MedExAgent: Training LLM Agents to Ask, Examine, and Diagnose in Noisy Clinical Environments

Yicheng Gao[1] **Xiaolin Zhou**[2] **Yahan Li**[1] **Yue Zhao**[1] **Ruishan Liu**[1]
[1]University of Southern California [2]Arizona State University
{gaoyiche,yahanli,yue.z,ruishanl}@usc.edu, xzhou226@asu.edu

Abstract

Real-world clinical diagnosis is a complex process in which the doctor is required to obtain information from both interaction with the patient and conducting medical exams. Additionally, the doctor needs to adapt to different patient personas, as well as noisy and incomplete information that can happen at any time during the process. However, existing benchmarks for medical LLMs and methods for automatic diagnosis largely simplify this process by reducing it to single-turn question answering, noise-free conversations, or sequential exam making, etc., ignoring the interactive and uncertain nature of clinical diagnosis. In this paper, we aim to address this gap by formalizing clinical diagnosis as a Partially Observable Markov Decision Process (POMDP) with three action types: questioning the patient, ordering medical exams as tool calls, and issuing a diagnosis. We also introduce a systematic noise model comprising seven patient noise types and three exam noise types. Using our proposed environment, we train an effective diagnosis agent, **MedExAgent**, through a two-stage pipeline that first performs supervised finetuning on synthetic conversations structured after the Calgary-Cambridge model for clinical interviews, and then applies DAPO to optimize a composite reward capturing diagnostic accuracy, tool call quality, and exam cost including financial cost and patient discomfort. Through extensive experiments and ablation studies, we demonstrate that MedExAgent achieves diagnostic performance comparable to larger models while maintaining cost-efficient examination strategies.

Code: https://github.com/EndlessCG/medexagent

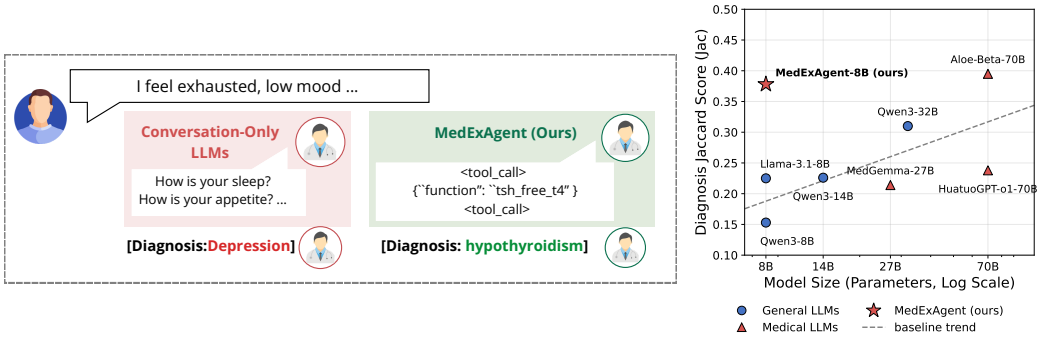

Figure 1: **MedExAgent overview.** (a) Tool-augmented diagnostic reasoning enables correct diagnoses where conversation-only baselines fail. (b) On the OOD test set AgentClinic-MedQA [42], Our 8B agent matches much larger baselines on Jac.

Disclaimer: MedExAgent is a research prototype. It must not be used for medical advice or patient care.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Diagnostic errors contribute to substantial patient harm each year, and access to timely, high-quality diagnostic care remains uneven across populations and geographies [33, 2, 17]. These pressures have motivated longstanding interest in computational support for clinical reasoning [45], and recent advances in large language models (LLMs) have brought this goal within closer reach. Medical LLMs now show strong performance on knowledge-intensive benchmarks such as MedQA [20] and MedMCQA [35], and a growing line of medical specialists [40, 26, 43, 10] are trained for these tasks; more recent diagnostic evaluations move closer to clinical use by asking models to reason over complete clinical cases [53, 12, 16]. Across this landscape, the model is handed a complete case description and asked to reason over it; however, in clinical practice, the case must be constructed through interaction with the patient. Building a deployable *diagnostic agent*, one that constructs the case itself, in real clinical conditions, is the subject of this work.

Clinical diagnosis as practiced differs from these benchmarks in three respects. It is *sequential*: a clinician begins with a chief complaint, asks questions to refine a working differential, orders exams as new evidence is needed, and commits to a diagnosis [4]. It is *noisy*: patients misreport symptom location or severity [38, 27], and laboratory results can be contaminated, partially missing, or ambiguous [44]. And it is *costly*: Exams range from a routine blood draw to an invasive biopsy, with corresponding differences in financial cost [32, 18] and patient discomfort that clinicians weigh against expected diagnostic yield [30]. Static benchmarks capture none of these properties.

Recent work has begun to move LLMs into interactive settings, but each effort handles only part of the picture. Benchmarks evaluate patient questioning [28, 42] or EHR tool-calling [19]; specialist agents are trained for conversational history-taking [47, 39] or for exam ordering [37]. Three properties of clinical practice consequently remain absent from the training signal of any single agent. First, no existing system optimizes a single agent over the combined action space of questioning, exam ordering, and diagnosis within one episode. Second, patient and exam observations are treated as clean rather than noisy. Third, the cost of investigation is rarely part of the objective. Whether a single agent can be trained to handle sequential interaction, noise, and cost together, and what such training requires, remains open.

We address this question along two axes: an environment that exposes the agent to all three properties, and a training recipe that teaches it to act within them. We formalize interactive diagnosis under observational uncertainty as a Partially Observable Markov Decision Process (POMDP). The action space unifies free-form patient questioning, exam ordering as tool calls, and final diagnosis under a single agent, closing the interleaving gap. Observational noise is injected stochastically into the observation function via a systematic noise model (seven patient noise types, three exam noise types), with diverse patient personas drawn from PatientSim [25]. Per-exam cost is grounded in CMS fee schedules [9, 8] for financial cost and in the access-and-invasiveness taxonomy of Whiteley et al. [48] for patient discomfort, so that an exam's cost reflects what it would impose on a real patient.

Within this environment we train **MedExAgent**, an 8B-parameter agent built on Meditron3-8B [40]. We draw on the Calgary–Cambridge model of medical interviewing [24, 23], a widely adopted framework in medical education that decomposes a clinical encounter into five stages: (1) initiating the session, (2) gathering information, (3) physical examination, (4) explanation and planning, and (5) closing. Doctor–patient conversations are synthesized following these stages and used as supervision. Training proceeds in two stages: supervised fine-tuning on these conversations, followed by DAPO [50] reinforcement learning against a composite reward over diagnostic accuracy, tool-call quality, and per-exam cost. Across in-distribution and out-of-distribution benchmarks, MedExAgent-8B matches or exceeds far larger baselines on diagnostic accuracy while maintaining cost efficiency.

Concretely, we make three contributions:

1. A POMDP environment for interactive clinical diagnosis that unifies patient questioning, exam ordering, and diagnosis, with patient and exam noise injected into the observation function and per-exam cost reflecting real-world financial burden and patient discomfort.

2. A training recipe that combines supervised fine-tuning on Calgary–Cambridge-structured doctor–patient conversations with reinforcement learning against a composite reward over diagnostic accuracy, tool-call quality, and per-exam cost.

3. An open 8B model showing that a small specialist trained in this environment matches or exceeds much larger baselines on diagnostic accuracy while maintaining cost-efficient examination strategies.

2 Related Works

2.1 Benchmarks and Environments for Interactive Diagnosis

Following the recent upsurge of LLM-based agents, a line of work aims to bridge the gap between text-based LLMs and interactive real-world clinical environments. MediQ [28] introduces a benchmark allowing the LLM to communicate with the patient. Results demonstrate that giving LLMs the flexibility to inquire the patient can boost performance significantly, highlighting the importance for agent-environment interaction. Similarly, AgentClinic [42] proposes a multi-agent framework to assess LLM's diagnosis performance in an environment allowing multi-modal information gathering. MedAgentBench [19] further evaluates LLMs in EHR-based clinical tasks such as patient information retrieval, lab test ordering, etc. However, existing environments and benchmarks often assume that the response to the action of LLMs are accurate, which ignores the noises that are common in real world. In fact, a large-scale study of 5,900 patients found only 5.5% complete agreement between self-reported data and EHR records [54]. Our work addresses this gap by systematically modeling patient noise, exam noise, and patient personalities in our environment.

2.2 LLM-based Systems for Clinical Diagnosis

Previous works in applying LLMs to conduct clinical diagnosis can be broadly split into two categories: multi-agent systems and finetuned expert models. Methods involving multi-agent systems typically assign different roles to a set of LLMs and design a workflow to orchestrate the diagnosis process. MAI-DxO [34] uses five LLMs to each role-play a step in the workflow (e.g. planning, hypothesis generation, cost management). MedAgents [46] and MDAgents [21] assign each LLM to a medical field (e.g. radiologist, oncologist, surgeon). These methods require extensively calling APIs of closed-source models, raising concerns about patient privacy which is highly sensitive in the field of medicine. Another line of work aims to finetune expert models for interactive clinical diagnosis. AMIE [47, 39] and ClinDiag-GPT [11] both train closed-source expert models for conversation-based information acquisition. DiagAgent [37] instead trains an agent to predict medical examinations for a given patient. Both directions only model part of the complete diagnosis procedure (either conversation or exam selection), which can restrict effectiveness and real-world authenticity.

3 Problem Formulation

In this paper, we study the problem of automatic diagnosis with medical exams as tool calls. We formalize this problem as a POMDP $(\mathcal{S}, \mathcal{A}, T, \Omega, O, R)$. The state space $\mathcal{S} = \mathcal{D}$ is a set of all possible diseases, from which a random disease d is sampled at the beginning of each episode as the ground truth disease the patient has. The action space $\mathcal{A}$ contains three types of actions the agent can take in each timestep t: $\mathcal{A} = \mathcal{A}_{\text{ask}} \cup \mathcal{A}_{\text{exam}} \cup \{a_{\text{dx}}\}$. If $a_t \in \mathcal{A}_{\text{ask}}$, the agent asks the patient a question; if $a_t \in \mathcal{A}_{\text{exam}}$, the agent chooses one exam from a predefined available exam list; if $a_t = a_{\text{dx}}$, the agent gives a final diagnosis $\hat{d} \in \mathcal{D}$. We assume that the patient's disease does not change or progress during the episode. This indicates that the transition function is deterministic: $T(s' \mid s, a) = \mathbb{1}(s' = s)$, where $\mathbb{1}$ is the indicator function. We detail other parts of the POMDP as follows.

3.1 Noise Modeling

In our POMDP definition, the observation space Ω contains two types of observations: $\Omega = \Omega_{\text{conv}} \cup \Omega_{\text{exam}}$, where Ω_{conv} represents all possible responses of the patient and Ω_{exam} represents all possible exam results. At timestep t, the observation $\omega_t \in \Omega_{\text{conv}}$ if and only if the action $a_t \in \mathcal{A}_{\text{ask}}$, and $\omega_t \in \Omega_{\text{exam}}$ if and only if the action $a_t \in \mathcal{A}_{\text{exam}}$.

In real world, a medical diagnosis session often contains noises. For example, the patient can mistakenly report the region of a pain (e.g. stomach pain $\rightarrow$ liver pain), and a medical exam can misreport certain symptoms (e.g. contaminated samples result in inaccurate blood test results). In

order to reflect this, we model such noises in the observation function $O(\omega \mid d, a)$. We consider seven types of patient noises and three types of exam noises detailed in Table 7. Each type of noise is applied independently with a fixed probability p_{conv} or p_{exam} on patient responses or exam results. Our noise taxonomy is derived from common error classifications in medical research, which repeatedly distinguish missing, incorrect or misinterpreted, and uncertain information. For example, Schiff et al. [41] separate failure to elicit information from inaccurate or misinterpreted information, Kim and Mansfield [22] distinguish under-reading and faulty reasoning in radiology, and Callen et al. [7] study uncertain findings in radiology reports. These correspond to the omission, substitution, and ambiguity noises we use. Our patient noises also align with recent taxonomies of difficult patient behaviors [29, 31].

3.2 Reward Design

Our reward function R is the total episode reward containing three parts: diagnosis reward, tool call matching reward, and exam cost reward. The **diagnosis reward** R_{dx} measures the agreement between the predicted diagnosis and the ground truth diagnosis. Since a single case may involve multiple conditions (e.g., "arm fracture with mild infection") and equivalent conditions can be expressed differently (e.g., "influenza" and "flu"), we use an LLM judge (`gpt-4.1-mini`) to extract the set of conditions from both the prediction and the ground truth, match semantically equivalent conditions, and compute a Jaccard score:

$$R_{\text{dx}} = \frac{|\mathcal{G} \cap \mathcal{P}|}{|\mathcal{G} \cup \mathcal{P}|} \tag{1}$$

where $|\mathcal{G}|$ and $|\mathcal{P}|$ denote the number of conditions identified by the judge in the ground truth and prediction, respectively, and $|\mathcal{G} \cap \mathcal{P}|$ is the number of matched conditions. The prompt templates are shown in Appendix B. We verify the accuracy of the LLM judge in Appendix C.

The **tool call reward** R_{tool} considers the overlap between the tool calls of the predicted trajectory and the ground truth trajectory. We follow ToolRL[36] and use a three-level matching reward:

$$R_{\text{tool}} = \frac{J_{\text{name}} + \sum_{i=1}^{|\mathcal{T}_\mathcal{G}|} \left(J_{\text{param}}^{(i)} + V^{(i)} \right)}{1 + \sum_{i=1}^{|\mathcal{T}_\mathcal{G}|} \left(1 + |\theta_i^{\mathcal{T}_\mathcal{G}}| \right)}. \tag{2}$$

where $J_{\text{name}} = \frac{\sum_x \min(c_{\mathcal{T}_\mathcal{P}}(x), c_{\mathcal{T}_\mathcal{G}}(x))}{\sum_x \max(c_{\mathcal{T}_\mathcal{P}}(x), c_{\mathcal{T}_\mathcal{G}}(x))}$ is the multiset Jaccard over tool names ($c_{\mathcal{T}_\mathcal{P}}, c_{\mathcal{T}_\mathcal{G}}$ are name counts in predicted/ground-truth), $J_{\text{param}}^{(i)}$ is the set Jaccard over parameter names for the i-th greedy-matched tool pair, $V^{(i)} = \sum_{k \in \theta_i^{\mathcal{T}_\mathcal{G}}} \mathbb{1}[\theta_i^{\mathcal{T}_\mathcal{G}}(k) = \theta_i^{\mathcal{T}_\mathcal{P}}(k)]$ is the number of exact parameter value matches, and $|\theta_i^{\mathcal{T}_\mathcal{G}}|$ is the number of parameters in the i-th ground-truth tool. We modify the ToolRL reward function by replacing the set Jaccard score in J_{name} with a multiset (count-aware) Jaccard, which penalizes the agent for calling the same tool multiple times unnecessarily.

The **exam cost reward** aims to prevent the agent from calling unnecessarily costly exams. We consider two aspects: financial cost of the exam, and patient discomfort induced by the exam. For each aspect, we classify the cost into three tiers on a three-point ordinal scale (Low=1, Medium=2, High=3). For financial cost, we follow CMS Physician Fee Schedule RVU values (2026) [9] for procedures and imaging, and CMS Clinical Laboratory Fee Schedule rates (2025Q4) [8] for laboratory tests. For patient discomfort, we follow [48] and distinguish procedures by varying degrees of penetration. Specifically, non-invasive exams (e.g., external physical exam, X-ray) are tier 1, minimally invasive exams involving superficial penetration (e.g., venipuncture, swabs, standard endoscopy) are tier 2, and invasive procedures (e.g., biopsy, catheterization, surgical exploration) are tier 3. Lab tests inherit the discomfort tier of their collection method. We sum both types of costs for all unnecessary exams and divide by 6, which is the maximum possible cost value for a single exam.

Finally, we combine the three reward terms by calculating a weighted sum:

$$R(d, a_{1:T}) = R_{\text{dx}} + w_{\text{tool}} R_{\text{tool}} - w_{\text{cost}} R_{\text{cost}}. \tag{3}$$

where w_{tool} and w_{cost} are hyperparameters.

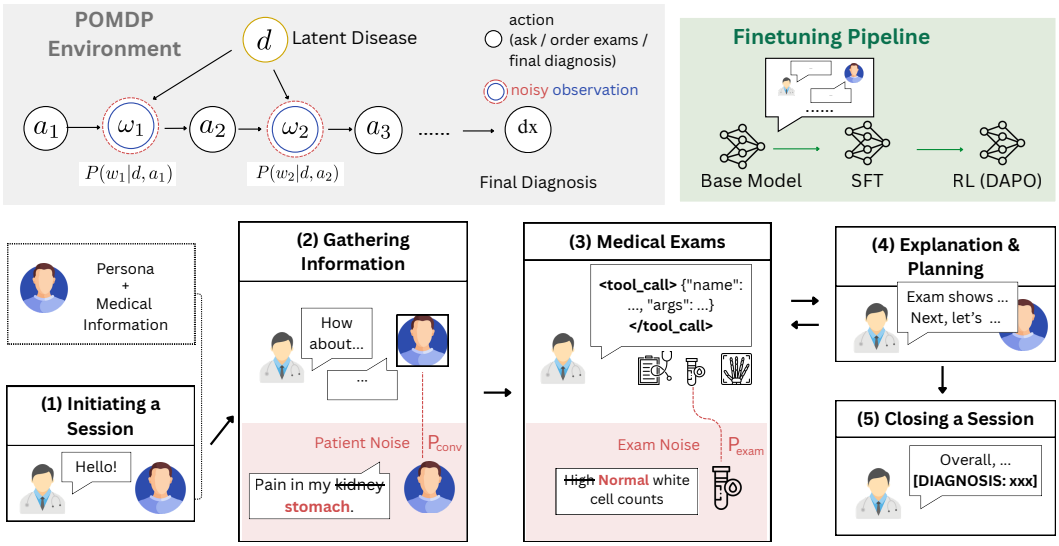

Figure 2: **Method overview.** Top left: The interactive diagnosis POMDP, with latent disease d generating noisy observations ω_t from agent actions a_t. Top right: Two-stage finetuning (SFT then DAPO RL). Bottom: Doctor–patient conversations follow the five-stage Calgary–Cambridge model, with patient and exam noise injected at rates p_{conv} and p_{exam}.

4 Dataset and Environment Construction

Training the agent against the POMDP defined in Section 3 requires two ingredients: a corpus of doctor–patient conversations for supervised fine-tuning, and an interactive environment for reinforcement learning. We describe both in this section.

4.1 Data Sources

DDxPlus. DDxPlus [13] is a large-scale synthetic differential diagnosis dataset of roughly 1.3M patients, covering 49 pathologies and 110 symptoms. DDxPlus is designed to reproduce authentic joint distributions over demographics, pathologies, and symptoms in the U.S. rather than a uniform or adversarial mix. This makes DDxPlus well-suited for optimizing an agent towards addressing "typical" patients, reflecting realistic base rates. However, its 49-pathology label space is narrow by design and under-represents rarer conditions.

PMC-Patients-v2. PMC-Patients-v2 [52] is derived from real clinical case reports published in PubMed Central, comprising approximately 250,294 patient records. Because the editorial threshold for publishing a case report favors unusual, diagnostically challenging, or instructive presentations, PMC-Patients-v2 is naturally enriched for rare diseases, atypical manifestations of common diseases, and multi-condition cases, complementing our dataset with long-tail cases.

AgentClinic-MedQA. AgentClinic [42] is a benchmark containing OSCE-style structured patient profiles, including medical history, symptoms, and test results of the patients. We use the MedQA subset of the benchmark since other subsets contain multimodal entries. AgentClinic-MedQA contains 201 cases grounded in the USMLE questions from the MedQA dataset. These cases are used as our Out-Of-Distribution (OOD) test set and held out from the training set.

4.2 SFT Conversation Generation

We now describe our pipeline to generate conversations for the SFT stage of our training pipeline. To begin with, we subsample DDxPlus and PMC-Patients-v2 to form a training set. We sample a total of 5,000 samples for SFT, 20,000 for RL, and 2,000 for test set, each evenly distributed between DDxPlus and PMC-Patients-v2. From each case, we construct a data point containing patient

demographics, medical history, self-reported symptoms, exams conducted, exam results, and the final diagnosis.

We generate conversations following the five-stage Calgary–Cambridge model [24] for medical interviews, using two LLMs to simulate the doctor and patient. As shown in Figure 2, each conversation can be split into five stages: initiating a session, gathering information, medical exams, explanation and planning, and closing a session. The doctor agent greets the patient and gathers self-reported symptoms, then enters a loop of ordering exams via tool calls and explaining results until it has enough evidence to commit to a diagnosis, summarize the session, and close. A stage-specific prompt is provided to each LLM at every turn (Appendix B).

In order to better reflect patient behavior in the real world, we randomly sample one patient persona for each conversation. We sample from a total of 36 patient personas proposed in PatientSim [25], excluding the "highly confused" persona where the patient is unable to communicate. The realism of these personas has been verified through expert evaluation in PatientSim, and we directly apply their taxonomy in our environment. In Appendix D.2, we show that the personas produce the intended differences in patient communication but do not significantly affect diagnostic performance. Therefore, we use them to increase the diversity of our patient simulator. Additionally, after generating the conversations, we apply the patient and exam noises described in Section 3.1.

4.3 RL Environment Construction

We instantiate the POMDP defined in Section 3. Each patient profile from Section 4.1 defines one episode instance with: (i) ground-truth disease d; (ii) demographics, medical history, and self-reported symptoms; (iii) an exam map $\mathcal{M}$ storing canonical findings for each recorded exam; and (iv) an available tool set $\mathcal{A}_{\text{exam}}$, consisting of ground-truth exams plus randomized distractors sampled from a predefined taxonomy.

The trained model is initialized with a system prompt specifying demographics, medical history, OpenAI-style schemas of $\mathcal{A}_{\text{exam}}$, and instructions for the conversation. Detailed prompt templates are shown in Appendix B.1. The trained model starts the conversation by greeting the patient as instructed in the system prompt. At each turn, the trained model emits one of three action types:

- a_{ask}: a free-form question forwarded to a patient simulator (an LLM conditioned on the case profile and a sampled persona same as in Section 4.2) that returns an in-character reply.
- a_{exam}: a `<tool_call>` block parsed against $\mathcal{M}$. Exact matches return canonical findings; available but non-ground-truth tools return "No significant findings"; tools outside $\mathcal{A}_{\text{exam}}$ return "This exam is not available".
- a_{dx}: a free-form text ending in `[DIAGNOSIS: ...]`, which terminates the episode and triggers reward computation (Equation 3).

Additionally, for each episode, we randomly sample patient and exam noises with probability p_{conv} and p_{exam} as described in Section 3.1. When a patient round is sampled for noise injection, we append a noise-specific prompt to the patient's system prompt. For exam noises, we use a deterministic rule to post-process exam findings. More details are described in Appendix B.2.

5 MedExAgent Training and Evaluation

5.1 Finetuning Pipeline

We use the synthesized dataset described in Section 4.2 to finetune an LLM-based agent. We use Meditron3-8B [40] as our base model since it's the state-of-the-art medical LLM with around 8B parameters, which allows for easier deployment in medical environments with lower compute resources. We use a two-stage pipeline with an SFT stage followed by a RL stage. In the SFT stage, we finetune the model with the generated conversations. In the RL stage, we apply DAPO [50] to train the model using the extracted patient information described in Section 4.2. For detailed hyperparameter settings, please refer to Appendix A.1.

5.2 Evaluation Pipeline

Test sets. We evaluate our trained model and baseline models on three benchmarks: the test splits of DDxPlus and PMC-Patients-v2, and AgentClinic-MedQA as an out-of-distribution (OOD) benchmark. For each evaluation set, we provide the model with the same system prompt used during SFT conversation generation, including instructions for the full diagnostic procedure and the list of available medical exams. We use 1000 samples each for DDxPlus and PMC-Patients-v2, and all (201) samples for AgentClinic-MedQA.

Baseline models. We compare MedExAgent against a total of seven baseline models. This include four general-purpose LLMs: Qwen3-8B, Qwen3-14B, Qwen3-32B [49], Llama-3-8B-Instruct [15], and three medical LLMs: Aloe-Beta-70B [14], MedGemma-27B-text-it [43] and HuatuoGPT-o1-70B [10]. Note that the base model Meditron-3-8B is not included since it's not an instruction-tuned model and can hardly follow evaluation instructions. We use default hyperparameters for baseline models, and all the thinking models are evaluated with thinking mode turned on. For each model, we adjust the tool-calling instruction and parsing logic to accept the supported tool-call format.

Evaluation environment. For evaluation, we use the same system prompts and action resolving logic as in RL (Section 4). For patient simulator, during evaluation we use `gpt-4.1-mini`, whereas `gpt-4o-mini` is used for SFT data generation and `Qwen3-30B-A3B` for RL rollouts. This separation prevents overfitting to a specific simulator style. We evaluate all models on the original (unperturbed) test sets and we analyze the effect of noise injection separately via training-time ablations in Section 6.2. To ensure a fair comparison with baseline models, noise is not included in the test sets. We evaluate the robustness of our model under test-time noise separately in Section 6.3.

Evaluation metrics. We employ an LLM-judge (`gpt-4.1-mini`) to evaluate diagnostic accuracy. The judge model first splits ground truth diagnosis and predicted diagnosis each into a set of individual medical conditions. Then, it judges the number of overlaps between the two sets. Given the judge's matched condition counts, we compute Jaccard similarity and strict diagnosis accuracy:

$$\text{Jac} = \frac{|\mathcal{G} \cap \mathcal{P}|}{|\mathcal{G} \cup \mathcal{P}|}, \qquad \text{Acc} = \mathbb{1}[\mathcal{G} \subseteq \mathcal{P}]. \tag{4}$$

where $\mathbb{1}$ is the indicator function, $\mathcal{G}$ and $\mathcal{P}$ denote the sets of ground-truth and predicted conditions, respectively. The Jaccard score naturally handles multi-label diagnoses and reduces to binary scoring for single-condition cases. Since the diagnosis reward R_{dx} (Equation 1) used during RL also utilizes Jaccard score, there is a possibility of our trained model overfitting to this metric. Therefore, to ensure a fair comparison, we also measure strict diagnosis accuracy (Acc), where the model is required to at least predict all ground truth conditions to be considered correct. Since the same judge model is used for R_{dx} during RL, we further verify our results with three other judge models in Appendix C. We also report embedding cosine similarity between predicted and ground-truth diagnoses:

$$\text{Sim} = \max\left(0, \boldsymbol{\phi}(p)^\top \boldsymbol{\phi}(g)\right) \tag{5}$$

where ϕ is MedEmbed-base-v0.1 [3], the embedding model we use.

6 Results and Discussions

6.1 Performance Comparison

We first measure the performance of all compared models on the three test sets we used. We test the head-to-head significance of MedExAgent-8B against the open-source baselines on the same split via paired percentile bootstrap with $B = 10{,}000$ resamples. Results are shown in Table 1.

As an in-distribution sanity check, MedExAgent-8B significantly outperforms every baseline on DDxPlus and PMC-Patients-v2 ($p < 0.001$ on all metrics). The result of interest, however, is the held-out AgentClinic-MedQA benchmark, which evaluates all models on cases drawn from a distribution unseen during training: MedExAgent-8B is statistically tied with the 70B Aloe-Beta on Sim, Jac, and Acc; it also ties Qwen3-32B on strict accuracy ($p = 0.857$) while outperforming

Table 1: Diagnostic performance across three benchmarks. **Sim** is cosine similarity between embeddings of the predicted and ground-truth diagnoses; **Jac** is the LLM-judge Jaccard similarity (matched concepts divided by the union of predicted and ground-truth concepts); **Acc** is LLM-judge strict accuracy (1 if every ground-truth concept was matched, 0 otherwise). **Bold** marks the best result per column, underline marks the second best. Superscripts denote paired-bootstrap significance of MedExAgent-8B vs. the runner-up on each metric ($B = 10{,}000$; $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$; no marker = not significant at $p < 0.05$).

	DDXPlus			PMC-Patients			AgentClinic (OOD)		
Model	Sim	Jac	Acc	Sim	Jac	Acc	Sim	Jac	Acc
General open-source									
Llama-3.1-8B-Instruct	0.640	0.383	0.605	0.507	0.122	0.157	0.575	0.225	0.298
Qwen3-8B	0.605	0.328	0.460	0.532	0.092	0.120	0.565	0.153	0.219
Qwen3-14B	0.659	0.399	0.538	0.565	0.113	0.143	0.614	0.226	0.284
Qwen3-32B	0.685	0.466	0.626	0.551	0.127	0.157	0.637	0.310	0.368
Medical open-source									
MedGemma-27B-text-it	0.498	0.241	0.325	0.469	0.108	0.123	0.544	0.214	0.249
Aloe-Beta-70B	0.716	0.503	0.670	0.590	0.171	0.219	**0.684**	**0.395**	**0.395**
HuatuoGPT-o1-70B	0.664	0.410	0.470	0.534	0.120	0.124	0.625	0.238	0.249
MedExAgent-8B (Ours)	**0.953*****	**0.937*****	**0.939*****	**0.626*****	**0.345*****	**0.314*****	0.672	0.378	0.378

Table 2: Ablation study on training data, training stages, and reward design. Same metrics as Table 1. All rows are 8B variants from our pipeline; **bold** marks the best value per column, underline marks the second best. Superscripts denote paired-bootstrap significance of MedExAgent-8B (Full) vs. the ablation variant on Sim ($B = 10{,}000$; $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$).

	DDXPlus			PMC-Patients			AgentClinic (OOD)		
Model	Sim	Jac	Acc	Sim	Jac	Acc	Sim	Jac	Acc
MedExAgent-8B	0.953	0.937	0.939	**0.626**	**0.345**	**0.314**	**0.672**	**0.378**	**0.378**
Training data									
w/o noise injection	**0.966**	0.943	0.944	0.624	0.269	0.254	0.669	0.318	0.318
Training stages									
SFT-only	0.957	**0.946**	**0.946**	0.506***	0.231	0.221	0.589***	0.286	0.289
Reward design									
w/o tool-match reward (w_{tool}=0)	0.962	0.930	0.929	0.593***	0.219	0.203	0.653	0.266	0.269
w/o cost penalty (w_{cost}=0)	0.846***	0.831	0.831	0.160***	0.102	0.094	0.383***	0.261	0.264

the remaining five baselines on Jac. At 8B parameters, MedExAgent matches the strongest medical specialist in the comparison at roughly $9\times$ smaller parameter count. We analyze the failure cases of MedExAgent in Appendix E.

6.2 Ablation Studies

Ablation of reward function terms. The RL reward (Equation 3) decomposes into a diagnosis term, a tool-call term, and an exam-cost penalty. To measure the contribution of each reward term, we train two otherwise identical variants with $w_{\text{tool}} = 0$ and $w_{\text{cost}} = 0$ each. Evaluation results are shown in Table 2. We can observe that removing either w_{tool} or w_{cost} degrades performance. As shown in Table 3, when w_{cost} is removed, even if w_{tool} still places some constraints on excessive tool calls, the model still learns to call much more unnecessary exams, potentially due to a tendency to hack the diagnosis reward by gathering as much information as possible. In contrast, when w_{tool} is removed, the model learns to call much less exams, potentially in response to the cost penalty w_{cost}. This demonstrates the necessity of including both w_{tool} and w_{cost} in our reward function. We also report the aggregate exam cost on PMC-Patients-v2 in Table 3. When w_{cost} is removed, the aggregate cost increases by $2.6\times$, since the model also orders more expensive exams. MedExAgent spends

Table 3: Tool-use efficiency for the reward function ablation variants. **Calls** is the average number of tool calls per case (reported for context). **Call F1** is the harmonic mean of call-weighted precision and set-level recall over the ground-truth exam set; **$ F1** is the harmonic mean of cost-weighted call-precision and cost-weighted recall. **Cost** is the average aggregate cost of exams per case on PMC-Patients-v2 (Section 3.2). **Bold** marks the column maximum.

	DDxPlus			PMC-Patients-v2				AgentClinic (OOD)		
Variant	Calls	Call F1	$ F1	Calls	Call F1	$ F1	Cost	Calls	Call F1	$ F1
MedExAgent-8B	3.75	**0.828**	**0.810**	4.03	**0.701**	**0.705**	12.68	3.05	**0.759**	**0.717**
w/o tool-match reward (w_{tool}=0)	2.38	0.768	0.755	1.99	0.592	0.581	6.44	1.59	0.484	0.461
w/o cost penalty (w_{cost}=0)	5.65	0.741	0.711	9.51	0.648	0.659	32.35	7.68	0.691	0.649

Table 4: Jac on PMC-Patients-v2 under false-negative noise with rate α.

α	0 (clean)	0.035	0.078	0.25	0.50	1.0
MedExAgent-8B	**0.345**	**0.334**	**0.339**	**0.313**	**0.271**	**0.202**
w/o noise injection	0.269	0.273	0.257	0.236	0.225	0.156

$1.11\times$ the cost of the recorded ground-truth exams and achieves the lowest cost per correct diagnosis among all baselines (Appendix D.4).

Ablation of training-data noise. In Section 3.1 we introduced two synthetic noise channels: patient noise and exam noise, which are applied to both SFT and RL training stages. To measure the contribution of noise injection, we train an otherwise identical variant with both noise channels disabled. Evaluation results are shown in Table 2. We can observe that removing noise injection degrades Jac by 0.076 on PMC-Patients-v2 and 0.060 on AgentClinic, while leaving DDxPlus essentially unchanged (± 0.005 Jac, within run-to-run variation).

Ablation of training stages. Our main pipeline finetunes the base model with 5K data entries for SFT followed by 20K entries for RL. To measure the contribution of the RL stage, we train an otherwise identical variant on all 25K entries with SFT only and no RL. Evaluation results are shown in Table 2. We can observe that removing RL degrades performance significantly on AgentClinic and PMC-Patients-v2, while matching the full model on the DDxPlus benchmark. This demonstrates that DDxPlus is an easier dataset that can be saturated by memorization in the SFT stage, while the more difficult in-distribution set PMC-Patients-v2 and the OOD set AgentClinic requires more generalization ability that SFT can not satisfy.

6.3 Robustness to Test-Time Noise

To evaluate the robustness of MedExAgent under test-time noise without leaking the noise types used in training, we use a different type of noise, false-negative, where the findings of each exam are removed with a probability of α. The reported false-negative rate is 7.8% for CT [1] and 17.7% for chest X-ray [6]. Results on PMC-Patients-v2 are shown in Table 4. We can observe that the performance of MedExAgent degrades gradually as α increases, with a Jac drop of at most 0.032 when $\alpha \leq 0.25$. Additionally, the variant trained without noise injection performs worse than MedExAgent at every noise level, suggesting that training with noise helps the model avoid over-relying on individual exam results.

7 Limitations and Future Work

Assumption of patient invariance over time. In our proposed environment and model (Section 3), we assume that the ground truth disease d does not change over the entire episode. Additionally, we also assume that the observation function $O(\omega \mid d, a)$ does not change over time. In reality, this can be false especially for patients with critical conditions, whose symptoms and ground truth disease can progress over time. We leave it for future work to extend towards this direction.

Data and rollout generation with simulated patients. In this work, we synthesized training data and RL rollout conversations with LLM-based simulated patients. Although we applied the persona from PatientSim [25] to improve the authenticity of our simulated patients, there could still be distribution shifts between our training and testing data and real-world patients. Our evaluation also does not include clinician review of the conversations or diagnostic decisions, and relies on an LLM judge, which agrees well with three other judges (Appendix C) but may not fully reflect clinical correctness. Validation with human patients remains necessary before deployment.

Exam-dominated diagnosis. In our environment, removing all useful information from the patient only decreases Jac by 0.013 (Appendix D.1), while in real-world encounters, history taking provides most of the diagnostic information. This might be because the information that a doctor obtains during the conversation (e.g., physical examination) is covered by tool calls in our environment. We leave it for future work to design more informative conversations.

Unrecorded exams. In our environment, exams outside the recorded exam list return "No significant findings", so alternative exam strategies that are also clinically valid cannot be rewarded.

English-only model. Our proposed model is trained and tested with English datasets only. Clinical communications often involve culturally and linguistically specific descriptions, which can be hardly resolved by naively translating inputs and outputs to target languages. We leave it for future work to extend MedExAgent to multilingual clinical settings.

8 Conclusion

In this work, we addressed the gap between existing medical LLM benchmarks and the interactive, uncertain nature of real-world clinical diagnosis. We formalized clinical diagnosis as a POMDP with three action types: questioning the patient, ordering medical exams as tool calls, and issuing a diagnosis. We also introduced a systematic noise model comprising seven patient noise types and three exam noise types, together with diverse patient personas [25]. On top of this environment, we proposed MedExAgent, trained through a two-stage pipeline that combines supervised fine-tuning on Calgary-Cambridge-structured synthetic conversations with DAPO reinforcement learning against a composite reward capturing diagnostic accuracy, tool call quality, and examination cost.

Our experiments on DDxPlus, PMC-Patients-v2, and the out-of-distribution AgentClinic-MedQA benchmark show that MedExAgent-8B achieves diagnostic performance comparable to or exceeding significantly larger open-source baselines, while maintaining cost-efficient examination strategies. Ablation studies further confirm that both the tool call and exam cost reward terms, as well as noise injection during training, contribute meaningfully to the model's effectiveness and robustness.

By releasing our environment, evaluation pipeline, and model checkpoint, we hope to provide a realistic testbed and a strong open-source baseline for future research on interactive clinical diagnosis. Promising directions include extending the formulation to time-varying patient states and multi-visit episodes, calibrating the noise model against empirical clinical data, and validating diagnostic agents with real patients and physicians in the loop.

References

[1] Q. S. Alhammami. Assessment of the diagnostic accuracy of CT as compared to MRI in detecting metastases in patients with colorectal cancer. *Cureus*, 16(8):e66125, 2024.

[2] Association of American Medical Colleges. Physician workforce projections. AAMC Data and Reports, 2024.

[3] A. Balachandran. MedEmbed: Medical-focused embedding models. `https://github.com/abhinand5/MedEmbed`, 2024.

[4] E. P. Balogh. The diagnostic process. In *Improving Diagnosis in Health Care*. National Academies Press, Dec. 2015.

[5] J. A. Baron, C. M. S.-B. Johnson, M. A. Schor, D. Olley, L. Nickel, V. Felix, S. M. Bello, C. Greene, R. Lichenstein, K. Bisordi, R. Koka, C. Bearer, R. Macatangay, N. Ada, K. Ballenger, E. Bliss, L. Colliver, G. Dobbins, H. Heitzig, S. Dixon, P. Semesky, J. Garth, M. Fairchild, P. Gaskin, S. Zahid, R. Castillo, S. Edwards, A. Widjaja, Y. Usui, E. Lynch, M. Clarkson, T. Detwiler, and L. M. Schriml. The DO-KB knowledgebase 2026 update: expanding programmatic and language access. *Nucleic Acids Research*, 54(D1):D1425–D1436, Jan. 2026.

[6] S. H. Bradley, B. S. K. Bhartia, M. E. J. Callister, W. T. Hamilton, N. L. F. Hatton, M. P. T. Kennedy, L. T. A. Mounce, B. Shinkins, P. Wheatstone, and R. D. Neal. Chest x-ray sensitivity and lung cancer outcomes: A retrospective observational study. *British Journal of General Practice*, 71(712):e862–e868, 2021.

[7] A. L. Callen, S. M. Dupont, A. Price, B. Laguna, D. McCoy, B. Do, J. Talbott, M. Kohli, and J. Narvid. Between always and never: Evaluating uncertainty in radiology reports using natural language processing. *Journal of Digital Imaging*, 33(5):1194–1201, 2020.

[8] Centers for Medicare & Medicaid Services. CY 2025 Q4 Clinical Laboratory Fee Schedule public use file (25CLABQ4). CMS Change Request 14211, Oct. 2025. `https://www.cms.gov/medicare/payment/fee-schedules/clinical-laboratory-fee-schedule-clfs/files/25clabq4`.

[9] Centers for Medicare & Medicaid Services. CY 2026 Physician Fee Schedule Relative Value Files (RVU26A), Jan. 2026. `https://www.cms.gov/medicare/payment/fee-schedules/physician/pfs-relative-value-files/rvu26a`.

[10] J. Chen, Z. Cai, K. Ji, X. Wang, W. Liu, R. Wang, J. Hou, and B. Wang. Huatuogpt-o1, towards medical complex reasoning with llms, 2024.

[11] X. Chen, H. Zhou, H. Yi, M. You, W. Liu, L. Wang, Z. Qin, H. Li, X. Zhang, Y. Guo, S. Li, Y. Hu, Q. Xiong, R. Li, L. Fan, Q. Lao, W. Fu, J. Li, and K. Li. Grounding large language models in clinical diagnostics. *Nature Communications*, Mar 2026.

[12] A. V. Eriksen, S. Möller, and J. Ryg. Use of gpt-4 to diagnose complex clinical cases. *NEJM AI*, 1(1), Dec 2023.

[13] A. Fansi Tchango, R. Goel, Z. Wen, J. Martel, and J. Ghosn. Ddxplus: A new dataset for automatic medical diagnosis. *Advances in neural information processing systems*, 35:31306–31318, 2022.

[14] D. Garcia-Gasulla, J. Bayarri-Planas, A. K. Gururajan, E. Lopez-Cuena, A. Tormos, D. Hinjos, P. Bernabeu-Perez, A. Arias-Duart, P. A. Martin-Torres, M. Gonzalez-Mallo, S. Alvarez-Napagao, E. Ayguadé-Parra, and U. Cortés. The aloe family recipe for open and specialized healthcare llms, 2025.

[15] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[16] T. Han, L. C. Adams, K. K. Bressem, F. Busch, S. Nebelung, and D. Truhn. Comparative analysis of multimodal large language model performance on clinical vignette questions. *JAMA*, 331(15):1320, Apr 2024.

[17] A. Javed. Bridging the health care gap in rural populations: Challenges, innovations, and solutions. *The American Journal of Medicine*, 138(5):761–762, May 2025.

[18] S. Ji and L. Carin. Cost-sensitive feature acquisition and classification. *Pattern Recognition*, 40(5):1474–1485, May 2007.

[19] Y. Jiang, K. C. Black, G. Geng, D. Park, J. Zou, A. Y. Ng, and J. H. Chen. Medagentbench: A realistic virtual ehr environment to benchmark medical llm agents, 2025.

[20] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*, 2020.

[21] Y. Kim, C. Park, H. Jeong, Y. S. Chan, X. Xu, D. McDuff, H. Lee, M. Ghassemi, C. Breazeal, and H. W. Park. Mdagents: An adaptive collaboration of llms for medical decision-making, 2024.

[22] Y. W. Kim and L. T. Mansfield. Fool me twice: Delayed diagnoses in radiology with emphasis on perpetuated errors. *American Journal of Roentgenology*, 202(3):465–470, 2014.

[23] S. Kurtz, J. Silverman, J. Benson, and J. Draper. Marrying content and process in clinical method teaching: enhancing the Calgary-Cambridge guides. *Acad. Med.*, 78(8):802–809, Aug. 2003.

[24] S. M. Kurtz and J. D. Silverman. The Calgary–Cambridge referenced observation guides: an aid to defining the curriculum and organizing the teaching in communication training programmes. *Medical Education*, 30(2):83–89, 1996.

[25] D. Kyung, H. Chung, S. Bae, J. Kim, J. H. Sohn, T. Kim, S. K. Kim, and E. Choi. Patientsim: A persona-driven simulator for realistic doctor-patient interactions, 2025.

[26] Y. Labrak, A. Bazoge, E. Morin, P.-A. Gourraud, M. Rouvier, and R. Dufour. Biomistral: A collection of open-source pretrained large language models for medical domains, 2024.

[27] J. M. Lackner, J. Jaccard, L. Keefer, R. Firth, A. M. Carosella, M. Sitrin, and D. Brenner. The accuracy of patient-reported measures for gi symptoms: A comparison of real time and retrospective reports. *Neurogastroenterology & Motility*, 26(12):1802–1811, Nov 2014.

[28] S. S. Li, V. Balachandran, S. Feng, J. Ilgen, E. Pierson, P. W. Koh, and Y. Tsvetkov. Mediq: Question-asking llms for adaptive and reliable clinical reasoning. In *NeurIPS 2024*, June 2024.

[29] Y. Li, X. Jie, W. Ruan, X. Zhang, H. Zhu, Y. Gao, C. Du, and R. Liu. Beyond idealized patients: Evaluating LLMs under challenging patient behaviors in medical consultations. arXiv:2603.29373, 2026.

[30] C. M. Lu. Benefits, risks, & costs of diagnostic tests. In M. A. Papadakis, S. J. McPhee, M. W. Rabow, K. R. McQuaid, and M. Gandhi, editors, *Current Medical Diagnosis & Treatment 2024*. McGraw-Hill Education, 2024.

[31] X. Luo, X. Jiang, and J. Wu. MedDialBench: Benchmarking LLM diagnostic robustness under parametric adversarial patient behaviors. arXiv:2604.06846, 2026.

[32] H. Lyu, T. Xu, D. Brotman, B. Mayer-Blackwell, M. Cooper, M. Daniel, E. C. Wick, V. Saini, S. Brownlee, and M. A. Makary. Overtreatment in the united states. *PLOS ONE*, 12(9), Sep 2017.

[33] D. E. Newman-Toker, N. Nassery, A. C. Schaffer, C. W. Yu-Moe, G. D. Clemens, Z. Wang, Y. Zhu, A. S. Saber Tehrani, M. Fanai, A. Hassoon, and et al. Burden of serious harms from diagnostic error in the usa. *BMJ Quality & Safety*, 33(2):109–120, Jul 2023.

[34] H. Nori, M. Daswani, C. Kelly, S. Lundberg, M. T. Ribeiro, M. Wilson, X. Liu, V. Sounderajah, J. Carlson, M. P. Lungren, B. Gross, P. Hames, M. Suleyman, D. King, and E. Horvitz. Sequential diagnosis with language models, 2025.

[35] A. Pal, L. K. Umapathi, and M. Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In G. Flores, G. H. Chen, T. Pollard, J. C. Ho, and T. Naumann, editors, *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR, 07–08 Apr 2022.

[36] C. Qian, E. C. Acikgoz, Q. He, H. Wang, X. Chen, D. Hakkani-Tür, G. Tur, and H. Ji. Toolrl: Reward is all tool learning needs, 2025.

[37] P. Qiu, C. Wu, J. Liu, Q. Zheng, Y. Liao, H. Wang, Y. Yue, Q. Fan, S. Zhen, J. Wang, J. Gu, Y. Wang, Y. Zhang, and W. Xie. Evolving diagnostic agents in a virtual clinical environment, 2025.

[38] D. A. Redelmeier, J. V. Tu, M. J. Schull, L. E. Ferris, and J. E. Hux. Problems for clinical judgement: 2. obtaining a reliable past medical history, Mar 2001.

[39] K. Saab, J. Freyberg, C. Park, T. Strother, Y. Cheng, W.-H. Weng, D. G. T. Barrett, D. Stutz, N. Tomasev, A. Palepu, V. Liévin, Y. Sharma, R. Ruparel, A. Ahmed, E. Vedadi, K. Kanada, C. Hughes, Y. Liu, G. Brown, Y. Gao, S. Li, S. S. Mahdavi, J. Manyika, K. Chou, Y. Matias, A. Hassidim, D. R. Webster, P. Kohli, S. M. A. Eslami, J. Barral, A. Rodman, V. Natarajan, M. Schaekermann, T. Tu, A. Karthikesalingam, and R. Tanno. Advancing conversational diagnostic ai with multimodal reasoning, 2025.

[40] A. Sallinen, A.-J. Solergibert, M. Zhang, G. B. Boyé, M. Dupont-Roc, X. Theimer-Lienhard, E. Boisson, B. Bernath, H. Hadhri, A. Tran, T. Rabbani, T. Brokowski, M. M. D. W. Group, T. G. J. Rudner, and M.-A. Hartley. Llama-3-meditron: An open-weight suite of medical LLMs based on llama-3.1. In *Workshop on Large Language Models and Generative AI for Health at AAAI 2025*, 2025.

[41] G. D. Schiff, O. Hasan, S. Kim, R. Abrams, K. Cosby, B. L. Lambert, A. S. Elstein, S. Hasler, M. L. Kabongo, N. Krosnjar, R. Odwazny, M. F. Wisniewski, and R. A. McNutt. Diagnostic error in medicine: Analysis of 583 physician-reported errors. *Archives of Internal Medicine*, 169(20):1881–1887, 2009.

[42] S. Schmidgall, R. Ziaei, C. Harris, E. Reis, J. Jopling, and M. Moor. Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments, 2024.

[43] A. Sellergren, S. Kazemzadeh, T. Jaroensri, A. Kiraly, M. Traverse, T. Kohlberger, S. Xu, F. Jamil, C. Hughes, C. Lau, et al. Medgemma technical report. *arXiv preprint arXiv:2507.05201*, 2025.

[44] J. Shreffler. Diagnostic testing accuracy: Sensitivity, specificity, predictive values and likelihood ratios, Mar 2023.

[45] R. T. Sutton, D. Pincock, D. C. Baumgart, D. C. Sadowski, R. N. Fedorak, and K. I. Kroeker. An overview of clinical decision support systems: Benefits, risks, and strategies for success. *npj Digital Medicine*, 3(1), Feb 2020.

[46] X. Tang, A. Zou, Z. Zhang, Z. Li, Y. Zhao, X. Zhang, A. Cohan, and M. Gerstein. Medagents: Large language models as collaborators for zero-shot medical reasoning, 2024.

[47] T. Tu, A. Palepu, M. Schaekermann, K. Saab, J. Freyberg, R. Tanno, A. Wang, B. Li, M. Amin, N. Tomasev, S. Azizi, K. Singhal, Y. Cheng, L. Hou, A. Webson, K. Kulkarni, S. S. Mahdavi, C. Semturs, J. Gottweis, J. Barral, K. Chou, G. S. Corrado, Y. Matias, A. Karthikesalingam, and V. Natarajan. Towards conversational diagnostic ai, 2024.

[48] M. S. Whiteley, S. E. Davey, and G. M. Placzek. The access and invasiveness-based classification of medical procedures to clarify non-invasive from different forms of minimally invasive and open surgery. *Journal of Minimal Access Surgery*, 20(3):301–310, July 2024.

[49] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[50] Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, W. Dai, T. Fan, G. Liu, L. Liu, X. Liu, H. Lin, Z. Lin, B. Ma, G. Sheng, Y. Tong, C. Zhang, M. Zhang, W. Zhang, H. Zhu, J. Zhu, J. Chen, J. Chen, C. Wang, H. Yu, Y. Song, X. Wei, H. Zhou, J. Liu, W.-Y. Ma, Y.-Q. Zhang, L. Yan, M. Qiao, Y. Wu, and M. Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025.

[51] E. Zelikman, Y. Wu, J. Mu, and N. D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[52] Z. Zhao, Q. Jin, F. Chen, T. Peng, and S. Yu. A large-scale dataset of patient summaries for retrieval-based clinical decision support systems. *Scientific Data*, 10(1):909, 2023.

[53] Y. Zhu, Z. Huang, L. Mu, Y. Huang, W. Nie, J. Liu, S. Zhang, P. Liu, and X. Zhang. Diagnosis-arena: Benchmarking diagnostic reasoning for large language models, 2025.

[54] M. N. Zozus, A. Walden, and C. F. Pieper. *Comparing the Accuracy of Health Record Data and Self-Reported Data*. Patient-Centered Outcomes Research Institute (PCORI), Washington, DC, Mar. 2023. Available from NCBI Bookshelf.

A Additional Experiment Details

A.1 Hyperparameters

In this section, we list the detailed hyperparameters used to finetune and evaluate our model in Table 5. For reward weights, we conducted a brief search over $(w_{\text{tool}}, w_{\text{cost}}) \in \{(1.0, 0.1), (0.5, 0.1), (0.1, 0.1), (0.5, 0.3)\}$ by running RL for 10 steps per configuration and selecting the configuration with the highest R_{dx} (the unweighted diagnosis reward). The configuration $(0.5, 0.1)$ was selected.

The DAPO-specific hyperparameters in Table 5 (group size, clip ratios, KL/entropy coefficients) are set to the defaults from [50]. SFT hyperparameters follow standard values for instruction tuning at this scale. Batch sizes were set to the largest values that fit on our hardware.

Noise injection probabilities $(p_{\text{conv}}, p_{\text{exam}}) = (0.3, 0.1)$ were chosen heuristically without tuning; the contribution of noise injection at these levels is validated by the ablation in Table 2.

A.2 Compute Resources

The compute resource used to train MedExAgent is listed in Table 6. With our settings and implementation, the SFT stage takes $\approx$ 4 hours and RL takes $\approx$ 5.5 days.

Table 5: Hyperparameters for SFT, RL, and evaluation.

Hyperparameter		**Value**
Supervised Fine-Tuning		
Base model		OpenMeditron/Meditron3-8B
Epochs		1
Per-device batch size		2
Grad. accumulation		2
Learning rate		2×10^{-5}
LR schedule		cosine, 5% warmup
Optimizer		AdamW
Reinforcement Learning (DAPO)		
Group size G		16
Rollout temperature		1.0
Clip ratio (low, high)		(0.2, 0.3)
KL coefficient		0.0
Entropy coefficient		0.002
Learning rate		1×10^{-6}
Global batch size		64
Rollout noise	patient / exam	0.3 / 0.1
Reward weights	$(w_{\text{tool}}, w_{\text{cost}})$	$(0.5,\ 0.1)$
Diagnosis verifier model		`gpt-4.1-mini`
Diagnosis verifier temperature		0.7
Evaluation		
Patient simulator model		`gpt-4.1-mini`
Patient simulator temperature		0.7

Table 6: Compute resource used to train MedExAgent.

Resource	**Value**
GPU model	Nvidia RTX PRO 6000 Blackwell Max-Q
Number of GPUs	4
VRAM per GPU	96 GB
CPU	AMD EPYC 9534, 128 cores
System memory	1.5 TiB
CUDA / PyTorch	12.8 / 2.9

B Prompt Templates

B.1 System Prompts

The system prompt templates applied in this paper are shown as follows. Patient prompt template is identical for both conversation generation and rollout generation during RL. The hidden canonical diagnosis in the doctor system prompt is hidden during rollout generation, and only kept for data generation.

Data generation system prompt (Doctor)

```
You are an experienced physician conducting a medical consultation. Your goal
is to gather information from the patient, order appropriate examinations,
interpret results, and arrive at a diagnosis.
PATIENT DEMOGRAPHICS: <demographics>
PATIENT'S MEDICAL HISTORY: <medical_history>
(Only shown during data generation) HIDDEN CANONICAL DIAGNOSIS
(for training/consistency checking only; do NOT reveal to patient):
<hidden_canonical_diagnosis>
AVAILABLE EXAMINATIONS: <tools_block>
INSTRUCTIONS:
- Only output what you would say directly to the patient. Do NOT include your
internal thoughts / reasoning / plan.
- Pace gently, do NOT overwhelm the patient with too many questions at once.
- Ask at most 1-2 short questions per turn.
- Order exams one at a time, do NOT order multiple exams in one turn.
- The hidden canonical diagnosis is ground truth for this case. Use it to guide
your reasoning and keep the conversation clinically consistent, but do NOT
reveal it to the patient before you have enough information and appropriate exam
results.
- When you give the final diagnosis, it must match the hidden canonical
diagnosis exactly.
- If no hidden canonical diagnosis is provided, reason from the conversation and
exam results alone.
- You MUST end with [DIAGNOSIS: ...] to conclude the consultation.
```

Data generation system prompt (Patient)

```
You are a patient visiting a doctor. You should role-play as a real patient
would behave during a medical consultation.
YOUR DEMOGRAPHICS: <demographics>
YOUR MEDICAL HISTORY: <medical_history>
YOUR CURRENT SYMPTOMS: <self_reported_symptoms>
SELECTED PATIENT PERSONA:
Personality: [<p_label>] <p_instruction>
Language Proficiency: [<lang_label>] <lang_instruction>
Medical History Recall: [<recall_label>] <recall_instruction>
INSTRUCTIONS:
- Follow the selected persona naturally, but do NOT mention the persona labels
to the doctor.
- Describe your symptoms using everyday, non-medical language when possible.
- Pace gently. Keep each reply brief and natural, usually 1-3 spoken sentences.
- Do NOT volunteer all your symptoms or history at once. Share information
gradually as the doctor asks.
- Even if your selected personality is Verbose, do NOT dominate the conversation
or dump long monologues unless the doctor explicitly asks for more detail.
- Only output spoken words. Do NOT include thoughts, reasoning, narration,
stage directions, bracketed text, labels, or body language.
- Stay faithful to your true demographics and current symptoms. For past
medical history, follow your selected recall setting exactly.
```

B.2 Noise Injection

As described in Section 3.1, we inject seven types of patient noise and three types of exam noise during conversation generation for SFT and rollout generation for RL. Detailed definitions and examples for each noise type are shown in Table 7. For each patient / exam, we first sample whether to introduce noise with designated noise levels (which are hyperparameters as shown in Table 5) as probability. If a patient is selected, we first decide 1-3 noise types according to eligibility, and assign each noise type to exactly one patient turn. If an exam is selected, we randomly assign one eligible noise type.

In terms of noise injection method, for patient noise, we append a hint prompt after the patient prompt shown in Table 8. For exam noise, we use a rule-based logic to postprocess the original exam results, as detailed in Table 9.

B.3 LLM-Judge Prompt and Implementation

We apply LLM-Judge to calculate both the diagnosis reward R_{dx} during RL (Equation 1) and the Jaccard score during evaluation (Equation 4). The prompt is shown below.

LLM Judge prompt

```
You are a medical expert evaluating a diagnosis prediction.
Ground truth diagnosis: {ground_truth} Predicted diagnosis: {predicted}
Instructions:
1. Identify the individual medical conditions in the ground truth. Note that a
comma may be part of a single condition name (e.g. "seminoma, classic type" is
ONE condition, "Follicular lymphoma, grade 2" is ONE condition). Semicolons or
"and" typically separate distinct conditions.
2. Identify the individual medical conditions in the prediction, using the same
logic.
3. For each ground truth condition, check if any predicted condition refers
to the same disease. Consider synonyms (e.g. "heart attack" = "myocardial
infarction"), abbreviations, and minor wording differences.
4. Count how many ground truth conditions have a match in the predictions.
You MUST respond in exactly this format (numbers only):
gt_count: <number of ground truth conditions>
pred_count: <number of predicted conditions>
matched: <number of matched conditions>
```

Note that in this implementation, instead of directly asking for $|\mathcal{G} \cap \mathcal{P}|$ and $|\mathcal{G} \cup \mathcal{P}|$, we ask the LLM judge for $|\mathcal{G}|$, $|\mathcal{P}|$, and $|\mathcal{G} \cap \mathcal{P}|$, and calculate the Jaccard score and Accuracy with

$$\text{Jac} = \frac{|\mathcal{G} \cap \mathcal{P}|}{|\mathcal{G}| + |\mathcal{P}| - |\mathcal{G} \cap \mathcal{P}|} \qquad \text{Acc} = \mathbb{1}(|\mathcal{G}| = |\mathcal{G} \cap \mathcal{P}|) \tag{6}$$

which is equivalent to Equation 4. In this way we are able to calculate both metrics in one call to the LLM Judge.

C Verification of LLM-judge accuracy

The diagnosis reward R_{dx} in Eq. (1) relies on an LLM judge (`gpt-4.1-mini`) to extract condition sets from the predicted and ground-truth diagnoses, semantically match equivalent conditions, and return the integer counts $|\mathcal{G}|$, $|\mathcal{P}|$, $|\mathcal{G} \cap \mathcal{P}|$ from which the Jaccard score is computed (more details in Appendix B.3). In this section we measure the judge's accuracy directly on conditions drawn from our test sets.

We sample 99 condition pairs from the Disease Ontology [5], restricted to DOID entries that match a diagnosis appearing in DDxPlus, PMC-Patients-v2, or AgentClinic-MedQA. From the matched terms we deterministically draw three uniform buckets with 33 samples each:

- **Synonym**: Ground truth is test-set diagnosis; prediction is a DOID `EXACT` synonym. Expected R_{dx}=1.

Table 7: Noise types applied to the observation function $O(\omega \mid d, a)$. Patient noises (white) affect conversational observations $\omega \in \Omega_{\text{conv}}$ and are applied with probability p_{conv}. Exam noises (gray) affect exam observations $\omega \in \Omega_{\text{exam}}$ and are applied with probability p_{exam}.

Category	Noise Type	Description	Example
Patient	Body Part Swap	Reports symptom location as an adjacent body part.	"stomach pain" → "liver pain"
	Symptom Confusion	Confuses a symptom descriptor with a related but incorrect term.	"burning sensation" → "tingling sensation"
	Severity Change	Misreports the intensity of symptoms, typically downplaying.	"severe headache" → "moderate headache"
	Temporal Change	Misreports symptom duration or onset time.	"3 weeks" → "1 week"
	Omission	Fails to mention one symptom from a multi-symptom presentation.	Omits "fatigue" when also presenting with fever and cough.
	Self-Diagnosis	Volunteers an incorrect self-diagnosis based on lay sources.	"I looked it up online and it seems like allergies."
	Vague Answer	Gives evasive, non-specific responses to direct questions.	"I'm not really sure about that."
Exam	Body Part Swap	Replaces an anatomical location in findings with an adjacent body part.	"Inflammation in the *stomach* lining" → "Inflammation in the *liver* lining"
	Omission	Drops one clause from multi-part findings.	"Elevated WBC; low hemoglobin; normal platelets" → "Elevated WBC; normal platelets"
	Ambiguity	Wraps findings in equivocal language indicating limited certainty.	"Nodule in right lung" → "Findings are equivocal — nodule in right lung. Cannot definitively rule out alternative interpretation."

- **Distractor**: Ground truth is test-set diagnosis; prediction is a DOID sibling under the `is_a` graph, with three coupling filters – substring-disjoint names, different acuity prefixes ({acute, chronic, recurrent, severe, mild}), different trailing integers. Expected R_{dx}=0.
- **Multi-condition partial**: Ground truth is "A and B" for two test-set-grounded terms with no shared parent and no `is_a` relation; prediction is A alone. Expected $|\mathcal{G}|$=2, $|\mathcal{P}|$=1, $|\mathcal{G} \cap \mathcal{P}|$=1, R_{dx}=$\frac{1}{2}$.

Each pair is scored independently by `gpt-4.1-mini` at temperature 0 using the prompt template of Appendix B. We parse the response $\{\texttt{gt_count}, \texttt{pred_count}, \texttt{matched}\}$ and compute $\widehat{R}_{\text{dx}}$ as in Equation 1.

The judge is exact on multi-condition counting and partial matching. Synonym sensitivity is $30/33 = 91\%$ and distractor specificity is $31/33 = 94\%$; the remaining failures are mostly obscure historical eponyms (e.g. "Rose's tamponade"; "insular sclerosis").

Agreement with other judges. To further verify the judge on the actual evaluation workload, we collect 2,992 (ground truth, prediction) pairs from our evaluation runs, including MedExAgent and the no-noise ablation on PMC-Patients-v2 and AgentClinic-MedQA, and Aloe-Beta-8B and Qwen3-32B on PMC-Patients-v2. We re-score each pair with three recent models using the same judge prompt. Results are shown in Table 11. We can observe that `gpt-4.1-mini` has high agreement with all three models.

Evaluation with other judges. Since `gpt-4.1-mini` is used for both the RL reward and evaluation, we further evaluate selected models with the three judges above. Results are shown in Table 12. We can observe that the four judges differ mainly in how permissive they are. The fraction of pairs

Table 8: Hint templates used for injecting noise into patient responses. Prompts are emitted verbatim; placeholders are resolved as described.

Category	Noise Type	Prompt (verbatim) and Placeholder Resolution
Patient	Body Part Swap	`When answering, refer to your {original} symptom as being in your {swapped} area instead.` *{original}*: first matched body-part keyword in symptoms; *{swapped}*: sampled adjacent body part.
	Symptom Confusion	`When describing the sensation, say '{confused}' instead of '{original}'.` *{original}*: first matched descriptor; *{confused}*: sampled confusion term.
	Severity Change	`When answering: {severity_instruction}.` Either: *report the pain/symptom as X instead of Y* (if severity word matched), or *downplay the severity of your symptoms slightly* (fallback).
	Temporal Change	`When answering: {temporal_instruction}.` Either: *say N units instead of M units* (if duration detected and perturbed), or *be vague about when symptoms started — say 'a while ago' or 'recently'* (fallback).
	Omission	`In this response, do NOT mention {omitted_symptom}. Talk only about your other symptoms or details.` *{omitted_symptom}*: sampled from symptom list (requires $\geq$2 symptoms).
	Self-Diagnosis	`Work the following into your response naturally: '{phrase}' Then continue answering the doctor's question.` *{phrase}*: template filled with a sampled common condition (e.g., cold, flu, allergies).
	Vague Answer	`For this response, be vague and evasive. Say something like '{vague}' Do not provide specific details for this question.` *{vague}*: sampled from a fixed pool of non-committal responses.

judged as a match is 0.243 for `gpt-5.5`, 0.257 for `claude-sonnet-5`, 0.345 for `gpt-4.1-mini`, and 0.365 for `qwen3.6-35b-a3b`, so our training judge is not the most permissive one. MedExAgent outperforms Aloe-Beta-8B and Qwen3-32B on PMC-Patients-v2 under all four judges. On AgentClinic, the variant without noise injection scores higher than MedExAgent under `gpt-5.5` and `claude-sonnet-5`.

D Additional Analyses

D.1 Ablation of Conversation and Exam Stages

To analyze how much the conversation and the exams contribute to the diagnosis in our environment, we ablate each of them by removing all useful information from that stage. Results are shown in Table 13. We can observe that removing the useful information from the patient only decreases Jac by 0.013, while making every ground-truth exam return a negative result decreases Jac by 0.138. Since the exam results are taken from the source cases rather than generated by LLMs, this shows that the effect of patient simulation on our results is limited.

D.2 Effect of Patient Personas

We first verify that the persona changes the behavior of the simulated patient as intended. Across 10,026 conversations, the average number of words per patient turn ranges from 78.8 (verbose) to 24.3 (plain), and the disclosure rate of medical history is 0.349, 0.243, and 0.130 for high, low, and

Table 9: Exam noise transformations applied to tool-returned findings. Unlike patient noise, these are not prompts; transformations are applied directly to the raw findings string before delivery.

Category	Noise Type	Transformation (verbatim) and Resolution
Exam	Body Part Swap	Replace the first body-part token with a randomly sampled adjacent term from a predefined body part list.
	Omission	If multiple findings exist, drop a random non-empty one.
	Ambiguity	Wrap the original findings using one of the following templates (uniformly sampled): • `Findings are equivocal - {original}. Cannot definitively rule out alternative interpretation.` • `Results show {original}. However, findings are not entirely clear and may warrant further evaluation.` • `{original}. Note: image quality/sample quality limits definitive interpretation.`

Table 10: Per-bucket accuracy on the 99-pair probe (`gpt-4.1-mini`, temperature 0). Accuracy is the ratio of the judge giving the correct answer. MAE is the mean absolute error between $\widehat{R}_{\text{dx}}$ and the expected score.

Bucket	n	**Accuracy**	**MAE**
Synonym (matched as equivalent)	33	91 %	0.197
Distractor (correctly rejected)	33	94 %	0.152
Multi-condition partial	33	100 %	0.000

Table 11: Agreement between `gpt-4.1-mini` and three other judge models on 2,992 evaluation pairs.

`gpt-4.1-mini` vs.	Agreement	Cohen's κ
`gpt-5.5`	93.98%	0.834
`claude-sonnet-5`	94.22%	0.844
`qwen3.6-35b-a3b`	96.02%	0.899

Table 12: Jac of selected models under four judge models, using the same judge prompt.

System / test set	`gpt-4.1-mini`	`gpt-5.5`	`claude-sonnet-5`	`qwen3.6`
MedExAgent-8B, PMC	0.337	0.248	0.267	0.359
w/o noise injection, PMC	0.253	0.219	0.226	0.259
Aloe-Beta-8B, PMC	0.114	0.110	0.118	0.132
Qwen3-32B, PMC	0.110	0.088	0.131	0.125
MedExAgent-8B, AgentClinic	0.359	0.261	0.281	0.407
w/o noise injection, AgentClinic	0.358	0.309	0.313	0.358

Table 13: Ablation of the conversation and exam stages for MedExAgent-8B on PMC-Patients-v2. Rows below the first show the change from the clean setting.

Condition	Jac	Acc	Sim
Clean	0.339	0.302	0.625
Patient withholds every symptom, location, and timing detail	−0.013	−0.003	−0.002
Every ground-truth exam returns a negative	−0.138	−0.113	−0.045
Both	−0.141	−0.118	−0.055

no recall, respectively. We then remove the persona from the patient prompt and re-run MedExAgent and Qwen3-8B on the same PMC-Patients-v2 cases, keeping all other settings fixed. Results are shown in Table 14. We can observe that neither model has a significant persona effect, and the difference between their effects is also not significant (Jac $p = 0.360$, Acc $p = 0.788$). Accuracy broken down by personality, language proficiency, and history recall also shows no significant difference for MedExAgent or three baselines on the same 1,000 cases (all $p \geq 0.224$). Therefore, the personas produce the intended differences in patient communication, but do not significantly affect the diagnostic difficulty in our environment.

Table 14: Performance with and without patient personas on PMC-Patients-v2, with paired p-values.

Model	n	Jac, on $\rightarrow$ off (Δ, p)	Acc, on $\rightarrow$ off (Δ, p)
MedExAgent-8B	1,000	0.3393 $\rightarrow$ 0.3446 ($+0.0053$, 0.622)	0.3020 $\rightarrow$ 0.3190 ($+0.0170$, 0.130)
Qwen3-8B	990	0.0895 $\rightarrow$ 0.1076 ($+0.0181$, 0.055)	0.1152 $\rightarrow$ 0.1374 ($+0.0222$, 0.066)

D.3 Conversation Generation without Ground Truth Diagnosis

During SFT data generation, the hidden canonical diagnosis is given in the doctor prompt (Appendix B) to make sure that the generated conversation does not diverge from the ground truth diagnosis. Using a known target to generate the path leading to it is a common practice in synthetic supervision. For example, STaR [51] gives the correct answer to the model when generating rationales for failed problems. To test whether the diagnosis is needed, we sample 60 PMC-Patients-v2 cases with a fixed random seed and generate their conversations twice, changing only whether the doctor model sees the diagnosis. Results are shown in Table 15. We can observe that without the diagnosis, 40% of the conversations end with a medically different diagnosis, and only 3.3% match the source diagnosis closely enough to pass the filter in our generation pipeline. Training on these conversations would introduce contradictory supervision.

Table 15: SFT conversation generation with and without the ground truth diagnosis in the doctor prompt, on 60 PMC-Patients-v2 cases.

	With diagnosis	Without diagnosis
Final diagnosis matches exactly	98.3%	3.3%
Final diagnosis correct under LLM judge	100%	60%
Judge Jaccard	1.000	0.560

D.4 Aggregate Exam Cost

We compare the aggregate exam cost of MedExAgent with the baseline models and the recorded ground-truth exams on PMC-Patients-v2. Cost per correct diagnosis is calculated as aggregate cost divided by strict accuracy. Results are shown in Table 16. We can observe that MedExAgent spends approximately the same cost as the recorded ground-truth exams ($1.11\times$), and achieves the lowest cost per correct diagnosis among all baseline models.

E Error Analysis

Failure stage. We first analyze at which stage of the conversation the failures happen. Specifically, we prompt `gpt-5.5`, `claude-sonnet-5`, and `gpt-4.1-mini` to classify which stage resulted in the final error, and report the majority vote. Results are shown in Table 17. For PMC-Patients-v2 and AgentClinic, most errors are due to exam selection, which is the most impactful and difficult step. For DDxPlus, where the cases are relatively easy, final reasoning becomes the main reason for errors.

Failure modes. To analyze the detailed reasons for failures, we sample 100 failure cases across the three datasets and manually read them to form a taxonomy of failure modes. We then prompt `gpt-5.5` to classify all the failure cases into these categories. Results are shown in Table 18. Insufficient examination is the most common failure mode, which is consistent with our stage analysis. We

Table 16: Aggregate exam cost of all models on PMC-Patients-v2.

System	Calls	Agg. cost	× GT workup	Jac	Acc	Cost / correct dx
Recorded ground-truth workup	3.46	11.46	1.00	–	–	–
Llama-3.1-8B-Instruct	6.59	19.15	1.67	0.122	0.157	122.0
MedGemma-27B-text-it	5.84	17.05	1.49	0.109	0.123	138.7
Aloe-Beta-70B	4.60	14.42	1.26	0.171	0.219	65.8
Qwen3-32B	4.66	14.05	1.23	0.127	0.157	89.5
HuatuoGPT-o1-70B	3.97	12.50	1.09	0.120	0.124	100.6
Qwen3-14B	3.14	9.45	0.82	0.113	0.143	66.1
Qwen3-8B	3.07	8.76	0.76	0.092	0.120	73.0
w/o cost penalty (w_{cost}=0)	9.51	32.35	2.82	0.102	0.094	344.1
w/o noise injection	3.44	10.53	0.92	0.269	0.254	41.5
w/o tool-match (w_{tool}=0)	1.99	6.44	0.56	0.219	0.203	31.7
MedExAgent-8B (ours)	4.03	12.68	1.11	0.345	0.314	40.4

Table 17: Primary failure stage of MedExAgent-8B errors (majority vote of three judges).

Primary failure stage	PMC (n=692)	DDxPlus (n=50)	AgentClinic (n=125)
Exam selection	50.7%	26.0%	52.8%
Final reasoning	29.3%	52.0%	28.8%
Exam interpretation	13.3%	6.0%	12.8%
Questioning	5.8%	16.0%	5.6%
Label or evidence inconsistency	0.7%	0.0%	0.0%

further map our taxonomy into the diagnostic process stages in Schiff et al. [41], which studies 583 physician-reported diagnostic errors (Table 19). We can observe that the error distribution of MedExAgent is similar to that of human physicians, where testing is the most common source of errors.

Table 18: Failure modes of MedExAgent-8B classified by `gpt-5.5`.

Failure mode	PMC	DDxPlus	AgentClinic	Overall
Insufficient examination	26.9%	10.7%	39.2%	27.6%
Incidental anchoring	16.9%	30.4%	12.0%	17.1%
Pattern synthesis	10.8%	30.4%	19.2%	13.3%
Evidence misread	15.2%	8.9%	10.4%	14.1%
Negative-evidence mishandling	11.0%	0.0%	6.4%	9.6%
History omission	7.4%	14.3%	8.0%	7.9%
Specificity failure	6.2%	1.8%	4.0%	5.6%
Other	5.6%	3.6%	0.8%	4.8%

Table 19: Errors of MedExAgent-8B and human physicians [41] by diagnostic process stage.

Diagnostic process stage	MedExAgent (n=867)	Physicians [41] (n=583)
Testing: ordering, processing, acting on results	43.9%	44%
Clinician assessment	42.5%	32%
History taking	4.6%	10%
Physical examination	7.4%	10%
Other	1.6%	3%

F Additional Experiment Result Tables

Table 20 reports 95% percentile-bootstrap confidence intervals (B = 10,000) for all models and ablation variants across the three benchmarks.

G Additional Statements

G.1 Broader Impact Statement

MedExAgent is a research prototype of an LLM-based agent for interactive clinical diagnosis. Our goal is to lower the compute barrier for medical agent research and provide an open-source baseline for on-premises deployment in privacy-sensitive research settings. We note that MedExAgent is not intended for clinical use. It is trained on synthetic conversations with LLM-simulated patients, and has not been validated in the real world. To prevent misuse, we gate access to the released checkpoint behind a usage agreement that restricts use to research and prohibits patient-facing deployment, and we include a prototype-only disclaimer in the paper, model card, and repository README.

G.2 Assets Statement

New assets. We release three artifacts: (i) the MedExAgent-8B model checkpoint, (ii) the dataset used to train MedExAgent, including synthesized conversations for SFT and extracted data tables for RL, and (iii) the code used for data generation, model finetuning, and evaluation.

MedExAgent-8B is finetuned from Meditron3-8B, which itself is finetuned from Llama-3.1-8B. Therefore, we release the checkpoint licensed under the Llama 3.1 Community License, with access gated behind a usage agreement that restricts use to research and prohibits patient-facing deployment. Our dataset is partially derived from PMC-Patients-v2 [52], which is licensed under CC BY-NC-SA 4.0. Therefore, we release the dataset licensed under CC BY-NC-SA 4.0. The code is released under the MIT license.

Existing assets. Table 21 lists all existing assets used in this work, along with their licenses. All assets are used in accordance with their original terms of use. For PatientSim, we only use the patient persona published in the paper, which is licensed under CC BY-NC-ND 4.0 per arXiv. For CMS PFS RVU and CMS CLFS values, we do not redistribute raw values in any of our released assets.

G.3 LLM Usage Statement

LLMs are a core component of this work and are used in four distinct roles, each documented in the body of the paper:

- **The trained agent.** MedExAgent itself is an LLM (Meditron3-8B fine-tuned via SFT and DAPO).
- **Synthetic conversation generation for SFT.** Section 4.2 and Appendix B document the use of `gpt-4o-mini` as both the doctor and patient simulator during SFT data generation.
- **RL rollout patient simulation.** Section 4.3 documents the use of `Qwen3-30B-A3B` as the patient simulator during DAPO rollouts.
- **Diagnosis judging.** Section 3.2 and Section 5.2 document the use of `gpt-4.1-mini` as an LLM judge that extracts condition sets and counts matches.

Table 20: 95% confidence intervals across datasets. Each cell reports mean ± half-width of the 95% percentile-bootstrap CI over per-sample scorer outputs ($B = 10{,}000$). Same metrics, bolding, and variants as Tables 1 and 2. **Bold** marks the column-best among non-ablation rows, underline marks the second best; ablation rows are listed for reference.

(a) AgentClinic (OOD, N=201)

Model / Variant	Sim	Jac	Acc
Llama-3.1-8B-Instruct	0.575 ± 0.033	0.225 ± 0.053	0.298 ± 0.065
Qwen3-8B	0.565 ± 0.017	0.153 ± 0.044	0.219 ± 0.057
Qwen3-14B	0.614 ± 0.022	0.226 ± 0.053	0.284 ± 0.062
Qwen3-32B	0.637 ± 0.028	0.310 ± 0.061	0.368 ± 0.067
MedGemma-27B-text-it	0.544 ± 0.038	0.214 ± 0.053	0.249 ± 0.060
Aloe-Beta-70B	$\mathbf{0.684 \pm 0.023}$	$\mathbf{0.395 \pm 0.061}$	$\mathbf{0.395 \pm 0.061}$
HuatuoGPT-o1-70B	0.625 ± 0.025	0.238 ± 0.059	0.249 ± 0.062
MedExAgent-8B (full)	$\underline{0.672 \pm 0.024}$	$\underline{0.378 \pm 0.067}$	$\underline{0.378 \pm 0.067}$
w/o noise injection	0.669 ± 0.024	0.318 ± 0.065	0.318 ± 0.065
SFT-only	0.589 ± 0.038	0.286 ± 0.063	0.289 ± 0.063
w/o tool-match reward	0.653 ± 0.026	0.266 ± 0.061	0.269 ± 0.061
w/o cost penalty	0.383 ± 0.055	0.261 ± 0.062	0.264 ± 0.062

(b) DDxPlus (N≈1000)

Model / Variant	Sim	Jac	Acc
Llama-3.1-8B-Instruct	0.640 ± 0.013	0.383 ± 0.023	0.605 ± 0.030
Qwen3-8B	0.605 ± 0.009	0.328 ± 0.024	0.460 ± 0.030
Qwen3-14B	0.659 ± 0.009	0.399 ± 0.026	0.538 ± 0.032
Qwen3-32B	0.685 ± 0.012	0.466 ± 0.025	0.626 ± 0.030
MedGemma-27B-text-it	0.498 ± 0.019	0.241 ± 0.023	0.325 ± 0.029
Aloe-Beta-70B	$\underline{0.716 \pm 0.009}$	$\underline{0.503 \pm 0.025}$	$\underline{0.670 \pm 0.029}$
HuatuoGPT-o1-70B	0.664 ± 0.013	0.410 ± 0.027	0.470 ± 0.031
MedExAgent-8B (full)	$\mathbf{0.953 \pm 0.008}$	$\mathbf{0.937 \pm 0.015}$	$\mathbf{0.939 \pm 0.015}$
w/o noise injection	0.966 ± 0.007	0.943 ± 0.014	0.944 ± 0.014
SFT-only	0.957 ± 0.010	0.946 ± 0.013	0.946 ± 0.013
w/o tool-match reward	0.962 ± 0.007	0.930 ± 0.016	0.929 ± 0.016
w/o cost penalty	0.846 ± 0.021	0.831 ± 0.023	0.831 ± 0.023

(c) PMC-Patients (N≈1000)

Model / Variant	Sim	Jac	Acc
Llama-3.1-8B-Instruct	0.507 ± 0.013	0.122 ± 0.018	0.157 ± 0.022
Qwen3-8B	0.532 ± 0.008	0.092 ± 0.016	0.120 ± 0.021
Qwen3-14B	0.565 ± 0.007	0.113 ± 0.017	0.143 ± 0.022
Qwen3-32B	0.551 ± 0.010	0.127 ± 0.018	0.157 ± 0.022
MedGemma-27B-text-it	0.469 ± 0.014	0.108 ± 0.017	0.123 ± 0.021
Aloe-Beta-70B	$\underline{0.590 \pm 0.008}$	$\underline{0.171 \pm 0.021}$	$\underline{0.219 \pm 0.026}$
HuatuoGPT-o1-70B	0.534 ± 0.010	0.120 ± 0.019	0.124 ± 0.021
MedExAgent-8B (full)	$\mathbf{0.626 \pm 0.010}$	$\mathbf{0.344 \pm 0.028}$	$\mathbf{0.314 \pm 0.029}$
w/o noise injection	0.624 ± 0.010	0.269 ± 0.026	0.254 ± 0.027
SFT-only	0.506 ± 0.018	0.231 ± 0.026	0.221 ± 0.026
w/o tool-match reward	0.593 ± 0.010	0.219 ± 0.025	0.203 ± 0.025
w/o cost penalty	0.160 ± 0.019	0.102 ± 0.018	0.094 ± 0.018

Table 21: Licenses and sources for existing assets used in this work.

Asset	License
DDxPlus [13]	CC BY 4.0
PMC-Patients-v2 [52]	CC BY-NC-SA 4.0
AgentClinic-MedQA [42]	MIT
PatientSim personas [25]	CC BY-NC-ND 4.0
Disease Ontology [5]	CC0 1.0
Meditron3-8B [40]	Llama 3.1 Community License
Llama-3.1-8B-Instruct [15]	Llama 3.1 Community License
Qwen3-8B/14B/32B [49]	Apache 2.0
MedGemma-27B-text-it [43]	Health AI Developer Foundations License
Aloe-Beta-70B [14]	CC BY-NC 4.0
HuatuoGPT-o1-70B [10]	Apache 2.0
MedEmbed-base-v0.1 [3]	Apache 2.0
CMS PFS RVU 2026 [9]	U.S. government work; CPT content © AMA
CMS CLFS 2025Q4 [8]	U.S. government work; CPT content © AMA

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state three contributions (POMDP environment, two-stage SFT+RL pipeline, comparable-to-larger-baselines results), which are supported by Sections 3, 4-5, and 6, respectively.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We included a Limitations and Future Work Section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our paper includes information about the dataset preprocessing process, finetuning, evaluation, and hyperparameters for reproducing the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The abstract includes a public URL for the code. Released assets are described in Appendix G.2.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Our paper includes information regarding datasets, hyperparmeters, etc. in Section 4 and Appendix A.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Statistical significance is reported in Table 1, Table 2, and Tables in Appendix F.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The compute resources is shown in Appendix A.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our paper uses publicly released datasets under their original licenses, does not involve human subjects or new data collection, and includes a disclaimer that MedExAgent is a research prototype and must not be used for medical advice or patient care.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impact is discussed in Appendix G.1

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Safeguard measures are discussed in Appendix G.1.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Existing assets are described in Appendix G.2

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: New assets are described in Appendix G.2.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All patient interactions are simulated by LLMs.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects. All patient interactions are simulated by LLMs.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLM usage is declared in Appendix G.3

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.