EHRTEMP: LEARNING DYNAMIC CLINICAL DECISION MAKING FROM LONGITUDINAL ELECTRONIC HEALTH RECORDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Clinical diagnosis requires physicians to acquire evidence and revise differential diagnoses as the patient's condition evolves over time. Existing sequential diagnostic agents often interact with a fixed patient snapshot, leaving the effects of clinical progression on observations and evidence acquisition underexplored. We formalize dynamic clinical decision making (DCDM) and introduce EHRTEMP, an interactive environment for studying this setting through longitudinal electronic health record replay. Within EHRTEMP, we train EHRTEMP-AGENT to update its diagnostic belief through differential diagnosis and select between laboratory test requests and final diagnosis. We initialize the agent with differential-diagnosis supervision and optimize its clinical decisions through reinforcement learning with By-Turn Process-Advantage Estimation (BT-PAE), which aligns process supervision with the clinical stage of each decision. On EHRSHOT, our 8B agent achieves the highest Sample F1, Micro F1, and diagnostic precision among the evaluated models, and remains competitive on an independent MIMIC-IV cohort.

1 INTRODUCTION

Clinical diagnosis is not made against a static patient snapshot. Physicians assess the patient, acquire evidence, and determine a diagnosis while the clinical state may evolve, altering symptoms, vital signs, and laboratory measurements. Through differential diagnosis (DDx), they form candidate diagnoses, seek discriminating evidence, and revise their assessment. Diagnosis is therefore dynamic in two senses: the clinician's information state changes as evidence is acquired, and the clinical state generating it may also change over time.

Large language model (LLM) agents offer a way to automate parts of this process by interacting with patients or medical systems, acquiring information, and selecting clinical actions (Teo et al., 2025). Recent work has extended medical question answering (Jin et al., 2020; Pal et al., 2022) to sequential agents that request tests and update diagnostic hypotheses (Nori et al., 2025; Bani-Harouni et al., 2025). Sequential interaction, however, does not by itself model clinical progression. An agent may acquire information over multiple turns while the underlying case remains fixed, so that interaction amounts to sequential revelation of a static patient snapshot. For example, MIMIC-CDM retains only the first recorded measurement of each laboratory test, preventing repeated requests from revealing changes in laboratory values over clinical time (Hager et al., 2024). Dynamic clinical decision making requires both the agent's information state and the clinical state generating its observations to evolve. Figure 1 illustrates this distinction and the additional diagnostic difficulty that arises when clinical time advances between decisions.

This setting creates two challenges. First, *clinical progression* changes both available evidence and its values: future measurements are inaccessible, and the same test may differ across clinical times, requiring belief revision as the trajectory unfolds. Second, *evidence acquisition* must account for elapsed clinical time. Clinicians often request several tests together rather than serially, since waiting for each result can delay diagnosis as the patient's condition evolves. The agent must therefore decide what evidence to acquire jointly at each decision point.

We formalize this problem as *dynamic clinical decision making* (DCDM) using a POMDP framework (Kaelbling et al., 1998) that distinguishes evolving clinical state, observations, and diagnostic

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

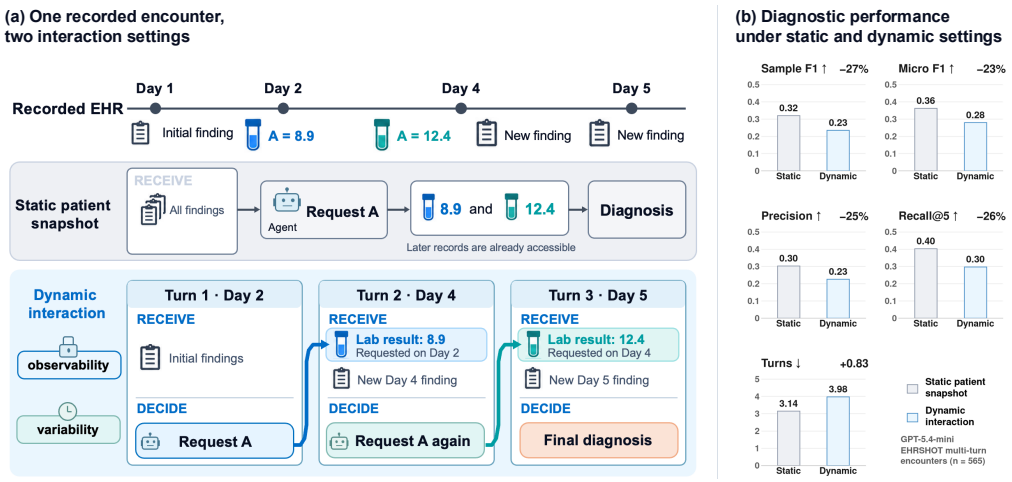


Figure 1: Motivation for dynamic clinical decision making (DCDM). (a) A static snapshot exposes all non-laboratory findings initially and permits requests for measurements from the complete encounter. Under dynamic interaction, observability prevents a Day 2 request from accessing the Day 4 measurement, while variability allows requests on different days to yield different values. Requested results arrive at the next decision point alongside new non-laboratory findings. (b) GPT-5.4-mini evaluated on the multi-turn subset of the EHRSHOT ID test set, comprising encounters with at least two recorded laboratory-ordering days, achieves lower diagnostic scores and requires more turns under dynamic interaction. This subset differs from the full ID test set used in Table 1.

belief. We then develop EHRTEMP, a longitudinal EHR replay environment for studying DCDM. EHRTEMP models temporal evolution through replay rather than intervention-dependent patient simulation: the recorded clinical trajectory advances with clinical time, while the agent controls which eligible laboratory evidence it acquires. Agent actions therefore affect information acquisition but do not alter the underlying recorded trajectory. Following a laboratory request, EHRTEMP advances clinical time and returns eligible requested measurements together with newly available non-laboratory observations; the agent may instead terminate by committing to a final diagnosis. This supports evaluation and reinforcement-learning-based policy training while preserving recorded EHR temporal constraints.

We train EHRTEMP-AGENT to assess currently available observations, maintain a DDx, and choose between acquiring laboratory evidence and committing to a diagnosis. EHR trajectories augmented with offline DDx labels initialize this behavior: recorded actions supervise action selection and DDx annotations supervise diagnostic belief. We then optimize diagnostic outcome and laboratory process rewards. Because process rewards at different turns arise under different clinical and information states, pooling them can compare actions from different encounter stages. *By-Turn Process-Advantage Estimation* (BT-PAE) instead compares rollouts of the same encounter at the same decision turn.

On EHRSHOT, the resulting 8B agent achieves the highest Sample F1, Micro F1, and diagnostic precision among evaluated systems and transfers competitively to an independent MIMIC-IV cohort. Laboratory-agreement metrics measure agreement with recorded testing and timing, not clinical optimality. Ablations and behavioral analyses examine DDx-guided initialization, process-supervision design, evidence acquisition, and diagnostic-belief revision.

Our contributions are threefold:

- We formalize DCDM and introduce EHRTEMP, a longitudinal EHR replay environment in which evidence availability and observed values evolve with clinical time while agents actively select laboratory evidence to acquire.
- We develop EHRTEMP-AGENT through DDx-guided initialization and reinforcement learning, and introduce BT-PAE, which conditions relative process-reward comparisons on encounter and decision turn instead of pooling actions from different clinical stages.

- We evaluate an 8B DCDM agent on EHRSHOT and an independent MIMIC-IV cohort, and provide ablations and behavioral analyses of diagnostic-belief evolution, evidence acquisition, and process-supervision design.

2 Related Work

2.1 Sequential Clinical Decision-Making Agents

Medical question-answering benchmarks evaluate diagnostic reasoning from supplied descriptions (Jin et al., 2020; Pal et al., 2022); EHR-based methods such as ColaCare (Wang et al., 2025), EHR2Path (Pellegrini et al., 2026), and EHR-R1 (Liao et al., 2025) incorporate longitudinal records. Sequential agents acquire information through dialogue, examination, or laboratory requests, including DoctorAgent-RL (Feng et al., 2025), Doctor-R1 (Lai et al., 2026), Baichuan-M2 (Team et al., 2025), SDBench (Nori et al., 2025), LA-CDM (Bani-Harouni et al., 2025), LDTL (Shen et al., 2026), MedChain (Liu et al., 2025), and DiaLLM (Ren et al., 2025). DCDM instead advances clinical time during interaction, changing evidence availability and observed values. EHRTemp models this through longitudinal EHR replay: the recorded trajectory evolves independently of agent actions, while requests determine which eligible laboratory evidence is revealed.

2.2 Reinforcement Learning for Medical Agents

Reinforcement learning has trained medical agents with feedback on final outcomes and intermediate decisions. Doctor-R1 (Lai et al., 2026) uses process feedback for clinical inquiry, while LA-CDM (Bani-Harouni et al., 2025) combines diagnostic outcomes with test-cost considerations, connecting to agentic RL (Zhang et al., 2026), turn-level supervision (Wei et al., 2025), and GRPO (Shao et al., 2024). Our method aligns DDx assessments and recorded actions with EHR decision points; RL evaluates requests against turn-specific evidence, and BT-PAE compares process rewards within the same encounter and turn.

3 Methodology

We formalize DCDM and its instantiation in EHRTemp through longitudinal EHR replay, then develop EHRTemp-Agent using DDx-guided supervised initialization and reinforcement learning with BT-PAE (Figure 2).

3.1 Dynamic Clinical Decision Making: Problem Formulation

We formulate dynamic clinical decision making (DCDM) in the diagnostic setting using the partially observable Markov decision process (POMDP) framework (Kaelbling et al., 1998). The process is represented by $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, \mathcal{O}, \Omega, R \rangle$, comprising world states, actions, transition dynamics, observations, an observation mechanism, and rewards. The agent maintains a belief about the patient's underlying clinical condition and selects actions according to a policy π_θ. This formulation permits action-dependent state transitions in general. EHRTemp implements the observational-replay case: clinical state progression follows the recorded trajectory, while agent actions govern evidence acquisition but do not alter that trajectory.

World State: Dynamically Evolving Patient State. The world state describes the patient's underlying clinical condition, which is only partially observable to the agent. When sequential diagnosis is modeled over a fixed patient snapshot, additional interactions reveal information about the same underlying condition. In DCDM, the patient's condition can also evolve with clinical time, so successive interactions correspond to different points along the clinical trajectory. Let $s_t \in \mathcal{S}$ denote the patient state at decision turn t, associated with clinical time τ_t. In the general DCDM formulation, state evolution may depend on both elapsed clinical time and the preceding action:

$$s_{t+1} \sim P(\cdot \mid s_t, a_t, \tau_{t+1}). \tag{1}$$

EHRTemp instantiates a restricted but observationally grounded case in which state progression is replayed from the EHR rather than generated in response to the agent. For a recorded encounter,

$$s_{t+1} \sim P_{\text{replay}}(\cdot \mid s_t, \tau_{t+1}), \tag{2}$$

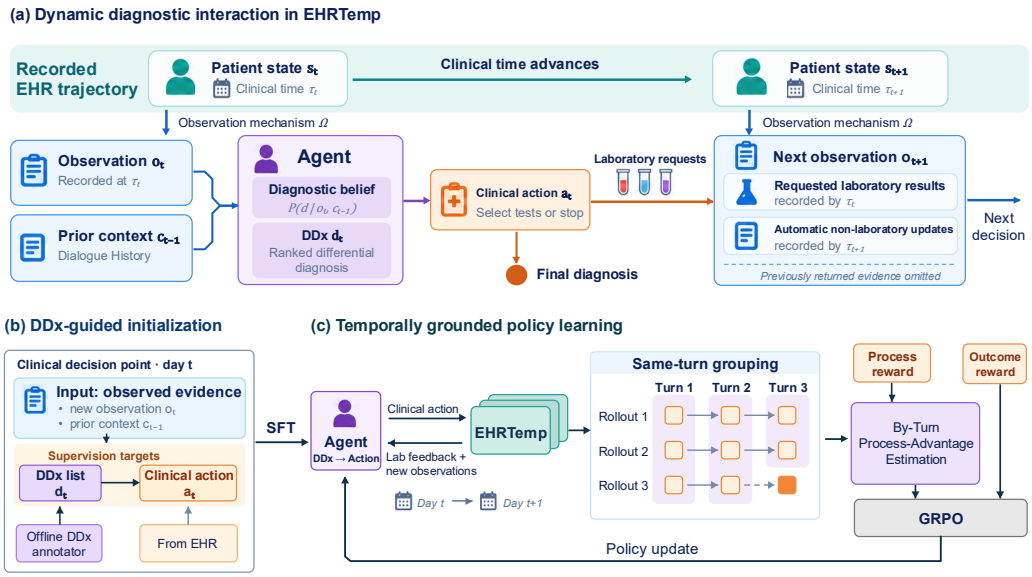


Figure 2: **Overview of dynamic diagnostic interaction in EHRTEMP and training of EHRTEMP-AGENT. (a) Dynamic interaction.** At turn t, the agent combines new observation o_t with prior context c_{t-1}, updates its DDx, and selects either laboratory requests or a final diagnosis. A laboratory action advances clinical time to the next recorded EHR decision point τ_{t+1}, where requested laboratory evidence and newly recorded non-laboratory findings form the next observation. **(b) DDx-guided initialization.** Offline DDx annotations supervise the representation of diagnostic belief, while recorded EHR decisions provide clinical-action targets at successive decision points. **(c) Policy learning.** The initialized agent interacts with EHRTEMP using diagnostic outcome and laboratory process rewards. BT-PAE compares process rewards among rollouts of the same encounter at the same decision turn, aligning advantage estimation with clinical stage before GRPO updates.

so the non-terminal action does not alter the underlying trajectory. Each dialogue turn corresponds to an EHR decision point, defined by a date with recorded laboratory tests or a final diagnosis. Following a non-terminal action, the dialogue index advances from t to $t+1$, while clinical time progresses from τ_t to the next recorded decision time τ_{t+1}. Consecutive dialogue turns therefore need not correspond to consecutive clinical days. The sequence of decision times is fixed by the recorded trajectory; rollouts of the same encounter that reach turn t consequently share τ_t. Time-indexed EHR findings provide observations of this evolving recorded state.

Action: Evidence Acquisition or Final Diagnosis. Actions specify how the agent interacts with the environment. In our diagnostic setting, let $\mathcal{L}$ denote the set of orderable laboratory tests. A laboratory action selects a set of tests, $a_t^{\text{lab}} \in \mathcal{A}_{\text{lab}} \subseteq 2^{\mathcal{L}}$, while a terminal diagnostic action belongs to $\mathcal{A}_{\text{diag}}$. The overall action space is

$$a_t \in \mathcal{A} = \mathcal{A}_{\text{lab}} \cup \mathcal{A}_{\text{diag}}. \tag{3}$$

A laboratory request acquires evidence that informs the agent's belief about the patient's clinical state, whereas a diagnostic action commits to one or more diagnoses and terminates the episode. In EHRTEMP, laboratory requests therefore control evidence delivery without altering the recorded patient trajectory. Since each laboratory action advances the interaction to the next clinical decision point, all tests intended to be requested under the same clinical-time conditions must be selected jointly within a single action.

Observation: Gated and Variable Information. Observations provide the partial evidence through which the agent assesses the patient. Let $o_t \in \mathcal{O}$ denote the new clinical evidence delivered at decision turn t, comprising requested laboratory measurements and automatically exposed non-

laboratory findings. The observation mechanism combines the evolving clinical state, the agent's request, and the preceding interaction context:

$$o_{t+1} \sim \Omega(\cdot \mid s_{t+1}, a_t, c_t), \tag{4}$$

where c_t denotes the interaction context accumulated through turn t, including observations, DDx lists, and actions. In EHRTEMP, the laboratory request a_t selects which tests to retrieve from the underlying EHR. The environment returns their measurements recorded at or before the request time τ_t, excluding those already delivered in c_t.

The dynamic nature of DCDM is reflected in two aspects of its observations:

(i) Observability. The evidence eligible for disclosure changes as clinical time advances, with future measurements excluded from the current observation:

$$v(\ell, \tau_i) \notin o_j, \qquad \tau_i > \tau_j, \tag{5}$$

where $v(\ell, \tau_i)$ denotes the recorded value of laboratory test ℓ at clinical time τ_i. For a request made at τ_t, the response at the next decision point includes all measurements of the requested tests recorded at or before τ_t that have not previously been returned.

(ii) Variability. Clinical variables may change as the patient's condition evolves, so measurements of the same test at different clinical times may satisfy

$$v(\ell, \tau_i) \neq v(\ell, \tau_j), \qquad \tau_i \neq \tau_j. \tag{6}$$

In addition to requested laboratory evidence, EHRTEMP automatically exposes newly recorded symptoms, vital-sign measurements, and other non-laboratory findings as the trajectory progresses. This channel represents routine clinical updates received without a separate laboratory request and is likewise time-gated: the response at τ_{t+1} includes newly exposed non-laboratory findings recorded up to τ_{t+1}. Each response thus combines laboratory evidence bounded by the preceding request time with non-laboratory updates bounded by the new clinical decision time.

Belief: Explicit DDx Representation. A POMDP agent maintains a belief b_t, a probability distribution over world states updated from its actions and observations. This distinguishes the patient's underlying condition from the agent's assessment based on its experience. We use a DDx list d_t as an explicit representation of the agent's *diagnostic belief*. At each decision point, the agent generates this list from the current observation and the preceding interaction context:

$$d_t \sim \pi_\theta(\cdot \mid c_{t-1}, o_t). \tag{7}$$

Access to earlier observations and DDx lists allows the agent to revise its diagnostic hypotheses as new evidence arrives. Generating and revising d_t serves a diagnostic state-estimation role, and the agent uses this assessment together with the observed evidence to select its next action.

Task Objective. The agent seeks to maximize expected final diagnostic performance:

$$\max_\theta \; J_{\text{diag}}(\theta) = \mathbb{E}_{\pi_\theta, \mathcal{M}} \left[U_{\text{diag}}(\hat{y}, y) \right], \tag{8}$$

where $\hat{y}$ is the diagnosis set submitted at termination, y is the reference diagnosis set, and U_{diag} is their set-level F1 score. At the task level, R assigns this utility to the terminal diagnostic decision and zero reward to intermediate actions. The policy-learning section specifies the training rewards, including auxiliary laboratory process feedback for intermediate decisions.

3.2 EHRTEMP-AGENT: LEARNING TO DIAGNOSE IN DYNAMIC CLINICAL ENVIRONMENTS

We parameterize EHRTEMP-AGENT as a policy π_θ that generates a DDx list d_t followed by a clinical action a_t, conditioned on the preceding interaction context c_{t-1} and current observation o_t:

$$\pi_\theta(d_t, a_t \mid c_{t-1}, o_t) = \pi_\theta(d_t \mid c_{t-1}, o_t)\, \pi_\theta(a_t \mid c_{t-1}, o_t, d_t). \tag{9}$$

This factorization reflects the response-generation order: the agent first forms an explicit diagnostic assessment and then selects its clinical action conditioned on that assessment and the available evidence.

DCDM requires the agent to update its diagnostic belief as new evidence arrives and use this assessment to guide clinical decisions. We initialize these behaviors through supervised learning from EHR trajectories augmented with DDx labels at successive decision points, then optimize the agent's decisions through reinforcement learning in EHRTEMP. Figure 2 illustrates this training procedure.

3.2.1 DIFFERENTIAL-DIAGNOSIS-GUIDED INITIALIZATION

Supervision Construction. Longitudinal EHRs record the evidence observed and actions taken by clinicians at successive stages of the patient's course. We preserve this sequence by constructing supervised trajectories according to the decision points and evidence-release rules of EHRTEMP. At clinical time τ_t, the input comprises the preceding interaction context c_{t-1} and the newly delivered observation o_t, while the target consists of a DDx list d_t followed by a clinical action a_t. Actions are derived from the recorded clinical decisions, and DDx labels are supplied by the offline annotator described below. This construction pairs stage-specific diagnostic assessments with recorded actions, providing supervision for belief formation and decision making as the patient's clinical evidence evolves.

DDx Annotation. The EHR trajectories provide observations and actions but lack explicit intermediate diagnostic hypotheses. We supplement them with DDx labels generated from the evidence exposed at each decision point. The annotator is a BioClinical ModernBERT (Sounack et al., 2025) classifier trained using disease-level DDx mappings extracted from clinical guidelines available through PubMed Central (PMC) (Roberts, 2001) and normalized through the ICD-10-CM hierarchy, together with patient cases from medical QA datasets and EHRSHOT. Its predictions provide reference diagnostic assessments for the supervised trajectories. Labels are precomputed before agent training, and the annotator is not invoked during agent training or inference. We initialize EHRTEMP-AGENT through LoRA-based supervised fine-tuning on the resulting trajectories. Data splits, DDx annotation and sampling, and fine-tuning details are provided in Appendices A.5, C, and D, respectively.

3.2.2 POLICY LEARNING WITH BY-TURN PROCESS-ADVANTAGE ESTIMATION

Following supervised initialization, we optimize EHRTEMP-AGENT through interaction with EHRTEMP using diagnostic outcome rewards and auxiliary laboratory process rewards. We adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which estimates advantages relative to sampled groups without training a separate critic. In DCDM, process rewards at different turns are obtained under different evidence-release boundaries and potentially different patient states. Pooling such rewards when constructing relative advantages can therefore confound action quality with systematic differences between clinical stages. We introduce By-Turn Process-Advantage Estimation (BT-PAE), which organizes process-reward comparisons by decision turn within each encounter while retaining GRPO as the underlying policy optimizer.

Outcome and Laboratory Process Rewards. The outcome reward evaluates the terminal diagnosis set $\hat{y}$ against the reference set y, supplementing the task-level F1 objective with an exact-match bonus:

$$r^{\mathrm{diag}} = \mathrm{F1}(\hat{y}, y) + 0.3\,\mathbf{1}[\hat{y} = y]. \tag{10}$$

For a laboratory action a_t, the process reward encourages evidence acquisition by rewarding requests that match temporally eligible recorded laboratory orders. Let $\mathcal{E}_t(\ell)$ denote recorded orders matching test ℓ at or before the request time τ_t that have not been credited in the rollout. We define

$$r_t^{\mathrm{lab}} = \sum_{\ell \in a_t} \begin{cases} \alpha, & \mathcal{E}_t(\ell) \neq \varnothing, \\ -\kappa, & \text{otherwise}, \end{cases} \qquad a_t \in \mathcal{A}_{\mathrm{lab}}. \tag{11}$$

Here, $\alpha > 0$ rewards agreement with eligible recorded orders, and $\kappa > 0$ penalizes requests without a match. This provides an order-based proxy for evidence acquisition, rather than requiring each rewarded request to return a measurement value. Eligibility is evaluated at the request time, and the reward is attributed to a_t. A turn that violates the required output format receives zero reward in place of its process or outcome reward; this rule applies to every variant with process rewards.

By-Turn Process-Advantage Estimation. For each encounter, we sample G rollouts using the behavior policy $\pi_{\theta_{\mathrm{old}}}$. Rollouts reaching the same turn t share the clinical time τ_t and the corresponding evidence-release boundaries, although their acquired laboratory evidence, DDx lists, and actions can differ. This correspondence provides a common clinical-stage reference for comparing process rewards.

Responses are classified as FINAL when a terminal-diagnosis structure is identified and as INTERMEDIATE otherwise. Let $\mathcal{G}_t$ contain the rollout indices with INTERMEDIATE responses at turn t, including malformed responses with zero process reward, and let $\mathcal{T} = \{t : |\mathcal{G}_t| > 0\}$. BT-PAE centers process rewards using the turn-specific mean $\bar{r}_t = |\mathcal{G}_t|^{-1} \sum_{i \in \mathcal{G}_t} r_t^{\text{lab},(i)}$. Pooling rewards across turns before centering would incorporate differences between turn-specific reward means into the resulting advantages, even when those differences arise from changing evidence availability rather than relative action quality. In the observed training rollouts, for example, mean process reward decreases from $+0.67$ at the first turn to -0.06 at the fourth turn (Appendix E.5). This turn dependence motivates estimating relative process advantages within, rather than across, decision turns.

Because rollouts can terminate at different turns, later groups may contain fewer samples. We estimate a shared within-turn scale using the pooled residual variance of a one-way ANOVA (Montgomery, 2019):

$$\sigma^2_{\text{pool}} = \frac{\sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{G}_t} \left(r_t^{\text{lab},(i)} - \bar{r}_t \right)^2}{\sum_{t \in \mathcal{T}} (|\mathcal{G}_t| - 1)}, \tag{12}$$

and compute

$$A^{(i)}_{t,\text{proc}} = \frac{r_t^{\text{lab},(i)} - \bar{r}_t}{\sigma_{\text{pool}} + \varepsilon_{\text{num}}}. \tag{13}$$

The pooled estimator treats within-turn dispersion as approximately shared across turns and excludes variation between turn means from the normalization scale. We set $\sigma_{\text{pool}} = 0$ when the residual degrees of freedom are zero; $\varepsilon_{\text{num}} > 0$ ensures a positive denominator.

Outcome advantages are estimated separately within the encounter's FINAL pool:

$$A^{(i)}_{\text{out}} = \frac{f^{(i)} - \bar{f}}{\max(\sigma_f, \delta_f) + \varepsilon_{\text{num}}}, \tag{14}$$

where $f^{(i)}$ is the terminal outcome reward, $\bar{f}$ and σ_f are the FINAL-pool mean and standard deviation, and δ_f is a standard-deviation floor. Invalid FINAL responses enter this pool with zero reward; rollouts without a FINAL response do not contribute to its statistics. For retained training turns, the combined advantage is

$$A^{(i)}_t = \begin{cases} A^{(i)}_{t,\text{proc}} + \lambda A^{(i)}_{\text{out}}, & \text{INTERMEDIATE}, \\ A^{(i)}_{\text{out}}, & \text{FINAL}, \end{cases} \tag{15}$$

where λ weights outcome feedback on intermediate decisions. No outcome term is injected when the rollout has no FINAL response or its FINAL pool is masked.

Policy Optimization. We optimize the policy using the clipped GRPO objective, assigning $A^{(i)}_t$ to the optimized response tokens at turn t:

$$\mathcal{J}(\theta) = \mathbb{E}\Big[\min\Big(\rho^{(i)}_{t,k} A^{(i)}_t, \text{clip}\Big(\rho^{(i)}_{t,k}, 1-\epsilon, 1+\epsilon \Big) A^{(i)}_t \Big) - \beta \mathcal{K}^{(i)}_{t,k} \Big], \tag{16}$$

where $\rho^{(i)}_{t,k}$ is the current-to-behavior policy probability ratio for token k at turn t of rollout i, and $\mathcal{K}^{(i)}_{t,k}$ is the KL divergence between the current and reference policies at the same token context. The expectation is over training encounters, sampled rollouts, and optimized response tokens; ϵ controls clipping and β weights KL regularization. Hyperparameters, pool-size filtering, boundary handling, token weighting, and loss masking are detailed in Appendix E.

4 EXPERIMENTS

Our evaluation addresses three questions: (1) How does EHRTEMP-AGENT compare with general-purpose LLMs and medical agents in diagnostic performance under DCDM? (2) How do DDx-guided initialization and the organization of process supervision affect diagnostic performance and

Table 1: In-distribution performance on EHRSHOT. Diagnosis metrics evaluate final diagnostic predictions; behavior metrics measure laboratory test–day agreement and diagnostic completion. Sample F1 is the primary metric. Lab–Day F1 is micro-averaged. Best results are shown in bold.

	Diagnosis				Behavior	
Method	Sample F1	Micro F1	Precision	Recall@5	Lab–Day F1	Completion
General-purpose LLMs						
GPT-5.4-mini	0.269	0.303	0.232	0.391	0.091	0.917
Claude Haiku 4.5	0.350	0.369	0.274	**0.580**	0.037	**0.995**
DeepSeek-V4-Flash	0.397	0.425	0.359	0.529	0.085	0.991
Qwen3-30B	0.230	0.299	0.202	0.332	0.021	0.770
Medical LLMs and agents						
EHR-R1-8B	0.211	0.213	0.168	0.378	0.017	0.990
Baichuan-M2-32B	0.236	0.296	0.181	0.412	0.064	0.742
HuatuoGPT-3-8B	0.173	0.200	0.148	0.252	0.004	0.889
DoctorAgent-8B	0.076	0.106	0.084	0.083	0.020	0.459
Qwen3-8B and task-trained variants						
Qwen3-8B (zero-shot)	0.051	0.110	0.048	0.065	0.029	0.126
Qwen3-8B (few-shot)	0.087	0.169	0.083	0.108	0.027	0.222
DDx-guided SFT	0.379	0.408	0.423	0.398	0.588	0.988
EHRTemp-Agent-8B	**0.415**	**0.445**	**0.452**	0.439	**0.617**	0.992

Table 2: Effects of DDx-guided initialization and process-supervision design on the ID test set. RL variants use the DDx-guided SFT checkpoint. Lab–Day F1 measures micro-averaged agreement in laboratory selection and clinical timing. Best observed results within each block are shown in bold. BT-PAE is reported over three training seeds in the text; the other RL variants use one seed each. Appendix E.8 specifies the reward and advantage configuration of each RL variant.

	Diagnosis				Behavior	
Variant	Sample F1	Micro F1	Precision	Recall@5	Lab–Day F1	Completion
Supervised initialization						
SFT without DDx	0.365	**0.417**	0.353	**0.443**	0.544	0.852
DDx-guided SFT	**0.379**	0.408	**0.423**	0.398	**0.588**	**0.988**
Reinforcement learning						
Outcome-only GRPO	0.406	0.437	0.425	**0.454**	0.595	0.995
Combined-reward estimation	0.400	0.430	0.451	0.409	**0.617**	0.991
Across-turn process estimation	0.406	0.437	0.435	0.445	0.612	**0.996**
By-turn estimation (BT-PAE)	**0.415**	**0.445**	**0.452**	0.439	**0.617**	0.992

evidence-acquisition behavior? (3) How do agents acquire evidence and revise their diagnostic beliefs as interactions progress along clinical trajectories? We address the first two questions through in-distribution and out-of-distribution evaluation and training-component ablations. Behavioral analysis addressing the third question appears in Appendix H.3. Experimental setup details are in Appendix F.

4.1 Diagnostic Performance

In-distribution Evaluation. Table 1 shows that EHRTemp-Agent achieves the highest Sample F1 (0.415), Micro F1 (0.445), and precision (0.452) among all evaluated models, on the in-distribution EHRSHOT dataset. Compared with the strongest external baseline, DeepSeek-V4-Flash, it improves these metrics by 1.8, 2.0, and 9.3 percentage points, respectively. These are system-level rather than matched-foundation-model comparisons: unlike external baselines, EHRTemp-Agent receives task-specific supervised and reinforcement learning in EHRTemp. They establish performance under a common interaction environment, not an architecture- or scale-controlled advantage.

EHRTemp-Agent also outperforms all evaluated medical LLMs and agents on all four diagnostic metrics, including 32B Baichuan-M2 and EHR-specialized EHR-R1-8B. Among general-purpose models, only Claude Haiku 4.5 and DeepSeek-V4-Flash achieve higher Recall@5, while

EHRTEMP-AGENT leads both in precision and F1. Relative to DDx-guided supervised initialization, policy learning further improves Sample F1 from 0.379 to 0.415 and Micro F1 from 0.408 to 0.445, while maintaining a completion rate of 0.992.

Beyond diagnosis, EHRTEMP-AGENT achieves Lab–Day F1 of 0.617, versus 0.588 for SFT and at most 0.091 among external baselines, indicating closer agreement with recorded test selection and timing. Lab–Day F1 measures behavioral agreement with the recorded trajectory; it does not establish that the recorded tests, or requests that match them, are clinically optimal.

Out-of-distribution Evaluation. On MIMIC-IV, EHRTEMP-AGENT achieves the second-highest Sample F1 (0.315), outperforming every medical baseline and all general-purpose baselines except DeepSeek-V4-Flash (0.362). Policy learning improves Sample F1 from 0.299 to 0.315 and Micro F1 from 0.323 to 0.341 over supervised initialization. The diagnostic improvement thus persists on an independent EHR source. After preprocessing, approximately 60% of OOD laboratory test types fall outside the ID executable action space, adding a laboratory-recommendation challenge. We interpret this experiment as cross-source transfer rather than unrestricted generalization to arbitrary clinical settings. Complete OOD comparisons are reported in Appendix A.6.

4.2 ABLATION STUDY

DDx-Guided Initialization. On the same filtered training encounters, adding DDx supervision raises Sample F1 from 0.365 to 0.379, precision from 0.353 to 0.423, Lab–Day F1 from 0.544 to 0.588, and completion from 0.852 to 0.988, while Micro F1 falls from 0.417 to 0.408 and Recall@5 from 0.443 to 0.398. Thus, DDx supervision favors precision, recorded-evidence agreement, and completion rather than uniformly improving diagnostic accuracy.

Process Supervision and By-Turn Advantage Estimation. From the same DDx-guided SFT checkpoint, we examine how laboratory process rewards enter policy learning. BT-PAE yields the strongest observed combination of Sample F1 and Lab–Day F1: relative to outcome-only GRPO, it improves both metrics ($0.406 \rightarrow 0.415$ and $0.595 \rightarrow 0.617$, respectively). The other process-supervised estimators improve laboratory agreement but do not improve diagnosis. Combining process and outcome rewards before estimating a trajectory-level advantage lowers Sample F1 to 0.400. Estimating process advantages separately across turns restores it to 0.406; grouping those comparisons by turn in BT-PAE raises it to 0.415. These comparisons involve complete estimators, including their boundary conventions (Appendix E.8). BT-PAE achieves Sample F1 of 0.4145 ± 0.0021 across three seeds; the other variants use one seed each, so the observed gaps do not establish statistically reliable superiority.

Behavior under Dynamic Clinical Progression. On the multi-turn analysis subset, the trained agent attains Lab–Day F1 of 0.516 and per-turn laboratory precision of 0.564, compared with 0.481 and 0.533 for DDx-guided SFT. Its DDx MRR also improves by approximately 12–14% over the first three decision turns relative to its first-turn value, consistent with progressive improvement in diagnostic belief during supervised initialization that is retained after RL. Because repeated-request and DDx analyses use model-dependent subsets, these are behavioral characterizations, not direct cross-model comparisons. Figure 6, metric definitions, and the analysis are provided in Appendix H.3.

5 CONCLUSION

We formalized dynamic clinical decision making and introduced EHRTEMP for diagnostic interaction through longitudinal EHR replay. EHRTEMP-AGENT learns to update its diagnostic belief and select actions through DDx-guided initialization and reinforcement learning with BT-PAE. Experiments show strong diagnostic performance, progressive improvement in diagnostic belief as evidence accumulates, and support for organizing process supervision by clinical stage. The framework connects evolving patient state, observations, and decisions; future work can extend beyond recorded trajectories to clinical interventions.

AI USE STATEMENT

The authors conceptualized and executed the study. Generative AI was not used to generate synthetic datasets, perform the experimental evaluations, statistical analyses, formulate theoretical claims, or assist with proofs. It was used to critique aspects of the methodology, experimental setup, and interpretation; assist with coding and figures; summarize, and retrieve relevant literature; and edit or rephrase text for clarity. All AI-assisted work and code were reviewed by the authors, and all citations were manually verified and added by the authors. The authors accept full responsibility for the manuscript and artifacts.

REPRODUCIBILITY STATEMENT

The Appendix provides implementation details and experimental settings. The anonymous supplementary material includes the environment, RL agent and training code, ablation implementation, evaluation-metric code, and lab/diagnose vocabulary. More code will be released upon acceptance of the manuscript for publication.

REFERENCES

David Bani-Harouni, Chantal Pellegrini, Ege Özsoy, Nassir Navab, and Matthias Keicher. Language Agents for Hypothesis-driven Clinical Decision Making with Reinforcement Learning, 2025.

Hanjie Chen, Zhouxiang Fang, Yash Singla, and Mark Dredze. Benchmarking large language models on answering and explaining challenging medical questions. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3563–3599, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.182. URL https://aclanthology.org/2025.naacl-long.182/.

Junying Chen, Xinyuan Xie, Ziniu Li, and Benyou Wang. Onepo: Direct one-stage policy optimization for sft-free domain adaptation. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026.

Yichun Feng, Jiawei Wang, Lu Zhou, Zhen Lei, and Yixue Li. DoctorAgent-RL: A Multi-Agent Collaborative Reinforcement Learning System for Multi-Turn Clinical Dialogue, October 2025.

Paul Hager, Friederike Jungmann, Robbie Holland, Kunal Bhagat, Inga Hubrecht, Manuel Knauer, Jakob Vielhauer, Marcus Makowski, Rickmer Braren, Georgios Kaissis, and Daniel Rueckert. Evaluation and mitigation of the limitations of large language models in clinical decision-making. *Nature Medicine*, 30(9):2613–2622, September 2024. ISSN 1546-170X. doi: 10.1038/s41591-024-03097-1.

IHTSDO. SNOMED-CT to ICD-10 Code Mapping Demonstrator. https://github.com/IHTSDO/snomed-to-icd10-mapper, 2024. Release 2.0.0; accessed 2026-06-19.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What Disease does this Patient Have? A Large-scale Open Domain Question Answering Dataset from Medical Exams, September 2020.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, and Brian Gow. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1, 2023.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Nikhil Khandekar, Qiao Jin, Guangzhi Xiong, Soren Dunn, Serina S. Applebaum, Zain Anwar, Maame Sarfo-Gyamfi, Conrad W. Safranek, Abid A. Anwar, Andrew Zhang, Aidan Gilson, Maxwell B. Singer, Amisha Dave, Andrew Taylor, Aidong Zhang, Qingyu Chen, and Zhiyong Lu. MedCalc-Bench: Evaluating Large Language Models for Medical Calculations, June 2024.

Yunghwei Lai, Kaiming Liu, Ziyue Wang, Weizhi Ma, and Yang Liu. Doctor-R1: Mastering Clinical Inquiry with Experiential Agentic Reinforcement Learning, March 2026.

Yusheng Liao, Chaoyi Wu, Junwei Liu, Shuyang Jiang, Pengcheng Qiu, Haowen Wang, Yun Yue, Shuai Zhen, Jian Wang, Qianrui Fan, Jinjie Gu, Ya Zhang, Yanfeng Wang, Yu Wang, and Weidi Xie. EHR-R1: A Reasoning-Enhanced Foundational Language Model for Electronic Health Record Analysis, November 2025.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. Self-Alignment Pretraining for Biomedical Entity Representations, April 2021.

Jie Liu, Wenxuan Wang, Zizhan Ma, Guolin Huang, Yihang SU, Kao-Jung Chang, Wenting Chen, Haoliang Li, Linlin Shen, and Michael Lyu. Medchain: Bridging the Gap Between LLM Agents and Clinical Practice with Interactive Sequence, October 2025.

Douglas C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, 10 edition, 2019.

National Library of Medicine. LOINC To CPT Mapping. `https://www.nlm.nih.gov/research/umls/mapping_projects/loinc_to_cpt_map.html`, 2017. Last reviewed: January 27, 2017; accessed: 2026-06-19.

Harsha Nori, Mayank Daswani, Christopher Kelly, Scott Lundberg, Marco Tulio Ribeiro, Marc Wilson, Xiaoxuan Liu, Viknesh Sounderajah, Jonathan Carlson, Matthew P. Lungren, Bay Gross, Peter Hames, Mustafa Suleyman, Dominic King, and Eric Horvitz. Sequential Diagnosis with Language Models, July 2025.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning*, pp. 248–260. PMLR, 2022.

Chantal Pellegrini, Ege Özsoy, David Bani-Harouni, Matthias Keicher, and Nassir Navab. EHR2Path: Scalable Modeling of Longitudinal Patient Pathways from Multimodal Electronic Health Records, March 2026.

Weijieying Ren, Tianxiang Zhao, Lei Wang, Tianchun Wang, and Vasant G Honavar. DiaLLMs: EHR-Enhanced Clinical Conversational System for Clinical Test Recommendation and Diagnosis Prediction. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25622–25635, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1313.

Richard J. Roberts. PubMed Central: The GenBank of the published literature. *Proceedings of the National Academy of Sciences*, 98(2):381–382, January 2001. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.98.2.381.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, April 2024.

Xuyang Shen, Haoran Liu, Dongjin Song, and Martin Renqiang Min. Uncertainty-Guided Latent Diagnostic Trajectory Learning for Sequential Clinical Diagnosis, April 2026.

Thomas Sounack, Joshua Davis, Brigitte Durieux, Antoine Chaffin, Tom J. Pollard, Eric Lehman, Alistair E. W. Johnson, Matthew McDermott, Tristan Naumann, and Charlotta Lindvall. BioClinical ModernBERT: A State-of-the-Art Long-Context Encoder for Biomedical and Clinical NLP, June 2025.

Baichuan-M2 Team, Chengfeng Dou, Chong Liu, Fan Yang, Fei Li, Jiyuan Jia, Mingyang Chen, Qiang Ju, Shuai Wang, Shunya Dang, Tianpeng Li, Xiangrong Zeng, Yijie Zhou, Chenzheng Zhu, Da Pan, Fei Deng, Guangwei Ai, Guosheng Dong, Hongda Zhang, Jinyang Tai, Jixiang Hong, Kai Lu, Linzhuang Sun, Peidong Guo, Qian Ma, Rihui Xin, Shihui Yang, Shusen Zhang, Yichuan Mo, Zheng Liang, Zhishou Zhang, Hengfu Cui, Zuyi Zhu, and Xiaochuan Wang. Baichuan-M2: Scaling Medical Capability with Large Verifier System, September 2025.

Zhen Ling Teo, Arun James Thirunavukarasu, Kabilan Elangovan, Haoran Cheng, Prasanth Moova, Brian Soetikno, Christopher Nielsen, Andreas Pollreisz, Darren Shu Jeng Ting, and Robert JT Morris. Generative artificial intelligence in medicine. *Nature medicine*, 31(10):3270–3282, 2025.

Zixiang Wang, Yinghao Zhu, Huiya Zhao, Xiaochen Zheng, Dehao Sui, Tianlong Wang, Wen Tang, Yasha Wang, Ewen Harrison, Chengwei Pan, Junyi Gao, and Liantao Ma. ColaCare: Enhancing Electronic Health Record Modeling through Large Language Model-Driven Multi-Agent Collaboration. In *Proceedings of the ACM on Web Conference 2025*, pp. 2250–2261, April 2025. doi: 10.1145/3696410.3714877.

Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. Reinforcing Multi-Turn Reasoning in LLM Agents via Turn-Level Reward Design, October 2025.

Michael Wornow, Rahul Thapa, Ethan Steinberg, Jason Fries, and Nigam Shah. EHRSHOT: An EHR Benchmark for Few-Shot Evaluation of Foundation Models. *Advances in Neural Information Processing Systems*, 36:67125–67137, December 2023.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita-Velez, Yue Liao, Hongru Wang, Mengyue Yang, Heng Ji, Jun Wang, Shuicheng Yan, Philip Torr, and Lei Bai. The Landscape of Agentic Reinforcement Learning for LLMs: A Survey, April 2026.

APPENDIX

Appendix A describes how the EHR cohorts and the laboratory action space are constructed. Appendix B specifies evidence release, request matching, clinical calculators, and prompts in EHRTEMP. Appendix C details the DDx annotator, and Appendices D and E report the supervised and reinforcement-learning procedures. Appendix G defines the evaluation protocol, and Appendix H presents additional results.

A EHR DATA CONSTRUCTION

A.1 CLINICAL SCOPE AND DIAGNOSIS LABEL SPACE

We focus on three ICD-10-CM chapters: endocrine, nutritional, and metabolic diseases; diseases of the circulatory system; and diseases of the respiratory system. These chapters include conditions that can be assessed using the clinical findings and laboratory measurements available in structured EHRs. In consultation with a physician, we selected 88 three-character ICD-10-CM categories considered suitable for this diagnostic setting. After applying the encounter-level eligibility criteria described in Appendix A.5, 82 of these categories are represented in the retained training, validation, and test sets combined. The remaining six categories have no retained encounters; they are not individually excluded by a disease-specific rule. The DDx annotator operates over the initial 88-category vocabulary, while agent training and evaluation use the 82 represented categories. This reduction precedes the annotator-based filtering applied only to SFT training data.

A.2 ENCOUNTER CONSTRUCTION

An encounter corresponds to a single clinical visit, with the clinical day as the unit of time. For EHRSHOT, we construct encounters by grouping consecutive record dates separated by no more than five days; a gap exceeding five days starts a new encounter. For MIMIC-IV, we use the encounter boundaries provided in the source data rather than applying this gap-based segmentation. Encounters are retained when all of the following conditions hold:

- the recorded diagnosis falls within the retained 88 label space (Appendix A.1);
- the encounter contains at least one symptom record or clinical observation;
- the encounter contains at most $1{,}000$ clinical records.

Medication and device-usage records are excluded from agent-observable input. These record types correlate strongly with the final diagnosis and would allow the agent to bypass evidence-based assessment. Each encounter additionally includes up to 15 of the patient's most recent prior diagnoses as clinical history, which provides relevant context while bounding prompt length.

A.3 CODING-SYSTEM NORMALIZATION

Diagnosis-related concepts are mapped to ICD-10-CM categories using SNOMED CT mapping resources (IHTSDO, 2024), and laboratory concepts are mapped to LOINC-based actions using UMLS terminology mappings where available (National Library of Medicine, 2017). Laboratory normalization is restricted to the high-frequency concepts that together account for 95% of laboratory events. The remaining long-tail records are kept as standalone observations and handled as described in Appendix A.4.

A.4 LABORATORY ACTION–OBSERVATION HIERARCHY

Hierarchy. LOINC distinguishes orderable laboratory concepts from component-level measurements. A concept can be an *action*, corresponding to an orderable test or panel; a *pure observation*, corresponding to a component reported within a larger order; or *both*, corresponding to a test that can be ordered on its own or reported as a panel component. This hierarchy allows EHRTEMP-AGENT to request tests at the level of clinical orders while the environment returns component-level values as observations.

Table 3: Filtering criteria for constructing diagnostic cases.

Criterion	Threshold	Split
`lab_actions_total`	≥ 3	Multi, Single
`lab_actions_max`	≤ 20 / ≤ 15	Multi / Single
`missing_obs_rate`	≤ 0.5	Multi, Single
`calcs_max`	≤ 10	Multi, Single
`true_target_count`	$[1, 5]$	All
`standalone_lab_mean`	> 0	Multi, Single
`turns_missing_ddx`	$= 0$	Multi, Single
`token_count`	< 7000	All
`turn_total`	$[2, 5]$	Multi

Table 4: Encounter counts used for training and evaluation. SFT and RL use potentially overlapping subsets of the training partition; neither shares patients with validation or test. For SFT, counts outside parentheses denote distinct encounters, while counts inside parentheses include repetitions introduced by oversampling. Validation and test counts are reported after subsampling.

Encounter type	SFT	RL	Validation	Test
Multi-turn	464 (568)	1,555	137	565
Single-turn	674 (807)	1,907	293	829
Direct-diagnosis	1,097 (1,261)	0	233	704
Total	2,235 (2,636)	3,462	663	2,098

Aggregation rules. Raw EHR laboratory records are typically stored at the component level even when a clinician ordered a panel, so we aggregate components into panel-level actions for each patient-day. Under **Rule 1**, when a parent panel and its components co-occur on the same clinical day, the panel becomes the executable action and the components are attached to it as observations. Under **Rule 2**, when multiple components of the same parent panel appear on the same day without the panel itself being recorded, the panel action is inferred. Because aggregation operates per patient-day, orders of the same panel on different clinical days remain distinct actions.

Standalone observations. Some laboratory records cannot be attached to an orderable action. These include observations without a parent panel, low-frequency tests outside the action vocabulary, and a small fraction of records whose CPT-4 or SNOMED CT codes cannot be reliably translated into the LOINC hierarchy. We mask such standalone records rather than exposing them without a request, so that every laboratory value surfaced to the agent is tied to an ordering action. In a sensitivity check with the strongest baseline: DeepSeek-V4-Flash on the full ID test set, revealing these records decreased Sample F1 from 0.3967 to 0.3878 ($\Delta = -0.0089$); the difference was not statistically significant under a paired t-test.

Executable action space. EHRSHOT contains more than $3{,}000$ distinct laboratory event types. After LOINC-based aggregation, frequency-based action selection, and removal of non-orderable observations, the executable space comprises approximately 260 orderable tests. At each turn the agent generates its request over this full space, and no candidate shortlist is provided.

A.5 Trajectory Filtering and Data Splits

After constructing the initial trajectories, we apply the trajectory-level quality criteria in Table 3. We then partition the EHRSHOT data by patient into training, validation, and test sets, targeting an approximately 7:1:2 ratio using the balanced partitioning procedure A.5. Encounters from the same patient remain in one partition.

From the training partition, we construct separate subsets for supervised initialization and reinforcement learning. For SFT, we retain encounters with at least one recorded diagnosis among the annotator's top three DDx predictions at the final decision point. For multi-turn encounters, that

Table 5: Out-of-distribution performance on MIMIC-IV. Diagnosis metrics evaluate final diagnostic predictions; behavior metrics measure laboratory test–day agreement and diagnostic completion. Sample F1 is the primary metric. Lab–Day F1 is micro-averaged. Best results are shown in bold.

	Diagnosis				Behavior	
Method	Sample F1	Micro F1	Precision	Recall@5	Lab–Day F1	Completion
General-purpose LLMs						
GPT-5.4-mini	0.186	0.240	0.193	0.221	0.040	0.835
Claude Haiku 4.5	0.303	0.335	0.286	**0.382**	0.021	0.961
DeepSeek-V4-Flash	**0.362**	**0.406**	**0.419**	0.373	0.037	**0.989**
Qwen3-30B	0.205	0.267	0.226	0.226	0.018	0.767
Medical LLMs and agents						
EHR-R1-8B	0.227	0.238	0.198	0.329	0.009	0.976
Baichuan-M2-32B	0.264	0.322	0.244	0.351	0.023	0.751
HuatuoGPT-3-8B	0.161	0.196	0.170	0.189	0.001	0.811
DoctorAgent-8B	0.056	0.075	0.076	0.052	0.007	0.421
Qwen3-8B and task-trained variants						
Qwen3-8B (zero-shot)	0.029	0.064	0.034	0.030	0.001	0.087
Qwen3-8B (few-shot)	0.066	0.125	0.074	0.068	0.006	0.163
DDx-guided SFT	0.299	0.323	0.390	0.288	**0.148**	0.911
EHRTemp-Agent-8B	0.315	0.341	0.400	0.311	0.142	0.916

diagnosis must also appear in the predictions at at least one intermediate decision point, regardless of rank. We additionally exclude encounters in which more than half of the laboratory actions across all decision turns have no corresponding observation with a recorded value.

We rebalance the retained SFT pool according to diagnosis frequency. Let $f(d)$ be the number of encounters containing diagnosis d in this pool, and let $f_{\min}$ and $f_{\max}$ denote the minimum and maximum frequencies among an encounter's diagnoses. If $f_{\min} \leq 15$, the encounter is included $\lceil 15/f_{\min} \rceil$ times. Otherwise, if all its diagnoses have frequencies greater than 200, it is retained with probability $200/f_{\max}$; all remaining encounters are included once. For RL, we exclude direct-diagnosis encounters to focus training on laboratory evidence acquisition, retaining all multi-turn encounters and a selected subset of single-turn encounters.

For evaluation, we subsample the validation and test set to control its size while retaining all three encounter types, with details listed in A.5. Selection uses recorded diagnoses and encounter structure, independently of model predictions, annotator outputs, or evaluation results. The resulting cohorts are fixed across models, and their sizes are reported in Table 4.

Patient-level balanced partitioning. We construct patient-disjoint partitions using the greedy procedure in Algorithm 1. Each patient is represented by a vector counting occurrences of each diagnosis category across their encounters. Given a selection fraction ρ, the procedure sets a target patient count of $\mathrm{round}(\rho|P|)$ and per-diagnosis targets by scaling the candidate pool's diagnosis counts by ρ and rounding elementwise. Patients are selected sequentially to maximize their contribution to unmet diagnosis-count targets. Ties are resolved first by smaller total overshoot beyond the targets, then by larger total diagnosis counts, and finally by a seeded random order. We use $\rho = 0.2$ to select test patients, followed by $\rho = 7/8$ to select training patients from the remainder; the remaining patients form the validation set. The random seed is 42.

Patient-level subsampling. For the held-out subsampling described above, we reuse the greedy selection procedure with seed 42. Within each selected encounter-type bucket, a nominal encounter target N and the existing encounter count E determine the selection fraction $\rho = N/E$. Patient diagnosis counts are computed within that bucket, and all bucket encounters belonging to selected patients are retained. The test-set targets are 500, 750, and 500 for multi-turn, single-turn, and direct-diagnosis encounters, respectively. For validation, only the direct-diagnosis bucket is subsampled, with $N = 160$. Because the procedure selects $\mathrm{round}(\rho|P|)$ patients, the resulting encounter count need not equal N.

Algorithm 1 Patient-Level Greedy Sampling

Input: patient set $\mathcal{P}$; encounters E_u of each patient u; selection ratio $\rho \in (0, 1]$ **or** target encounter count T
Output: selected patients S; remaining patients R

Let $\mathcal{K}$ be the diagnosis labels occurring in $\mathcal{P}$, $K = |\mathcal{K}|$, and $p_{u,k}$ the number of encounters in E_u carrying label k.
if T is given **then**
  $\rho \leftarrow \min\big(T / \sum_{u \in \mathcal{P}} |E_u|,\, 1\big)$
end if
$\mathbf{m} \leftarrow \sum_{u \in \mathcal{P}} \mathbf{p}_u$; $\mathbf{m}^* \leftarrow \mathrm{round}(\rho\, \mathbf{m})$; $n^* \leftarrow \mathrm{round}(\rho\, |\mathcal{P}|)$
$S \leftarrow \emptyset$; $\mathbf{a} \leftarrow \mathbf{0}$; $R \leftarrow \mathcal{P}$
while $R \neq \emptyset$ and $|S| < n^*$ **do**
  $\mathbf{q} \leftarrow \max(\mathbf{m}^* - \mathbf{a}, \mathbf{0})$
  for each $u \in R$ in a freshly shuffled order **do**
    $\mathrm{gain}(u) \leftarrow \sum_{k=1}^{K} \min(q_k, p_{u,k})$
    $\mathrm{overshoot}(u) \leftarrow \sum_{k=1}^{K} \max(a_k + p_{u,k} - m_k^*, 0)$
    $\mathrm{total}(u) \leftarrow \sum_{k=1}^{K} p_{u,k}$
  end for
  Select u^* maximizing $\big(\mathrm{gain}(u), -\mathrm{overshoot}(u), \mathrm{total}(u)\big)$ lexicographically; remaining ties go to the patient that comes first in the shuffled order.
  $S \leftarrow S \cup \{u^*\}$; $\mathbf{a} \leftarrow \mathbf{a} + \mathbf{p}_{u^*}$; $R \leftarrow R \setminus \{u^*\}$
end while
return (S, R)

A.6 Out-of-Distribution Cohort

The MIMIC-IV (Johnson et al., 2023) evaluation cohort covers the same 82 diagnosis categories, and its laboratory records are aggregated with Rules 1 and 2 (Appendix A.4). Because MIMIC-IV laboratory records do not identify the ordered panel, two further operations are applied. When a panel can be identified from the measurements recorded on a clinical day, compatible measurements in the *both* category are also grouped under that panel. Under **Rule 3**, when more than three components of an in-distribution laboratory action appear on the same day, they are aggregated under that action. These operations change the granularity of laboratory actions while preserving the underlying measurements and their timestamps, so the evidence available to agents is unchanged. All models interact with the same processed environment, and a request retrieves measurements whenever it matches either the action or the observation terminology under the evaluation matcher (Appendix B.2). Complete results on the MIMIC-IV evaluation cohort are reported in Table 5.

B Environment Interface

B.1 Evidence Release

EHRTemp never generates laboratory values absent from the underlying record. A request at clinical time τ_t returns, at the next decision point, the measurements of each requested test that were recorded at or before τ_t and have not been returned earlier in the rollout; a matched panel returns its component observations. Deduplication operates on individual measurement records, so new measurements of a previously requested test can still be returned. Measurements recorded after τ_t are not returned by that request, although a later request made at or after their recording day can retrieve them. A request matching no recorded laboratory event returns no value. Non-laboratory findings, including newly documented symptoms and re-measured vital signs, enter the observation automatically once the trajectory reaches the day on which they were recorded. Laboratory evidence is therefore action-conditioned, while all released content remains faithful to the recorded timeline.

Each episode allows at most five decision turns, and the prompt instructs the agent to request at most 15 distinct tests per turn. An episode that reaches the turn limit without a final diagnosis is treated as unfinished (Appendix G.1).

Table 6: Laboratory matching protocols used in different experimental stages. Action-level quantities, such as RL rewards and lab-test precision/recall/F1, are matched against the executable lab-test action space. Reveal-related quantities are matched against the combined action–observation space, since ordered panels may reveal component-level laboratory observations. During training, we use rule-based fuzzy matching only; during evaluation, we additionally enable SapBERT matching for fair comparison with prompting-based baselines. *Our trained checkpoints interact with the in-distribution environment under fuzzy matching only.

Stage	Matching Rule	Matching Scope
ID Env training	Fuzzy	Action + observation
ID reward training	Fuzzy	Action
ID Env evaluation	Fuzzy + SapBERT*	Action + observation
OOD Env evaluation	Fuzzy + SapBERT	Action + observation
Metrics: lab Pre/Rec/F1	Fuzzy + SapBERT	Action

B.2 Laboratory Request Matching

Agents express requests as free text, so each requested name is mapped to the laboratory vocabulary before evidence release, reward computation, and evaluation. The matching scope depends on the quantity being computed (Table 6). Action-level quantities, including laboratory metrics and the process reward, are matched against the executable action space, whereas reveal-related quantities are matched against the combined action–observation space because a requested panel can expose component-level measurements.

Matching is deliberately stricter during reinforcement learning, and for our trained checkpoints on the in-distribution test set, than in the other evaluation runs. During training, requests are resolved against the standardized LOINC-derived actions by the rule-based fuzzy matcher alone, so a request earns process reward only when its name resolves without embedding similarity, which encourages standardized action names. During evaluation, SapBERT embedding similarity (Liu et al., 2021) serves as a fallback for requests that fuzzy matching fails to resolve, recovering clinically valid surface variants produced by prompting-based baselines. Because this fallback only adds matches that fuzzy matching misses and never removes or replaces one, enabling it uniformly relaxes matching; our trained checkpoints, which interact with the in-distribution environment under fuzzy matching alone, are thus evaluated under the stricter setting.

B.3 Clinical Calculators

The system prompt exposes 17 clinical calculators together with their required inputs (Table 7). All calculator implementations are taken from MedCalc-Bench (Khandekar et al., 2024) without modifying their formulas; two entries return several related quantities (Ideal-and-Adjusted-Body-Weight returns both ideal and adjusted body weight, and AnionGap additionally returns the albumin-corrected anion gap, delta gap, and delta ratio when albumin is supplied), so the 17 entries are backed by 21 MedCalc-Bench implementations. The agent is instructed to request a calculator only when all of its inputs are available and the calculator is relevant to the current decision, and to repeat a calculation only after one of its inputs has changed. Calculator results are computed by the environment, not reported by the model: the agent names the calculator and copies the input values and units from the record, generation pauses at the end of the calculator block, and the environment parses the arguments (averaging repeated measurements, filling missing units and optional inputs with registry defaults, e.g., QTc uses the Bazett formula), executes the implementation, and writes the result into the `Calculator Results` field before the agent continues its turn.

B.4 Prompt Templates

All agents are evaluated on the same recorded encounters. Their observations depend on their requests and the matching configuration described in Appendix B.2. The invocation formats are specified below. Table 8 lists which prompt each model receives.

Table 7: Clinical calculator outputs and required input variables used in EHRTEMP.

Calculator output	**Required inputs**
BMI (kg/m^2)	Weight, height
Creatinine clearance (CrCl; mL/min)	Sex, age, weight, height, creatinine
eGFR (mL/min/1.73 m^2)	Sex, age, creatinine
GFR (mL/min/1.73 m^2)	Sex, age, creatinine
Mean arterial pressure (MAP; mmHg)	Systolic blood pressure, diastolic blood pressure
Corrected calcium (mg/dL)	Calcium, albumin, normal albumin
Ideal and adjusted body weight	Sex, height, weight
Corrected QT interval (QTc; ms)	Heart rate, QT interval, QTc formula
FIB-4 score	Age, AST, ALT, platelet count
Child–Pugh score/class	Total bilirubin, albumin, INR, ascites, encephalopathy
MELD-Na score	Creatinine, total bilirubin, INR, sodium, dialysis status
Corrected sodium (mEq/L)	Sodium, glucose
LDL cholesterol (mg/dL)	Total cholesterol, HDL cholesterol, triglycerides
Fractional excretion of sodium (FENa; %)	Serum sodium, serum creatinine, urine sodium, urine creatinine
Anion gap (mEq/L)	Sodium, chloride, bicarbonate, albumin
Serum osmolality (mOsm/kg)	Sodium, BUN, glucose
HOMA-IR	Insulin, glucose

Model	System prompt
EHRTEMP-AGENT (SFT and RL)	Base system prompt (Prompt 1)
Prompting baselines (Qwen3-8B zero-shot, Qwen3-30B-A3B, GPT-5.4-mini, Claude Haiku 4.5, DeepSeek-V4-Flash)	Base prompt + diagnosis label space (Prompt 2)
Qwen3-8B few-shot	As above + four worked examples (Prompt 4)
Medical LLMs (HuatuoGPT-3, DoctorAgent-RL, Baichuan-M2)	As the prompting baselines, without calculators (Prompt 3)
EHR-R1	Native-format prompt (Prompts 7–9)

Table 8: System prompt used by each model. All models except EHR-R1 receive the same per-turn user messages (Prompts 5–6).

Base system prompt. Prompt 1 is the system prompt of EHRTEMP-AGENT, used at both training and inference time. It does not enumerate the candidate diagnoses; EHRTEMP-AGENT acquires the label space from its training trajectories.

Prompt 1: System prompt of EHRTEMP-AGENT.

```
You are a clinical decision-making agent. You will receive EHR-format patient observations
↪  over multiple clinical decision turns.

As the interaction proceeds, clinical time advances. The patient's condition and other
↪  clinical observations may evolve over time. You have at most 5 decision turns. You must
↪  provide a final diagnosis no later than the fifth decision turn.

At each turn, follow this workflow:
1. Consider clinically applicable calculators that may support the diagnostic decision,
↪  following the calculator use policy below.
2. List up to 10 distinct ICD-10 category-level differential diagnoses, ranked by diagnostic
↪  confidence from most likely to least likely. Do not repeat diagnoses.
3. Decide whether to recommend additional laboratory tests to collect more evidence or
↪  provide one or more final ICD-10 category-level diagnoses.

Calculator use policy:
- Request a calculator only when all of its required inputs are available and the calculator
↪  is clinically relevant to the current decision.
- Repeat a calculator only when at least one required input has changed since the previous
↪  calculation.
- Newly available laboratory results or observations may enable additional calculators.
- If no calculator is clinically applicable at the current turn, write "None." inside the
↪  <Calculator> block.
```

```
- Do not include commentary about whether calculator-related information is missing or
↪  available.

Available calculators:
{
    calculator.BMI: weight, height;
    calculator.Cockcroft-Gault: sex, age, weight, creatinine, height;
    calculator.2021-CKD-EPI: sex, age, creatinine;
    calculator.Mean-Arterial-Pressure: systolic_bp, diastolic_bp;
    calculator.Corrected-Calcium: calcium, albumin, normal_albumin;
    calculator.MDRD-GFR: sex, age, creatinine;
    calculator.Ideal-and-Adjusted-Body-Weight: sex, height, weight;
    calculator.QTc: heart_rate, qt_interval, QTc_formula;
    calculator.FIB-4: age, AST, ALT, platelet_count;
    calculator.Child-Pugh-Score: total_bilirubin, albumin, INR, ascites, encephalopathy;
    calculator.Corrected-Sodium-Hyperglycemia: sodium, glucose;
    calculator.LDL-Calculated: total_cholesterol, hdl_cholesterol, triglycerides;
    calculator.MELD-Na: dialysis_twice, creatinine, total_bilirubin, INR, sodium;
    calculator.FENa: sodium, creatinine, urine_sodium, urine_creatinine;
    calculator.AnionGap: sodium, chloride, bicarbonate, albumin;
    calculator.serum_osmolality: sodium, bun, glucose;
    calculator.HOMA_IR: insulin, glucose;
}

Clinical decision policy:
- Use only information available up to the current clinical decision turn.
- If the evidence is insufficient for final diagnosis, recommend up to 15 distinct
↪  clinically useful laboratory tests.
- Recommend clinically orderable laboratory tests or panels using standardized LOINC test
↪  names.
- Do not repeat laboratory tests within the same decision turn.
- Laboratory tests recommended in the current decision may have results available at the
↪  next decision turn.
- If no result is currently available for a test requested in the immediately preceding
↪  decision, decide whether to request it again, request alternative laboratory tests, or
↪  proceed to a final diagnosis based on the currently available evidence.
- If the evidence is sufficient, provide one to five distinct final ICD-10 category-level
↪  diagnoses.
- Both premature diagnosis without sufficient evidence and unnecessary delay in diagnosis
↪  carry clinical costs. Balance the value of acquiring additional evidence against the
↪  need to make a timely final diagnosis.

Your response must use the following format:

<Calculator>
None.
</Calculator>

or, if one or more calculators are clinically applicable:

<Calculator>
Requested Calculators:
{calculator.name:
    param1=value (unit), param2=value (unit)}

Calculator Results:
{calculator.name: value (unit)}
</Calculator>

<Differential Diagnosis>
    1. ICD-10 code - Diagnosis name
    2. ICD-10 code - Diagnosis name
</Differential Diagnosis>

<Decision Rationale>
Brief rationale based only on currently available evidence, using no more than 2 sentences.
</Decision Rationale>

<Decision: Lab Test Recommendation>
None.
</Decision: Lab Test Recommendation>

or:

<Decision: Lab Test Recommendation>
    1. Standardized orderable laboratory test or panel name
    2. Standardized orderable laboratory test or panel name
</Decision: Lab Test Recommendation>

<Decision: Final Diagnosis>
Not yet determined.
</Decision: Final Diagnosis>
```

```
or, if sufficient evidence is available:

<Decision: Final Diagnosis>
The final ICD-10 category-level diagnoses are:
    1. ICD-10 code - Diagnosis name
    2. ICD-10 code - Diagnosis name
</Decision: Final Diagnosis>
```

Diagnosis label space. Models not trained on our trajectories are given the candidate set explicitly: the block below is inserted into the clinical decision policy of Prompt 1, immediately before the bullet beginning "Both premature diagnosis". The same 82-code list is used for the OOD evaluation.

Prompt 2: Diagnosis label-space block (prompting and medical-LLM baselines).

```
Diagnosis label space:
All differential diagnoses and final diagnoses must be selected from the following 82 ICD-10
↪  category-level codes. Do not output any diagnosis (in either the differential diagnosis
↪  list or the final diagnosis) outside this set.
E01 - Iodine-deficiency related multinodular (endemic) goiter
E02 - Subclinical iodine-deficiency hypothyroidism
...  [82 lines in total, one "<ICD-10 code> - <name>" per line]
E86 - Volume depletion
```

No-calculator variant. For the medical LLMs, the calculator components are removed from the label-space prompt: workflow step 1, the calculator use policy, the list of available calculators, and the `<Calculator>` block of the output format. The workflow becomes:

Prompt 3: Workflow of the no-calculator variant (remaining text as in Prompts 1–2).

```
At each turn, follow this workflow:
1. List up to 10 distinct ICD-10 category-level differential diagnoses, ranked by diagnostic
↪  confidence from most likely to least likely. Do not repeat diagnoses.
2. Decide whether to recommend additional laboratory tests to collect more evidence or
↪  provide one or more final ICD-10 category-level diagnoses.
```

Few-shot variant. The few-shot baseline appends the following instruction and four worked assistant responses (two final-diagnosis turns and two lab-recommendation turns, each from a different patient) to the label-space prompt.

Prompt 4: Few-shot preamble.

```
The following 4 cases illustrate the required workflow and output format. Note: these
↪  examples are illustrative only. Each example reflects a different patient with different
↪  findings. Do NOT copy specific diagnoses, lab recommendations, or differential diagnosis
↪  rankings from the examples. For the new patient below, derive your answer entirely from
↪  that patient's own observations. The new patient's final diagnosis is unlikely to match
↪  any example case.

Case 1 (multi-turn, ending in final diagnosis):

Assistant:
{complete assistant response in the required output format}

... Cases 2-4 follow the same pattern.
```

Per-turn user messages. At each decision turn the environment renders the newly disclosed observations into the user message. The first turn contains demographics, prior diagnoses, and the initial observations (Prompt 5). Later turns contain the results of the laboratory tests requested in the previous turn together with newly recorded observations (Prompt 6). Braces denote fields filled from the patient record; text after "<−" is an annotation and is not part of the prompt. When an assistant turn requests calculators, generation pauses at `</Calculator>`, the environment writes the computed values into the `Calculator Results` field, and generation resumes (Appendix B.3).

Prompt 5: First-turn user message.

```
[Clinical Day {d}: {d-1} days since initial presentation]
```

```
This is the prior diagnosis of this patient:        <- only if prior diagnoses exist
1. {prior diagnosis name}
...

Here's the demographic & initial observations of the patient:

Demographic Information:
'''
Date | Value | Unit | Name
Day0 | {value} |  | {age / race / gender / ethnicity}
'''

Initial Observations:

{Section}:                 <- Visit Information / Symptoms / Vital Signs /
'''                          Clinical Observations / Procedures (non-empty only)
Date | Value | Unit | Name
Day{k} | {value} | {unit} | {observation name}
...
'''

Available standalone laboratory observations:
'''
Date | Value | Unit | Name
Day{k} |  |  | {lab name}           <- names only; values are not shown
...
'''

Please make the next diagnostic decision using only information available up to this
↪  clinical day. Follow the required output format.
```

Prompt 6: Subsequent-turn user message.

```
[Clinical Day {d}: {d-1} days since initial presentation]

Results for last turn recommended laboratory tests:
'''
1. {requested test name}:
        - Day{k} | {component name}: {value}-{unit}
        ...

2. {requested test name}: no result is currently available

'''

Other observations now include:

{Section}:                 <- newly recorded observations of this turn, same
'''                          table format and sections as in the first turn
Date | Value | Unit | Name
Day{k} | {value} | {unit} | {observation name}
'''

Available standalone laboratory observations:
'''
Date | Value | Unit | Name
Day{k} |  |  | {lab name}
'''

Please make the next diagnostic decision using only information available up to this
↪  clinical day. Follow the required output format.
```

EHR-R1 native mode. EHR-R1 was trained on EHR-Bench's Markdown event format rather than our structured prompt. To keep it in-distribution, we render the same information it would receive under our protocol into that format (Prompt 7). The system message is EHR-R1's default, "`You are a helpful assistant.`", and no calculators are offered. Each turn is an independent query that contains the complete record up to the current clinical day: observations for days $0, \ldots, t$ plus the results returned for the model's own earlier laboratory requests, with their clinical days. On non-final turns the model chooses between ordering labs and diagnosing (Prompt 8); on the last allowed turn it must diagnose (Prompt 9). As for all other models, standalone laboratory observations are shown as names only. Thinking mode is enabled.

Prompt 7: EHR-R1 record format.

```
## Patient Demographics [None]
```

```
- Age: {age}
- Gender: {gender}
- Medical_History:
  - {prior diagnosis}

## {Visit | Symptoms | Vitalsign | Clinical Observations | Procedure_Occurrence} [Day{k}]
| Item_Name | Valuenum | Valueuom |
| ------ | ------ | ------ |
| {observation name} | {value} | {unit} |

## Standalone Laboratory Observations [Day{k}]
- {lab name}

## Labotary Test Events [available results]
| Day | Item_Name | Valuenum | Valueuom | Ref_Range_Lower | Ref_Range_Upper |
| ------ | ------ | ------ | ------ | ------ | ------ |
| Day{k} | {component name} | {value} | {unit} | nan | nan |
```

Prompt 8: EHR-R1 user message, non-final turn.

```
{EHR Markdown up to the current clinical day}
You are a clinical diagnostic agent managing this patient over multiple decision turns.
↪  Based on all the information above, take EXACTLY ONE action this turn---either order
↪  labs OR give the final diagnosis. You MUST choose only one: output ONE block only, never
↪  both blocks, never neither. Do NOT write both `LABS` and `DIAGNOSIS`.
- To gather more evidence: write a line that is exactly `LABS`, then list up to 15 laboratory
↪  tests to order next, one per line---preferably using standardized LOINC test names where
↪  you can, but plain test names are acceptable.
- To make the diagnosis (only when the evidence is already sufficient): write a line that is
↪  exactly `DIAGNOSIS`, then list the one to five most likely diagnoses, one per line, most
↪  likely first. Write each diagnosis EXACTLY as `<ICD-10 code> - <name>`, copying both the
↪  code and the name verbatim from the candidate list. Do not invent codes or names.
You should choose the item from the candidate list below. Candidate List: {82 candidates,
↪  each "<ICD-10 code> - <name>", comma-separated}.
For the diagnosis option, copy each diagnosis EXACTLY as `<ICD-10 code> - <name>` from the
↪  candidate list above (both code and name).
```

Prompt 9: EHR-R1 user message, final turn.

```
{EHR Markdown up to the current clinical day}
You are a clinical diagnostic agent. Based on all the patient information above, provide
↪  your final diagnosis. If the evidence is sufficient, give the one to five most likely
↪  ICD-10 category-level diagnoses, ranked from most to least likely. Select from the
↪  candidate list below.
You should choose the item from the candidate list below. Candidate List: {82 candidates,
↪  each "<ICD-10 code> - <name>", comma-separated}.
Output ONLY the final diagnoses, AT MOST FIVE, one per line, most likely first. Write each
↪  EXACTLY as `<ICD-10 code> - <name>`, copying both the code and the name verbatim from the
↪  candidate list. Do not output reasoning or any other text.
```

C DDx Annotator

C.1 Motivation

Supervising the DDx-guided workflow requires, at each decision point, the observed evidence, the requested tests, and the differential diagnosis, and at termination the final diagnosis. Longitudinal EHRs record all of these except the intermediate differential, which in clinical practice remains an unrecorded judgment of the physician. We therefore approximate it with an annotator that produces a differential diagnosis from the evidence available at any point along the trajectory. The resulting labels serve as reference diagnostic assessments for supervision and do not reconstruct the differential held by the treating physician.

C.2 Knowledge Sources and Training

Disease-level DDx mapping. We first construct a differential-diagnosis mapping for the 88 categories. We query the PubMed Central (PMC) open-access subset (Roberts, 2001) for documents discussing the differential diagnosis of each target disease, collecting 23,066 unique documents that together cover all targets. Text- and table-parsing scripts extract the paragraphs, sections, and tables describing differentials, which we aggregate into a per-disease list, retaining the five most

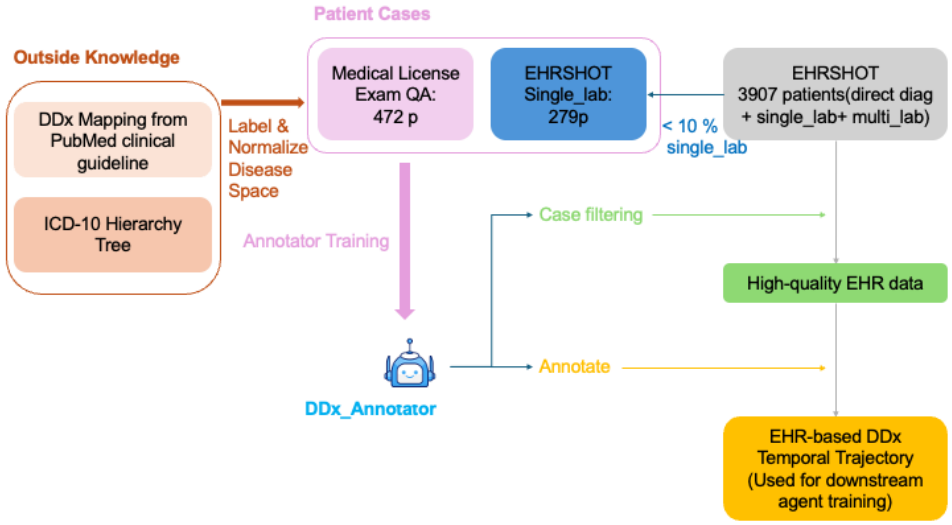


Figure 3: Overview of the DDx annotator. **Training (left):** the annotator learns from differential-diagnosis mappings extracted from PMC clinical literature and normalized through the ICD-10-CM hierarchy, together with patient cases from medical QA datasets (472 cases) and single-lab EHRSHOT encounters (1,806 encounters from 279 patients). **Application (right):** the trained annotator is applied offline to the EHRSHOT cohort (3,907 patients; 35,637 encounters) to annotate per-decision-point differentials for supervised trajectories and to filter the supervised training subset.

Table 9: Training configuration and dataset statistics for the ICD-10 classification model.

Item	Description
Training Setting	
Base model	BioClinical ModernBERT
Model head	ModernBertForSequenceClassification
Task	Multi-label ICD-10 classification, 88 codes
Loss	BCEWithLogitsLoss with class-frequency positive weights
Optimizer	AdamW
Learning rate	2×10^{-5}
LR scheduler	Cosine
Warmup ratio	0.1
Weight decay	0.01
Batch size	16
Epochs	5
Max input length	1024 tokens
Dataset	
QA patients	629 samples from MedMCQA, MedQA, and MedBullets; 472 unique patients
EHR patients	3,612 samples from EHRSHOT; 279 unique patients
Total training samples	4,241
Validation / Test	ID: single-lab-turn EHR encounter; OOD: multi-lab-turn EHR encounter

frequently mentioned differentials when more are available. The ICD-10-CM hierarchy normalizes naming across sources: fine-grained diagnoses are aggregated to their three-character parent code (e.g., I10), and a higher-level code inherits the differentials of its children.

Patient cases. We collect patient-style cases from medical QA datasets, namely MedQA (Jin et al., 2020), MedMCQA (Pal et al., 2022), and Medbullets (Chen et al., 2025), retaining 472 cases whose diagnosis falls within the 88 categories or their parent codes. To familiarize the annotator with

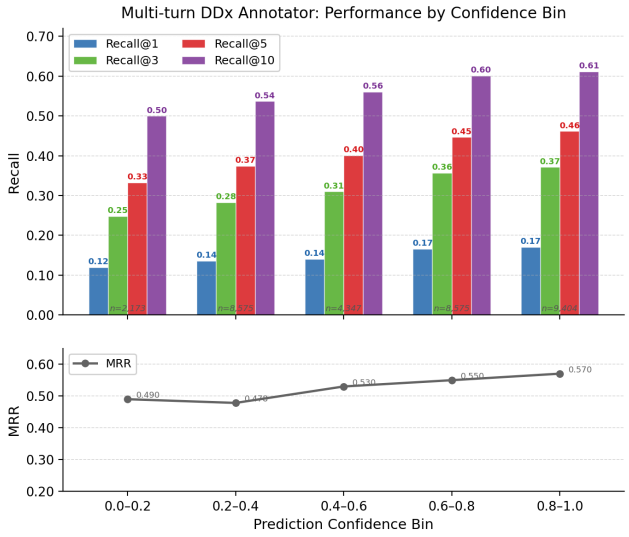


Figure 4: DDx annotator performance on multi-lab-turn encounters along the trajectory. Each decision point is binned by its relative position in the encounter, computed as the current turn divided by the total number of turns; the axis label "Prediction Confidence Bin" denotes this relative position. Recall@k and MRR generally increase toward later positions as clinical evidence accumulates.

tabular EHR input, we add 1,806 single-lab EHRSHOT encounters from 279 patients (Wornow et al., 2023), i.e., 10% of the available patients, to avoid distributional dominance.

Model. The annotator is a BioClinical ModernBERT (Sounack et al., 2025) encoder whose token representations are mean-pooled into a sequence representation and passed to a linear classification head. It is trained in a multi-task setting to predict both the differential diagnosis, supervised by the DDx mapping, and the final diagnosis. Both tasks are formulated as multi-label classification with binary cross-entropy over the candidate ICD-10-CM codes, and differentials are obtained by ranking the predicted probabilities. Training hyperparameters are listed in Table 9.

C.3 Reliability Evaluation

We evaluate the annotator on held-out EHRSHOT encounters. Because it is trained only on single-lab-turn encounters, encounters requiring multiple laboratory turns before the final diagnosis lie outside its training distribution and test whether its output tracks evidence revealed over successive decision points. The annotator attains recall@5 of 0.512 on held-out encounters of the training types and 0.485 on multi-lab-turn encounters. Figure 4 shows that performance increases as more evidence is revealed along the trajectory, consistent with the annotator responding to accumulating evidence rather than relying on a fixed prior.

C.4 Use in Supervised Initialization

The annotator is used only during offline data preparation and is never invoked during agent training or inference; the trained agent generates its own DDx lists. It serves two purposes.

Annotation. For each decision point of a training trajectory, the annotator generates the DDx list from the evidence exposed up to that point, forming the DDx component of the supervised targets (Section 3.2.1).

Filtering and resampling. Supervised fine-tuning uses a filtered subset of the training set. An encounter is retained when (i) at least one recorded final diagnosis appears among the top three candidates of the annotated DDx at its final decision point and, for multi-turn encounters, also appears (at any rank) in the annotated DDx of at least one intermediate decision point; and (ii) at most 50%

of the laboratory actions in its reference trajectory lack results, where the denominator is the number of laboratory orders (individual tests or panels) across the encounter's non-final decision points and an order counts as missing when none of its component observations has a recorded value; encounters without laboratory orders are not affected by this criterion. Encounters without a recorded final diagnosis are excluded. This criterion selects trajectories whose final annotated assessment agrees with a recorded diagnosis; it does not establish that excluded diagnoses are unrecoverable from the EHR. Filtering shifts the class distribution toward presentations the annotator recognizes, so we rebalance at the encounter level using per-category encounter frequencies f in the filtered pool: an encounter whose rarest diagnosis has $f_{\min} \leq 15$ is replicated $\lceil 15/f_{\min} \rceil$ times, and an encounter whose diagnoses all have $f > 200$ is kept with probability $200/f_{\max}$ (seed 42). This reduces 6,306 filtered encounters to 2,235 unique encounters (2,636 after replication) and lowers the share of the most frequent category (I10) from 64.5% to 45.7%. Filtering applies to the training set only. Validation and test sets are unfiltered, so all models are evaluated on identical cases, and reinforcement learning uses a broader training cohort (all multi-turn training encounters plus a disease-balanced half of single-turn training patients) that includes encounters excluded by this filter. Dataset counts are given in Table 4.

D Supervised Initialization Details

We fine-tune Qwen3-8B on the 2,235 unique annotated encounters selected in Appendix C.4 (2,636 after oversampling), decomposed into $4{,}752$ per-decision-point examples whose input is (c_{t-1}, o_t) and whose target is the DDx list followed by the recorded action. We use LoRA with rank $r = 16$, scaling factor $\alpha_{\text{LoRA}} = 32$, and dropout 0.05. Training uses AdamW with a learning rate of 2×10^{-4}, cosine decay, a warmup ratio of 0.05, and weight decay 0.01, with an effective batch size of 16 and a maximum sequence length of $8{,}192$ tokens, on two A100 GPUs with bf16 precision and gradient checkpointing. The checkpoint with the lowest validation loss (step 425) initializes reinforcement learning. The no-DDx variant used in the ablation is trained on the same encounters with DDx lists removed from the targets, under the same configuration and random seed of 42.

E Reinforcement Learning Details

E.1 Rewards and Response Classification

For a valid laboratory request, each requested test receives $\alpha = 0.15$ if it matches a clinician order recorded at or before the request time that has not been credited earlier in the rollout, and $-\kappa = -0.05$ otherwise. Matching follows EHRTemp's observation logic, using the same name-matching procedure and time-based availability rules, but is evaluated against recorded laboratory orders rather than the values returned by the environment. Consequently, an order without a recorded result can still receive credit: recommending that test at the same or any later eligible time is rewarded until the order has been credited. The penalty therefore applies when a test was never ordered in the encounter, when all matching orders occur after the request time, when all eligible orders have already been credited, or when the test is repeated within the same turn. A test receives a single positive reward regardless of how many eligible orders it matches. The outcome reward is set-level F1 plus an exact-match bonus of 0.3. A malformed turn receives zero reward regardless of its type, so it neither earns process credit nor contributes a diagnosis to the outcome reward.

A response is classified as FINAL when the parser identifies a terminal-diagnosis structure and as INTERMEDIATE otherwise; classification itself excludes no responses. Malformed INTERMEDIATE responses remain in the process-reward pool with zero reward. A FINAL response with an empty, unparseable, or format-invalid diagnosis receives zero outcome reward and remains in the FINAL pool. A rollout that reaches the turn limit without a FINAL response contributes no sample to that pool rather than a zero-outcome sample, while its INTERMEDIATE responses still enter their process-reward groups.

E.2 Pool Construction and Minimum Pool Size

Reward pools are constructed separately for each encounter and response type. The INTERMEDIATE pool contains the intermediate responses of the encounter's sampled rollouts across decision

turns, within which BT-PAE groups responses by turn to compute turn-specific means. The FINAL pool contains the terminal responses of the sampled rollouts.

A pool with fewer than four responses is pruned: its advantages are set to zero and its response tokens are masked out of the training loss. This threshold applies to the entire encounter-level pool of a response type rather than to each decision-turn group, so a retained INTERMEDIATE pool can contain turns represented by a single rollout. A pruned FINAL pool injects no outcome advantage into intermediate responses, and rollouts without a FINAL response likewise receive no outcome injection. Both cases omit the outcome contribution rather than adding artificial zero-reward samples to the FINAL pool.

E.3 Pooled Variance and Degenerate Groups

For a retained INTERMEDIATE pool, let $\mathcal{G}_t$ denote its samples at decision turn t, $n_t = |\mathcal{G}_t|$, and $\mathcal{T} = \{t : n_t > 0\}$. The turn-specific means are

$$\bar{r}_t = \frac{1}{n_t} \sum_{i \in \mathcal{G}_t} r_t^{\mathrm{lab},(i)}. \tag{17}$$

The within-turn residual sum of squares and residual degrees of freedom are

$$\mathrm{SS} = \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{G}_t} \left(r_t^{\mathrm{lab},(i)} - \bar{r}_t \right)^2, \qquad \nu = \sum_{\substack{t \in \mathcal{T} \\ n_t > 1}} (n_t - 1), \tag{18}$$

and the pooled scale and process advantages are

$$\sigma_{\mathrm{pool}} = \begin{cases} \sqrt{\mathrm{SS}/\nu}, & \nu > 0, \\ 0, & \nu = 0, \end{cases} \qquad A_{t,\mathrm{proc}}^{(i)} = \frac{r_t^{\mathrm{lab},(i)} - \bar{r}_t}{\sigma_{\mathrm{pool}} + \varepsilon_{\mathrm{num}}}, \tag{19}$$

where $\varepsilon_{\mathrm{num}} = 10^{-4}$. Singleton turn groups contribute zero to both SS and ν, so summing $(n_t - 1)$ over all nonempty groups yields the same degrees of freedom.

If $\nu = 0$, every nonempty turn group is a singleton, each reward equals its turn-specific mean, and all process advantages are zero. If $\nu > 0$ but $\sigma_{\mathrm{pool}} = 0$, the residual sum of squares is zero, which again implies zero centered rewards throughout the pool. The positive numerical constant keeps the ratio well defined in both cases. A zero process advantage removes the process-reward contribution to the clipped objective without necessarily implying a zero gradient, since a retained response can still receive outcome feedback and KL regularization. Responses in pruned pools are instead masked out of the loss.

E.4 Outcome Advantages and Injection

Let $\mathcal{F}$ denote the FINAL pool of an encounter. For a retained pool, outcome advantages are computed as

$$A_{\mathrm{out}}^{(i)} = \frac{f^{(i)} - \bar{f}}{\max(\sigma_f, \delta_f) + \varepsilon_{\mathrm{num}}}, \qquad i \in \mathcal{F}, \tag{20}$$

where $f^{(i)}$ is the diagnostic outcome reward, $\bar{f}$ and σ_f are the FINAL-pool mean and standard deviation, and $\delta_f = 0.05$. The standard-deviation floor limits the scale of normalized differences when terminal rewards are nearly identical, which would otherwise turn negligible reward differences into large advantages broadcast to every intermediate turn of the rollout.

For retained INTERMEDIATE responses, the outcome advantage of the corresponding rollout is added with weight $\lambda = 1.0$, and FINAL responses use their outcome advantage directly. Outcome injection is omitted when the rollout has no FINAL response or its FINAL pool is pruned, and pool masks remain in effect during optimization.

E.5 Reward Statistics across Decision Turns

Turn-specific centering and pooled scale estimation address different properties of the process-reward distribution, and Table 10 illustrates both. The mean process reward decreases from $+0.673$

Table 10: Process-reward statistics of INTERMEDIATE responses by decision turn, collected during training of the main run (one epoch, 216 steps, six rollouts per encounter). Statistics are computed on rewards and include all intermediate responses, including those in pools pruned from the policy loss (fewer than four intermediate responses in the group). Within-group SD is the pooled standard deviation across responses at the same turn within the same rollout group; groups with a single response at a turn contribute to the count and mean but not to the SD. The fifth turn ($n = 42$), reached only by rollouts that had not committed earlier, is omitted from this table.

Turn	Responses	Mean reward	Within-group SD
1	4,265	+0.673	0.170
2	1,713	+0.212	0.201
3	531	+0.042	0.162
4	128	−0.059	0.128

at the first turn to -0.059 at the fourth, consistent with eligible recorded orders being credited at earlier turns. Centering with a single mean pooled across turns would therefore credit turn position as advantage, a confound that turn-specific centering removes. The number of responses falls from $4{,}265$ to 128 while the within-group standard deviation remains between 0.13 and 0.20. Per-turn scale estimates thus become unreliable at later turns, whereas the pooled estimate draws on residuals from all turns. The estimator treats within-turn dispersion as approximately shared across clinical stages; these aggregate statistics support this approximation but do not establish equal variances within individual encounter groups.

E.6 Token-Level Policy Optimization

Let $z_t^{(i)}$ denote the generated response at turn t of rollout i, comprising the DDx list and the clinical action. For its k-th token, the importance ratio is

$$\rho_{t,k}^{(i)}(\theta) = \frac{\pi_\theta\left(z_{t,k}^{(i)} \mid c_{t-1}^{(i)}, o_t^{(i)}, z_{t,<k}^{(i)}\right)}{\pi_{\theta_{\text{old}}}\left(z_{t,k}^{(i)} \mid c_{t-1}^{(i)}, o_t^{(i)}, z_{t,<k}^{(i)}\right)}. \tag{21}$$

The combined turn advantage $A_t^{(i)}$ is assigned to the optimized tokens of that response, and each retained token contributes

$$\min\left(\rho_{t,k}^{(i)}(\theta)A_t^{(i)}, \text{clip}\left(\rho_{t,k}^{(i)}(\theta), 1-\epsilon, 1+\epsilon\right) A_t^{(i)}\right) - \beta\mathcal{K}_{t,k}^{(i)}(\theta). \tag{22}$$

Turns in pruned pools are removed from the optimization mask, and episodes longer than $8{,}192$ tokens are right-truncated. The KL penalty uses the sampled-token k3 form, $\mathcal{K}_{t,k}^{(i)} = \frac{\pi_{\text{ref}}}{\pi_\theta} - \log\frac{\pi_{\text{ref}}}{\pi_\theta} - 1$, with both probabilities evaluated at the same token context; π_{ref} is the frozen SFT policy, $\beta = 0.1$, and no KL clipping is applied. The loss is first averaged over the retained tokens of each rollout, jointly across all turns, so turns are weighted by their token counts. It is then averaged uniformly over the 96 rollouts of a training step (16 encounters $\times$ 6 rollouts). Each batch of rollouts is used for a single gradient step, with clipping parameter $\epsilon = 0.2$.

E.7 Training Configuration

The main method uses turn-specific process-reward means with the ANOVA pooled within-turn standard deviation. Table 11 summarizes the RL configuration. We train for one epoch and evaluate the final checkpoint.

E.8 Configurations of the RL Ablation Variants

All four RL variants start from the DDx-guided SFT checkpoint and share the optimization, batching, rollout, and KL settings in Table 11, including six rollouts per encounter and one training epoch of 216 steps. These variants differ only in how rewards are formed and converted into token-level advantages, as summarized in Table 12.

Table 11: Configuration of the main RL run: reward and advantage estimation, optimization, batching, and rollout generation.

Parameter	Value
Reward	
Positive laboratory reward α	0.15
Laboratory penalty magnitude κ	0.05
Diagnostic exact-match bonus	0.3
Advantage estimation	
Rollouts per encounter G	6
Outcome-injection weight λ	1.0
Minimum pool size	4
Turn-specific process centering	Enabled
Pooled within-turn standard deviation	Enabled
Outcome standard-deviation floor δ_f	0.05
Numerical constant ε_{num}	10^{-4}
Policy and reference	
Base model	Qwen3-8B (bf16)
Initialization	SFT checkpoint (step 425)
Parameter-efficient tuning	LoRA, $r = 16$, $\alpha_{\text{LoRA}} = 32$, dropout 0.05
Reference policy	Frozen copy of the SFT LoRA adapter
Optimization	
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$), weight decay 0
Learning rate	2×10^{-6}, cosine schedule, 10 warm-up steps
Gradient clipping	max norm 1.0
KL coefficient β	0.1 (k3 estimator)
Clipping parameter ϵ	0.2
Training	1 epoch, 216 steps (3,462 encounters)
Batching	
Encounters / rollouts per step	16 / 96
Micro-batch	1 rollout per GPU, 24 accumulation steps
Rollout generation	
Maximum decision turns	5
Maximum new tokens per turn	1,024 per generation segment (calculator block + remainder)
Maximum episode length (loss)	8,192 tokens
Sampling	temperature 0.7, top-p 0.9
Thinking mode	Disabled
Rollout engine	vLLM (colocated)

Outcome-only GRPO and combined-reward estimation follow standard trajectory-level GRPO. Each rollout receives one scalar reward, which is centered and scaled by the mean and standard deviation of the six rollouts of the same encounter, and the resulting advantage is assigned to every policy-sampled token of that rollout. Combined-reward estimation defines this scalar as the sum of the turn rewards used by BT-PAE, so a malformed turn contributes zero. Outcome-only GRPO uses r^{diag} in Eq. (10) computed from the last well-formed final diagnosis and subtracts 0.5 for each malformed turn, with the total penalty capped at 1.5.

This graded penalty compensates for the absence of turn-level feedback. In the process-reward variants, a malformed turn forfeits its process reward and therefore typically falls below the turn-specific or trajectory reward it would otherwise earn, particularly at early turns where process rewards are predominantly positive (Table 10). Outcome-only training lacks this channel, and a malformed intermediate turn would otherwise share the positive advantage of a correct final diagnosis.

Neither trajectory variant applies the standard-deviation floor δ_f. In BT-PAE, the floor limits the outcome advantage before it is added to separately standardized process advantages, preserving the relative weight of these two terms. Trajectory variants normalize a single combined scalar, so no such balance arises; near-identical rewards within a group can still inflate their advantages, a property of standard GRPO normalization that we retain for both baselines.

Pool pruning and the treatment of unfinished rollouts likewise follow the structure of each estimator. Trajectory variants always normalize over six rollouts, whereas by-turn pools vary in size, so by-turn variants prune pools with fewer than four responses and remove the pruned turns from both

the policy loss and the KL term. A rollout that reaches the turn limit without a final diagnosis receives zero diagnostic reward in the trajectory variants but contributes no FINAL sample under by-turn estimation (Appendix E.1), since a fabricated zero-reward sample would alter the statistics that normalize completed diagnoses.

This asymmetry penalizes non-commitment less directly under by-turn estimation, because an unfinished rollout loses its outcome term instead of receiving a negative one. In practice, completion at the final checkpoint remains above the SFT level of 0.988 for all variants, ranging from 0.991 to 0.996 (Table 2).

Table 12: Reward and advantage configuration of the RL variants in Table 2. All variants share the settings in Table 11.

Variant	Reward	Advantage	Malformed turn	Unfinished rollout	δ_f / min pool
Outcome-only GRPO	r^{diag}	Trajectory	-0.5 per turn (cap 1.5)	$r^{\text{diag}} = 0$	–
Combined-reward	$\sum_t r_t$	Trajectory	0	No diagnostic term	–
Across-turn process	r_t per turn	Pooled across turns $+ \lambda A_{\text{out}}$	0	No FINAL sample	0.05 / 4
BT-PAE	r_t per turn	Turn-specific, Eq. (13) $+ \lambda A_{\text{out}}$	0	No FINAL sample	0.05 / 4

F Experimental Setup

Datasets. We use EHRSHOT (Wornow et al., 2023) for training and in-distribution (ID) evaluation, and the hospital module of MIMIC-IV (Johnson et al., 2023) for out-of-distribution (OOD) evaluation. Both provide de-identified, time-stamped clinical records for constructing DCDM encounters. EHRSHOT supplies longitudinal records for learning and evaluating diagnostic policies, while MIMIC-IV provides an independent institutional source for assessing transfer. We construct encounters using EHRTemp as described in Section 3.1, with patient-disjoint training, validation, and test partitions for EHRSHOT. The OOD cohort uses the same 82 diagnosis categories and a harmonized laboratory action space. Each test set contains 2,098 encounters.

Baselines. We compare against two groups of external models: (1) general-purpose LLMs, comprising GPT-5.4-mini, Claude Haiku 4.5, DeepSeek-V4-Flash, and Qwen3-30B; and (2) medical LLMs and agents, comprising EHR-R1-8B (Liao et al., 2025), Baichuan-M2-32B (Team et al., 2025), HuatuoGPT-3-8B (Chen et al., 2026), and DoctorAgent-RL (Feng et al., 2025). These comparisons assess our trained 8B agent against general-purpose and medically specialized systems, including larger open models and proprietary models. We additionally evaluate zero-shot, few-shot, and supervised-only variants of Qwen3-8B to contextualize the effects of task-specific training. All external baselines are provided with the candidate diagnosis list and evaluated in EHRTemp under the same clinical action interface.

Metrics. **Diagnostic performance** is evaluated using Sample F1 (primary), Micro F1, Precision, and Recall@5 against recorded ICD-10 category sets. Sample F1, Precision, and Recall@5 are averaged across encounters, whereas Micro F1 is computed globally. **Agent behavior** is summarized by Lab–Day F1, which measures agreement with recorded laboratory events as $(\text{test}, \text{clinical day})$ pairs, and Completion Rate, the proportion of encounters ending with a valid diagnosis within the turn budget. Unfinished encounters receive zero diagnostic scores. Detailed definitions are provided in Appendix G.1.

Implementation. We initialize EHRTemp-Agent from Qwen3-8B and apply rank-16 LoRA in bf16 precision during both supervised fine-tuning and reinforcement learning, using two A100 and four A40 GPUs, respectively. The SFT checkpoint is selected by minimum validation loss, and RL uses the final checkpoint after one training epoch. The same checkpoints are used for ID and OOD evaluation without OOD-specific selection. Training configurations are provided in Appendix E, with evaluation protocols and metric definitions in Appendix G. Across environment interaction and evaluation, we normalize laboratory requests to the environment's test vocabulary through keyword-based matching, supplemented by SapBERT at evaluation to accommodate terminology variants and aliases; Appendix B.2 specifies the matching procedure.

G Evaluation Protocol

G.1 Metrics

Diagnostic metrics. Each encounter's final diagnosis set $\hat{y}$ is compared with the reference set y at the three-character ICD-10-CM level. Sample F1, the primary metric, averages the per-encounter set F1 over encounters, precision is averaged in the same way, and micro F1 pools true positives, false positives, and false negatives across encounters. Recall@5 measures the fraction of reference diagnoses that appear among the first five codes of the final diagnosis list, in the order the agent wrote them; because agents are instructed to give one to five final diagnoses, it coincides with set recall unless more than five codes are listed. An encounter that reaches the turn limit without a final diagnosis receives zero on all diagnostic metrics rather than being excluded, since exclusion would favor agents that fail to commit.

Laboratory metrics. Laboratory metrics are computed over encounters with at least one recorded laboratory action (1,394 of the 2,098 test encounters). Requested test names are resolved to recorded laboratory actions by exact, fuzzy, and then SapBERT matching (Appendix B.2).

Lab–Day F1 compares requests and records as sets of (laboratory action, clinical day) pairs. Each distinct request is paired with the clinical day of the turn at which it was made and counts as correct only if the same action was recorded on that same day. Only requests that resolve to a laboratory action are credited; a request that resolves to an individual component earns no credit for its parent panel, and requests made beyond the recorded timeline cannot match. True positives, requests, and recorded pairs are summed over all encounters before precision, recall, and F1 are computed (micro-averaging), so an encounter in which the agent requests nothing still contributes its recorded pairs to recall.

Per-turn precision is the fraction of distinct tests requested at a turn that match a recorded laboratory action. Here, a requested component is credited to its parent action, and each recorded (clinical day, action) entry can be credited only once, so a re-order with no remaining entry counts as a miss; the day of the request is not checked. Per-turn precision is averaged over turns with at least one request and then over encounters.

Non-reorder rate is computed on the multi-turn subset used in Figure 6(a). A case–test pair qualifies when the test is recorded on at least two distinct clinical days in the encounter and the model makes at least one further laboratory-ordering turn after first requesting it; qualification does not require that later turn to reach a subsequent recording day. The non-reorder rate is the fraction of qualifying pairs that the model requests in only one turn. Because qualification depends on each model's own trajectory, both the number n of qualifying pairs and the pairs themselves differ across models, so the rate describes repeated-request behavior rather than serving as a directly comparable performance score.

These metrics quantify agreement with recorded clinical practice and do not establish the clinical optimality of either the record or the agent's requests.

Interaction metrics. Completion rate is the fraction of encounters that end with a parseable final diagnosis within the turn limit.

G.2 Baseline Invocation

All models interact with the same environment through an identical in-prompt action protocol rather than provider-specific function-calling interfaces, and all decisions are scored by the same extraction pipeline and matcher. General-purpose models follow the full task output format, including the DDx list, with reasoning disabled, greedy decoding, and a budget of 1,024 new tokens per turn. Medical-specialist models follow the same format without the calculator components, using their recommended sampling parameters and a larger per-turn token budget; for models with a native reasoning channel, the channel is enabled and its content is removed before scoring.

For the two medical models with an alternative invocation available, EHR-R1 and Baichuan-M2, we ran a preliminary probe and used the native mode only if the model completed fewer than 80% of encounters under our format. EHR-R1 met this criterion (5% under our format, 2/40, versus 5/5

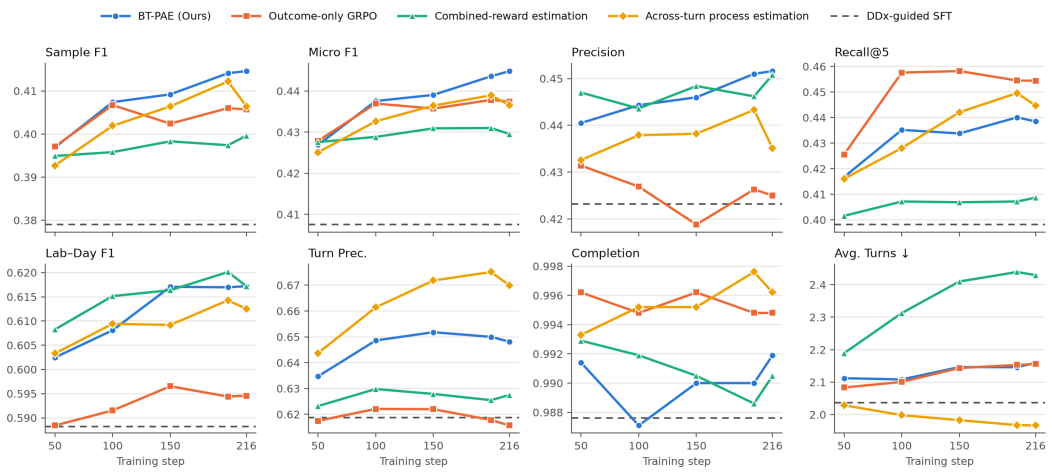


Figure 5: Test-set performance of each reinforcement-learning variant across training checkpoints (steps 50, 100, 150, 200, and 216), with the DDx-guided SFT checkpoint shown as a dashed reference. These curves are not used for checkpoint selection: each reinforcement-learning run reports its final checkpoint, while supervised fine-tuning reports the checkpoint with the minimum validation loss. BT-PAE is still improving in Sample F1 and Micro F1 at the final step. Because the curves show test-set trajectories and are not used for checkpoint selection, we report them descriptively rather than using intermediate checkpoints for cross-variant model selection. Combined-reward estimation lengthens interactions over training, reaching 2.43 average turns, and matches the highest Lab–Day F1 while ending with the lowest Sample F1 and Recall@5. Across-turn process estimation instead slightly shortens interactions and attains the highest per-turn precision throughout training.

natively) and is run natively: it receives the same clinical information re-serialized into its EHR-Bench Markdown format and chooses between ordering laboratory tests and diagnosing at each turn, without an explicit DDx list (Appendix B.4). Baichuan-M2 did not (80% with its reasoning channel enabled and 65% without, versus 100% natively; 20 encounters each), so it runs in our format with reasoning enabled. HuatuoGPT-3 and DoctorAgent-RL were run directly in our format. DoctorAgent-RL was trained to question a simulated patient in dialogue, an interaction that has no counterpart in the EHR environment; its low completion rate reflects this mismatch with the structured diagnostic task.

Completion rates (encounters ending in a final diagnosis) on the ID/OOD test sets are 99.0%/97.6% for EHR-R1, 88.9%/81.1% for HuatuoGPT-3, 74.2%/75.1% for Baichuan-M2, and 45.9%/42.1% for DoctorAgent-RL; encounters without a final diagnosis receive zero on all diagnostic metrics (Appendix G.1). All models are evaluated on the same recorded encounters using common scoring definitions; invocation and matching differences are described above and in Appendix B.2.

G.3 Checkpoint Selection

The SFT checkpoint is selected by the lowest validation loss. For reinforcement learning, we use the final checkpoint after one training epoch. The selected checkpoints are used for both in-distribution and out-of-distribution evaluation, without separate checkpoint selection on the out-of-distribution data.

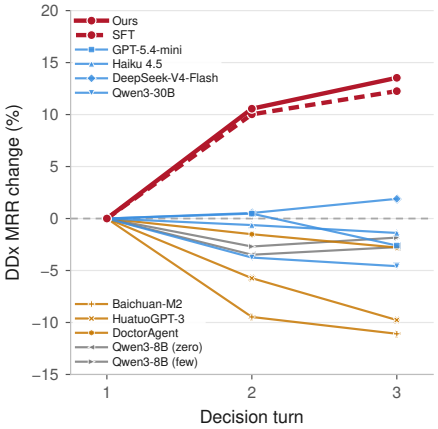
H Additional Results

H.1 Performance across Checkpoints

Figure 5 tracks test-set performance of each reinforcement-learning variant during training. These trajectories are reported for transparency and play no role in checkpoint selection (Appendix G.3).

Model	Lab–Day F1↑	Turn Prec.↑	Non-reorder Rate	n
GPT-5.4-mini	0.106	0.111	45.5%	516
Haiku 4.5	0.048	0.064	54.0%	437
DeepSeek-V4-Flash	0.099	0.113	44.4%	637
Qwen3-30B	0.027	0.044	48.6%	393
EHR-R1-8B	0.015	0.091	82.8%	29
Baichuan-M2	0.075	0.111	31.7%	426
DoctorAgent-8B	0.020	0.110	61.3%	160
HuatuoGPT-3-8B	0.004	0.010	82.8%	87
DDx-guided SFT	0.481	0.533	24.0%	712
Ours	**0.516**	**0.564**	28.1%	758

(a) Evidence acquisition



(b) Diagnostic belief evolution

Figure 6: **Agent behavior during dynamic clinical decision making.**
(a) Evidence acquisition. Laboratory behavior on the EHR-defined multi-turn ID subset. Lab–Day F1 is micro-averaged; turn precision measures the fraction of tests requested at a turn that match a recorded laboratory action, without checking the recording day. Non-reorder rate is the fraction of n qualifying case–test pairs requested in only one turn. Qualification requires recordings on at least two days and a further laboratory-ordering turn after the first request, but not reaching the next recording day; these model-dependent rates are descriptive rather than directly comparable performance scores. Best F1 and precision are bold.
(b) Diagnostic belief evolution. DDx MRR is normalized to each model's first-turn value. Each curve uses a fixed, model-specific subset of encounters lasting at least three turns, supporting comparisons of relative change rather than absolute DDx quality. EHR-R1 is excluded because it does not produce per-turn DDx lists.

H.2 Out-of-Distribution Evaluation

We evaluate the same checkpoints on MIMIC-IV (Johnson et al., 2023), which shares the 82-category diagnosis space but comes from a different institution, without out-of-distribution tuning or checkpoint reselection (Table 5). Under this shift, EHRtemp-Agent ranks second among all evaluated models on Sample F1 (0.315), Micro F1 (0.341), and precision (0.400), behind only DeepSeek-V4-Flash (0.362, 0.406, and 0.419), and outperforms every medical baseline on these three metrics. Its Recall@5 (0.311) is lower than that of four baselines, with Claude Haiku 4.5 attaining the highest value (0.382). Laboratory agreement remains well above the external models (Lab–Day F1 of 0.142 against at most 0.040), although the supervised checkpoint retains a slight margin over reinforcement learning on this metric (0.148 against 0.142), while reinforcement learning improves on it across all four diagnostic metrics. One possible explanation is that process supervision rewards agreement with tests recorded at the training institution; differences in ordering practice or terminology across institutions could therefore reduce transfer of laboratory-agreement metrics. The present experiment does not isolate the source of this shift.

H.3 Agent Behavior under Dynamic Clinical Progression

We examine how agents acquire evidence and revise diagnostic beliefs as interactions progress along clinical trajectories. These analyses characterize stage-specific laboratory requests, repeated acquisition, and changes in the quality of intermediate DDx hypotheses.

Evidence-Acquisition Behavior. On the multi-turn analysis subset, our trained checkpoints achieve higher Lab–Day F1 and per-turn laboratory precision than the external models in Figure 6(a). Alongside Lab–Day F1, which additionally requires a request to fall on the day the test was recorded, we report per-turn precision, the fraction of tests requested at a turn that match a recorded, not-yet-credited laboratory action in the encounter. Non-reorder rate is a descriptive measure of repeated-request behavior: it is the fraction of qualifying case–test pairs requested in only one turn.

Because qualification depends on each model's interaction trajectory, both the denominator n and the qualifying pairs differ across models; non-reorder rates should therefore not be interpreted as directly comparable performance scores. Detailed definitions are provided in Appendix G.1.

EHRTEMP-AGENT achieves a per-turn precision of 0.564, compared with at most 0.113 among external models. On this multi-turn subset, RL increases Lab–Day F1 from 0.481 to 0.516 and per-turn precision from 0.533 to 0.564. The corresponding non-reorder rates are 24.0% and 28.1%, respectively, but this difference is descriptive rather than a direct performance comparison: qualification is model-dependent, and the metric does not determine whether a new measurement was accessible when another request could have been made.

Diagnostic Belief Evolution. Both trained checkpoints show sustained improvement in DDx quality relative to their initial assessments (Figure 6(b)). We track DDx MRR over the first three decision turns for encounters lasting at least three turns, normalizing each model's curve to its own first-turn MRR. Each model is evaluated on a fixed subset across these turns, although the subsets differ between models.

The SFT and RL checkpoints improve by approximately 12–14% over their first-turn MRR. Most external models remain flat or decline, with DeepSeek-V4-Flash showing a smaller increase of approximately 2%. This pattern is consistent with progressive improvement in diagnostic belief after supervised initialization that is retained after RL. Because each model's curve is computed on its own fixed multi-turn subset, the analysis supports within-model statements about belief evolution but not direct comparison of absolute DDx quality across models. Together with the acquisition results, it characterizes how the trained checkpoints use successive evidence-acquisition turns while revising their diagnostic assessments.