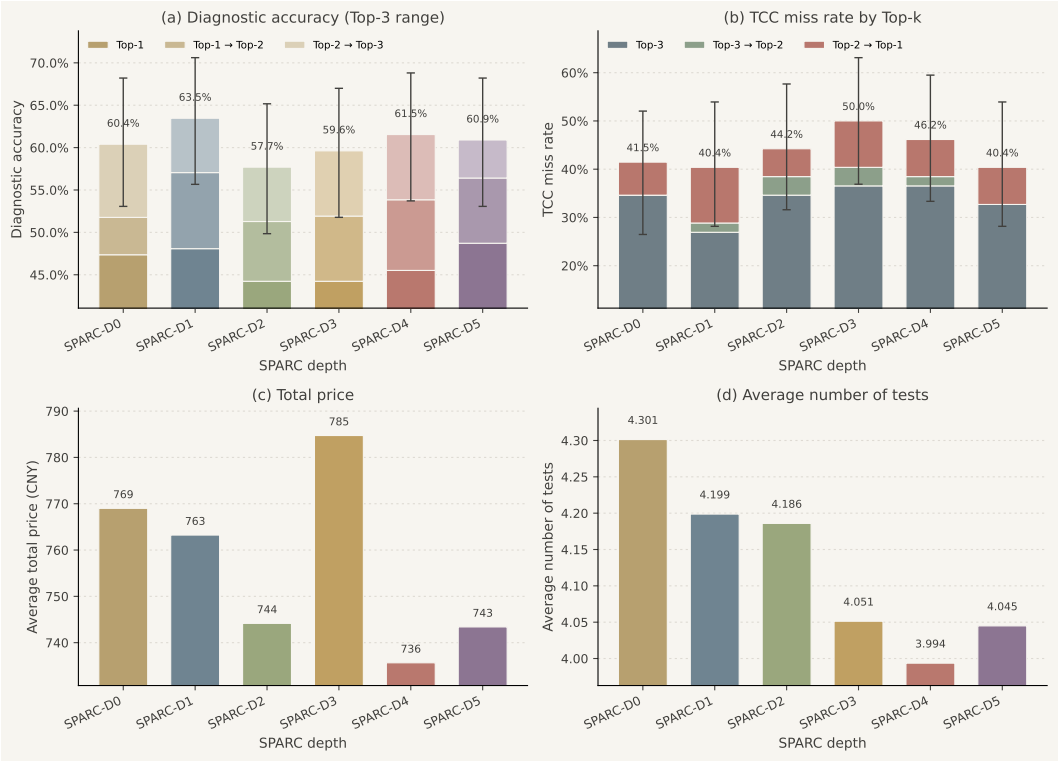

Figure 4: Effect of planning depth on diagnostic performance and resource use. D0 denotes no lookahead, while D1–D5 denote search depths of one to five. (a) Top-k diagnostic accuracy; (b) Top-k TCC miss rate; (c) average total testing cost per case; and (d) average number of tests per case. Error bars in (a) and (b) show 95% Wilson score confidence intervals.

6.2 RQ2: How Does Planning Depth Affect Diagnostic Performance and Resource Use?

As shown in Figure 4, one-step planning (D1) improves diagnostic accuracy while reducing the TCC miss rate, testing cost, and number of tests compared with no lookahead (D0). This suggests that even limited lookahead can improve diagnostic decision-making. Increasing the planning depth further does not lead to monotonic improvements: deeper search reduces resource use in some settings but can be accompanied by a higher TCC miss rate. Planning depth should therefore be chosen by considering diagnostic quality, resource use, and safety together.

6.3 RQ3: How Does Introducing Simulated Test Results Affect Diagnostic Performance and TCC Miss Rate?

In the real-result environment (R), a selected test returns its recorded result when available and "no result" otherwise. In the real-plus-simulated environment (R+S), a selected test returns its recorded result when available, while a simulated result is generated when no recorded result exists. The simulator protocol is provided in Appendix H.6. As shown in Figure 5, R+S achieves higher Top-1 diagnostic accuracy than R, while their Top-2 and Top-3 accuracies are similar. R+S has a lower Top-1 TCC miss rate but slightly higher Top-2 and Top-3 miss rates. One possible explanation is that simulated results provide additional feedback constrained by the patient's recorded information, allowing the model to update its diagnosis instead of proceeding with "no result."

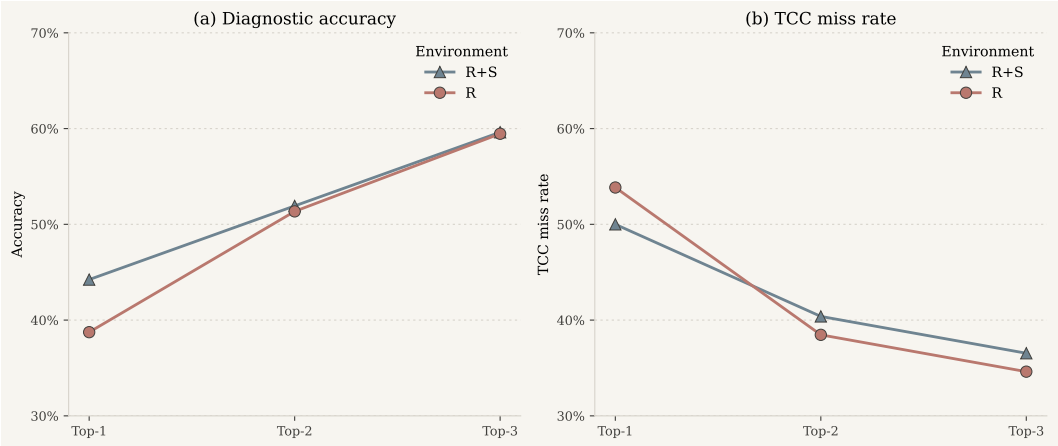

Figure 5: Effect of the test-result environment on diagnostic performance. (a) Top-k diagnostic accuracy; (b) Top-k TCC miss rate. R returns recorded results for tests performed in the original patient record and "no result" otherwise. R+S provides simulated results for tests without recorded results.

7 Discussion

Lookahead planning. Compared with no lookahead (D0), one-step planning (D1) improves diagnostic accuracy while reducing the observed TCC miss rate, testing cost, and number of tests in our experiments. This suggests that even limited lookahead can improve diagnostic decisions. SPARC predicts the possible outcomes of candidate tests and evaluates how those outcomes may affect subsequent test choices and stopping decisions.

Increasing planning depth further does not lead to consistent improvements. Some deeper settings reduce resource use, but diagnostic accuracy and TCC recognition do not improve accordingly. Longer planned pathways rely on more predicted test outcomes and diagnostic score updates, whose errors may affect action selection. The current experiments do not establish whether such errors explain the differences observed across depths.

Efficiency and diagnostic safety. SPARC considers testing costs alongside the risk of missing time-critical conditions. Compared with the lowest-cost baselines, it incurs higher testing costs but shows lower observed TCC miss rates while selecting substantially fewer tests. This pattern is consistent with a strategy that favors a small number of potentially informative tests, although the selected test trajectories require further analysis to establish that mechanism.

The results also illustrate why diagnostic efficiency cannot be assessed by cost or test count alone. A low-cost policy may leave important conditions unresolved, whereas additional spending may be warranted when it helps reduce the risk of a missed TCC. Because the observed differences in miss rates have not been established as statistically significant, these findings should be interpreted as an empirical trade-off rather than proof of a safety advantage.

8 Limitations

The study uses retrospective cases and has not been prospectively validated in clinical practice. Consequently, the observed diagnostic performance and resource use may differ in prospective settings, where patient conditions, test availability, and clinical workflows can change over time. Some unrecorded test results in the R+S environment are simulated rather than measured; errors in these results may propagate through subsequent belief updates, test selection, and stopping decisions. These findings therefore require validation with prospectively observed test results in clinical workflows.

Ethics statement

This study uses retrospective clinical cases that have undergone de-identification. Data processing followed GB/T 37964-2019, *Information Security Technology—Guide for De-identifying Personal Information*. Patient identifiers were removed or transformed before FISC was constructed, and access to the original records was restricted. To protect patient privacy, the original clinical records will not be released. The proposed method is evaluated for research purposes; simulated test results must not replace clinical measurements or physicians' judgment.

Reproducibility statement

The main text and appendices describe SPARC's planning procedure, FHIR-based test representation, comparison methods, evaluation metrics, experimental settings, and prompts. An anonymized code repository is available at `https://anonymous.4open.science/r/022121-F55B`. The original clinical records are not publicly available at this time due to patient privacy and data-use restrictions.

AI use statement

We used generative AI tools to assist with identifying relevant literature, discussing the rigor of the methodology and experimental design, writing and debugging code, and polishing the manuscript, including checks for spelling, grammar, and clarity. The authors independently verified the cited literature, reviewed and tested AI-assisted code, and revised all AI-assisted suggestions and text. We take full responsibility for the final content, findings, code, and claims of this work.

References

Corban Allenbrand. DECIDE-Lab: A value-of-information and POMDP framework for diagnostic laboratory test selection. *Diagnostics*, 16(15):2356, 2026. doi: 10.3390/diagnostics16152356. URL https://doi.org/10.3390/diagnostics16152356.

Joseph Bingham and Netanel Arussy. Cost-aware active feature acquisition for differential diagnosis under realistic clinical availability constraints. *medRxiv*, 2026. doi: 10.64898/2026.08.30.26361745. URL https://doi.org/10.64898/2026.08.30.26361745.

Zhong Cao and Shuying Chen. VeriDx: Earning the right to diagnose with disease-centric verification, 2026. URL https://arxiv.org/abs/2609.14018.

Xi Chen, Hanyu Zhou, Huahui Yi, Mingke You, WeiZhi Liu, Li Wang, Ziyuan Qin, Hairui Li, Xue Zhang, Yingman Guo, Songlin Li, Yunan Hu, Qianjiang Xiong, Rongxiao Li, Lei Fan, Qicheng Lao, Weili Fu, Jian Li, and Kang Li. Grounding large language models in clinical diagnostics. *Nature Communications*, 17:4401, 2026. doi: 10.1038/s41467-026-70274-w. URL https://doi.org/10.1038/s41467-026-70274-w.

Allan S. Detsky. Learning the art and science of diagnosis. *JAMA*, 327(18):1759–1760, 2022. doi: 10.1001/jama.2022.4650. URL https://doi.org/10.1001/jama.2022.4650.

A. V. Gheorghe, H. N. Bali, W. J. Hill, and E. R. Carson. Dynamic decision models for clinical diagnosis. *International Journal of Bio-Medical Computing*, 7(2):81–92, 1976. doi: 10.1016/0020-7101(76)90008-8. URL https://doi.org/10.1016/0020-7101(76)90008-8.

Hsin-Ling Hsu, Zizheng Wang, Donghua Zhang, Nai-Chia Chen, Jerry Wang, Jun-En Ding, Chia-Hsuan Hsu, Guoan Wang, Feng Liu, Fang-Ming Hung, Chenwei Wu, and Liyue Shen. MedAction: Towards active multi-turn clinical diagnostic LLMs, 2026. URL https://arxiv.org/abs/2605.07305.

Moritz Lehne, Stefan Luijten, Paul Vom Felde Genannt Imbusch, and Sylvia Thun. The use of FHIR in digital health – a review of the scientific literature. *Studies in Health Technology and Informatics*, 267:52–58, 2019. doi: 10.3233/SHTI190805. URL https://doi.org/10.3233/SHTI190805.

Yang Li and Junier Oliva. Active feature acquisition with generative surrogate models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6450–6459. PMLR, 2021. URL https://proceedings.mlr.press/v139/li21p.html.

Joshua C. Mandel, David A. Kreda, Kenneth D. Mandl, Isaac S. Kohane, and Rachel B. Ramoni. SMART on FHIR: A standards-based, interoperable apps platform for electronic health records. *Journal of the American Medical Informatics Association*, 23(5):899–908, 2016. doi: 10.1093/jamia/ocv189. URL https://doi.org/10.1093/jamia/ocv189.

Milad Memarzadeh and Matteo Pozzi. Value of information in sequential decision making: Component inspection, permanent monitoring and system-level scheduling. *Reliability Engineering & System Safety*, 154:137–151, 2016. doi: 10.1016/j.ress.2016.05.014. URL https://doi.org/10.1016/j.ress.2016.05.014.

Joris L. J. M. Müskens, Rudolf Bertijn Kool, Simone A. van Dulmen, and Gert P. Westert. Overuse of diagnostic testing in healthcare: A systematic review. *BMJ Quality & Safety*, 31 (1):54–63, 2022. doi: 10.1136/bmjqs-2020-012576. URL https://doi.org/10.1136/bmjqs-2020-012576.

National Academies of Sciences, Engineering, and Medicine. *Improving Diagnosis in Health Care*. The National Academies Press, Washington, DC, 2015. doi: 10.17226/21794. URL https://doi.org/10.17226/21794.

Alexander Luke Ian Norcliffe, Changhee Lee, Fergus Imrie, Mihaela van der Schaar, and Pietro Liò. Stochastic encodings for active feature acquisition. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 46784–46814. PMLR, 2025. URL https://proceedings.mlr.press/v267/norcliffe25a.html.

Julianna Piskorz, Nicolás Astorga, Jeroen Berrevoets, and Mihaela van der Schaar. Active feature acquisition for personalised treatment assignment. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 4330–4338. PMLR, 2025. URL https://proceedings.mlr.press/v258/piskorz25a.html.

Silas Ruhrberg Estévez, Nicolás Astorga, and Mihaela van der Schaar. Timely clinical diagnosis through active test selection. In *Advances in Neural Information Processing Systems*, volume 38, pp. 95632–95670, 2025. doi: 10.52202/085713-2879. URL https://doi.org/10.52202/085713-2879.

Samuel Schmidgall, Rojin Ziaei, Carl Harris, Ji Woong Kim, Eduardo Pontes Reis, Jeffrey Jopling, and Michael Moor. AgentClinic: A multimodal benchmark for tool-using clinical AI agents. *npj Digital Medicine*, 9:499, 2026. doi: 10.1038/s41746-026-02674-7. URL https://doi.org/10.1038/s41746-026-02674-7.

Hajin Shim, Sung Ju Hwang, and Eunho Yang. Joint active feature acquisition and classification with variable-size set encoding. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/e5841df2166dd424a57127423d276bbe-Abstract.html.

Shaoting Tan, Ning Liu, Yuntao Du, Shuyue Wei, Wu Shuai, Qian Li, Yanyu Xu, Wei Zhang, Lizhen Cui, and Haitao Yuan. GraphDx: A cost-aware knowledge-enhanced multi-agent framework for sequential diagnosis. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 21721–21736. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1092. URL https://aclanthology.org/2026.findings-acl.1092/.

Michael Valancius, Maxwell Lennon, and Junier Oliva. Acquisition conditioned oracle for nongreedy active feature acquisition. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48957–48975. PMLR, 2024. URL https://proceedings.mlr.press/v235/valancius24a.html.

Shicheng Xu, Xin Huang, Zihao Wei, Liang Pang, Huawei Shen, and Xueqi Cheng. DxDirector: An agentic large language model driving the full-process clinical diagnosis. *Nature Communications*, 17:5614, 2026. doi: 10.1038/s41467-026-71928-5. URL https://doi.org/10.1038/s41467-026-71928-5.

Li Zhang, Georg Wölflein, Dyke Ferber, Junhao Liang, Zunamys I. Carrero, Xuewei Wu, Julien Vibert, Jan Clusmann, Lino Möhrmann, Elena E. Möhrmann, Catharina Wichmann, Fabian Wolf, Tim Lenz, and Jakob Nikolas Kather. On-premise medical AI agents for reliable clinical decision-making. *Nature Medicine*, 2026. doi: 10.1038/s41591-026-04609-x. URL https://doi.org/10.1038/s41591-026-04609-x.

A NOTATION

Table 1: Notation used in the problem formulation, planning procedure, execution environment, and evaluation.

Symbol	Definition
$n,\ N$	Case index and total number of evaluated cases.
r	Index of an experimental repetition or simulated-result realization.
$t,\ T,\ T_n$	Decision-step index, number of executed tests in a trajectory, and number of tests executed for case n, respectively.
I_0	Patient information available at the initial decision point.
H_t	Visible clinical history at step t, including I_0 and previously selected tests with their returned results.
$z^{(n)}$	Hidden case context for case n, unavailable to the planning module.
S_t	Planning state at step t, defined as $S_t = (H_t, b_t)$.
$b_t(d)$	Unnormalized confidence score assigned to candidate disease d at step t, bounded in $[0, 1]$.
$f_\omega(d, H_t)$	Language-model-based diagnostic scoring function; ω denotes its model parameters.
$\mathcal{D}$	Fixed candidate disease set.
$\mathcal{D}_{\mathrm{TCC}}$	Predefined subset of time-critical conditions (TCCs).
$\hat{d}(S_t)$	Candidate diagnosis with the highest confidence score in state S_t.
$B(S_t)$	Leading-diagnosis confidence score, equal to $\max_{d\in\mathcal{D}} b_t(d)$.
$d^{(n)}_{\mathrm{gold}}$	Reference diagnosis for case n.
$\mathrm{Match}(\hat{d}, d)$	Indicator that predicted diagnosis $\hat{d}$ matches disease d under the predefined diagnosis-matching rule.
$m(d, \hat{d})$	Diagnosis miss indicator, $1 - \mathrm{Match}(\hat{d}, d)$.
$\mathbb{I}[\cdot]$	Indicator function, equal to one when its condition holds and zero otherwise.
Actions and test results	
$a,\ a_t$	Candidate test action and action selected at step t.
a_{stop}	Explicit action to stop testing and issue a diagnosis.
$\mathcal{A}_{\mathrm{FHIR}}$	Fixed catalog of FHIR-compliant diagnostic test actions.
K	Number of test actions in $\mathcal{A}_{\mathrm{FHIR}}$.
$\mathcal{A}_{\mathrm{done}}(S_t)$	Tests already executed in the trajectory leading to S_t.
$A(S_t)$	Available unused tests at state S_t.
$\bar{A}(S_t)$	Candidate tests considered by the search, with a_{stop} included only when $\mathrm{Safe}(S_t) = 1$.

Table 1 (continued)

Symbol	Definition
$\mathcal{A}_{\text{real}}^{(n)}$	Tests with recorded results for case n.
$x,\ x_{t+1}$	Possible test result and result returned after executing a_t, respectively.
$\mathcal{X}(S,a)$	Finite set of predicted result branches for test a in state S.
M	Maximum number of predicted result branches retained per test.
$P_{\theta_{\text{plan}}}(x \mid S,a)$	Planning-stage probability assigned to result branch x using the visible state.
$\mathcal{T}_\psi(S,a,x)$	State-update function applied to state S, action a, and result x.
$x_{\text{real}}^{(n)}(a)$	Recorded result of test a for case n.
$x_{\text{sim}}^{(n,r)}(a)$	Simulated execution result of test a for case n in repetition r.
$P_{\theta_{\text{env}}}(x \mid a, z^{(n)}; r)$	Execution simulator's result distribution given the selected test and permitted hidden case context.
$\text{HasReal}^{(n)}(a)$	Indicator that test a has a recorded result for case n.
$\text{ResultEnv}^{(n,r)}(a)$	Result returned by the execution environment for the selected test.
$\text{CacheKey}(n,a,r)$	Key used to cache a simulated result for a case, test, and repetition.
Costs, planning, and stopping	
$C(a)$	Resource cost of performing test a.
C_{test}	Cumulative testing cost along a diagnostic trajectory.
C_{diag}	Fixed weight of the confidence-based general diagnostic loss.
$C_{\text{TCC}}^{\text{extra}}(d)$	Fixed penalty weight for missing TCC d.
$L_{\text{diag}}(S,\hat{d})$	Confidence-based general diagnostic planning loss.
$L_{\text{TCC}}^{\text{extra}}(S,\hat{d})$	TCC risk-weighted planning loss based on unnormalized disease scores.
$L_{\text{terminal}}(S,\hat{d})$	Terminal planning loss combining general diagnostic and TCC risk-weighted losses.
$\pi,\ \pi^*$	Diagnostic decision policy and policy minimizing the expected planning objective.
Π_{adm}	Policies satisfying the action, step-limit, and stopping constraints.
$\widehat{Q}(S,a)$	Estimated value of taking action a in state S.
$\widehat{V}(S)$	Estimated remaining value of state S during lookahead search.
$\widehat{V}_{\text{leaf}}(S)$	Approximate value assigned to a search branch truncated at the depth limit.
D_{search}	Maximum lookahead depth at each decision step.
$H_{\max}$	Maximum number of tests that may be executed for one case.
K_{action}	Maximum number of candidate tests passed to the search.
τ_B	Minimum leading-diagnosis confidence score required for active safe stopping.
τ_M	Maximum permitted TCC risk-weighted loss for active safe stopping.
$\text{Safe}(S)$	Indicator that state S satisfies the active safe-stopping conditions.
$\text{Boundary}(S)$	Indicator that a forced stopping boundary has been reached.
Evaluation measures	
$\text{Correct}^{(n)}$	Indicator that the final diagnosis matches the reference diagnosis for case n.
$\text{TCCMiss}^{(n)}$	Indicator that a reference TCC is missed for case n.

Table 1 (continued)

Symbol	Definition
Accuracy	Proportion of evaluated cases with a matching final diagnosis.
TCCMissRate	Proportion of reference-TCC cases whose final diagnosis misses the TCC.
AvgCost	Mean cumulative testing cost per case.
AvgTestCount	Mean number of executed tests per case.
ActiveSafeStopRate	Proportion of cases terminated by an active safe-stopping decision.
BoundaryStopRate	Proportion of cases terminated at a stopping boundary.
UnsafeForcedStopRate	Proportion of cases forced to stop without satisfying the safe-stopping conditions.
RealResultRate	Proportion of executed tests that returned recorded results.
SimulatedResultRate	Proportion of executed tests that returned simulated results.

B Detailed Problem Formulation

B.1 Sequential Diagnostic Trajectory

Given the initial patient information I_0, the diagnostic process consists of a sequence of test-selection actions, returned test results, state updates, and stopping decisions:

$$S_0 \xrightarrow{a_0,x_1} S_1 \xrightarrow{a_1,x_2} \cdots \xrightarrow{a_{T-1},x_T} S_T.$$

Here, S_t denotes the planning state at decision step t, a_t denotes the selected diagnostic action, and x_{t+1} denotes the result returned after executing a_t. The number of executed tests is bounded by

$$T \leq H_{\max},$$

where $H_{\max}$ is the maximum number of tests allowed for one diagnostic trajectory. The process terminates when the planner selects the stopping action, when no unused test remains, or when $H_{\max}$ is reached.

B.2 Patient State and Hidden State

Let $z^{(n)}$ denote the hidden state of case n, including the underlying disease and other clinically relevant information. The hidden state cannot be directly observed by the planning module.

The visible clinical history at step t is

$$H_t = \{I_0, (a_0, x_1), (a_1, x_2), \ldots, (a_{t-1}, x_t)\}.$$

The approximate planning state is defined as

$$\boxed{S_t = (H_t, b_t)}$$

where b_t is the diagnostic belief generated from the currently visible clinical history. The planning module only receives S_t and does not access $z^{(n)}$, the gold-standard diagnosis, or the results of unselected tests.

B.3 Diagnostic Belief and Diagnosis Matching

Let $\mathcal{D}$ denote the candidate disease set. Given the currently visible clinical history H_t, a language-model-based diagnostic module f_ω assigns a confidence score to each candidate disease:

$$b_t(d) = f_\omega(d, H_t), \qquad d \in \mathcal{D}.$$

Here, ω denotes the parameters of the language model used by the diagnostic module. Each score satisfies $b_t(d) \in [0, 1]$. The scores are elicited independently and are not normalized across diseases; in particular, $\sum_{d\in\mathcal{D}} b_t(d)$ need not equal one. The diagnostic belief prompt is provided in Appendix H.8.

B.4 FHIR Action Space

The available diagnostic tests are restricted to a fixed FHIR-compliant catalog:

$$\mathcal{A}_{\mathrm{FHIR}} = \left\{a^{(1)}, a^{(2)}, \ldots, a^{(K)}\right\}.$$

Each test action has a standardized identifier, a FHIR code, a test name, a result representation, and a predefined cost. At state S_t, the set of unused tests is

$$A(S_t) = \mathcal{A}_{\mathrm{FHIR}} \setminus \mathcal{A}_{\mathrm{done}}(S_t),$$

where $\mathcal{A}_{\mathrm{done}}(S_t)$ denotes the tests already executed on the current trajectory.

The stopping action is denoted by a_{stop}, and the legal action set is

$$\bar{A}(S_t) = A(S_t) \cup \{a_{\mathrm{stop}}\}.$$

The planner cannot generate a test outside $\mathcal{A}_{\mathrm{FHIR}}$. Selected tests are represented as FHIR `ServiceRequest` resources, and returned results are represented using the corresponding FHIR result resources.

B.5 Time-Critical Conditions

Time-critical conditions (TCC) refer to diseases for which delayed recognition or intervention may lead to rapid deterioration, severe disability, or death. Let

$$\mathcal{D}_{\mathrm{TCC}} \subseteq \mathcal{D}$$

denote the predefined TCC disease set.

For a candidate diagnosis $\hat{d}$, the additional expected loss associated with missing TCC is

$$L^{\mathrm{extra}}_{\mathrm{TCC}}(S, \hat{d}) = \sum_{d\in\mathcal{D}_{\mathrm{TCC}}} b(d)\, m(d, \hat{d})\, C^{\mathrm{extra}}_{\mathrm{TCC}}(d),$$

Here, $C^{\mathrm{extra}}_{\mathrm{TCC}}(d)$ is the fixed penalty weight for missing TCC d. Because $b(d)$ is an unnormalized confidence score, this quantity is a planning penalty rather than an expected monetary loss. The complete TCC list, ICD mapping, and case-level correspondence are provided in Section G.2.

B.6 Testing Cost and Diagnostic Loss

Let $C(a)$ denote the resource cost of test a. The cumulative testing cost of a trajectory is

$$C_{\mathrm{test}} = \sum_{t=0}^{T-1} C(a_t).$$

The expected general diagnostic error loss is

$$L_{\text{diag}}(S, \hat{d}) = C_{\text{diag}} \left[1 - b(\hat{d})\right],$$

where C_{diag} is the unit loss of a general diagnostic error.

The terminal loss is

$$L_{\text{terminal}}(S, \hat{d}) = L_{\text{diag}}(S, \hat{d}) + L_{\text{TCC}}^{\text{extra}}(S, \hat{d}).$$

The overall sequential planning objective is

$$\boxed{\pi^* = \arg \min_{\pi \in \Pi_{\text{adm}}} \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} C(a_t) + L_{\text{diag}} + L_{\text{TCC}}^{\text{extra}} \right]}$$

where Π_{adm} denotes the set of policies satisfying the FHIR action, maximum-step, and stopping constraints.

C Detailed SPARC Planning Procedure

C.1 Diagnostic Belief Generation

Given the visible clinical history H_t, the language-model-based diagnostic module assigns an independent confidence score $b_t(d) = f_\omega(d, H_t)$ to each candidate disease. The module returns the candidate diseases and their scores, from which SPARC identifies the leading diagnosis $\hat{d}(S_t)$ and its score $B(S_t)$. The scores are not normalized across diseases.

When a new result is added to the history, the module regenerates the scores from the updated history.

C.2 Candidate Test Generation

At each state S_t, SPARC removes tests that have already been executed and retains only valid actions from the FHIR catalog:

$$A(S_t) = \mathcal{A}_{\text{FHIR}} \setminus \mathcal{A}_{\text{done}}(S_t).$$

The stopping action a_{stop} is added to the candidate action set. Invalid actions, duplicate tests, and tests without a valid FHIR representation are excluded.

C.3 Counterfactual Test-Outcome Prediction

For each candidate test a, SPARC predicts a finite set of possible outcomes:

$$\mathcal{X}(S, a) = \left\{ x^{(1)}, x^{(2)}, \ldots, x^{(M)} \right\}.$$

The planning-stage outcome distribution is

$$P_{\theta_{\text{plan}}}(x \mid S, a),$$

with

$$\sum_{x \in \mathcal{X}(S,a)} P_{\theta_{\text{plan}}}(x \mid S, a) = 1.$$

For each possible result x, SPARC constructs a hypothetical successor state:

$$S' = \mathcal{T}_\psi(S, a, x).$$

The hypothetical result is used only for planning and is not treated as an observed clinical result until the corresponding test is actually executed.

C.4 Finite-Horizon AND–OR Search

SPARC evaluates candidate tests using a stochastic AND–OR search. The OR node represents the selection of a test or the stopping action, whereas the AND node represents the possible outcomes of the selected test.

For a candidate test a, the estimated action value is

$$\widehat{Q}(S, a) = C(a) + \sum_{x \in \mathcal{X}(S,a)} P_{\theta_{\text{plan}}}(x \mid S, a) \widehat{V}\left(\mathcal{T}_\psi(S, a, x)\right).$$

The search is limited to a predefined depth D_{search}. If a search branch reaches this depth before the diagnostic process terminates, its value is approximated by

$$\widehat{V}_{\text{leaf}}(S) = L_{\text{terminal}}\left(S, \hat{d}(S)\right).$$

This leaf value is used only to estimate action values at the search horizon. Reaching the depth limit does not terminate the executed diagnostic trajectory or override the safety-constrained stopping rule. Because this approximation omits tests and observations beyond the search horizon, it may bias the estimated value of actions whose benefits emerge later. SPARC executes only the selected first action and replans after observing its result.

The complete recursive search procedure is given in Algorithm E.1.

C.5 Risk-Sensitive Value Function

The value function combines testing cost, general diagnostic error, and TCC miss loss:

$$\widehat{V} = \text{testing cost} + \text{general diagnostic loss} + \text{TCC miss loss}.$$

At a terminal state, the value is

$$L_{\text{terminal}}(S, \hat{d}) = C_{\text{diag}}[1 - B(S)] + L_{\text{TCC}}^{\text{extra}}(S, \hat{d}).$$

Therefore, a test may be selected even when it is not the cheapest available test if its possible future results reduce the expected diagnostic or TCC miss loss.

C.6 Safety-Constrained Stopping

Stopping is represented as an explicit action. SPARC allows active safe stopping only when

$$B(S_t) \geq \tau_B$$

and

$$L_{\text{TCC}}^{\text{extra}}\left(S_t, \hat{d}(S_t)\right) \leq \tau_M.$$

Here, τ_B is the minimum diagnostic belief threshold and τ_M is the maximum acceptable expected TCC miss loss. Their fixed values for this study are reported in Appendix E.3.

C.7 Receding-Horizon Replanning

SPARC executes only the first action selected by the current search. After receiving the result x_{t+1}, it updates the patient state:

$$S_{t+1} = \mathcal{T}_\psi(S_t, a_t, x_{t+1}).$$

The search is then performed again from the updated state. The resulting closed-loop process is

$$S_t \to a_t \to x_{t+1} \to S_{t+1} \to a_{t+1}.$$

D Execution Environment and Information Control

D.1 Real and Simulated Test Results

For case n, define

$$\mathrm{HasReal}^{(n)}(a) = \mathbb{I}\left[a \in \mathcal{A}^{(n)}_{\mathrm{real}}\right],$$

where $\mathcal{A}^{(n)}_{\mathrm{real}}$ is the set of tests with available real results.

For repetition r, the execution environment returns

$$\mathrm{ResultEnv}^{(n,r)}(a) = \begin{cases} x^{(n)}_{\mathrm{real}}(a), & \mathrm{HasReal}^{(n)}(a) = 1, \\ x^{(n,r)}_{\mathrm{sim}}(a), & \mathrm{HasReal}^{(n)}(a) = 0. \end{cases}$$

In the R+S environment, when a real result is unavailable, the execution environment generates a simulated result according to

$$\widetilde{x}^{(n,r)}_{\mathrm{sim}}(a) \sim P_{\theta_{\mathrm{env}}}\left(x \mid a, z^{(n)}; r\right).$$

Here, $z^{(n)}$ denotes the permitted de-identified case context available to the execution environment after the test has been selected. It is not available to the planning module before action selection and does not expose the gold diagnosis or unobserved future test results.

The simulated result is then converted into the corresponding FHIR result representation. In the R environment, the corresponding missing result is represented as "no result".

D.2 Result Caching

Each simulated result is indexed by

$$\mathrm{CacheKey}(n, a, r).$$

If the cache entry already exists, the environment returns the stored result. Otherwise, it generates one result, stores it, and returns it. Thus,

$$x^{(n,r)}_{\mathrm{sim}}(a) = \mathrm{Cache}\left[\mathrm{CacheKey}(n, a, r)\right].$$

For a fixed case n, test a, and repetition r, all compared methods receive the same simulated result. The planning module cannot query the cache before selecting the corresponding test.

D.3 FHIR Result Representation

All returned results are converted into a common FHIR representation. Laboratory and measurement results are represented using `Observation`, whereas report-level findings can be represented using `DiagnosticReport` or an associated FHIR resource.

The result source, whether real or simulated, is recorded internally for auditing but is not provided as an input to the planning module.

D.4 Agent–Environment Information Boundary and Leakage Control

We enforce a strict information boundary between the diagnostic agent and the patient environment. All agent–environment interactions are mediated through the FHIR action–observation interface. At decision step t, the planning module can access only the current visible state S_t, the remaining legal FHIR actions, their costs, and the planning-stage outcome distributions. It cannot directly query the underlying patient record or the execution simulator.

After the planner selects an action a_t, the environment returns only the FHIR-encoded result associated with that selected action. This returned result is then appended to the visible history and becomes available at the next decision step. No other information from the environment is exposed to the planner. Recorded and simulated results therefore enter the diagnostic agent through the same action–observation interface.

The planning module can access only the current visible state S_t, the remaining FHIR actions, their costs, and the planning-stage outcome distribution. The execution environment may access the permitted de-identified case context only after the planner has selected a test that requires result generation.

The planning module cannot access:

- the gold-standard diagnosis;
- the hidden state $z^{(n)}$;
- the complete future medical record;
- results of unselected tests;
- the set of tests with available real results;
- simulation-cache contents;
- the source of the returned result;
- the future action sequence of another method.

The hidden state $z^{(n)}$ is accessed only by the execution environment after the planner has selected a test.The execution environment uses the permitted hidden context only to produce the result of the selected test. The generated result is converted into the same FHIR-compatible representation as a recorded result and is then added to the visible case history.

E Algorithm and Implementation Details

E.1 SPARC Algorithm

[t] SPARC sequential diagnostic planning [1]

Initial information I_0, FHIR catalog $\mathcal{A}_{\mathrm{FHIR}}$, maximum number of tests $H_{\max}$, search depth D_{search} Final diagnosis and executed test trajectory

Initialize $H_0 \leftarrow I_0$ Generate $b_0 \leftarrow f_\omega(\cdot, H_0)$ Set $S_0 \leftarrow (H_0, b_0)$

$t = 0, \ldots, H_{\max} - 1$

Construct $A(S_t) \leftarrow \mathcal{A}_{\mathrm{FHIR}} \setminus \mathcal{A}_{\mathrm{done}}(S_t)$

$A(S_t) =$ **break**

Generate up to K_{action} candidate tests from $A(S_t)$

each candidate test a Generate up to M possible outcomes $\mathcal{X}(S_t, a)$ Estimate $\widehat{Q}(S_t, a) \leftarrow C(a) + \sum_{x \in \mathcal{X}(S_t,a)} P_{\theta_{\text{plan}}}(x \mid S_t, a)\widehat{V}(\mathcal{T}_\psi(S_t, a, x))$

Set $\bar{A}(S_t)$ to the candidate test set

$\text{Safe}(S_t) = 1$ Set $\widehat{Q}(S_t, a_{\text{stop}}) \leftarrow L_{\text{terminal}}(S_t, \hat{d}(S_t))$ Add a_{stop} to $\bar{A}(S_t)$

Select $a_t \leftarrow \arg\min_{a \in \bar{A}(S_t)} \widehat{Q}(S_t, a)$

$a_t = a_{\text{stop}}$ Output $\hat{d}(S_t)$ **terminate**

Convert a_t into a FHIR `ServiceRequest` Execute a_t in the result environment Receive $x_{t+1} \leftarrow \text{ResultEnv}^{(n,r)}(a_t)$ Update $S_{t+1} \leftarrow \mathcal{T}_\psi(S_t, a_t, x_{t+1})$

Output the leading diagnosis at the final state

E.2 Search Parameters

The maximum number of executed tests per case is $H_{\max} = 20$. At each decision step, the search retains up to $M = 3$ outcome branches per test and evaluates up to $K_{\text{action}} = 3$ candidate tests. These parameters are held fixed across all experiments. The lookahead depth D_{search} is specified separately for each experimental condition. In the environment ablation, $D_{\text{search}} = 3$. These values were specified before evaluation and were not selected through a validation-set search.

E.3 Stopping Parameters

Active stopping is permitted only when the leading diagnostic belief satisfies $B(S_t) \geq \tau_B$ and the expected additional loss from missing a time-critical condition satisfies

$$L_{\text{TCC}}^{\text{extra}}\left(S_t, \hat{d}(S_t)\right) \leq \tau_M.$$

For the experiments in this study, we fix $\tau_B = 0.85$ and $\tau_M = 2{,}000$ CNY. We also set $C_{\text{TCC}}^{\text{extra}}(d) = 100{,}000$ CNY for every $d \in \mathcal{D}_{\text{TCC}}$. These values are held fixed across all experiments and were not selected through a validation-set search.

A case also terminates when no unused test remains or the maximum number of executed tests is reached. Such boundary termination does not imply that the safety conditions were satisfied.

E.4 Invalid Output Handling

The following fixed rules are applied:

1. A test outside the FHIR catalog is rejected.
2. A previously executed test is removed from the candidate set.
3. Duplicate outcome branches are merged.
4. Outcome probabilities are checked and renormalized.
5. Invalid structured outputs are retried a fixed number of times.
6. If the retry limit is exceeded, the action is marked invalid.
7. If the search budget is exceeded, the planner falls back to single-step action evaluation.
8. All failures, retries, invalid actions, and fallback decisions are recorded in the audit log.

F Evaluation Metrics and Additional Experiments

F.1 Diagnostic Accuracy

For Top-k evaluation, a reference diagnosis is covered if at least one of the first k predicted ICD-10 codes matches its first three characters. Exact full-code agreement is not required. The same matching rule is used for Top-k TCC miss rate. For case n, the correctness indicator is

$$\mathrm{Correct}^{(n)} = \mathbb{I}\left[\mathrm{Match}\left(\hat{d}_T^{(n)}, d_{\mathrm{gold}}^{(n)}\right) = 1\right].$$

The final diagnostic accuracy is

$$\mathrm{Accuracy} = \frac{1}{N}\sum_{n=1}^{N} \mathrm{Correct}^{(n)}.$$

F.2 TCC Miss Rate

A TCC miss occurs when the gold-standard diagnosis belongs to $\mathcal{D}_{\mathrm{TCC}}$ but the final diagnosis does not match it:

$$\mathrm{TCCMiss}^{(n)} = \mathbb{I}\left[d_{\mathrm{gold}}^{(n)} \in \mathcal{D}_{\mathrm{TCC}} \wedge \mathrm{Match}\left(\hat{d}_T^{(n)}, d_{\mathrm{gold}}^{(n)}\right) = 0\right].$$

The TCC miss rate is

$$\mathrm{TCCMissRate} = \frac{\sum_{n=1}^{N} \mathrm{TCCMiss}^{(n)}}{\sum_{n=1}^{N} \mathbb{I}\left[d_{\mathrm{gold}}^{(n)} \in \mathcal{D}_{\mathrm{TCC}}\right]}.$$

F.3 Testing Cost and Test Count

The average testing cost is

$$\mathrm{AvgCost} = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T_n-1} C(a_t^{(n)}).$$

The average number of executed tests is

$$\mathrm{AvgTestCount} = \frac{1}{N}\sum_{n=1}^{N} T_n.$$

F.4 Stopping Statistics

The active safe stopping rate is

$$\mathrm{ActiveSafeStopRate} = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[a_T^{(n)} = a_{\mathrm{stop}} \wedge \mathrm{Safe}(S_T^{(n)}) = 1\right].$$

The boundary stopping rate is

$$\mathrm{BoundaryStopRate} = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[\mathrm{Boundary}(S_T^{(n)}) = 1\right].$$

Table 2: Paired comparison between SPARC-D0 and SPARC-D1. Values in the final column show paired differences with 95% confidence intervals.

Outcome	SPARC-D0	SPARC-D1	Paired Δ [95% CI]
Top-1 accuracy (%)	60.4	63.5	$+3.0$ $[-2.2, +8.3]$
TCC miss rate (%)	41.5	40.4	-1.1 $[-10.0, +7.8]$
Average total cost (CNY)	769	763	-6 $[-25, +13]$
Average number of tests	4.301	4.199	-0.101 $[-0.178, -0.022]$

The unsafe forced stopping rate is

$$\mathrm{UnsafeForcedStopRate} = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[\mathrm{Boundary}(S_T^{(n)}) = 1 \wedge \mathrm{Safe}(S_T^{(n)}) = 0\right].$$

F.5 Result Environment Statistics

The real-result rate is

$$\mathrm{RealResultRate} = \frac{\sum_{n=1}^{N}\sum_{t=0}^{T_n-1} \mathbb{I}\left[\mathrm{HasReal}^{(n)}(a_t^{(n)}) = 1\right]}{\sum_{n=1}^{N} T_n}.$$

The simulated-result rate is

$$\mathrm{SimulatedResultRate} = 1 - \mathrm{RealResultRate}\,.$$

We report the proportions of executed tests that return recorded or simulated results, as defined in Appendix F. The fidelity of simulated results to recorded clinical measurements was not evaluated in this study.

F.6 Paired Comparison of No Lookahead and One-Step Planning

To more directly assess the effect of introducing lookahead, we performed a paired case-level comparison between SPARC-D0 and SPARC-D1. These two settings use the same belief model, candidate generation procedure, objective function, stopping rule, action space, and execution environment, differing only in the planning depth. D0 performs no lookahead, whereas D1 evaluates one-step anticipated outcomes before selecting the next action.

Because both settings are evaluated on the same cases, uncertainty in their difference was estimated using paired case-level resampling rather than by comparing the marginal confidence intervals of the two methods. For diagnostic accuracy and TCC miss rate, we report the absolute paired difference in percentage points. For cumulative testing cost and number of executed tests, we report the paired mean difference. The 95% confidence intervals of these differences were obtained using paired bootstrap resampling of cases with replacement.

The paired analysis evaluates whether the observed differences between D0 and D1 are stable across cases while controlling for the shared evaluation set. Rather than treating overlapping marginal confidence intervals as evidence for or against a difference, the analysis directly quantifies uncertainty in the within-case change induced by introducing one-step lookahead.

F.7 Additional Experiments

We consider the following controlled variants in the experimental design:

1. **Single-step planning:** set $D_{\mathrm{search}} = 1$;
2. **No testing cost:** remove $C(a)$ from the value function;

3. **No TCC penalty:** remove $L_{\mathrm{TCC}}^{\mathrm{extra}}$;
4. **No safety-constrained stopping:** remove the τ_B and τ_M constraints;
5. **Full simulation environment:** generate all selected results using the same execution environment;
6. **Real-result-only environment:** evaluate only tests with available real results.

All comparison methods and ablations use the same FHIR catalog, diagnosis matching rule, cost table, and evaluation metrics.

G Dataset Description

This appendix describes the clinical-domain distribution and predefined time-critical conditions (TCCs) in FISC, our institutional dataset of 156 clinical cases.

G.1 Clinical-Domain Distribution

The 156 FISC cases span ten clinical domains. Figure 6 shows the distribution of cases across these domains. The domain definitions and representative diagnoses are listed in Table 3.

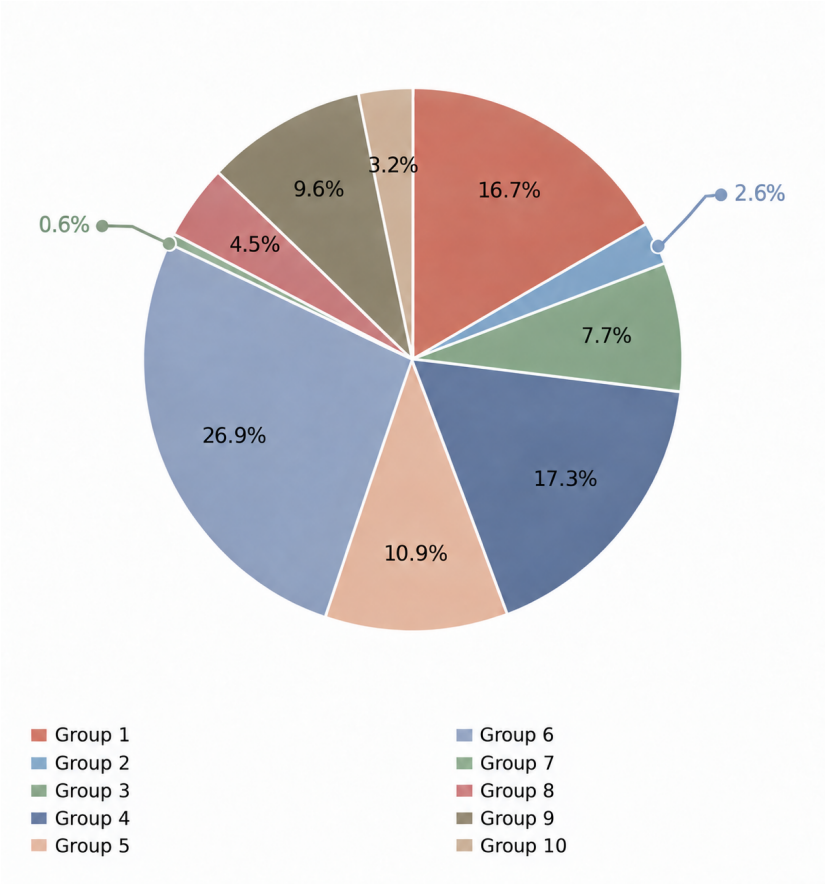

Figure 6: Clinical-domain distribution of the 156 cases in FISC.

G.2 Time-Critical Conditions

Table 4 lists the predefined TCC diagnoses in FISC and their associated case identifiers. ICD-10 categories are represented by the first three characters of each code.

Table 3: Clinical domains and representative diagnoses in the FISC dataset.

Group	Clinical domain	Representative diagnoses
1	Gastrointestinal and acute abdominal emergencies	Acute pancreatitis; acute appendicitis; acute suppurative appendicitis; colonic diverticulitis; cholelithiasis with acute cholecystitis; perforated peptic ulcer
2	Gynecological and obstetric emergencies	Acute pelvic inflammatory disease; ovarian torsion; hemorrhagic tubal pregnancy
3	Respiratory and reflux-related diseases	Cough-variant asthma; gastroesophageal reflux disease; eosinophilic airway disease
4	Pulmonary and thoracic diseases	Bronchiectasis; pulmonary tuberculosis; lung cancer; other pulmonary diseases
5	Cardiovascular and thromboembolic emergencies	Acute pulmonary embolism; pulmonary embolism; acute myocardial infarction
6	Hepatobiliary and pancreatic diseases	Pancreatic cancer; malignant neoplasm of the head of the pancreas; biliary obstruction; common bile duct stone; acute liver failure; chronic hepatitis B
7	Urological diseases	Urinary tract obstruction
8	Endocrine and thyroid diseases	Graves' disease; hyperthyroidism; subacute thyroiditis
9	Neurological and neurosurgical emergencies	Cerebral infarction; lacunar cerebral infarction; intracerebral hemorrhage; cerebellar hemorrhage; subarachnoid hemorrhage; traumatic brain injury
10	Neuro-ophthalmic diseases	Optic neuritis; neuromyelitis optica; retinal vein occlusion

Table 4: Time-critical conditions (TTCs) identified in the FHIR-MSCC dataset. ICD-10 categories are represented using the first three characters of the code.

ICD-10 category	Disease or diagnosis	Related case IDs
I21	Acute myocardial infarction	074, 082
I26	Acute pulmonary embolism	070, 071, 072, 075, 076, 077, 079, 083
I26	Pulmonary embolism	078, 080, 081, 084, 085, 086
I26	Pulmonary embolism without mention of acute cor pulmonale	073
I60	Subarachnoid hemorrhage	140, 141, 142
I61	Intracerebral hemorrhage	148
I61	Cerebellar hemorrhage	149
I63	Cerebral infarction due to occlusion of cerebral artery	144
I63	Cerebral infarction	143, 147
I63	Lacunar cerebral infarction	138, 139, 145, 151
I63	Cerebellar infarction	150
N83	Ovarian torsion	027
O00	Hemorrhagic tubal pregnancy	030

H Experimental Setup

H.1 Compared Methods and Evaluation Protocol

We compare SPARC with Greedy, ACTMED, `self_depending`, and `primary_info`. All methods receive the same initial patient information and use the same candidate test catalog, test-cost table, and test-result environment.

At each decision step, a method either selects one unused diagnostic test or terminates and returns a ranked list of diagnoses. Once a test is selected, its result is added to the visible patient history

and becomes available for subsequent decisions. Tests that have already been executed are removed from the candidate set. Reference diagnoses are withheld during decision making and are used only for evaluation.

SPARC is evaluated with planning depths $D_{\text{search}} \in \{0, 1, 2, 3, 4, 5\}$. A planning depth of one corresponds to single-step action evaluation, whereas larger depths allow SPARC to consider downstream tests and possible result branches.

Greedy selects the test with the highest information-gain-per-cost score, defined as

$$\text{Score}_{\text{Greedy}}(a) = \frac{\text{IG}(a)}{\sqrt{\text{Cost}(a)}}.$$

ACTMED selects tests using its uncertainty-driven active test-selection utility. The test cost is retained for resource accounting but is not included in the ACTMED selection utility.

The `self_depending` and `primary_info` methods follow their respective decision rules while using the same action catalog, execution interface, and stopping constraints as the other methods. All methods produce Top-1, Top-2, and Top-3 diagnostic predictions.

H.2 Action Space and Test Prices

At each step, a method selects one unused diagnostic action from the shared action catalog. The selected action is removed from the candidate set after execution. Test prices are taken from the dataset price table and are used to compute cumulative testing cost. A selected test counts toward the test total and incurs its catalog cost even when the environment returns "no result".ACTMED selects actions using information gain only; its price is retained for accounting but is not provided to the selection utility. Greedy uses the information-gain-per-cost rule $\text{KL}/\sqrt{\text{price}}$.

A diagnostic test action in FISC corresponds to a clinically ordered examination or laboratory panel rather than to a single scalar measurement. Accordingly, one selected action may return multiple clinically related findings. For example, a complete blood count is treated as one diagnostic test action, while its returned result may contain multiple hematological measurements such as white blood cell count, hemoglobin, platelet count, and differential cell counts.

This design follows the granularity of examination ordering in the source hospital system. The reported number of tests therefore counts ordered diagnostic examinations or panels, not the number of individual variables contained in their returned results. Consequently, a trajectory with four selected tests may expose the diagnostic agent to substantially more than four clinical measurements.

H.3 FHIR-Based Examination Representation and Execution

We represent clinical examinations using standardized FHIR resources so that all cases and decision-making strategies share the same examination action space. Each available examination is associated with a unique LOINC code, a display name, a cost, and an expected result schema. The planner cannot freely generate arbitrary examination names or codes. Instead, it must select the next examination from the validated set of legal catalog entries that have not yet been executed for the current case.

At the resource level, an examination order is represented as a FHIR `ServiceRequest`, an individual test finding as an `Observation`, and, when appropriate, the corresponding report as a `DiagnosticReport`. Cost information is associated with the relevant examination or report resource and is used to evaluate action cost. The original FHIR case files are first parsed into a unified examination catalog and patient history. The planner therefore operates on structured standard actions rather than unprocessed natural-language examination names.

At each decision step, the controller provides the language model with the currently visible clinical state, the available examination catalog, and the required number of candidates. The language model does not directly return a FHIR resource and does not directly access a FHIR server. It returns a structured JSON object containing candidate LOINC codes, for example:

```
{"loinc_codes": ["<code 1>", "<code 2>", "<code 3>"]}.
```

The controller then validates the response as JSON, checks that every code is present in the catalog, enforces the requested cardinality, and rejects codes that have already been executed. After validation, the selected action is converted into a FHIR `ServiceRequest` representation. This design ensures that the language model can influence examination selection without bypassing the standardized catalog or inventing an unavailable examination.

After an action is selected, the environment returns an examination outcome according to whether a corresponding real result is available in the case. If a real result exists, the environment returns that result. Otherwise, an execution environment model generates a simulated result using only the permitted de-identified hidden context. Both sources are converted into the same FHIR-compatible `Observation`/`DiagnosticReport` format and appended to the visible case history.

The complete execution pipeline is therefore:

> FHIR resource parsing → standardized examination catalog construction → LLM generation of candidate LOINC codes → controller-side action validation → `ServiceRequest` construction → real or simulated `Observation`/`DiagnosticReport` return.

H.4 Example of Examination Execution and Result Feedback

This appendix presents a concrete example of how the environment converts a diagnostic action selected by SPARC into a FHIR examination request and returns the corresponding result for subsequent decision-making. The example is taken from a case involving a 43-year-old woman who presented with progressive exertional dyspnea and a small amount of hemoptysis. Previous chest computed tomography (CT) examinations had shown multiple pulmonary nodules and thin-walled lucencies in both lungs. The main diagnostic considerations included lymphangioleiomyomatosis (LAM), Birt–Hogg–Dubé syndrome, and pulmonary Langerhans cell histiocytosis.

At one decision step, SPARC selected a chest CT examination. The environment resolved this action against the standardized examination catalogue and created a FHIR `ServiceRequest` resource. The request was associated with the patient, the current encounter, and the requesting practitioner:

[language=json, caption=FHIR ServiceRequest generated for a chest CT examination, label=lst:chest-ct-service-request] ”resourceType”: ”ServiceRequest”, ”status”: ”active”, ”intent”: ”order”, ”category”: [”coding”: [”system”: ”http://terminology.hl7.org/CodeSystem/service-category”, ”code”: ”diagnostic”, ”display”: ”Diagnostic procedure”]], ”code”: ”text”: ”Chest CT” , ”subject”: ”reference”: ”Patient/369” , ”authoredOn”: ”2026-07-22T11:13:42Z”, ”requester”: ”reference”: ”Practitioner/368” , ”performer”: [”reference”: ”Organization/367”], ”encounter”: ”reference”: ”Encounter/370”

The examination catalogue mapped “Chest CT” to the catalogue item “Chest X-ray computed tomography without contrast”, with the local service code `EBAJT001`. The quantity was one examination and the unit price was 135 CNY. Because SPARC did not specify whether contrast enhancement was required, the environment used the plain CT variant and recorded this as a defaulted pricing decision. The pricing metadata was attached to the request using FHIR extensions:

[language=json, caption=Pricing metadata associated with the chest CT request, label=lst:chest-ct-pricing] ”url”: ”https://reapulmo.local/fhir/StructureDefinition/pricing-status”, ”valueCode”: ”defaulted”

In another decision step, SPARC selected a pulmonary function test. The environment generated a second `ServiceRequest`:

[language=json, caption=FHIR ServiceRequest generated for pulmonary function testing, label=lst:pft-service-request] ”resourceType”: ”ServiceRequest”, ”status”: ”active”, ”intent”: ”order”, ”category”: [”coding”: [”system”: ”http://terminology.hl7.org/CodeSystem/service-category”, ”code”: ”diagnostic”, ”display”: ”Diagnostic procedure”]], ”code”: ”text”: ”Pulmonary function test” , ”subject”: ”reference”: ”Patient/369” , ”authoredOn”: ”2026-07-22T11:14:20Z”, ”requester”: ”reference”: ”Practitioner/368” , ”performer”: [”reference”: ”Organization/367”], ”encounter”: ”reference”: ”Encounter/370”

This examination was associated with two catalogue items: flow–volume curve measurement, coded as `FJE02401`, with a unit price of 30 CNY, and pulmonary diffusion-capacity measurement, coded as `W0213001003`, with a unit price of 75 CNY. The estimated total cost was therefore 105 CNY. In contrast to the chest CT request, the pulmonary function examination was marked as an estimated rather than a defaulted price because the catalogue mapping was composed of multiple related procedures.

After execution, the environment returned the examination result as a FHIR `Observation` resource. The result was linked to the same patient and was marked as final:

[language=json, caption=FHIR Observation returned for the pulmonary function test, label=lst:pft-observation] ”resourceType”: ”Observation”, ”status”: ”final”, ”meta”: ”tag”: [”system”: ”https://reapulmo.local/CodeSystem/result-origin”, ”code”: ”observed-real”] , ”code”: ”coding”: [”system”: ”https://reapulmo.local/CodeSystem/test-catalog”, ”code”: ”pulmonary-function”, ”display”: ”Pulmonary function test”], ”text”: ”Pulmonary function test” , ”subject”: ”reference”: ”Patient/369” , ”effectiveDateTime”: ”2026-07-22T11:13:00Z”, ”performer”: [”reference”: ”Organization/367”], ”valueString”: ”Ventilatory and diffusion functions were normal. The bronchodilator test was negative. FeNO was 13 ppb.”

The environment then appended the returned `Observation` to the current case record. Consequently, the next SPARC decision step could use the following new facts:

- pulmonary ventilatory function was normal;
- pulmonary diffusion function was normal;
- the bronchodilator test was negative; and
- fractional exhaled nitric oxide (FeNO) was 13 ppb.

These findings did not independently establish a final diagnosis, but they provided additional evidence for the subsequent diagnostic update. In particular, the normal pulmonary function results suggested that the patient did not have substantial functional impairment at that stage, while the previously documented combination of multiple thin-walled pulmonary cysts, small nodules, and a history of renal angiomyolipoma continued to support LAM as the leading diagnosis. The environment therefore completed the loop

SPARC action $\rightarrow$ examination catalogue lookup $\rightarrow$ FHIR ServiceRequest $\rightarrow$ examination result $\rightarrow$ case-state update.

In this example, the returned examination result is represented by an `Observation`. For examinations that produce an interpreted report rather than a single measured result, the same environment can instead return a FHIR `DiagnosticReport`, whose `result` field references one or more `Observation` resources. Thus, both quantitative findings and narrative diagnostic reports can be incorporated into the case record and exposed to SPARC during the next decision round.

H.5 Model and Inference Settings

All methods use the same configured language model and inference environment. Model sampling parameters, the number of samples, random seed, request timeout, and retry policy are fixed across methods. API failures are retried; cases that remain incomplete after retry or recovery are excluded from metrics that require a final diagnosis.

H.6 Test-Result Simulator

The test-result simulator is used only when a selected diagnostic test has no recorded result in the original patient record. For a selected action a in case n, the simulator generates a standardized result

$$x_{\mathrm{sim}}^{(n,r)}(a) \sim P_{\theta_{\mathrm{env}}}\left(x \mid a, z^{(n)}; r\right),$$

where $z^{(n)}$ denotes the permitted de-identified case context available to the execution environment after action selection. It is separate from the planner-visible state S_t and is never exposed to the

planning module. The restricted context excludes the gold diagnosis, the target result being generated, results of unselected tests, future clinical information, and the physician's final diagnostic conclusion. And r denotes the experimental repetition.

The simulator is invoked only after the agent has selected the test. It is not used to provide the agent with the result of an unselected test, and it does not participate in SPARC's counterfactual planning. During planning, SPARC uses its own predicted outcome branches; after execution, the environment returns either the recorded real result or the simulated result.

After the agent selects a test, the simulator receives the selected test, the permitted de-identified case context, and the expected result schema. The simulator does not receive the planning state, the gold diagnosis, the results of unselected tests, or the future action sequence.It must return a result conforming to the schema, including the required field names, value types, units, and reference ranges. It cannot reveal the gold diagnosis, future test results, the physician's diagnostic conclusion, or the agent's predicted planning branches.

For a test with an available real result, the environment returns $x^{(n)}_{\text{real}}(a)$. For a test without a recorded result, it returns $x^{(n,r)}_{\text{sim}}(a)$. Thus, the execution result is defined as

$$\text{ResultEnv}^{(n,r)}(a) = \begin{cases} x^{(n)}_{\text{real}}(a), & a \in \mathcal{A}^{(n)}_{\text{real}}, \\ x^{(n,r)}_{\text{sim}}(a), & a \notin \mathcal{A}^{(n)}_{\text{real}}. \end{cases}$$

H.7 Masked-result fidelity validation

To evaluate whether the execution simulator produces clinically consistent observations without access to the target result, we performed a masked-result validation experiment using examinations with recorded results.

For each evaluated examination, the recorded target result was removed from the simulator input and treated as unobserved. The simulator was then asked to generate the result using the same execution protocol used in the R+S environment. The generated result was compared against the held-out recorded result, which was used only for evaluation and was never provided to the simulator. This experiment evaluates the fidelity of the execution environment rather than the diagnostic performance of SPARC.

The simulator was not designed to reconstruct exact numerical measurements. Accordingly, exact numerical agreement was lower than agreement in decision-relevant clinical direction. In particular, normal/abnormal and positive/negative classifications were substantially more stable than exact values.

H.8 Prompt Design

This appendix reports the functional prompt templates used by the diagnostic, planning, execution, and summarization modules. Text in angle brackets denotes runtime-filled fields. The prompts constrain every module to the information available at the corresponding decision state and prohibit access to the gold diagnosis or unobserved future results.

The system uses separate prompts for diagnosis, action selection, counterfactual planning, result execution, result summarization, and final diagnosis. The same prompt family is used across methods whenever the module has the same role; method-specific differences are limited to the action utility and planning information supplied to the module.

H.8.1 EXECUTION SIMULATION PROMPT

Execution Simulation Prompt

You are the test execution environment. After a test action has been selected, generate one standardized result using only the permitted de-identified hidden context. Do not reveal the gold disease, future tests, physician conclusions, planning branches, strategy names, or current predictions. Return only a JSON object with root key `value`. Every returned field must match `expected_result_schema`, including names, types, units, and reference ranges. The output must be a concrete test result and must not use placeholders such as "unknown" or "not available". Return only `{"value":[{"name":"<field>","value":"<result>", "reference_range":[]}]}` and no additional fields.

H.8.2 DIAGNOSTIC BELIEF PROMPT

Diagnostic Belief Prompt

You are a diagnostic belief module. Based only on the initial clinical information and the returned results of tests already executed, rank the most relevant ICD-10 differential diagnoses. The initial candidate diseases define only a group-level candidate range and are not confirmed diagnoses. Do not use the gold diagnosis, hidden records, unexecuted test results, future clinical information, or the physician's true pathway. Select codes only from `coarse_candidates`. Return exactly `differential_count` independent confidence values in the following JSON format: `"probabilities": {"<ICD-10 code>": 0.0}`. Each value must be in [0,1]; the values need not sum to one. Return only `"probabilities":{"<ICD-10 code>":0.0}` and no additional fields.

H.8.3 CANDIDATE TEST SELECTION PROMPT

Candidate Test Selection Prompt

You are a clinical test re-ranking module. Using only the currently visible state and the current differential diagnoses, select the tests that are most useful for distinguishing the candidate diagnoses from the supplied `candidate_tests`. Never use the gold diagnosis, hidden records, future results, physician pathways, or tests outside the supplied catalog. Return only a JSON object containing exactly `candidate_count` distinct legal LOINC codes: `"loinc_codes":["<code 1>", "<code 2>", "<code 3>"]`. Do not return Markdown, explanations, or additional fields.

H.8.4 COUNTERFACTUAL PLANNING PROMPT

Counterfactual Planning Prompt

You are a forward-planning test-outcome prediction module. Given the current visible state and one specified candidate test, predict distinct plausible result branches and their probabilities for search-tree evaluation. These branches are planning hypotheses, not patient observations. Do not use the gold diagnosis, hidden records, future executed results, or cached execution results. Return exactly `branch_count` branches whose probabilities sum to one and whose fields match `expected_result_schema`.

Return only the following JSON structure:

```
{"branches":[
{"value":[...],"probability":0.5},
{"value":[...],"probability":0.5}]}
```

Do not return additional fields.

H.8.5 CLINICAL RESULT SUMMARY PROMPT

Clinical Result Summary Prompt

You are a clinical test-result summarization module. Based only on the current returned test result, produce a short visible summary for the next planning step and final diagnosis. Keep abnormal values with units and reference ranges, key positive or negative findings, and decision-relevant imaging or pathology findings. Omit normal or repetitive details. Do not add diagnoses, treatment recommendations, future results, or gold-standard labels. Return only `"summary":"<Chinese clinical summary, at most 250 characters>"` and no additional fields.

H.8.6 FINAL DIAGNOSIS PROMPT

Final Diagnosis Prompt

You are the final clinical diagnosis module. Using only the initial visible information and returned test results, produce exactly three distinct differential diagnoses in descending confidence order. The Top-1 diagnosis is the final diagnosis. Every code must be a valid six-character Chinese ICD-10 extension code selected from `coarse_candidates`; do not use the gold diagnosis, hidden results, future tests, pathology conclusions, or treatment response. Return only: `"top_differential_diagnoses": [{"icd10_code":"", "confidence":0.0, "reason":""}, {"icd10_code":"", "confidence":0.0, "reason":""}, {"icd10_code":"", "confidence":0.0, "reason":""}]`. Do not return Markdown or additional fields.

H.8.7 ACTMED CONFIDENCE PROMPT

ACTMED Confidence Prompt

You are a clinical diagnostic confidence assessor. Based only on the current visible case information and returned test results, estimate the confidence of the most likely diagnosis. Return one number between 0 and 1 and no explanation. Return only a single numeric value, for example `0.73`.

H.8.8 ACTMED CANDIDATE-RESULT SAMPLING PROMPT

ACTMED Candidate-Result Sampling Prompt

Based on the current visible case information, simulate one reasonable possible result for the named test. Return only the result text, without explanation or JSON. This sampled result is used only to rank candidate tests and is not treated as the patient's observed result. Return only the result text, without Markdown or additional fields.

H.9 PLANNING AND STOPPING CONDITIONS

The maximum number of executed tests is 20. This is a ceiling rather than a required number of tests. A method may stop earlier when its confidence and safety conditions are satisfied, when no unused action remains, or when the maximum number of tests is reached. Planning depth controls the lookahead search and is distinct from the maximum number of executed tests.

H.10 EXECUTION ENVIRONMENT

Experiments are executed in the project Python environment using the same FHIR data, price table, model configuration, and random seed. ACTMED and Greedy recovery runs use separate result directories and are executed in parallel with a fixed number of workers per policy. Each case writes a checkpoint and step-level records so that interrupted runs can be resumed. Checkpoints record the executed history, selected actions, returned results, diagnostic predictions, completion status, and cumulative testing cost.

H.11 Overall Statistical Analysis

All reported statistics are aggregated at the dataset level over the 156 cases. We report case-level means for cost and test count and case-level percentages for diagnostic accuracy, TCC miss rate, and completion rate. Top-k denominators are reported explicitly when the corresponding diagnostic list is unavailable. We do not perform a separate case-level subgroup analysis in this appendix.

H.12 Environment Sensitivity

We evaluate two result environments: R, which returns recorded results when available and "no result" otherwise, and $R + S$, which returns recorded results when available and simulated results otherwise. Environment sensitivity is evaluated using the same Top-k accuracy and TCC miss-rate definitions. All other case information, method configurations, and evaluation rules remain fixed.