EHR-RobustGym: Benchmarking and Training Agents for Robust Clinical Reasoning

Anonymous authors
Paper under double-blind review

Abstract

In hospital workflows, electronic health records (EHRs) are often noisy, and may not contain the evidence needed to confirm events or measurements referenced in a clinical query. Even when database retrieval succeeds, clinical agents can overlook such discrepancies and return plausible but unsupported answers. We introduce `EHR-RobustGym`, a scalable and interactive environment for evaluating and training robust clinical agents grounded in noisy EHRs. Built on MIMIC-IV hospital records (365K patients, 31 tables, and over 500M records), `EHR-RobustGym` comprises 5,486 Clean-Noise pairs spanning six clinical intents and both patient-level and population-level queries. The pairs test robustness to Record-level, Value-level, and Query-level noise, while interactive SQL/Python execution and outcome verification support trajectory collection and training. Evaluating multiple LLMs reveals substantial robustness gaps: average task success across proprietary and large-scale open-weight models drops from 62.2% on Clean questions to 37.9% on Noise questions. At $k = 4$, pass^k consistency falls below 50% for most evaluated models, exposing instability in clinical task completion. Supervised fine-tuning and reinforcement learning in `EHR-RobustGym` improve performance, with gains generalizing to five external EHR benchmarks. Together, these results position `EHR-RobustGym` as a testbed for evaluating and improving the evidence-grounded robustness of clinical agents.

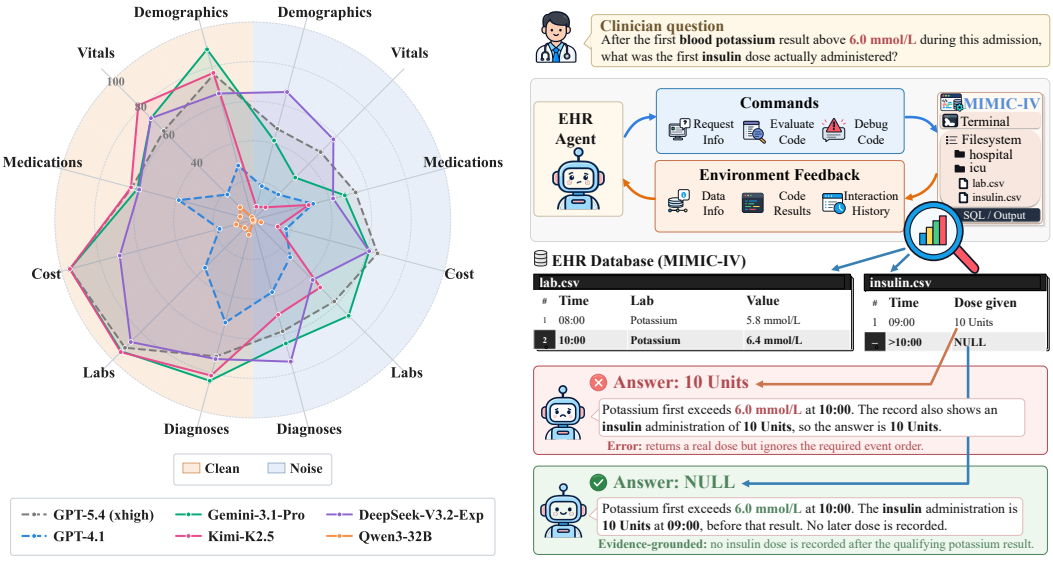


Figure 1: Performance Comparisons under Clean and Noise Conditions. **(Left):** Base success rates (%) for selected models across six clinical intents. **(Right):** A Query-level Noise request asks for the first insulin dose administered after blood potassium first exceeds 6.0 mmol/L. The incorrect response reports 10 Units given at 09:00, overlooking that the qualifying potassium result occurs at 10:00. In contrast, the evidence-grounded response returns `NULL` because the EHR contains no insulin administration after that result.

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.

1 INTRODUCTION

Electronic health records (EHRs) support the reconstruction of patient histories and the analysis of patient populations (Lee et al., 2022; Qiao et al., 2026). Yet they may omit relevant events, contain outdated or inconsistent copied information, or fail to support assumptions embedded in clinical questions (Lin et al., 2018; Weiskopf et al., 2013; Tsou et al., 2017). Although large language model (LLM) agents provide a natural-language interface to EHR databases (Shi et al., 2024; Xu et al., 2026), these limitations complicate reliable answering. Therefore, robust clinical agents must do more than retrieve records or execute queries: they must verify that the evidence satisfies the question's constraints. For example, an agent asked for the first insulin dose administered after blood potassium first exceeds 6.0 mmol/L might return a dose given before the qualifying result (Figure 1). The answer may appear plausible, but it violates the required temporal order. The agent should identify this mismatch and abstain if no supported answer is available.

Recent work has advanced clinical database agents through interactive execution and trajectory-based training (Shi et al., 2024; Xu et al., 2026). EHR-ChatQA studies question clarification and terminology mismatches through user and tool interactions, while EHR-Complex evaluates longitudinal, multi-table reasoning over hospital-scale records (Lee et al., 2026b; Qiao et al., 2026). We focus on a complementary requirement: recognizing when a question's necessary conditions are unsupported, without changing the request or rejecting answerable questions. Evaluation requires examining supported and unsupported requests, since an agent that refuses more often may appear cautious without becoming more reliable. Improving it requires interactive training environments with verifiable learning signals for accurate answering and evidence-grounded abstention.

To this end, we introduce `EHR-RobustGym`, a scalable and interactive environment for benchmarking and training robust clinical agents. Built on MIMIC-IV records containing 365K patients, 31 tables, and over 500M records (Johnson et al., 2024), `EHR-RobustGym` comprises 5,486 Clean-Noise pairs spanning six clinical intents and both patient- and population-level queries. Each pair shares a clinical goal but differs in one necessary condition, contrasting a supported Clean question with an unsupported Noise counterpart without modifying the underlying database. Record-level, Value-level, and Query-level Noise capture unavailable records, missing or unsupported attributes of existing records, and relations that available records do not jointly satisfy, respectively.

`EHR-RobustGym` couples these paired tasks with interactive SQL/Python execution, allowing agents to inspect results and refine operations using database feedback. Completed trajectories are evaluated with a shared outcome verifier that checks answer correctness on Clean questions and requires evidence-grounded abstention on Noise questions. A query failure or a refusal alone does not constitute successful handling of an unsupported request: abstention must be justified by evidence obtained during interaction. Trajectory collection and outcome-based rewards enable supervised fine-tuning and reinforcement learning for both accurate answering and justified abstention.

Extensive benchmarking of 12 LLMs reveals substantial robustness gaps, with average task success across proprietary and large-scale open-weight models dropping from 62.2% on Clean questions to 37.9% on Noise questions. Repeated-run evaluation further shows that pass^k consistency, which requires all k attempts to succeed, falls below 50% for most evaluated models at $k = 4$. These findings highlight that occasional success does not imply dependable task completion across repeated executions. Supervised fine-tuning and reinforcement learning with `EHR-RobustGym` yield substantial performance gains. The improvements extend to five external EHR benchmarks, demonstrating the environment's value as a training ground beyond its own clinical evaluation tasks.

Our contributions are as follows:

- **A comprehensive benchmark for clinical agent robustness.** We introduce a hospital-grounded benchmark of 5,486 Clean-Noise pairs covering Record-level, Value-level, and Query-level Noise, jointly evaluating accurate answering and recognition of unsupported clinical requests.
- **An interactive environment for evidence-grounded agent training.** We integrate SQL/Python execution, outcome verification, and trajectory collection to support supervised fine-tuning and reinforcement learning for accurate answering and evidence-grounded abstention.
- **Extensive benchmarking and effective training of clinical agents.** We evaluate 12 LLMs, characterize their robustness and repeated-run consistency, and demonstrate gains from trajectory-based training on `EHR-RobustGym` and five external EHR benchmarks.

Table 1: Comparison with existing EHR benchmarks.

Dataset	Agentic Mode	Robustness Evaluation		Database Scale			Task Instances		
		Noise Conditions	Trajectory-level Evaluation	Patients	Tables	Elements	Types	Test	Train
MedAgentBench (Jiang et al., 2025)	✔	✗	✔	100	-	700K	10	300	-
EHR-Complex (Qiao et al., 2026)	✔	✗	✔	365K	31	>500M	12	3,915	48,092
TREQS (Wang et al., 2020)	✗	✗	✗	100	5	2.5M	4	996	8,988
EHRSQL (eICU) (Lee et al., 2022)	✗	✗†	✗	<1K	10	1.5M	9	611	6,213
EHRSQL (MIMIC-III) (Lee et al., 2022)	✗	✗†	✗	<1K	17	1.4M	9	1,122	9,318
EHR-SeqSQL (Ryu et al., 2024)	✗	✗	✗	<1K	17	1.4M	4	7,913	18,950
EHR-RobustGym (Ours)	✔	✔	✔	**365K**	**31**	**>500M**	**12**	**2,016**	**8,956**

† EHRSQL uses databases simplified through patient subsampling and measurement filtering. Its unanswerable questions concern schema incompatibility or external knowledge, without explicit noise evaluation within schema-valid requests.

2 Related Work

EHR Database Agents. Recent work has developed LLM agents that retrieve and reason over EHR databases. EHRSQL (Lee et al., 2022) studies text-to-SQL over hospital records, while EHRAgent (Shi et al., 2024) uses iterative code generation and execution for multi-table reasoning. MedAgentBench (Jiang et al., 2025), MedAgentGym (Xu et al., 2026), EHR-ChatQA (Lee et al., 2026b), and EHR-Complex (Qiao et al., 2026) extend this setting to interactive EHR tasks, agent training, clarification, and complex longitudinal reasoning. EHR-Complex provides fine-grained analysis of failed trajectories, while MedAgentGym includes an outcome-supervised verifier that reads interleaved agent trajectories and an analysis of agent errors (Qiao et al., 2026; Xu et al., 2026). Our evaluation distinguishes supported answering from evidence-grounded abstention, while trajectory analysis examines how agents acquire and use evidence from EHR databases.

Robustness under Noisy Conditions. EHRSQL (Lee et al., 2022) includes questions that are incompatible with the database schema or require external knowledge. SCARE (Lee et al., 2026a) studies answerability and SQL correction, while EHR-ChatQA (Lee et al., 2026b) examines ambiguous requests and value mismatches. TrustEHRAgent (Song et al., 2025) studies confidence-based abstention, and CuraView (Ye et al., 2026) verifies generated clinical text against patient records. We study schema-valid requests with unavailable records, missing or conflicting field values, and unsatisfied relations among records. `EHR-RobustGym` pairs these Noise requests with supported Clean requests and requires retrieved evidence to justify abstention. It supports training on interaction trajectories and task outcomes to improve accurate answering and evidence-grounded abstention.

3 EHR-RobustGym

3.1 Problem Formulation

Task Formulation. Given a natural-language clinical question q over an EHR database D with schema Σ, an agent produces a multi-turn trajectory τ by executing SQL queries or Python code and using environment feedback to refine its query plan and debug its code. The objective is to derive the requested answer from database evidence or return `NULL` with evidence identifying the unmet task requirement. We distinguish three levels of missing or conflicting evidence. For population-level questions, these definitions apply within each candidate unit before aggregation. Population-level answers must also satisfy the minimum population size requirement specified in Appendix C.3.

Record-Level Noise. A patient, encounter, or clinical-event record required by q is absent within the specified scope. $D[u] = \varnothing$, where $D[u]$ denotes records for a required reference u in q.

Value-Level Noise. The required record exists, but a field value needed to answer q is missing or contradicts an explicit assumption about that record. For a required field with stored value v, this corresponds to $v = \texttt{NULL}$ or, when q asserts a value or state v^q, to $v \neq v^q$. When multiple records are candidates, this condition must hold for every candidate within the scope of q.

Query-Level Noise. The required records and field values are available, but no combination satisfies the event-selection or relational conditions in q, including thresholds, temporal order, and encounter

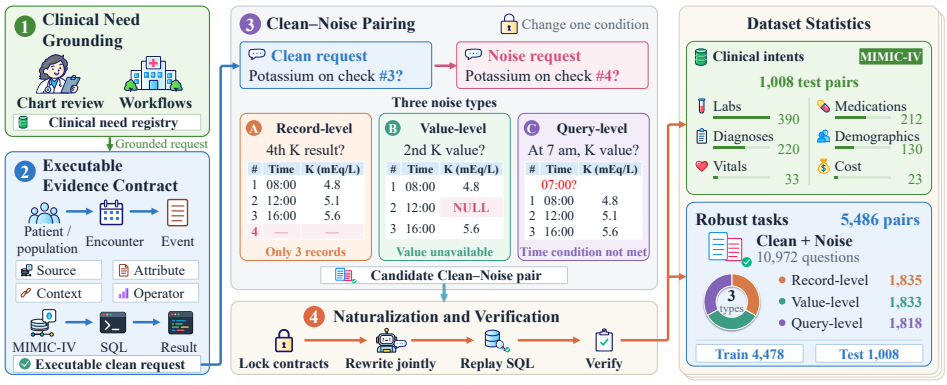


Figure 2: Construction pipeline and task composition of EHR-RobustGym. Step-1. Based on chart-review needs and hospital workflows, clinical requests are derived. Step-2. These requests are grounded in executable evidence contracts with reference SQL and verified answers. Step-3. Clean-Noise pairs are created by changing one necessary condition without modifying the database. Step-4. The paired questions are jointly rewritten with fixed contracts, then verified by SQL execution and semantic checks. The right panel shows clinical intents, noise types, and data splits.

links. Let $\mathcal{C}_q(D)$ denote candidate record combinations satisfying the record- and value-level requirements, and let $\phi_q(C) \in \{0, 1\}$ indicate whether C satisfies these query conditions. Query-level Noise occurs when $\mathcal{C}_q(D) \neq \varnothing$ but $\phi_q(C) = 0$ for every candidate $C \in \mathcal{C}_q(D)$.

3.2 Data Construction

Clinical Task Construction. Figure 2 illustrates the construction of Clean-Noise pairs from clinical evidence and executable SQL queries. EHR-RobustGym focuses on verifiable structured EHR tasks grounded in clinical information needs reflected in DiSCQ chart-review questions and EHRSQL workflows (Lehman et al., 2022; Lee et al., 2022). Following the clinical evidence-path construction in EHR-Complex (Qiao et al., 2026), we programmatically instantiate patient- and population-level queries over MIMIC-IV 3.1 (Johnson et al., 2024). Each query encodes the intended temporal constraints and order of operations. Population-level queries compute within each eligible unit before aggregation. We retain instances whose reference queries execute successfully and yield answers satisfying the specified task requirements as Clean questions.

Clean-Noise Pairing. We construct a Noise counterpart $\bar{q}_t$ for each Clean question q by deterministically changing exactly one necessary condition while preserving the clinical goal and all remaining task requirements. Here, $t \in \{\mathrm{R}, \mathrm{V}, \mathrm{Q}\}$ denotes the Record-, Value-, or Query-level Noise type, determined by the modified condition. We retain a pair only when both reference SQL queries execute successfully and confirm that q has a supported answer and $\bar{q}_t$ requires abstention.

Question Rewriting and Verification. To improve naturalness and diversity, we use a language model to jointly rewrite each pair while preserving the meaning and computation encoded by the reference SQL. Reference answers are extracted from database execution. Clean answers provide the requested values and units, while Noise answers return NULL and identify the unmet task requirement. We replay the reference queries and conduct an independent language-model audit to verify execution consistency and semantic alignment for both conditions. As an additional semantic check, two reviewers independently audit 344 challenging test pairs (688 questions), with 92.15% judged valid after adjudication (Appendix B). Identified issues are corrected and reverified, and affected tasks are reevaluated so that all reported results correspond to the finalized test set.

Dataset Statistics. Table 1 compares EHR-RobustGym with existing EHR benchmarks. EHR-RobustGym contains 5,486 Clean-Noise pairs across 51 programs and six clinical intents, split into 4,478 training and 1,008 test pairs (8,956 and 2,016 questions, respectively). The test set is balanced across two query scopes (504 pairs each) and three Noise types (336 pairs each). The in-domain and structure-disjoint tracks each contain 504 test pairs, with six programs held out for the latter. Each pair remains within a single split. Patient-level tasks use disjoint patient identifiers

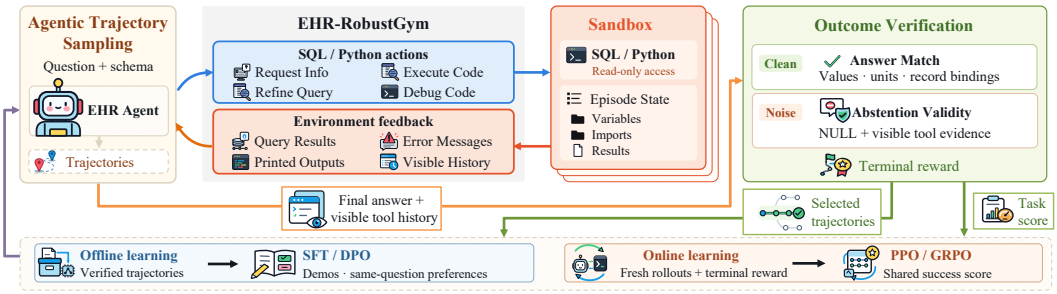


Figure 3: Overview of `EHR-RobustGym`. It contains paired Clean and Noise tasks with interactive SQL/Python execution and outcome verification for evaluating and training LLM agents.

across training and test sets, while population-level tasks use disjoint query specifications but may share patients. Further dataset construction details are provided in Appendix A.3.

3.3 Interactive Evaluation and Training

Interactive Environment. `EHR-RobustGym` provides an executable SQL/Python environment with interactive feedback and repeated trajectory sampling (Figure 3). For each episode, the agent receives $x = (q, s)$, where s contains the database schema and shared task instructions. The agent iteratively refines its queries using execution results and error feedback, then submits a final response containing `Answer` and `Reasoning`. Clean and Noise counterparts are presented in separate episodes under the same interface. Condition labels, pairing information, reference SQL, and reference answers are hidden from the agent throughout each inference episode.

Outcome Verification. For each trajectory, the binary success score is formulated as

$$R(x, \tau) = \begin{cases} \text{Answer Match}(\tau), & q \text{ is Clean}, \\ \text{Abstention Validity}(\tau), & q \text{ is Noise}. \end{cases} \tag{1}$$

Answer Match checks the requested values, units, and associated records against the execution-verified reference answer. Abstention Validity requires `NULL` and visible evidence from tool calls establishing an unmet task requirement within the scope specified by the question.

Table 2: Task success rates of unadapted EHR agents on `EHR-RobustGym`.

Model	Demographics	Vitals	Medications	Cost	Labs	Diagnoses	Avg.
Proprietary API models							
GPT-5.4 (xhigh)	76.2 / 47.7	63.6 / 48.5	63.2 / 53.8	95.7 / 65.2	91.3 / 58.2	71.4 / 58.2	78.3 / 55.8
GPT-4.1	28.5 / 17.7	18.2 / 18.2	38.7 / 31.6	17.4 / 17.4	34.1 / 26.7	53.6 / 37.7	37.7 / 28.5
Gemini-3.1-Pro	89.2 / 41.5	72.7 / 30.3	60.4 / 48.1	95.7 / 60.9	93.8 / 68.5	84.1 / 64.5	83.4 / 58.4
Claude Sonnet 4.6	62.3 / 28.5	60.6 / 24.2	58.5 / 22.6	65.2 / 13.0	64.4 / 19.2	63.2 / 42.7	62.5 / 26.3
Open-weight models ≥100B							
Kimi-K2.5	76.9 / 6.9	81.8 / 9.1	63.7 / 29.2	95.7 / 13.0	94.4 / 48.2	81.4 / 49.5	82.4 / 37.1
DeepSeek-V3.2	66.2 / 66.9	72.7 / 57.6	59.4 / 42.0	69.6 / 60.9	87.2 / 42.8	72.7 / 74.1	74.6 / 53.5
Qwen3-235B-A22B	3.1 / 0.0	3.0 / 0.0	23.6 / 4.7	13.0 / 8.7	6.7 / 5.1	38.2 / 10.9	16.7 / 5.6

Note. Clean / Noise success rates (Pass@1, %, ↑). indicates rates below 5%.

Agent Training. `EHR-RobustGym` supports repeated sampling of successful and unsuccessful trajectories for agent training. For SFT, we fine-tune models on verified successful teacher trajectories from both Clean and Noise tasks after removing teacher-only reference guidance. Offline DPO (Rafailov et al., 2023) contrasts successful teacher trajectories with unsuccessful model trajectories for the same question. We further consider online PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024), enabling agents to actively explore tasks and generate new training trajectories through environment interaction. Both methods use the same success score $R(x, \tau)$ for optimization, with separate KL regularization and no intermediate or format rewards. We explore self-improvement by refining each model on its own trajectories using RS+SFT and iterative DPO. PE provides a prompt-only baseline by adding evidence-check instructions without parameter updates. Training settings and prompts are provided in Section 4 and Appendices H and D.2.

Table 3: EHR agent performance after prompting and training with `EHR-RobustGym`.

Model	Method	Demographics	Vitals	Medications	Cost	Labs	Diagnoses	Avg.	Δ vs. Base
Prompt-only mitigation (API-based models)									
GPT-5.4 (xhigh)	PE	81.5 / 43.8	69.7 / 45.5	65.1 / 52.4	82.6 / 52.2	91.5 / 59.5	78.2 / 59.1	80.9 / 55.3	+2.6 / -0.5
GPT-4.1	PE	30.0 / 23.1	18.2 / 6.1	38.2 / 31.1	21.7 / 21.7	35.4 / 23.1	53.2 / 40.5	38.3 / 28.0	+0.6 / -0.5
Gemini-3.1-Pro	PE	89.2 / 37.7	75.8 / 48.5	63.2 / 56.1	91.3 / 65.2	95.6 / 70.8	85.0 / 66.4	84.9 / 61.6	+1.5 / +3.2
Claude Sonnet 4.6	PE	66.2 / 33.8	78.8 / 24.2	63.2 / 25.0	87.0 / 17.4	68.2 / 29.0	70.0 / 47.3	68.1 / 32.3	+5.6 / +6.1
Kimi-K2.5	PE	77.7 / 16.2	87.9 / 18.2	59.9 / 33.5	91.3 / 8.7	93.8 / 51.0	80.9 / 49.5	81.5 / 40.5	-0.9 / +3.4
DeepSeek-V3.2	PE	74.6 / 66.9	81.8 / 57.6	57.1 / 43.4	91.3 / 73.9	83.6 / 44.6	79.5 / 77.3	76.1 / 55.5	+1.5 / +2.0
Qwen3-235B-A22B	PE	2.3 / 0.0	3.0 / 0.0	23.6 / 7.1	4.3 / 4.3	7.9 / 6.7	42.7 / 15.9	17.9 / 7.6	+1.2 / +2.1
Prompting and training (open-weight models <100B)									
Qwen3-4B	Base	1.5 / 0.0	0.0 / 0.0	2.4 / 0.0	0.0 / 0.0	1.3 / 0.0	0.0 / 0.0	1.2 / 0.0	- / -
	PE	0.8 / 0.0	0.0 / 0.0	2.8 / 0.0	0.0 / 0.0	1.3 / 0.0	0.0 / 0.0	1.2 / 0.0	0.0 / 0.0
	SFT	17.7 / 13.1	12.1 / 3.0	29.2 / 23.6	39.1 / 13.0	28.7 / 26.4	44.5 / 12.3	30.6 / 19.9	+29.4 / +19.9
	+DPO	9.2 / 6.9	15.2 / 3.0	29.2 / 9.9	39.1 / 8.7	45.1 / 7.7	35.9 / 9.5	34.0 / 8.3	+32.8 / +8.3
	+PPO	16.2 / 10.8	3.0 / 6.1	29.2 / 39.6	43.5 / 30.4	52.3 / 39.7	45.0 / 36.8	39.4 / 34.0	+38.2 / +34.0
	+GRPO	33.1 / 24.6	27.3 / 21.2	35.8 / 48.6	60.9 / 56.5	60.0 / 29.5	46.4 / 41.4	47.4 / 35.8	**+46.2** / **+35.8**
Qwen3-8B	Base	0.0 / 0.0	0.0 / 0.0	6.1 / 0.0	0.0 / 0.0	0.5 / 1.3	9.1 / 0.0	3.5 / 0.5	- / -
	PE	0.0 / 0.0	0.0 / 0.0	3.3 / 0.0	0.0 / 4.3	0.5 / 2.6	3.6 / 0.5	1.7 / 1.2	-1.8 / +0.7
	SFT	21.5 / 18.5	15.2 / 15.2	26.4 / 25.9	21.7 / 47.8	42.1 / 20.5	37.7 / 25.0	33.8 / 22.8	+30.4 / +22.3
	+DPO	19.2 / 30.8	9.1 / 27.3	17.5 / 25.9	39.1 / 60.9	26.4 / 43.8	27.3 / 24.1	23.5 / 33.9	+20.0 / +33.4
	+PPO	16.9 / 20.8	6.1 / 3.0	24.1 / 21.2	30.4 / 43.5	36.7 / 30.5	34.5 / 12.7	29.9 / 22.8	+26.4 / +22.3
	+GRPO	38.5 / 47.7	33.3 / 12.1	41.0 / 45.8	69.6 / 69.6	63.6 / 40.5	50.0 / 27.7	51.8 / 39.5	**+48.3** / **+39.0**
Qwen3-14B	Base	0.0 / 0.0	0.0 / 0.0	0.9 / 0.0	8.7 / 0.0	14.9 / 4.9	27.3 / 18.2	12.1 / 5.9	- / -
	PE	0.8 / 0.8	3.0 / 0.0	0.5 / 0.0	8.7 / 17.4	14.9 / 3.6	8.6 / 8.2	8.1 / 3.7	-4.0 / -2.2
	SFT	23.8 / 23.8	27.3 / 18.2	41.0 / 33.0	34.8 / 34.8	49.2 / 33.1	45.5 / 48.2	42.4 / 34.7	+30.3 / +28.9
	+DPO	28.5 / 43.8	27.3 / 27.3	44.8 / 44.3	65.2 / 65.2	50.0 / 55.4	37.7 / 56.8	43.1 / 51.2	+31.0 / +45.3
	+PPO	65.4 / 69.2	51.5 / 39.4	60.4 / 61.8	95.7 / 95.7	77.2 / 59.0	71.4 / 65.9	70.4 / 62.6	**+58.3** / **+56.7**
	+GRPO	63.1 / 72.3	60.6 / 51.5	47.2 / 44.8	91.3 / 78.3	78.2 / 42.1	70.9 / 74.5	67.9 / 54.8	+55.8 / +48.9
Qwen3-32B	Base	1.5 / 0.0	9.1 / 0.0	6.6 / 0.5	8.7 / 4.3	5.6 / 0.5	7.7 / 0.9	6.0 / 0.6	- / -
	PE	1.5 / 0.8	9.1 / 0.0	7.1 / 0.5	13.0 / 0.0	5.1 / 0.5	2.3 / 0.9	4.8 / 0.6	-1.2 / 0.0
	SFT	36.9 / 31.5	21.2 / 15.2	47.6 / 45.3	30.4 / 26.1	57.9 / 45.1	46.8 / 37.3	48.8 / 40.3	+42.9 / +39.7
	+DPO	23.8 / 32.3	24.2 / 18.2	45.3 / 57.5	30.4 / 39.1	67.9 / 61.5	41.8 / 60.5	49.5 / 54.8	+43.6 / +54.2
	+PPO	40.8 / 61.5	39.4 / 39.4	48.6 / 57.1	95.7 / 87.0	72.6 / 52.6	72.3 / 68.6	62.8 / 58.5	**+56.8** / **+57.9**
	+GRPO	60.0 / 54.6	42.4 / 33.3	42.5 / 25.0	78.3 / 56.5	78.7 / 39.0	57.3 / 36.4	62.8 / 37.7	**+56.8** / +37.1
Llama-3.1-8B	Base	0.0 / 0.0	0.0 / 0.0	0.0 / 0.9	0.0 / 0.0	0.3 / 0.3	0.0 / 0.0	0.1 / 0.3	- / -
	PE	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	1.0 / 0.0	0.0 / 0.0	0.4 / 0.0	+0.3 / -0.3
	SFT	41.5 / 35.4	54.5 / 30.3	48.1 / 39.2	69.6 / 73.9	59.2 / 26.2	49.1 / 28.6	52.5 / 31.8	**+52.4** / +31.5
	+DPO	32.3 / 26.2	15.2 / 30.3	37.7 / 48.1	60.9 / 60.9	11.5 / 59.7	31.4 / 28.6	25.3 / 45.2	+25.2 / **+44.9**
	+PPO	0.0 / 0.8	9.1 / 21.2	29.7 / 27.8	4.3 / 47.8	22.1 / 32.1	45.5 / 13.6	25.1 / 23.1	+25.0 / +22.8
	+GRPO	49.2 / 53.8	54.5 / 42.4	45.3 / 58.5	87.0 / 65.2	53.8 / 30.8	39.5 / 25.9	49.1 / 39.7	+49.0 / +39.4

Note. Clean / Noise success rates (Pass@1, %, ↑). indicates training results. indicates rates below 5%.

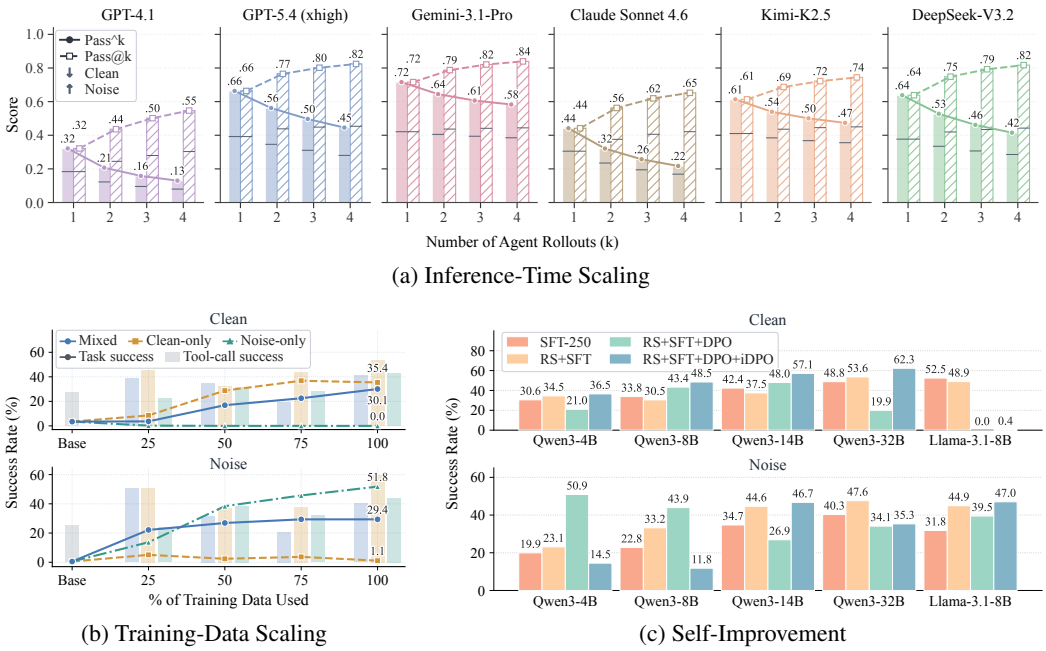
4 Experiments

We benchmark EHR agents under Clean and Noise conditions, then evaluate prompting and trajectory-based training with `EHR-RobustGym`. Analyses examine evidence-grounded abstention, inference-time and training-data scaling, capability retention, and tool interaction errors.

4.1 Experimental Setup

Tasks, Models and Implementation Details. We evaluate seven models through external APIs and five locally deployed open-weight models. The API-based models include GPT-5.4 (xhigh) (OpenAI, 2026a), GPT-4.1 (`gpt-4.1-2025-04-14`) (OpenAI, 2025a), Gemini-3.1-Pro (`gemini-3.1-pro-preview`) (Google, 2026), Claude Sonnet 4.6 (Anthropic, 2026), Kimi-K2.5 (Kimi Team et al., 2026a), DeepSeek-V3.2 (`DeepSeek-V3.2-Exp`) (DeepSeek-AI, 2025), and Qwen3-235B-A22B (`Qwen3-235B-A22B-Instruct-2507`) (Qwen Team, 2025; Yang et al., 2025). For parameter training, we use Qwen3-4B, Qwen3-8B, Qwen3-14B, Qwen3-32B, and Llama-3.1-8B (`Llama-3.1-8B-Instruct`) (Yang et al., 2025; Grattafiori et al., 2024). All models are evaluated on the same 1,008 Clean-Noise test pairs across six clinical intents. The training partition contains 4,478 pairs. All models interact with the same environment described in Section 3.3. Prompts and implementation details appear in Appendices H and C.

Evaluation Metrics. We report *Task Success Rate* (SR) separately for Clean and Noise requests, using the criterion in Equation 1. GPT-5.2 (high) (OpenAI, 2025b) performs this verification. Evaluation and online RL use the same scorer settings. Appendix C.3 provides the scoring criteria.



(a) Inference-Time Scaling

(b) Training-Data Scaling

(c) Self-Improvement

Figure 4: Scaling and Self-Improvement in `EHR-RobustGym`.

4.2 Evaluation Results

Overall Performance. Table 2 reports Clean and Noise success rates for seven unadapted API-based models. All seven perform worse on Noise requests, with average success dropping from 62.2% to 37.9%. Gemini and Kimi achieve similar Clean success rates of 83.4% and 82.4%, but differ under Noise, reaching 58.4% and 37.1%. These results suggest that evidence-grounded reasoning remains challenging when supporting records are missing or inconsistent with the request. Previous evaluations focused on answerable tasks do not adequately capture this capability.

Noise Types and Evidence Verification. Query-level Noise is the most challenging for all seven Base models. GPT-5.4 achieves 43.2% success, compared with 58.3% on Record-level and 65.8% on Value-level Noise, suggesting difficulty in verifying relations among records (Appendix E.1). Abstention alone does not imply task success. GPT-4.1 returns `NULL` on 80.6% of Noise requests, but only 35.3% of these responses pass evidence verification, yielding 28.5% success.

Clinical Intent and Scope. On Clean requests, GPT-5.4 performs best on Cost and Labs, at 95.7% and 91.3%. Medications and Vitals show lower success, at 63.2% and 63.6%, and involve reasoning over medication states and ordered observations. Moving from patient-level to population-level requests, its Noise success drops from 65.5% to 46.0%. Population-level requests require agents to verify cohort eligibility and reporting support. Full results are reported in Table 16.

4.3 Training EHR Agents with EHR-RobustGym

Training Setup. We use trajectory-based training to jointly strengthen task completion and evidence-grounded abstention. To assess training effectiveness across model scales and families, we select four Qwen3 backbones (4B, 8B, 14B, and 32B) and Llama-3.1-8B. We randomly sample 1,000 Clean-Noise pairs from the training set, comprising 2,000 questions, and adopt a two-stage fine-tuning framework. Models first learn from successful trajectories through SFT for 250 updates (half an epoch). For each backbone, DPO, PPO, and GRPO independently continue from the same SFT checkpoint for another 250 updates using the remaining 1,000 questions. Offline DPO contrasts successful teacher trajectories with unsuccessful model trajectories for the same question, while online PPO and GRPO sample eight trajectories per question through environment interaction. We include PE as a prompt-only baseline for both the five backbones and seven API-based models. Full training configurations and additional experiments are provided in Appendix D.

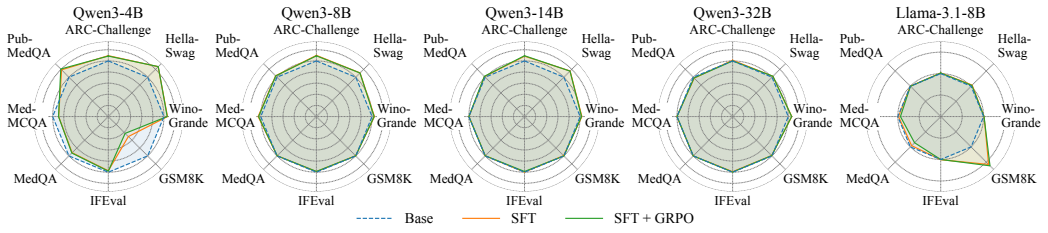


Figure 5: General and Medical Capabilities after EHR Training.

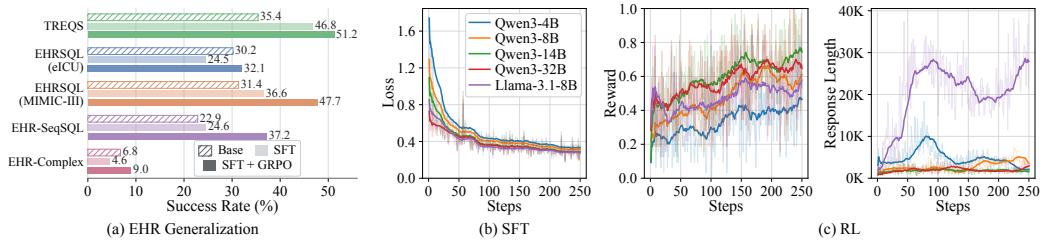


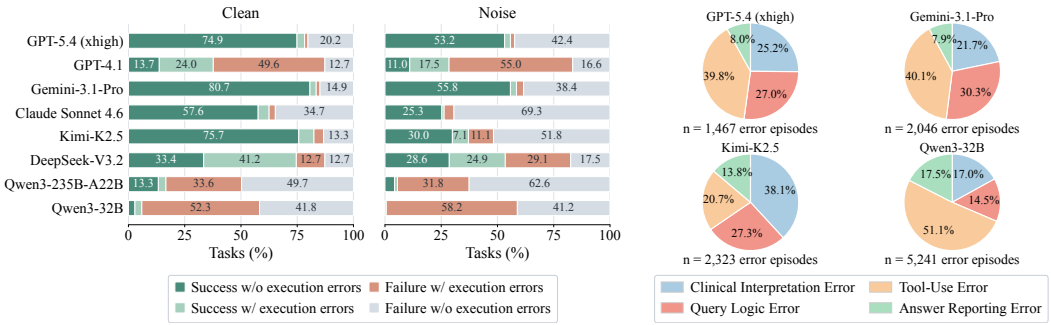
Figure 6: External EHR Generalization and Training Dynamics. (a) Qwen3-14B success rates on five external EHR benchmarks. (b) SFT loss and (c) subsequent GRPO reward and response length.

Training Results. Table 3 shows that SFT on successful trajectories improves Clean and Noise success across all five backbones. Qwen3-32B improves from 6.0%/0.6% to 48.8%/40.3% Clean/Noise success, while Llama-3.1-8B improves from near-zero success to 52.5%/31.8%. By contrast, PE yields little improvement on smaller models and mixed results on API-based models. The SFT gains demonstrate the effectiveness of trajectory supervision for task completion and evidence-grounded abstention across model scales and families. Subsequent online RL yields the strongest results on the Qwen3 backbones, with GRPO performing best on the 4B and 8B models and PPO on the 14B and 32B models. PPO raises Qwen3-14B's Clean/Noise success from 42.4%/34.7% to 70.4%/62.6% with outcome feedback from `EHR-RobustGym`. DPO also improves both conditions for Qwen3-14B and Qwen3-32B but remains below PPO. However, subsequent optimization does not consistently improve both conditions. On Qwen3-32B, PPO and GRPO both reach 62.8% Clean success, but their Noise scores are 58.5% and 37.7%, with GRPO falling below SFT. On Llama-3.1-8B, DPO raises Noise success to 45.2% but reduces Clean success to 25.3%, while PPO decreases both. Figure 6(b,c) presents the training dynamics of SFT and subsequent GRPO.

Inference-Time Scaling. We study whether models can consistently complete tasks across repeated rollouts. Pass@k measures the fraction of questions solved by at least one of k attempts, while pass^k requires all k attempts to succeed. Figure 4(a) shows that GPT-5.4 reaches 82.4% Pass@4 but only 44.5% pass^4, while Gemini achieves 83.9% and 58.3%, respectively. These results show that additional sampling improves task coverage, but even strong models struggle to complete the same tasks consistently. Sampling details and per-condition results are provided in Appendix E.5.

Training-Data Scaling and Composition. Figure 4(b) compares one-epoch SFT on Qwen3-8B with 250-1,000 trajectories, using one trajectory per pair across Clean-only, Noise-only, and Mixed training. At 1,000 trajectories, Mixed training achieves 30.1%/29.4% Clean/Noise success, while single-condition training yields near-zero success on the other condition. With both trajectories per pair, increasing Mixed training from 500 to 2,000 trajectories raises Clean/Noise success from 16.9%/21.9% to 39.8%/42.8% (Appendix E.3). At 1,000 trajectories, Mixed training improves tool-call success while roughly halving calls per question. However, Clean-only training achieves 59.6% tool-call success on Noise questions but only 1.1% task success. Increasing Mixed training from 500 to 750 trajectories lowers tool-call success despite higher task success on both conditions. Successful tool execution alone does not ensure correct answers or evidence-grounded abstention.

Self-Improvement with Model-Generated Trajectories. We explore self-improvement by refining each model using its own outputs. Starting from SFT-250, we perform RS+SFT followed by DPO and iDPO for one epoch each, selecting training trajectories from eight fresh rollouts per question



(a) Tool Execution Errors and Task Outcomes

(b) Distribution of Error Episodes

Figure 7: Trajectory Error Analysis.

at every stage. Figure 4(c) shows that RS+SFT improves Noise success for all five models, while Clean success improves only for Qwen3-4B and Qwen3-32B. Subsequent DPO stages yield less consistent gains across models. After the full sequence, Qwen3-14B improves from 42.4%/34.7% to 57.1%/46.7% Clean/Noise success, while Qwen3-8B and Qwen3-32B finish below their SFT Noise scores. The final iDPO set for Llama-3.1-8B is highly imbalanced, containing only three Clean questions and 241 Noise questions. Further details are provided in Appendix D.3.

General Capabilities and EHR Generalization. Across eight general and medical benchmarks (Figure 5), four of five models improve their average over Base after SFT and remain above Base after GRPO. Qwen3-4B is the exception, with GSM8K success falling from 78.9% to 33.7% after SFT and subsequent GRPO. Full results are provided in Appendix E.4. Figure 6(a) further evaluates Qwen3-14B on five external EHR benchmarks without target-specific fine-tuning. After GRPO, the model outperforms both Base and SFT on all five benchmarks. Relative to Base, success rises from 35.4% to 51.2% on TREQS (Wang et al., 2020), with gains extending to the challenging EHR-Complex (Qiao et al., 2026), from 6.8% to 9.0%. Although absolute success on EHR-Complex remains low, improvements across all five benchmarks support the effectiveness of `EHR-RobustGym` as a training environment for EHR reasoning that transfers to other benchmarks.

Error Analysis. Figure 7(a) groups Base trajectories by task outcome and the presence of tool execution errors, with timeouts and response-format errors counted separately. For GPT-5.4, failures without tool execution errors account for 42.4% of Noise requests, compared with 20.2% of Clean requests. This suggests that tool execution errors alone do not explain the additional difficulty under Noise. We use LLM-based annotation to analyze 12,096 Base trajectories from GPT-5.4, Gemini-3.1-Pro, Claude Sonnet 4.6, Kimi-K2.5, DeepSeek-V3.2, and Qwen3-32B, covering both successful and failed trajectories under Clean and Noise conditions. Figure 7(b) summarizes error episode distributions across four categories for four models; Claude and DeepSeek appear in Appendix F. A trajectory can contain multiple error episodes, including those later corrected. For GPT-5.4, clinical interpretation and query logic errors account for 25.2% and 27.0% of annotated error episodes, respectively. Representative trajectories show errors in interpreting procedure counts, preserving event order, and selecting the requested hospitalization (Appendix G). These cases show that agents can retrieve relevant records yet fail to preserve clinical constraints when using the evidence.

5 Conclusion

We presented `EHR-RobustGym`, a scalable and interactive environment for evaluating and training robust clinical agents grounded in noisy EHRs. With Clean-Noise pairs spanning Record-level, Value-level, and Query-level noise, `EHR-RobustGym` jointly evaluates accurate answering on supported questions and evidence-grounded abstention on unsupported requests. Our evaluation reveals substantial robustness gaps between Clean and Noise questions and unstable clinical task completion across repeated executions. Supervised fine-tuning and reinforcement learning improve performance, with gains generalizing to five external EHR benchmarks. Together, these findings position `EHR-RobustGym` as a testbed for evaluating and improving the evidence-grounded robustness of clinical agents, while highlighting the need to preserve accurate answering on supported questions.

AI USE STATEMENT

AI assistants were used to support language polishing. The use of language models in task construction, trajectory generation, outcome verification, and error analysis is described in the corresponding methodology and experimental sections. All scientific claims, experimental results, analyses, citations, and final manuscript content were reviewed and verified by the authors, who take full responsibility for the work.

ETHICS STATEMENT

`EHR-RobustGym` is constructed from MIMIC-IV v3.1, a de-identified EHR database released through PhysioNet for research under credentialed access (Johnson et al., 2024). The underlying data have undergone privacy-preserving processing, including removal of protected health information, random identifier replacement, and patient-level date shifting (Johnson et al., 2023). Our study uses these retrospective records for research on evidence-grounded clinical agents. Access to the underlying records remains subject to the original MIMIC-IV credentialing and data use requirements. Any released artifacts, including task definitions, SQL programs, trajectories, model checkpoints, and code, will be shared only in forms compatible with these requirements, with appropriate access controls for restricted derivatives. `EHR-RobustGym` is intended as a research testbed for evaluating and training clinical agents, not as a clinical decision-support system or a substitute for professional clinical judgment.

REPRODUCIBILITY STATEMENT

Section 3 defines the clinical tasks, the three noise types, and the construction of Clean-Noise pairs, together with the interactive environment and outcome verification used for evaluation and training. Appendix A details the construction programs, reference generation, question rewriting, and data splits, including the separation of patient-level tasks and the construction programs reserved for structure-disjoint evaluation. Appendix B documents the selection of challenging test pairs, human review criteria, and adjudication procedure.

Section 4 describes the evaluated models, metrics, and training comparisons. Appendix C specifies model identifiers, sampling settings, interaction limits, and outcome evaluation criteria for accurate answering and evidence-grounded abstention. It also describes the datasets and protocols used to assess general capabilities and transfer to external EHR benchmarks. Appendix D provides the demonstration selection procedure, training-data allocation, checkpoint initialization, and optimization settings for SFT and subsequent DPO, PPO, and GRPO, along with the trajectory selection and training schedule for self-improvement. Appendix E reports additional comparisons and per-benchmark scores, while Appendix E.5 specifies the repeated-run protocol and the computation of Pass@k and pass^k. The error taxonomy and trajectory annotation procedure are provided in Appendix F, and the shared agent instructions and response format appear in Appendix H.

The supplementary code provides database preparation, the interactive SQL/Python environment, and an evaluation framework with answerable-question scoring. It also provides the EHR-specific training integration and configurations through verl. Dependency requirements and usage instructions accompany these components. The underlying MIMIC-IV records must be obtained separately through PhysioNet; the supplied database conversion procedure prepares them for use with the environment.

REFERENCES

Anthropic. Introducing Claude Sonnet 4.6, 2026. URL https://www.anthropic.com/news/claude-sonnet-4-6.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI. DeepSeek-V3.2-Exp: Boosting Long-Context Efficiency with DeepSeek Sparse Attention, 2025. URL https://github.com/deepseek-ai/DeepSeek-V3.2-Exp.

Google. Gemini 3.1 Pro Preview, 2026. URL https://ai.google.dev/gemini-api/docs/models/gemini-3.1-pro-preview.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 Herd of Models, 2024. URL https://arxiv.org/abs/2407.21783.

Yixing Jiang, Kameron C. Black, Gloria Geng, Danny Park, James Zou, Andrew Y. Ng, and Jonathan H. Chen. MedAgentBench: A Virtual EHR Environment to Benchmark Medical LLM Agents. *NEJM AI*, 2(9):AIdbp2500144, 2025. doi: 10.1056/AIdbp2500144. URL https://doi.org/10.1056/AIdbp2500144.

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. *Applied Sciences*, 11(14):6421, 2021. doi: 10.3390/app11146421. URL https://www.mdpi.com/2076-3417/11/14/6421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A Dataset for Biomedical Research Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1259. URL https://aclanthology.org/D19-1259/.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Brian Gow, Benjamin Moody, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV. *PhysioNet*, October 2024. doi: 10.13026/kpb9-mt58. URL https://doi.org/10.13026/kpb9-mt58. Version 3.1.

Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-wei H. Lehman, Leo A. Celi, and Roger G. Mark. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1, 2023. doi: 10.1038/s41597-022-01899-x. URL https://doi.org/10.1038/s41597-022-01899-x.

Kimi Team et al. Kimi K2.5: Visual Agentic Intelligence, 2026a. URL https://arxiv.org/abs/2602.02276.

Kimi Team et al. Kimi K3: Open Frontier Intelligence, 2026b. URL https://arxiv.org/abs/2607.24653.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626. Association for Computing Machinery, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Gyubok Lee, Hyeonji Hwang, Seongsu Bae, Yeonsu Kwon, Woncheol Shin, Seongjun Yang, Minjoon Seo, Jong-Yeup Kim, and Edward Choi. EHRSQL: A Practical Text-to-SQL Benchmark for Electronic Health Records. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15589–15601. Curran Associates, Inc., 2022. doi: 10.52202/068431-1134. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/643e347250cf9289e5a2a6c1ed5ee42e-Paper-Datasets_and_Benchmarks.pdf.

Gyubok Lee, Woosog Chay, and Edward Choi. SCARE: A Benchmark for SQL Correction and Question Answerability Classification for Reliable EHR Question Answering. In *Proceedings of the Fifth Machine Learning for Health Symposium*, volume 297 of *Proceedings of Machine Learning Research*, pp. 1446–1466. PMLR, 2026a. URL https://proceedings.mlr.press/v297/lee26b.html.

Gyubok Lee, Woosog Chay, Heeyoung Kwak, Yeong Hwa Kim, Haanju Yoo, Oksoon Jeong, Meong Hi Son, and Edward Choi. From Conversation to Query Execution: Benchmarking User and Tool Interactions for EHR Database Agents. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=hLweUPBz7k.

Eric Lehman, Vladislav Lialin, Katelyn Edelwina Legaspi, Anne Janelle Sy, Patricia Therese Pile, Nicole Rose Alberto, Richard Raymund Ragasa, Corinna Victoria Puyat, Marianne Katharina Taliño, Isabelle Rose Alberto, Pia Gabrielle Alfonso, Dana Moukheiber, Byron Wallace, Anna Rumshisky, Jennifer Liang, Preethi Raghavan, Leo Anthony Celi, and Peter Szolovits. Learning to ask like a physician. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*, pp. 74–86. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.clinicalnlp-1.8. URL https://aclanthology.org/2022.clinicalnlp-1.8/.

Kueiyu Joshua Lin, Robert J. Glynn, Daniel E. Singer, Shawn N. Murphy, Joyce Lii, and Sebastian Schneeweiss. Out-of-system care and recording of patient characteristics critical for comparative effectiveness research. *Epidemiology*, 29(3):356–363, 2018. doi: 10.1097/EDE.0000000000000794. URL https://pubmed.ncbi.nlm.nih.gov/29283893/.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, 2019. URL https://arxiv.org/abs/1711.05101.

OpenAI. Introducing GPT-4.1 in the API, 2025a. URL https://openai.com/index/gpt-4-1/.

OpenAI. Introducing GPT-5.2, 2025b. URL https://openai.com/index/introducing-gpt-5-2/.

OpenAI. Introducing GPT-5.4, 2026a. URL https://openai.com/index/introducing-gpt-5-4/.

OpenAI. Introducing GPT-5.5, 2026b. URL https://openai.com/index/introducing-gpt-5-5/.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition, 2026c. URL https://openai.com/index/gpt-5-6/.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A Large-scale Multi-Subject Multi-Choice Dataset for Medical domain Question Answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260. PMLR, 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Yitong Qiao, Lei Liu, Yue Shen, Jian Wang, Jinjie Gu, Zhixuan Chu, and Kui Ren. EHR-Complex: Benchmarking Medical Agents for Complex Clinical Reasoning, 2026. URL https://arxiv.org/abs/2606.23301.

Qwen Team. Qwen3-235B-A22B-Instruct-2507, 2025. URL https://huggingface.co/Qwen/Qwen3-235B-A22B-Instruct-2507.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory Optimizations Toward Training Trillion Parameter Models, 2020. URL https://arxiv.org/abs/1910.02054.

Jaehee Ryu, Seonhee Cho, Gyubok Lee, and Edward Choi. EHR-SeqSQL : A Sequential Text-to-SQL Dataset For Interactively Exploring Electronic Health Records. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 16388–16407. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.971. URL https://aclanthology.org/2024.findings-acl.971/.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740, April 2020. doi: 10.1609/aaai.v34i05.6399. URL https://ojs.aaai.org/index.php/AAAI/article/view/6399.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

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, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297. Association for Computing Machinery, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce C. Ho, Carl Yang, and May Dongmei Wang. EHRAgent: Code Empowers Large Language Models for Few-shot Complex Tabular Reasoning on Electronic Health Records. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22315–22339. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.1245. URL https://aclanthology.org/2024.emnlp-main.1245/.

Yongwoo Song, Minbyul Jeong, and Mujeen Sung. Trustworthy Agents for Electronic Health Records through Confidence Estimation, 2025. URL https://arxiv.org/abs/2508.19096.

Lintang Sutawika, Hailey Schoelkopf, Leo Gao, Baber Abbasi, Stella Biderman, Jonathan Tow, ben fattori, Charles Lovering, farzanehnakhaee70, Jason Phang, Anish Thite, Fazz, Thomas Wang, Niklas, Aflah, sdtblck, nopperl, gakada, tttyuntian, researcher2, James A. Michaelov, Julen Etxaniz, Taksh Kothari, Chris, Hanwool Albert Lee, Leonid Sinev, Janna, Zdeněk Kasner, Kiersten Stokes, and Khalid. Eleutherai/lm-evaluation-harness: v0.4.13, August 2026. URL https://doi.org/10.5281/zenodo.22214970.

Amy Y. Tsou, Christoph U. Lehmann, Jeremy Michel, Ronni Solomon, Lorraine Possanza, and Tejal Gandhi. Safe practices for copy and paste in the ehr: Systematic review, recommendations, and novel model for health it collaboration. *Applied Clinical Informatics*, 8(1):12–34, 2017. doi: 10.4338/ACI-2016-09-R-0150. URL https://pubmed.ncbi.nlm.nih.gov/28074211/.

Ping Wang, Tian Shi, and Chandan K. Reddy. Text-to-SQL generation for question answering on electronic medical records. In *Proceedings of The Web Conference 2020*, pp. 350–361. Association for Computing Machinery, 2020. doi: 10.1145/3366423.3380120. URL https://doi.org/10.1145/3366423.3380120.

Nicole G. Weiskopf, George Hripcsak, Sushmita Swaminathan, and Chunhua Weng. Defining and measuring completeness of electronic health records for secondary use. *Journal of Biomedical Informatics*, 46(5):830–836, 2013. doi: 10.1016/j.jbi.2013.06.010. URL https://pubmed.ncbi.nlm.nih.gov/23820016/.

Ran Xu, Yuchen Zhuang, Yishan Zhong, Yue Yu, Zifeng Wang, Xiangru Tang, Hang Wu, May Dongmei Wang, Peifeng Ruan, Donghan Yang, Tao Wang, Guanghua Xiao, Xin Liu, Carl Yang, Yang Xie, and Wenqi Shi. MedAgentGym: A Scalable Agentic Training Environment for Code-Centric Reasoning in Biomedical Data Science. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jHDZEUgS4r.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report, 2025. URL https://arxiv.org/abs/2505.09388.

Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, Qiaobo Chen, Yinyuting Yin, Hao Zhang, Tengfei Shi, Liang Wang, Qiang Fu, Wei Yang, and Lanxiao Huang. Mastering Complex Control in MOBA Games with Deep Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6672–6679, April 2020. doi: 10.1609/aaai.v34i04.6144. URL https://ojs.aaai.org/index.php/AAAI/article/view/6144.

Severin Ye, Xiao Kong, Xiaopeng He, Guangsu Yan, and Dongsuk Oh. CuraView: A Multi-Agent Framework for Medical Hallucination Detection with GraphRAG-Enhanced Knowledge Verification, 2026. URL https://arxiv.org/abs/2605.03476.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel, 2023. URL https://arxiv.org/abs/2304.11277.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. SWIFT: A Scalable Lightweight Infrastructure for Fine-Tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(28):29733–29735, April 2025. doi: 10.1609/aaai.v39i28.35383. URL https://ojs.aaai.org/index.php/AAAI/article/view/35383.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-Following Evaluation for Large Language Models, 2023. URL https://arxiv.org/abs/2311.07911.

APPENDIX CONTENTS

A Dataset Details **16**

A.1 Database Preprocessing 16

A.2 Dataset Composition 16

A.3 Construction Details 17

A.4 Question Vocabulary Visualization 18

A.5 Data Splits, Provenance, and Validation 18

B Human Validation of Task Quality **18**

C Experimental Details **21**

C.1 Environment and Interaction Settings 21

C.2 Models and Sampling Settings 21

C.3 Outcome Evaluation 22

C.4 Reporting and Interaction Statistics 23

C.5 Capability Evaluation 23

D Training Details **24**

D.1 Training Data and Checkpoints 24

D.2 Training Configuration 25

D.3 Self-Improvement 25

E Additional Results **26**

E.1 Performance by Noise Type 26

E.2 Performance by Scope, Test Track, and Population Size 27

E.3 Training-Data Scaling and Self-Improvement Results 28

E.4 Capability Retention 29

E.5 Repeated Evaluations 31

F Trajectory Error Analysis **32**

G Case Studies **33**

G.1 Selected Model Trajectories 33

H Prompts **35**

H.1 Agent System Prompt and Response Format 35

H.2 Evidence-Check Instruction 37

I Limitations and Scope **37**

A Dataset Details

A.1 Database Preprocessing

We rename database tables and columns to reduce reliance on memorized MIMIC-IV schemas. Descriptive table names and consistent shared identifiers also help clarify the links between patient, admission, and ICU records. Table 4 lists every table and a representative column mapping. Shared identifiers include `patientid`, `admissionid`, and `icuadmissionid`. The conversion retains all 342 source columns, their values, data types, and temporal precision. The only derived field is admission `age`, computed as `anchor_age` plus the admission year minus `anchor_year`. Fields absent from MIMIC-IV 3.1, such as date of birth or diagnosis timestamps, are not reconstructed. Source and converted tables are checked for matching row counts and contents. Both conditions use the same converted database; Noise construction changes question conditions rather than stored records. The full column mapping is provided with the code.

Table 4: Table and column renaming for MIMIC-IV 3.1. All 31 tables are listed; the final column shows one example per table.

Source table	Renamed table	Example column mapping
Hospital tables (22)		
admissions	hospitaladmissions	admittime → admitdatetime
d_hcpcs	billingprocedurecodes	code → billingcode
diagnoses_icd	admissiondiagnoses	icd_version → codeversion
d_icd_diagnoses	diagnosiscodes	icd_code → icdcode
d_icd_procedures	procedurecodes	long_title → description
d_labitems	labtesttypes	label → itemname
drgcodes	diagnosisrelatedgroups	drg_type → groupingsystem
emar_detail	medicationadministrationdetails	dose_given → dosegiven
emar	medicationadministrations	charttime → administrationdatetime
hcpcsevents	billingprocedureevents	hcpcs_cd → billingcode
labevents	labresults	valuenum → resultvalue
microbiologyevents	microbiologyresults	org_name → organismname
omr	outpatientmeasurements	result_name → measurementname
patients	demographics	anchor_age → anchorage
pharmacy	pharmacyorderdetails	medication → medicationname
poe_detail	providerorderdetails	field_name → fieldname
poe	providerorders	order_type → ordertype
prescriptions	medicationorders	drug → medicationname
procedures_icd	admissionprocedures	chartdate → proceduredate
provider	providers	provider_id → providerid
services	hospitalservices	curr_service → currentservice
transfers	patienttransfers	eventtype → transfertype
ICU tables (9)		
caregiver	caregivers	caregiver_id → caregiverid
chartevents	clinicalevents	value → textvalue
datetimeevents	clinicaldatetimeevents	value → datetimevalue
d_items	clinicalitemtypes	linksto → itemtype
icustays	icuepisodes	first_careunit → initialcareunit
ingredientevents	infusioningredients	amount → administeredingredientamount
inputevents	intakerecords	amount → administeredamount
outputevents	outputrecords	value → volume
procedureevents	clinicalprocedures	value → procedureduration

A.2 Dataset Composition

Tables 5 and 6 summarize `EHR-RobustGym` by query scope, noise type, and clinical intent. The benchmark contains 4,478 training pairs and 1,008 test pairs, with one Clean and one Noise question

in each pair. Its 51 construction programs cover six primary clinical intents. Query scope and noise type define the sampling strata, whereas clinical intents describe task coverage. Program counts in the All column are distinct unions of the training and test programs rather than sums. Each pair has one primary clinical intent; background bars in Table 6 show relative pair counts within that table.

Table 5: Composition of `EHR-RobustGym` by query scope and noise type.

	Question pairs			Programs		
Noise type	Train	Test	**All**	Train	Test	All
Patient-level queries						
Record-level	750	168	**918**	3	4	4
Value-level	750	168	**918**	3	4	4
Query-level	750	168	**918**	6	7	7
Population-level queries						
Record-level	749	168	**917**	10	11	11
Value-level	747	168	**915**	12	13	13
Query-level	732	168	**900**	11	12	12
Total	**4,478**	**1,008**	**5,486**	**45**	**51**	**51**

Table 6: Coverage of the six primary clinical intents.

	Question pairs			Programs		
Clinical intent	Train	Test	**All**	Train	Test	All
Demographics & Tracking	392	130	522	6	7	7
Vital Signs & Observations	214	33	247	4	4	4
Medications & Orders	1,465	212	1,677	10	11	11
Cost & Billing	203	23	226	3	3	3
Labs & Microbiology	1,625	390	2,015	18	20	20
Diagnoses & Procedures	579	220	799	4	6	6
Total	**4,478**	**1,008**	**5,486**	**45**	**51**	**51**

The six primary clinical intents follow the naming used in EHR-Complex (Qiao et al., 2026). We assign each pair a primary intent based on its clinical information need and requested output. Cross-domain requests retain their additional output categories. In the current corpus, 94 laboratory/microbiology–medication pairs have two output categories, while all other pairs have one. ICU datetime events concerning devices, airways, procedures, or wound care are assigned the primary intent Diagnoses & Procedures rather than Vital Signs & Observations.

A.3 Construction Details

Semantics and condition substitution. We use Kimi-K2.5 (Kimi Team et al., 2026a) to construct clinical requests, Clean-Noise pairs, and reference SQL. Construction programs deterministically instantiate the requests and execute their reference queries. Each pair changes one necessary condition while preserving the clinical topic, query scope, requested output, and other conditions. The noise label identifies the support condition that fails, not merely the type of locator changed. For example, changing an eMAR event ordinal yields Value-level Noise when the event exists but its state does not support an administered dose. Selecting an event before checking its status differs from selecting among events already satisfying that status. Population queries likewise compute within each eligible unit before aggregating across units, rather than pooling all measurement rows.

Clinical semantics. Blood-culture tasks jointly interpret specimen source and assay identity, excluding non-blood fluids in culture bottles, serological assays, and stem-cell product cultures. ICU

intake tasks combine codes for the same named therapy before computing within-stay summaries. A recorded-unit eligibility restriction is stated explicitly and is not implied by a requested reporting unit. Medication doses require a compatible administration state. Dose references normalize unit aliases and aggregate original product-component rows within the same administration event and unit before clinical joins. Distinct product rows with equal doses both contribute. Non-numeric dose expressions require explicit interpretation rather than being silently omitted.

Reference generation. A deterministic renderer converts execution results into Clean reference values, units, and record context. Noise references identify an unsupported necessary condition and distinguish it from related evidence. Population references distinguish zero eligible units from one to four. Both require `NULL`, and the latter withhold exact counts and aggregates. Lack of support is relative to the requested records and reporting rules, not absolute nonexistence in the real world. Evaluation also accepts another sufficient reason established by the agent's visible evidence.

Question rewriting and review. Kimi-K2.5 rewrites both canonical questions together using the locked evidence contract, selected substitution, required locators, and permitted language exemplars. Rewriting preserves record identities, linked conditions, data sources, units, temporal direction, and the order of filtering and ordinal selection. Exemplars supply phrasing, not clinical facts. Automatic checks reject information leakage, locator or linkage errors, altered operation order, excessive length, copied exemplar wording, and duplicate questions. Kimi-K2.5 separately reviews and repairs wording against the same contract, after which checks are repeated.

A.4 Question Vocabulary Visualization

Figure 8 summarizes the final 4,478 training and 1,008 test pairs, pooling the two splits but keeping conditions and noise types separate. We lowercase the question text, retain alphabetic tokens (including internal apostrophes and hyphens), and remove tokens of at most two letters and a fixed stopword list. Numeric identifiers, numerical values, and numeric date components therefore do not contribute. We use only questions, not SQL, reference answers, or teacher instructions. Each panel displays up to 160 words with its own frequency scaling and random seed 42. The visualization describes vocabulary rather than demonstrating absence of leakage or equal difficulty.

A.5 Data Splits, Provenance, and Validation

Split design. The test set contains 504 in-domain and 504 structure-disjoint pairs. Each track has 252 patient-level and 252 population-level pairs, with 168 pairs per noise type. The in-domain track uses training construction programs, whereas the structure-disjoint track reserves six programs, one per query-scope–noise-type stratum, with 84 pairs each (Table 7). The latter also separates program-level information-need IDs, surface-form families, SQL structures without literal values, and semantic contracts from training. These IDs are not the six clinical intents. Patient-level questions are patient-disjoint. Population questions use disjoint query contracts, but contributing patients may overlap. We check exact reference SQL and final questions across partitions and keep pairs intact.

Sources and validation. The 51 construction programs draw on 94 distinct DiSCQ chart-review question sources and 35 EHRSQL information-need templates. These mappings motivate clinical information needs rather than independently validate every generated question. Compound queries are executable MIMIC-IV operationalizations, not requests claimed to appear verbatim in either source. Reference queries and results are checked against MIMIC-IV, alongside partition and provenance checks. Split restrictions also govern rewriting exemplars, and revisions to clinical scope or reference semantics require re-execution.

B Human Validation of Task Quality

Task construction in `EHR-RobustGym` is grounded in database evidence and execution-verified reference outcomes. Each Clean question is instantiated from observed records and paired with a reference SQL query and its execution result. Its Noise counterpart is constructed by modifying one necessary question condition against the same database, followed by verification of the unmet requirement and the corresponding abstention outcome. After joint question rewriting, query replay and an independent language-model audit further check execution consistency and semantic align-

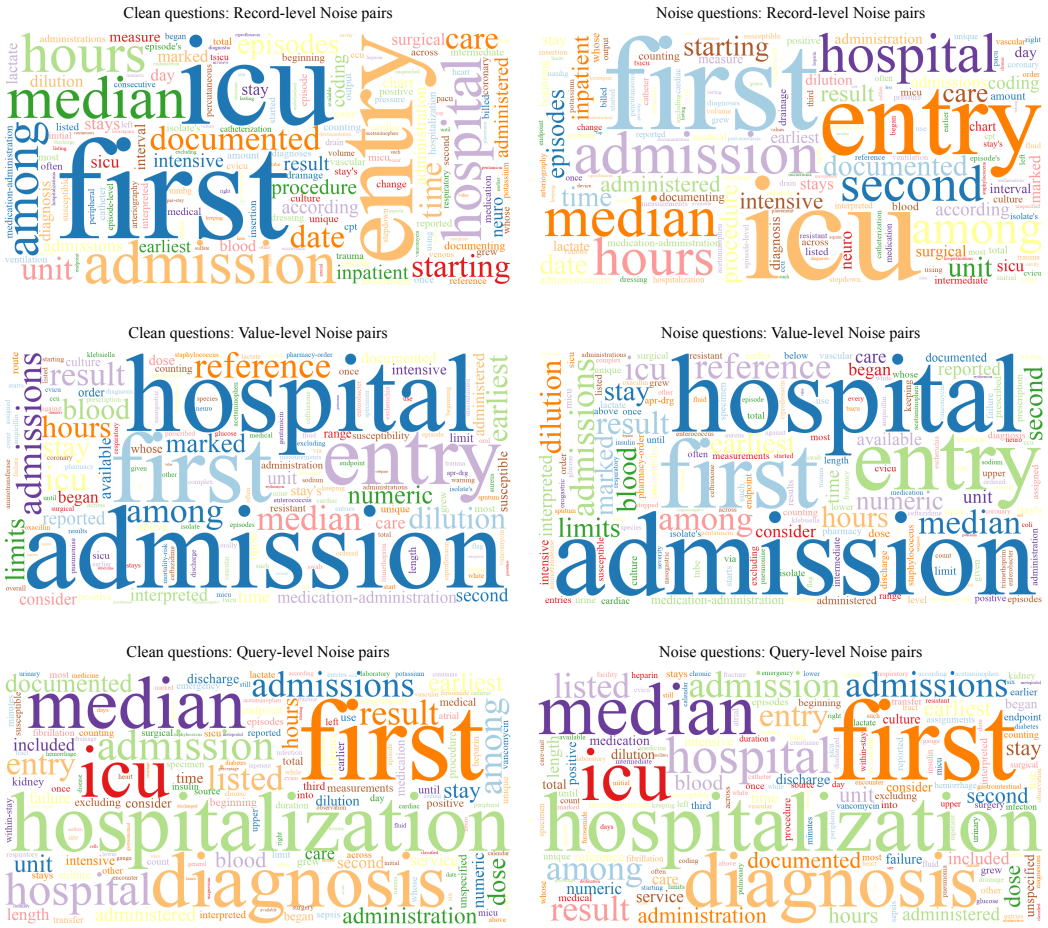


Figure 8: Question vocabulary by noise type. Rows show Record-level, Value-level, and Query-level pairs. Columns show Clean and Noise questions from the final training and test sets. Word size indicates frequency within each panel.

Table 7: Construction programs reserved for the structure-disjoint test track. Construction tasks appear below their clinical intents.

Noise type	Clinical intent / construction task	Pairs
Patient-level queries		
Record	Diagnoses & Procedures	84
Hospital admission order		
Value	Labs & Microbiology	84
Initial ICU care unit		
Query	Labs & Microbiology	84
ICU unit and lactate across admissions		
Population-level queries		
Record	Diagnoses & Procedures	84
ICU procedure records		
Value	Medications & Orders	84
Medication event status		
Query	Demographics & Tracking	84
Diagnosis and admission-source conditions		

ment. Building on these checks, we conduct an additional human audit of challenging test pairs, focusing on clinical plausibility and semantic alignment rather than SQL syntax alone.

Selection of challenging pairs. We identify candidates using the initial single-run Base evaluations of GPT-5.4 (xhigh) and Gemini-3.1-Pro under the configurations used for the main comparison. GPT-5.4 uses its default sampling configuration, while Gemini-3.1-Pro uses temperature 0.0. A pair is included if it contains at least one question that both models fail to solve. This criterion identifies 109 jointly failed Clean questions and 281 jointly failed Noise questions, with 46 pairs appearing in both groups. The candidate set contains $109 + 281 - 46 = 344$ distinct pairs. We review all 344 pairs, covering 688 questions, including counterparts that do not themselves meet the joint-failure criterion. This screening identifies challenging cases without certifying task validity.

Annotation procedure. Each selected pair is independently reviewed by two annotators with medical AI backgrounds and experience in EHR data analysis and SQL-based database querying. Annotators are shown both natural-language questions, the reference SQL queries, execution results, extracted answers, relevant schema information, and the database evidence used during construction. They assess whether each question faithfully expresses the intended clinical information need and whether the reference outcome is justified by the available records. For Clean questions, the review focuses on the evidence supporting the requested answer. For Noise questions, it focuses on whether the modified condition leads to the stated unmet requirement that justifies abstention. A pair is marked valid only if both questions and their pairing satisfy all four criteria below:

- **Question clarity and semantic alignment.** Both questions are clear, and reference queries implement the intended clinical concepts, temporal constraints, query scopes, and computations.
- **Clean answer support.** The reference execution and supporting records justify the Clean answer, including the relevant values, units, and aggregation rules where applicable.
- **Noise abstention support.** The available records support the stated unmet requirement and justify abstention under the question's intended semantics. An empty or `NULL` query result alone is insufficient without evidence establishing why the requested answer cannot be supported.
- **Pair consistency.** The two questions preserve the intended clinical goal and differ in the designated condition. Question rewriting does not introduce unintended changes to other constraints, and the difference between answering and abstaining follows from the intended modification.

Annotators assign a binary pair-level validity label: 1 indicates that both questions and their pairing satisfy all criteria, while 0 indicates a substantive concern requiring further inspection. Minor wording issues that do not affect task interpretation, reference logic, or the final outcome are recorded separately and are not counted as invalid. Disagreements are resolved through joint adjudication after inspecting the questions, reference queries, execution results, and supporting evidence.

Results. Table 8 summarizes the human validation results. After adjudication and before correction, 317 of the 344 audited pairs are judged valid, corresponding to a pair-level validity rate of 92.15%. The remaining 27 pairs (7.85%) contain substantive issues requiring correction. This rate describes the quality of the original audited pairs, rather than their validity after issues have been corrected. Because the audit targets pairs containing jointly failed questions, the result provides a targeted assessment of challenging cases rather than a random-sample estimate for the benchmark.

Inter-annotator agreement. Before adjudication, the two annotators agree on 338 of the 344 binary pair-level labels, yielding 98.26% raw agreement. Disagreements on the remaining 6 pairs (1.74%) are resolved through joint adjudication. Both the validity rate and inter-annotator agreement use pairs as the unit of analysis, with a denominator of 344, although all 688 questions are inspected.

Correction and final evaluation. All 27 pairs with confirmed substantive issues are corrected rather than removed. We revise affected question wording, reference queries, or reference outcomes, then replay the queries and recheck both questions against the annotation criteria. This process verifies that the Clean answer remains supported, the Noise abstention is justified, and the intended relationship between the two questions is preserved. We rerun the affected tasks for all evaluated agents under the same experimental settings, including repeated-run evaluations where applicable.

Table 8: Human validation summary for challenging EHR-RobustGym test pairs. Validity is assessed after adjudication but before correction; agreement is measured before adjudication.

Item	Count (rate)
Jointly failed Clean questions	109
Jointly failed Noise questions	281
Pairs with joint failure on both sides	46
Audited pairs / questions	344 / 688
Valid pairs before correction	317 / 344 (92.15%)
Pairs requiring substantive correction	27 / 344
Pairs with initial reviewer agreement	338 / 344 (98.26%)
Pairs with initial reviewer disagreement	6 / 344

Updated scores and trajectories replace their pre-correction versions, and affected aggregate results are recomputed. All reported EHR-RobustGym results correspond to the finalized test set.

C Experimental Details

C.1 Environment and Interaction Settings

Each episode uses a task-scoped, read-only DuckDB connection named `conn`. Python variables, imports, and intermediate results persist across actions within the episode. A new episode starts with fresh execution state against the same database. The agent sees printed outputs and execution feedback, with omitted portions of long outputs explicitly marked. Table 9 summarizes the limits.

Table 9: Environment and interaction settings.

Setting	Value or behavior
Input limit	65,536 tokens in the accumulated model input
Per-response limit	4,096 output tokens under the model service's completion-token accounting
Interaction limit	20 accepted model responses, including invalid responses and final submission
Tool timeout	120 seconds of execution, excluding queue waiting

The question triggers the first turn. Each accepted model completion counts as one turn, including invalid or empty responses and final submission. Tool observations and transport retries without a completion do not add turns. Input limits use the tokenizer locally and reported API input usage for external models. Exhausting the input or turn limit without a valid final answer scores zero.

Model-generated SQL/Python errors and ordinary tool timeouts remain in the trajectory as feedback and do not alone determine task success. Infrastructure or judge failures must be resolved before evaluation is reported and are not converted to zero scores or removed from the denominator. During online training, unresolved failures stop the affected update rather than supplying a false reward.

C.2 Models and Sampling Settings

Table 10 lists model identifiers and reasoning settings. API-based denotes access, not whether weights are public. The five local open-weight models use vLLM (Kwon et al., 2023) for EHR evaluation.

The seven API-based models and Qwen3-32B each have four Base runs on the same 1,008 Clean-Noise test pairs: two at temperature 0 and two at 0.6, except GPT-5.4, which uses four service-default xhigh runs. The main tables use the first Base run, and PE, other local Base models, and trained checkpoints use one temperature-zero run per setting (service-default xhigh for GPT-5.4).

Table 10: Model identifiers and reasoning settings.

Display name	Model identifier	Reasoning setting
API-based models		
GPT-5.4 (xhigh)	`gpt-5.4`	xhigh
GPT-4.1	`gpt-4.1-2025-04-14`	No override
Gemini-3.1-Pro	`gemini-3.1-pro-preview`	No override
Claude Sonnet 4.6	`claude-sonnet-4-6`	Thinking disabled
Kimi-K2.5	`Kimi-K2.5`	Thinking disabled
DeepSeek-V3.2	`DeepSeek-V3.2-Exp`	Thinking disabled
Qwen3-235B-A22B	`Qwen3-235B-A22B-Instruct-2507`	Thinking disabled
Locally deployed open-weight models		
Qwen3-4B/8B/14B/32B	Corresponding `Qwen3-` instruction checkpoint	Thinking disabled
Llama-3.1-8B	`Llama-3.1-8B-Instruct`	Native instruction

C.3 Outcome Evaluation

Evaluation and online training use the same outcome scorer. Clean questions use string matching to compare the submitted Answer with the execution-verified reference answer. Noise questions require an Answer of `NULL`, with an LLM judge checking whether visible tool evidence supports abstention. The final Reasoning is not a separate task-score component and cannot replace an incorrect Answer or missing tool evidence. Trajectory error analysis separately examines interaction and reporting errors.

Answer extraction and matching. The scorer extracts the complete Answer. Ambiguous or competing Answer fields fail. Clean answer matching preserves the requested values, units, and associated records. For Noise questions, after whitespace, markup, and case normalization, the whole Answer must be `NULL` or `NULL.`. A mixed partial answer is not accepted as a NULL submission.

Unrequested auxiliary counts may be omitted. Supplied counts for the requested result must agree in both population and counting unit. The size of an eligible cohort is different from the number of admissions reporting a modal value. Clearly identified supplementary results for another population cannot replace the requested answer and are assessed against their own evidence and reporting requirements in error analysis. Unresolved alternatives for the same requested field do not constitute a complete answer.

Evidence for abstention. After the NULL gate, the scorer checks the ordered tool calls and feedback shown to the agent. A trajectory without any executed tool call fails directly. Otherwise, one valid GPT-5.2 verdict determines Abstention Validity and cites the supporting tool turns. The question and evidence contract define the required scope. Reference SQL provides an accepted implementation but does not supply evidence acquired by the agent. Codebook definitions and precomputed record-scope equivalences clarify names and linkage only within their stated domains. Evidence verification uses executed tool calls and observations actually shown during the interaction. Agent-generated tables, unprinted variable contents, and reference replays cannot replace missing observations. Equivalent queries and stepwise computations are accepted when their visible outputs establish a sufficient reason for abstention. A correctly scoped empty result or a complete event list lacking the requested ordinal can suffice. A failed query, wrong-scope empty result, or truncated preview alone cannot. Earlier sufficient evidence remains usable after a later failed query.

Population reporting and numerical zero. Population summaries with zero or one to four eligible units require withholding the aggregate. The latter also withhold the exact count. A visible count for a broader population can establish that the requested subgroup is below the threshold only when the subgroup relation is valid. Such a bound does not establish an exact subgroup count or distinguish zero from one to four. A broader population above the threshold does not establish that the requested subgroup is reportable. First/last selection is not monotone, so enlarging the source before selecting an event does not automatically yield a valid bound on the requested population.

A defined, reportable zero is a supported value, not the NULL marker. A zero count showing missing records cannot replace an unavailable measurement or undefined aggregate. Tasks concern attributes, ordered events, and summaries, rather than standalone count or existence questions. Population disclosure restrictions appear in the agent prompt and reporting-error taxonomy.

Judge configuration and failures. For Noise evidence verification, GPT-5.2 uses high reasoning effort, service-default sampling, and an 8,192-token completion limit. One format-repair response is permitted after an invalid judgment, but a valid verdict is not resampled. Transient failures may be retried. Unresolved infrastructure or judge failures raise an error rather than assigning zero task success to the agent.

C.4 Reporting and Interaction Statistics

Tables 2 and 3 use 1,008 questions per condition. In column order, the six intents contain 130, 33, 212, 23, 390, and 220 questions. Avg. weights by question count. Scores use the first complete temperature-zero run, or first service-default xhigh run for GPT-5.4. Base has no task-specific prompting or training. For mitigation, SFT uses 1,000 demonstrations over 250 updates, and each continuation adds 250 updates independently from SFT-250. Size groups use total parameter counts.

Success rates are computed from integer successes over the fixed question set. The Noise NULL Rate uses all Noise questions as its denominator. Conditional Abstention Validity uses only submitted NULL answers and is undefined when there are none. Thus Noise success is the product of NULL Rate and conditional validity before rounding. Aggregated validity is weighted by NULL counts, not by averaging subgroup percentages. Displayed scores and Δ values are independently rounded from exact quantities. Light-red cells indicate displayed success below 5% for readability, not significance. Bold Δ includes exact ties. Training comparisons use one training seed, and smoothed update curves do not represent uncertainty across independent training seeds.

Table 11: Interaction statistics and their counting units.

Statistic	Counted event	Rate denominator or exclusion
Model responses and tool interaction		
Interaction Turns	Accepted model completions	Includes invalid and final responses
Tool Calls	Dispatched code actions	Excludes final submissions
Response Format Errors	Responses rejected by the action parser	Accepted model responses
Tool Call Format Errors	Malformed responses attempting a tool action	Dispatched calls plus malformed tool requests
Tool Execution Errors	Failed dispatched calls	Dispatched calls, excluding timeouts from the numerator
Tool Timeouts	Calls interrupted at the execution limit	Dispatched calls

A successful tool call may contain a caught query error, and an empty result is not automatically an error. Figure 7(a) groups first-run Base trajectories from eight models. Within each condition, all four outcome/error shares use 1,008 questions as the denominator. The groups without execution errors can include trajectories that make no tool calls or encounter timeouts or response-format errors. Interaction statistics are descriptive and do not contribute extra reward terms.

C.5 Capability Evaluation

We evaluate eight general and medical benchmarks with 10,072 questions per checkpoint. The reported mean weights these eight benchmarks equally, not by question count. Table 12 lists subsets and metrics, including five external EHR benchmarks. Figure 6(a) reports Qwen3-14B's task success on these external benchmarks before and after SFT and subsequent GRPO. These external benchmarks use their respective final-answer string-matching criteria.

For the eight general and medical benchmarks, sampling is without replacement using seed 42, retaining complete evaluation splits smaller than 2,000 questions. MedMCQA preserves subject proportions, and PubMedQA preserves answer-label proportions. GSM8K uses five fixed training

Table 12: Evaluation subsets and metrics for general, medical, and external EHR benchmarks.

Benchmark	Evaluation subset	Questions	Metric
General capabilities			
ARC-Challenge (Clark et al., 2018)	Test	1,172	Length-normalized accuracy
HellaSwag (Zellers et al., 2019)	Validation	2,000	Length-normalized accuracy
WinoGrande XL (Sakaguchi et al., 2020)	Validation	1,267	Accuracy
GSM8K (Cobbe et al., 2021)	Test	1,319	Exact match, strict extraction
IFEval (Zhou et al., 2023)	Official prompts (named train)	541	Prompt-level strict accuracy
Medical question answering			
MedQA (English) (Jin et al., 2021)	Four-option test	1,273	Accuracy
MedMCQA (Pal et al., 2022)	Validation	2,000	Accuracy
PubMedQA PQA-L (Jin et al., 2019)	Original test	500	Accuracy
External EHR benchmarks			
TREQS (Wang et al., 2020)	Test subset	613	Task success rate
EHRSQL (eICU) (Lee et al., 2022)	Test subset	371	Task success rate
EHRSQL (MIMIC-III) (Lee et al., 2022)	Test subset	421	Task success rate
EHR-SeqSQL (Ryu et al., 2024)	Test subset	293	Task success rate
EHR-Complex (Qiao et al., 2026)	Fixed subset	500	Task success rate

examples. Compared checkpoints of each model use that model's Base tokenizer, and all comparisons share the same sample identities. The evaluator is lm-evaluation-harness 0.4.13 (Sutawika et al., 2026), using native chat templates without an extra system instruction. Qwen3 thinking is disabled, and few-shot examples are included within one user prompt rather than as separate turns.

Multiple-choice tasks compare conditional continuation log-likelihoods. ARC-Challenge and HellaSwag use the harness's character-length normalization. GSM8K uses greedy generation with a 1,024-token limit and strict extraction after `####`. IFEval uses greedy generation with a 1,280-token limit and requires all instructions in a prompt to pass for prompt-level strict accuracy. MedQA and PubMedQA require field-name conversion to the native templates, without changing questions, options, or labels. Answer-bearing auxiliary fields do not enter model inputs.

The capability evaluations use a documented IFEval implementation difference: the letter-frequency check counts the specified literal character, including punctuation, instead of replacing a nonalphabetic target. Only this check is changed; the native result processor is otherwise used. Two of the 541 prompts are affected by this character-handling rule. Saved Base, SFT, and SFT-initialized continuation evaluations use this rule. Original generated responses are preserved when metrics are recomputed. Raw scores for the main capability comparisons appear in Appendix E.4.

D Training Details

D.1 Training Data and Checkpoints

The selected training trajectories (demonstrations) come from GPT-5.4, GPT-5.5 (OpenAI, 2026b), GPT-5.6-sol (OpenAI, 2026c), Gemini-3.1-Pro, and Kimi K3 (Kimi Team et al., 2026b). The GPT teachers use xhigh reasoning with service-default sampling, Gemini uses temperature 0.6, and Kimi uses service-default sampling. Of the 2,000 demonstrations, 1,150 are unguided and 850 use teacher-only reference explanations and queries. Candidates must pass task evaluation and include successful tool-based evidence acquisition. Recovered tool errors are allowed, but three consecutive equivalent failed code executions are excluded. Guided candidates additionally pass naturalness and self-containment checks by Gemini-3.1-Pro at temperature zero. Noise curation by Gemini-3.1-Pro requires a quality score of at least four out of five, cited supporting tool feedback, and no unresolved issues. Student inputs preserve the original dialogue but exclude teacher-only guidance, separately returned reasoning fields, and evaluator metadata.

We randomly sample 1,000 Clean-Noise pairs from the training set, comprising 2,000 questions, with one verified demonstration per question. SFT-250 processes the first 1,000 demonstrations in a fixed shuffled order over 250 updates, half an epoch of the full dataset. DPO, PPO, and GRPO independently continue from this checkpoint for another 250 updates on the remaining 1,000 questions, comprising 495 Clean and 505 Noise requests. The stages share no question IDs, although counterparts from the same pair may occur in different stages. All five students use the same fixed preference set for offline DPO, pairing verified teacher demonstrations with scored unsuccessful trajectories from the five local models for the same question. Online PPO and GRPO collect fresh rollouts through environment interaction. Each branch starts a fresh optimizer and is evaluated at its specified endpoint without test-score-based checkpoint selection. See Table 13 for settings. We also retain an additional SFT-500 checkpoint after one epoch over all 2,000 demonstrations. Its capability scores and Qwen3-8B training-data scaling appear in Appendices E.4 and E.3.

D.2 Training Configuration

Table 13: Training settings shared across methods and specific to each stage.

Scope	Settings
All methods	Full-parameter training; 250 updates; policy learning rate 10^{-6}; AdamW (Loshchilov & Hutter, 2019) ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.01); constant schedule without warm-up; BF16; seed 42.
SFT	Instruction initialization; batch of four dialogues; ms-swift 4.5.3 (Zhao et al., 2025); 69,632 encoded tokens without truncation or packing.
DPO	SFT-250 policy and frozen reference; batch of four preferences; $\beta = 0.1$; the same framework and token limit as SFT.
PPO and GRPO	SFT-250 policy and frozen reference; verl (Sheng et al., 2025); four questions per update, eight rollouts per question; temperature 0.6 and top-p 1.0; initial-prompt limit 8,192 tokens and engine context 69,632 tokens.
Online policy loss	KL coefficient 0.001; actor clip 0.2; dual clip 3 (Ye et al., 2020); entropy coefficient 0; low-variance KL estimator; log-ratio bounds $[-20, 20]$ and KL cap 10.
PPO critic	Learning rate 10^{-5}; $\gamma = \lambda = 1$; no warm-up; value clip 0.5.
GRPO normalization	Sample standard deviation with $\epsilon = 10^{-6}$.

Note. DPO, PPO, and GRPO independently continue from SFT-250. Interaction limits are listed in Table 9.

Losses apply only to assistant tokens, with tool observations retained as context and inserted template-only tokens masked. DPO uses summed assistant-token log-probabilities without length normalization or an auxiliary SFT loss. PPO and GRPO use token-mean reduction and KL regularization in the policy loss, separately from the terminal task reward. PPO initializes the critic backbone from SFT-250 with a new scalar value head. PPO policy advantages are whitened, while value targets use unwhitened advantages and skip tool-observation positions. SFT and DPO use eight GPUs with ZeRO-3 (Rajbhandari et al., 2020) and sequence parallelism 4. GRPO uses 16 GPUs with FSDP (Zhao et al., 2023). PPO uses FSDP on eight GPUs for Qwen3-4B/8B, 16 for Qwen3-14B and Llama, and 32 for Qwen3-32B. Training-data scaling settings appear in Appendix E.3.

D.3 Self-Improvement

Starting from SFT-250, each model undergoes RS+SFT, DPO, and one additional DPO iteration. Before each stage, the current policy samples eight trajectories per question from the remaining 1,000-question pool at temperature 0.6. RS+SFT retains one successful trajectory per selected question, while DPO and iDPO retain one successful and one unsuccessful trajectory for the same question. Each stage trains for one epoch using the shared outcome scorer for selection, a fresh optimizer, batch size four, and seed 42. Other settings follow SFT or DPO in Table 13, with the DPO reference reset to the stage-start checkpoint. Table 14 reports the selected question counts.

Table 14: Number of questions selected at each self-improvement stage.

Model	RS+SFT	DPO	iDPO
Qwen3-4B	683	614	519
Qwen3-8B	736	670	647
Qwen3-14B	832	675	702
Qwen3-32B	864	685	631
Llama-3.1-8B	882	586	244

Note. Each stage trains for one epoch on its selected questions. Bars share a scale of 1,000 candidate questions.

E Additional Results

Tables 15 and 16 use the same evaluations as Tables 2 and 3. Here, SFT denotes SFT-250; +DPO, +PPO, and +GRPO are independent 250-update continuations. Reporting follows Appendix C.4.

E.1 Performance by Noise Type

Table 15 reports noise types and their Clean counterparts, with 336 questions per condition and type. Noise type describes the unsupported condition, not the agent error. Noise success requires NULL and sufficient evidence; high conditional abstention validity does not imply high success.

Table 15: Performance by noise type and on paired Clean questions.

		Record-level		Value-level		Query-level	
Model	**Method**	Success	NR / AV	Success	NR / AV	Success	NR / AV
GPT-5.4 (xhigh)	Base	73.5/58.3	66.7/87.5	78.3/65.8	74.4/88.4	83.0/43.2	52.1/82.9
	PE	78.0/58.6	67.3/87.2	79.5/65.5	72.9/89.8	85.1/41.7	50.0/83.3
GPT-4.1	Base	51.2/40.5	79.8/50.7	29.2/23.8	84.5/28.2	32.7/21.1	77.4/27.3
	PE	51.2/39.3	82.4/47.7	27.7/22.0	85.1/25.9	36.0/22.6	78.3/28.9
Gemini-3.1-Pro	Base	82.1/61.9	66.1/93.7	78.6/62.2	66.1/94.1	89.6/51.2	53.9/95.0
	PE	85.4/66.4	70.8/93.7	79.2/67.9	72.9/93.1	90.2/50.6	51.8/97.7
Claude Sonnet 4.6	Base	64.6/39.9	42.3/94.4	64.3/26.2	29.5/88.9	58.6/12.8	15.2/84.3
	PE	73.8/46.1	50.0/92.3	68.8/35.7	41.1/87.0	61.6/15.2	15.8/96.2
Kimi-K2.5	Base	83.0/41.7	42.6/97.9	78.9/48.5	48.5/100.0	85.4/21.1	21.1/100.0
	PE	83.6/46.1	46.7/98.7	75.0/49.7	50.0/99.4	86.0/25.6	25.9/98.9
DeepSeek-V3.2	Base	75.3/66.1	72.9/90.6	70.2/52.1	61.0/85.4	78.3/42.3	47.0/89.9
	PE	79.5/69.0	75.9/91.0	68.8/54.2	60.4/89.7	80.1/43.2	48.2/89.5
Qwen3-235B-A22B	Base	33.0/9.2	16.7/55.4	14.0/6.5	14.9/44.0	3.0/0.9	4.5/20.0
	PE	37.2/13.7	24.4/56.1	15.2/7.1	14.9/48.0	1.2/2.1	8.3/25.0
Qwen3-4B	Base	0.0/0.0	52.7/0.0	0.3/0.0	27.7/0.0	3.3/0.0	18.8/0.0
	PE	0.3/0.0	58.3/0.0	0.6/0.0	39.3/0.0	2.7/0.0	29.2/0.0
	SFT	42.3/18.5	33.3/55.4	31.3/24.1	37.8/63.8	18.2/17.3	42.3/40.8
	+DPO	36.9/7.4	8.0/92.6	37.2/10.1	12.2/82.9	28.0/7.4	9.5/78.1
	+PPO	45.2/38.4	55.1/69.7	41.4/36.6	49.4/74.1	31.5/27.1	47.0/57.6
	+GRPO	50.3/45.5	74.7/61.0	48.5/36.3	61.3/59.2	43.5/25.6	48.5/52.8
Qwen3-8B	Base	9.2/0.9	13.7/6.5	0.6/0.6	16.1/3.7	0.6/0.0	14.9/0.0
	PE	4.5/1.8	17.6/10.2	0.3/1.8	22.6/7.9	0.3/0.0	34.8/0.0
	SFT	39.6/25.0	52.1/48.0	34.2/25.6	54.8/46.7	27.7/17.9	56.3/31.7
	+DPO	26.8/31.0	40.5/76.5	22.9/33.3	41.1/81.2	20.8/37.5	48.5/77.3

Continued on next page

Table 15 (continued)

Model	Method	Record-level		Value-level		Query-level	
		Success	NR / AV	Success	NR / AV	Success	NR / AV
	+PPO	33.6/22.9	24.4/93.9	33.3/22.0	27.1/81.3	22.6/23.5	29.2/80.6
	+GRPO	52.7/36.6	52.7/69.5	51.5/47.0	54.8/85.9	51.2/34.8	50.3/69.2
Qwen3-14B	Base	20.5/16.7	67.6/24.7	12.8/0.9	67.0/1.3	3.0/0.0	72.0/0.0
	PE	9.2/8.6	91.1/9.5	13.7/1.8	82.4/2.2	1.5/0.6	90.5/0.7
	SFT	48.8/50.9	76.8/66.3	45.2/28.3	58.0/48.7	33.0/25.0	66.7/37.5
	+DPO	43.2/56.8	74.7/76.1	44.0/51.2	75.3/68.0	42.0/45.5	66.7/68.3
	+PPO	74.4/67.6	78.6/86.0	70.8/65.8	75.3/87.4	66.1/54.5	64.6/84.3
	+GRPO	71.4/73.2	79.5/92.1	62.5/42.6	48.5/87.7	69.6/48.5	55.7/87.2
Qwen3-32B	Base	6.8/1.2	20.5/5.8	6.3/0.6	17.9/3.3	4.8/0.0	22.6/0.0
	PE	3.0/0.9	48.8/1.8	5.7/0.3	32.4/0.9	5.7/0.6	31.8/1.9
	SFT	52.1/43.8	71.4/61.3	49.1/44.9	70.8/63.4	45.2/32.1	64.3/50.0
	+DPO	49.7/61.0	90.5/67.4	51.8/57.7	85.7/67.4	47.0/45.5	83.0/54.8
	+PPO	73.5/69.0	81.5/84.7	61.6/55.7	72.3/77.0	53.3/50.9	67.3/75.7
	+GRPO	63.4/37.8	47.3/79.9	61.9/39.9	45.2/88.2	63.1/35.4	42.9/82.6
Llama-3.1-8B	Base	0.0/0.0	9.2/0.0	0.0/0.9	32.7/2.7	0.3/0.0	21.1/0.0
	PE	0.0/0.0	11.9/0.0	0.9/0.0	16.4/0.0	0.3/0.0	17.3/0.0
	SFT	53.3/33.6	52.4/64.2	53.9/39.3	47.9/82.0	50.3/22.6	43.8/51.7
	+DPO	35.1/39.6	60.7/65.2	19.9/54.8	78.6/69.7	20.8/41.4	69.3/59.7
	+PPO	41.4/24.4	66.4/36.8	28.9/25.3	65.2/38.8	5.1/19.6	70.2/28.0
	+GRPO	48.2/32.7	54.8/59.8	51.8/50.0	59.2/84.4	47.3/36.3	50.0/72.6

Note. All values are percentages. Success is Clean / Noise, with 336 questions per condition and noise type. NR is the NULL rate among Noise questions; AV is abstention validity conditional on NULL. marks success below 5%; marks parameter training.

E.2 Performance by Scope, Test Track, and Population Size

Table 16 reports patient- and population-level questions and the in-domain and structure-disjoint test tracks. Each partition contains 504 questions per condition. The two tracks differ in task composition, so their score difference does not isolate the causal effect of structural generalization. Table 16 distinguishes zero eligible units from one to four eligible units. The latter accounts for 260 of 504 population Noise questions. Both require evidence-grounded abstention, but only the former establishes that no units qualify. A small eligible population does not imply missing records.

Table 16: Task success by query scope, test track, and eligible population size.

Model	Method	Query scope		Test track		Population
		Patient	Population	In-domain	Disjoint	Zero / 1–4
GPT-5.4 (xhigh)	Base	88.5/65.5	68.1/46.0	79.4/48.8	77.2/62.7	72.1/21.5
	PE	91.1/66.9	70.6/43.7	80.2/47.2	81.5/63.3	73.8/15.4
GPT-4.1	Base	56.2/40.7	19.2/16.3	39.1/25.0	36.3/31.9	25.4/7.7
	PE	56.3/41.1	20.2/14.9	39.9/23.8	36.7/32.1	25.0/5.4
Gemini-3.1-Pro	Base	94.8/78.6	72.0/38.3	83.1/49.2	83.7/67.7	38.9/37.7
	PE	96.4/79.8	73.4/43.5	85.1/53.6	84.7/69.6	47.1/40.0
Claude Sonnet 4.6	Base	76.2/35.7	48.8/16.9	63.3/22.0	61.7/30.6	29.9/4.6
	PE	81.3/43.7	54.8/21.0	71.2/25.2	64.9/39.5	38.9/4.2
Kimi-K2.5	Base	97.2/59.5	67.7/14.7	82.9/28.4	81.9/45.8	27.9/2.3
	PE	98.8/63.5	64.3/17.5	83.5/31.9	79.6/49.0	33.6/2.3

Continued on next page

Table 16 (continued)

		Query scope		Test track		Population
Model	**Method**	Patient	Population	In-domain	Disjoint	Zero / 1–4
DeepSeek-V3.2	Base	93.7/56.3	55.6/50.6	75.0/49.6	74.2/57.3	49.6/51.5
	PE	93.8/58.5	58.3/52.4	75.2/49.8	77.0/61.1	54.5/50.4
Qwen3-235B-A22B	Base	31.3/10.5	2.0/0.6	20.6/6.9	12.7/4.2	0.8/0.4
	PE	34.7/15.1	1.0/0.2	21.6/9.9	14.1/5.4	0.4/0.0
Qwen3-4B	Base	1.8/0.0	0.6/0.0	1.8/0.0	0.6/0.0	0.0/0.0
	PE	1.8/0.0	0.6/0.0	2.4/0.0	0.0/0.0	0.0/0.0
	SFT	50.4/31.0	10.7/8.9	32.3/19.6	28.8/20.2	15.6/2.7
	+DPO	59.1/12.3	8.9/4.4	33.1/10.5	34.9/6.2	4.5/4.2
	+PPO	69.4/56.3	9.3/11.7	34.7/39.7	44.0/28.4	20.5/3.5
	+GRPO	77.2/49.8	17.7/21.8	45.8/34.9	49.0/36.7	32.4/11.9
Qwen3-8B	Base	6.9/1.0	0.0/0.0	2.8/0.8	4.2/0.2	0.0/0.0
	PE	3.4/2.2	0.0/0.2	1.8/1.2	1.6/1.2	0.4/0.0
	SFT	56.3/29.8	11.3/15.9	36.1/24.8	31.5/20.8	21.7/10.4
	+DPO	36.5/50.0	10.5/17.9	18.1/33.5	29.0/34.3	15.6/20.0
	+PPO	52.4/36.3	7.3/9.3	28.2/28.8	31.5/16.9	8.6/10.0
	+GRPO	80.0/52.4	23.6/26.6	52.8/45.6	50.8/33.3	29.1/24.2
Qwen3-14B	Base	23.2/11.7	1.0/0.0	4.0/3.4	20.2/8.3	0.0/0.0
	PE	14.5/6.3	1.8/1.0	5.2/3.4	11.1/4.0	0.8/1.2
	SFT	66.3/52.0	18.5/17.5	46.0/33.1	38.7/36.3	22.5/12.7
	+DPO	57.7/74.0	28.4/28.4	48.8/50.4	37.3/52.0	25.4/31.2
	+PPO	87.3/73.2	53.6/52.0	68.3/61.5	72.6/63.7	63.5/41.2
	+GRPO	89.5/65.5	46.2/44.0	67.7/57.1	68.1/52.4	36.9/50.8
Qwen3-32B	Base	9.3/0.8	2.6/0.4	6.5/0.6	5.4/0.6	0.8/0.0
	PE	6.9/0.8	2.6/0.4	7.1/0.6	2.4/0.6	0.8/0.0
	SFT	67.7/55.8	30.0/24.8	49.2/40.5	48.4/40.1	31.6/18.5
	+DPO	76.0/78.6	23.0/31.0	52.0/52.2	47.0/57.3	37.3/25.0
	+PPO	85.3/72.4	40.3/44.6	63.1/60.5	62.5/56.5	44.3/45.0
	+GRPO	81.7/46.2	43.8/29.2	62.7/41.5	62.9/33.9	23.0/35.0
Llama-3.1-8B	Base	0.0/0.0	0.2/0.6	0.2/0.2	0.0/0.4	1.2/0.0
	PE	0.6/0.0	0.2/0.0	0.4/0.0	0.4/0.0	0.0/0.0
	SFT	77.0/37.9	28.0/25.8	55.2/36.9	49.8/26.8	28.3/23.5
	+DPO	34.1/71.4	16.5/19.0	29.8/46.6	20.8/43.8	14.8/23.1
	+PPO	48.8/40.3	1.4/6.0	23.0/30.6	27.2/15.7	7.4/4.6
	+GRPO	65.5/43.8	32.7/35.5	52.0/44.4	46.2/34.9	40.6/30.8

Note. All values are percentages. Scope and track columns report Clean / Noise success, with 504 questions per condition in each partition. The last column reports Noise success for zero / one to four eligible units, with 244 / 260 questions. Colors follow Table 15.

E.3 Training-Data Scaling and Self-Improvement Results

Training-data scaling protocol. Qwen3-8B uses nested subsets of 250, 500, 750, and 1,000 training pairs, selected with seed 42. Each subset balances query scopes and covers all 45 training programs. Paired Mixed retains both demonstrations per pair, while Clean-only and Noise-only retain the corresponding question only. Every point starts from the instruction checkpoint and trains for one epoch with the SFT settings in Table 13. Final partial batches are retained, giving 63 and 188 updates for 250 and 750 single-condition demonstrations. Paired Mixed 100% reuses SFT-500 after all 2,000 demonstrations, not an independent run. At equal pair fractions, Mixed uses twice as many demonstrations and approximately twice as many updates, so equal-example comparisons instead match Mixed 25% with single-condition 50%, or Mixed 50% with single-condition 100%.

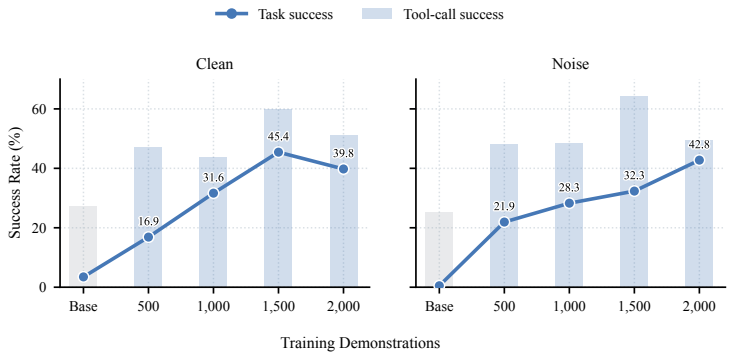


Figure 9: Paired Mixed SFT scaling on Qwen3-8B. Lines show task success; bars show successful executions divided by all tool calls, including errors and timeouts. Each selected pair contributes one Clean and one Noise demonstration; each checkpoint is trained for one epoch. Base uses no training demonstrations.

The Mixed setting in Figure 4(b) uses one demonstration from each of the same 1,000 pairs as the full Clean-only and Noise-only datasets. We assign 500 pairs to Clean and the remaining 500 to Noise, stratifying by query scope, noise type, and training program with seed 42. Each condition contains 250 patient-level and 250 population-level demonstrations. The demonstrations and pair order are unchanged before the same sampler shuffle. All three runs therefore use 1,000 demonstrations, the same pair coverage, and 250 updates; their token counts need not be identical. This run starts from the original instruction checkpoint and is distinct from both the 500-complete-pair Mixed subset and SFT-250 after 1,000 shuffled questions from the 2,000-example dataset. The 250-, 500-, and 750-demonstration Mixed subsets retain the fixed arm assignment and the same pairs as the single-condition subsets, with Clean/Noise counts of 127/123, 251/249, and 377/373. Each subset has equal numbers of patient-level and population-level requests and starts from the original instruction checkpoint for one epoch, giving 63, 125, and 188 updates. Figure 4(b) includes only these one-demonstration-per-pair settings. Figure 9 separately shows the paired Mixed scaling results.

Figure 4(b,c) and Figure 9 report results on the same 1,008 Clean and 1,008 Noise questions. All training results use a single seed. Table 17 reports training updates for sequential self-improvement. At an equal budget of 1,000 demonstrations drawn from the same 1,000 pairs, Mixed reaches 30.1% Clean and 29.4% Noise success, whereas Clean-only reaches 35.4% and 1.1%, and Noise-only reaches 0.0% and 51.8%. Mixed retains success on both conditions. The 500-complete-pair Mixed subset achieves 31.6%/28.3% with the same number of demonstrations but covers fewer pairs.

Table 17: Training updates during sequential self-improvement.

Model	SFT-250	RS+SFT	DPO	iDPO
Updates: stage / cumulative				
Qwen3-4B	250/250	171/421	154/575	130/705
Qwen3-8B	250/250	184/434	168/602	162/764
Qwen3-14B	250/250	208/458	169/627	176/803
Qwen3-32B	250/250	216/466	172/638	158/796
Llama-3.1-8B	250/250	221/471	147/618	61/679

Note. Stages proceed sequentially from SFT-250. Each stage uses newly collected trajectories and one epoch over its selected set. Cumulative updates include the initial 250 SFT updates.

E.4 Capability Retention

Figure 10 extends the main capability comparison to DPO and PPO. Each method starts from SFT-250 and performs 250 further updates. Base and SFT-250 are shared with Figure 5, and all panels use the same eight fixed capability subsets, totaling 10,072 questions per completed checkpoint. Both comparisons include all five models at their final checkpoints. Table 18 reports the eight raw

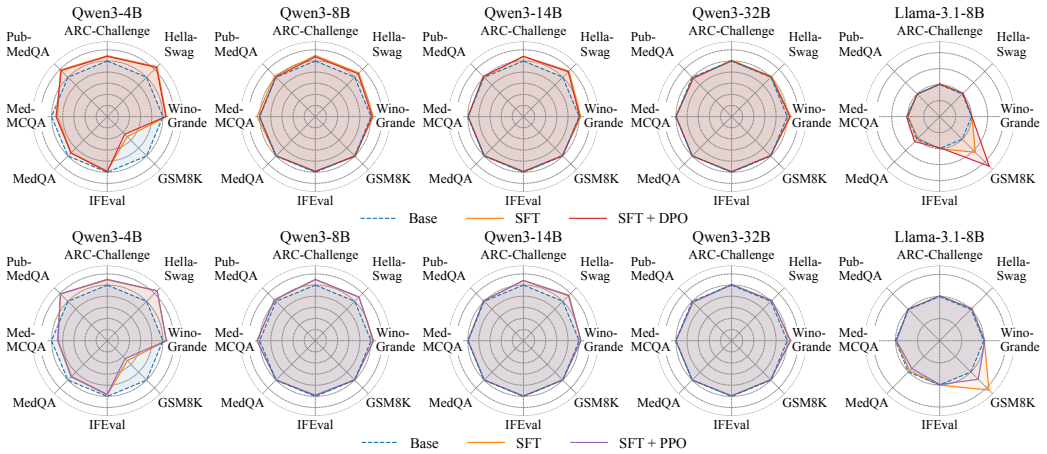


Figure 10: Capability retention after SFT-initialized DPO and PPO. Top: DPO. Bottom: PPO. Both continue from SFT-250 for 250 updates. Radii show scores relative to each model's Base score. Raw scores appear in Table 18; radial limits vary across models.

benchmark scores and their equal-weight mean. The ratio used for radar radii is separate from these percentage scores, and polygon areas are not directly comparable across model-specific radial limits.

Table 18: Raw scores on the eight general and medical benchmarks.

	General capabilities					Medical QA			
Checkpoint	ARC-C	Hella-Swag	Wino-Grande	GSM8K	IFEval	Med-QA	Med-MCQA	Pub-MedQA	**Mean**
Qwen3-4B									
Base	37.4	46.3	55.9	78.9	79.1	31.2	39.1	58.8	**53.3**
SFT-250	40.9	58.6	59.4	40.2	77.1	28.7	34.9	69.8	**51.2**
SFT-500	41.0	58.6	60.1	42.4	75.6	28.6	35.1	70.8	**51.5**
+DPO	40.4	57.9	58.9	35.2	78.4	28.8	36.0	68.8	**50.5**
+PPO	41.2	58.8	59.4	35.8	76.3	28.4	34.6	70.0	**50.5**
+GRPO	40.6	58.7	59.1	33.7	77.6	28.9	35.0	71.0	**50.6**
Qwen3-8B									
Base	42.3	59.8	61.9	87.4	81.3	28.0	35.3	66.4	**57.8**
SFT-250	46.5	66.3	64.6	87.0	80.4	27.9	36.9	68.8	**59.8**
SFT-500	46.7	66.9	64.7	86.9	79.1	27.9	36.8	69.8	**59.8**
+DPO	45.5	65.1	63.1	88.3	79.9	27.9	35.5	66.8	**59.0**
+PPO	46.6	66.3	64.7	87.0	79.5	28.0	36.9	69.2	**59.8**
+GRPO	46.2	66.2	64.2	87.3	80.2	27.9	36.6	68.6	**59.6**
Qwen3-14B									
Base	42.5	58.9	61.3	91.4	84.8	27.8	32.0	76.2	**59.4**
SFT-250	46.1	68.2	63.2	90.7	84.7	27.7	31.9	77.4	**61.2**
SFT-500	47.1	69.0	64.6	91.4	83.5	27.7	31.9	77.6	**61.6**
+DPO	45.9	67.0	62.0	90.5	83.5	27.7	31.8	77.8	**60.8**
+PPO	45.9	67.3	63.5	91.1	84.1	27.7	31.8	77.2	**61.1**
+GRPO	46.4	68.0	62.8	91.5	83.7	27.7	32.0	77.8	**61.3**
Qwen3-32B									
Base	49.0	71.0	62.8	94.5	84.3	28.0	32.0	76.0	**62.2**
SFT-250	49.6	72.6	66.8	93.7	83.5	28.0	32.1	74.2	**62.5**
SFT-500	49.4	73.1	66.3	93.6	82.4	28.0	32.0	74.4	**62.4**

Checkpoint	General capabilities					Medical QA			Mean
	ARC-C	Hella-Swag	Wino-Grande	GSM8K	IFEval	Med-QA	Med-MCQA	Pub-MedQA	
+DPO	49.1	71.5	65.9	93.6	83.5	27.9	32.0	74.2	**62.2**
+PPO	49.3	72.6	66.3	93.8	83.4	28.0	32.0	74.6	**62.5**
+GRPO	48.9	72.7	66.6	93.9	84.1	28.0	32.0	74.6	**62.6**
Llama-3.1-8B									
Base	53.6	73.7	67.8	34.0	74.3	34.2	48.6	74.6	**57.6**
SFT-250	54.4	76.3	68.4	53.6	74.7	32.8	47.8	74.6	**60.3**
SFT-500	52.8	75.7	69.1	59.7	74.5	31.0	46.4	74.2	**60.4**
+DPO	54.8	76.0	68.4	74.5	72.3	37.9	50.2	73.6	**63.5**
+PPO	54.4	75.7	69.3	41.9	73.6	30.5	47.4	75.4	**58.5**
+GRPO	54.3	75.5	68.4	55.1	74.3	29.5	45.6	74.4	**59.6**

Note. Scores are percentages under the metrics in Table 12. Mean weights the eight benchmarks equally. +DPO, +PPO, and +GRPO each add 250 updates from SFT-250. SFT-500 is included as an additional checkpoint, not as their initialization.

E.5 Repeated Evaluations

Each of the eight models in Appendix C.2 has four Base runs of 2,016 questions. Temperature-configurable models have two runs at 0 and two at 0.6, while GPT-5.4 has four service-default runs. Figure 12 separates the conditions for the six models in the main figure, and Figure 11 adds the two Qwen3 models. The first run uses the main tables' Base trajectories and scores.

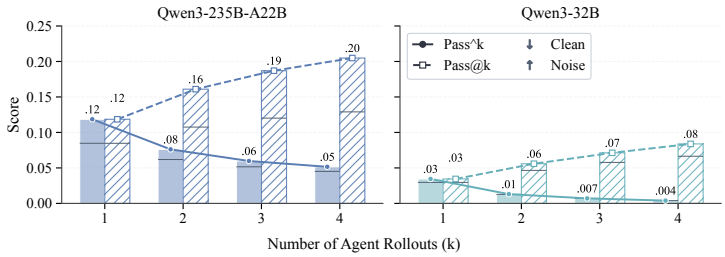


Figure 11: Repeated Base evaluations for two additional Qwen3 models. Each model uses two runs at temperature 0 and two at 0.6. Bars stack equally weighted Clean and Noise contributions over 2,016 questions. Metrics match Figure 4(a), with the vertical axis restricted to 0–0.25.

Let $c_i = \sum_{r=1}^{4} \text{Success}(\tau_{ir})$ count successes for question i, with N evaluated questions in total. Pass@k and pass^k average any-success and all-success over observed k-run subsets, respectively:

$$\text{pass@}\,k = \frac{1}{N}\sum_{i=1}^{N}\left[1 - \frac{\binom{4-c_i}{k}}{\binom{4}{k}}\right],$$
$$\text{pass}\,\hat{}\,k = \frac{1}{N}\sum_{i=1}^{N}\frac{\binom{c_i}{k}}{\binom{4}{k}}, \qquad k \in \{1,2,3,4\}, \tag{2}$$

$\binom{a}{k} = 0$ for $a < k$. Separate-condition scores use $N = 1{,}008$, and combined scores use $N = 2{,}016$ questions. At $k = 1$, both metrics report the four-run mean, not the first-run score. These observations mix sampling temperatures, so they summarize the run combination rather than estimate probabilities for a fixed-temperature distribution. Pass@k does not select a final response. Separate-condition plots report scores directly; stacked plots give each condition one-half weight.

Additional Qwen3 models. Figure 11 reports Qwen3-235B-A22B-Instruct-2507 and Qwen3-32B under the same four-run protocol. Across all 2,016 test questions, their Pass@4 scores are 20.5%

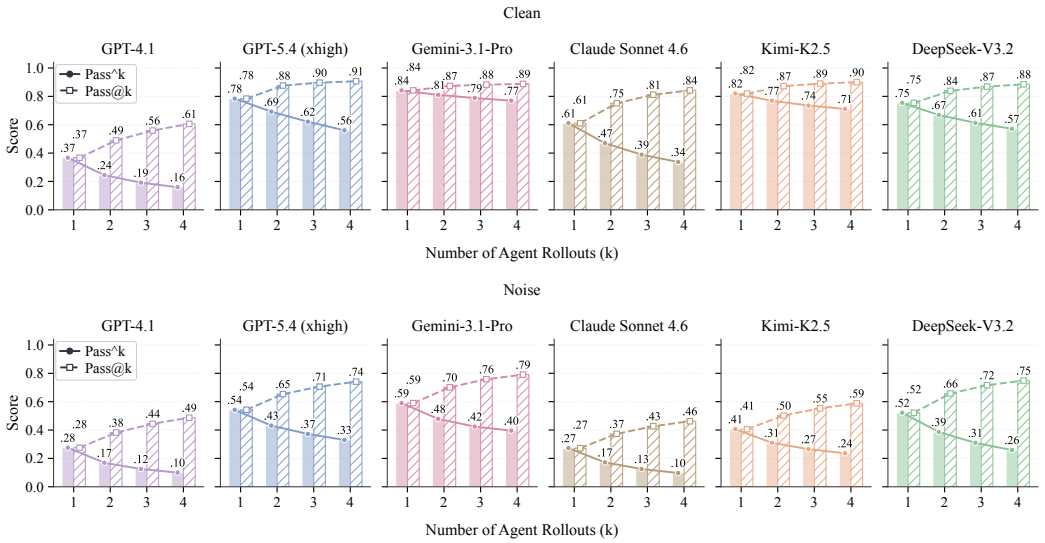


Figure 12: Repeated Base evaluations by condition. Clean and Noise results are shown separately for the six models in Figure 4(a), using 1,008 questions per condition and four runs per model. Pass@k and pass^k measure any-success and all-success across observed run subsets. Scores are shown directly, without the one-half weighting used in the stacked figure.

and 8.4%, while pass^4 scores are 5.2% and 0.4%, respectively. These statistics describe question coverage and consistency across the observed runs, not performance after training.

F Trajectory Error Analysis

Table 19: Trajectory error categories and their subtypes. Each episode is counted once.

Category	Scope and subtypes
C1 Clinical Interpretation Error	Incorrect clinical identity or meaning: concept mapping (C1.1), medication events (C1.2), laboratory measurements (C1.3), microbiology and susceptibility (C1.4), and other clinical facts (C1.5).
C2 Query Logic Error	Incorrect operations on understood evidence: patient or encounter binding (C2.1), event linkage (C2.2), temporal or logical constraints (C2.3), statistics or numerical operations (C2.4), and query targets (C2.5).
C3 Tool-Use Error	Action formulation (C3.1), result access (C3.2), unproductive loops (C3.3), and voluntary premature termination (C3.4), unless an identified C1 or C2 error explains the defect. Empty results and external truncation alone are not agent errors.
C4 Answer Reporting Error	New factual errors (C4.1), omission of observed required results (C4.2), and prohibited disclosures (C4.3). Repeating an earlier error is not counted again; clinical misinterpretation remains C1.

Annotation procedure. We analyze 12,096 Base trajectories from GPT-5.4, Gemini-3.1-Pro, Claude Sonnet 4.6, Kimi-K2.5, DeepSeek-V3.2, and Qwen3-32B: 1,008 Clean and 1,008 Noise trajectories per model, including successful outcomes. The annotation model, GPT-5.4, uses high reasoning effort, service-default sampling, and a 32,768-token output budget. It receives the question, instructions, reference evidence, trajectory, and final response, but no model identity, condition metadata, or task score. Up to four database checks may clarify facts; their results do not count as agent-observed evidence. A second automatic pass re-examines 65 flagged trajectories using the same classifier and taxonomy. The distributions incorporate these reviews and one exclusion of an unsupported error episode. These diagnoses do not affect task scores or training rewards.

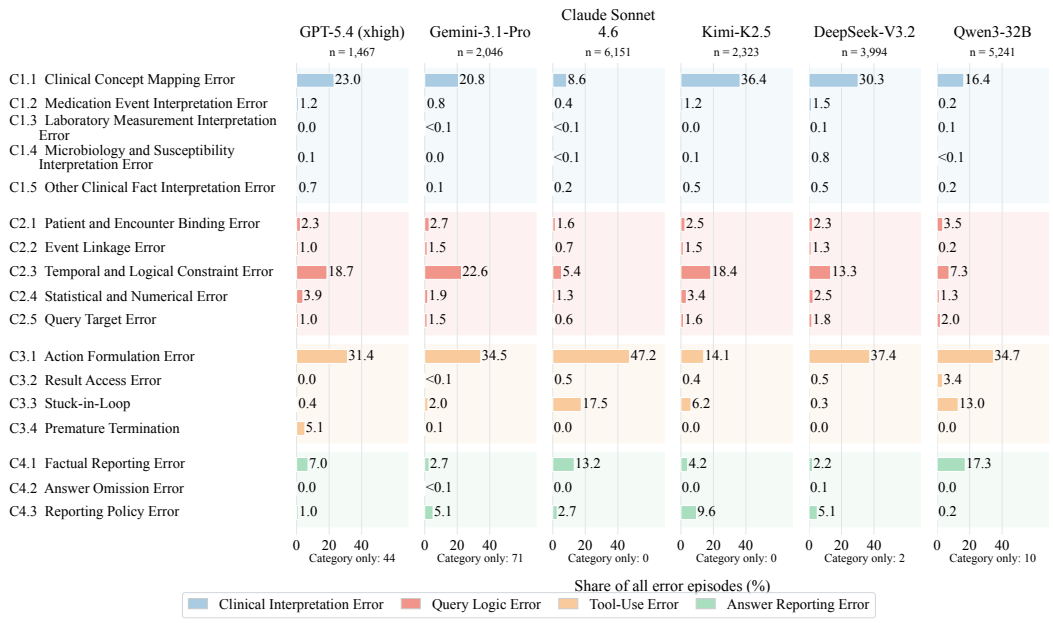


Figure 13: Error subtype composition across six models, using the episodes in Figures 7(b) and 14. Bars show percentages of all distinct error episodes, pooling Clean and Noise trajectories and including corrected errors. Colors mark the four categories; category-only counts below each panel remain in the denominator.

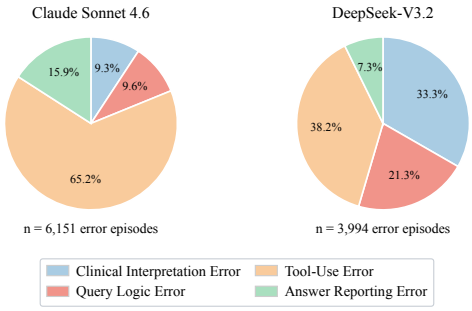


Figure 14: Error episode composition for Claude Sonnet 4.6 and DeepSeek-V3.2 under Base. Distributions use the same annotation protocol as Figure 7(b), pooling Clean and Noise trajectories and including corrected errors. Each distinct episode is counted once.

Annotation unit and boundaries. Each distinct error episode has one category and its most specific supported subtype (Table 19). Consequences of earlier errors are not counted again. We accept equivalent SQL and valid abstention evidence, but not agent-generated observations as tool feedback. External failures are recorded separately. Successful trajectories may contain corrected errors; error composition differs from task failure. The taxonomy adapts EHR-Complex (Qiao et al., 2026). An error episode is corrected only when later visible action or explicit retraction resolves its specific defect. It is uncorrected if the defect remains at the observable end, and unclear if resolution cannot be established. Eventual task success alone does not establish error correction.

G Case Studies

G.1 Selected Model Trajectories

Tables 20–22 present selected GPT-5.4 xhigh Base trajectories from the first default-sampling run. Each case illustrates an unsupported answer after successful retrieval. Questions, Answer fields,

and quoted explanation excerpts retain their original wording except that patient, admission, and medication-event identifiers use display labels. Evidence rows summarize the visible tool outputs. These cases illustrate specific failure mechanisms rather than their prevalence. Verification confirms that all four tool calls across the three trajectories succeeded without timeouts or truncated feedback, and that replayed tool and reference queries reproduced the recorded results.

Table 20: A procedure-list position is mistaken for a repeated occurrence.

Paired requests	
Clean request	During hospital admission A1 for patient P1, when was invasive catheter-based electrophysiologic testing first documented, according to the inpatient procedure coding record?
Clean reference	2117-04-07.
Noise request	During hospital admission A1 for patient P1, when was invasive catheter-based electrophysiologic testing documented for the second time, according to the inpatient procedure coding record?
Retrieved evidence	
Returned procedure list	Tool call 1 returns all four coded procedures for the admission. Sequence numbers 1, 2, 3 and 4 have codes 3734, 3726, 3727 and 3728, respectively; all are dated 2117-04-07. Only code 3726 describes catheter based invasive electrophysiologic testing.
Model response	
Model Answer	2117-04-07
Reasoning excerpt	"In the inpatient procedure coding record for patient P1 during hospital admission A1, the entry with sequencenumber 2 is ICD-9 procedure code 3726, described as "Catheter based invasive electrophysiologic testing." The proceduredate attached to that second listed inpatient procedure code is 2117-04-07."
Reference and observed error	
Noise reference	NULL. The requested procedure appears once in the complete returned list. The second position among different procedure codes is not a second occurrence of that procedure.
Observed error	The model explicitly uses sequencenumber 2 to answer a question about the second occurrence of one named procedure. The required list is already visible; the error is the interpretation of its ordering, not failure to retrieve the record.

Table 21: A later administered dose replaces the requested non-administered event.

Paired requests	
Clean request	During hospital admission A2 for patient P2, the first overall medication-administration record entry for insulin was marked as administered. What dose was recorded for that entry?
Clean reference	2 Units on the first overall insulin entry, at 2162-02-03 12:31.
Noise request	During hospital admission A2 for patient P2, the second overall medication-administration record entry for insulin was marked as administered. What dose was recorded for that entry?
Retrieved evidence	
Returned event order	Tool call 1 lists all ten insulin events in chronological order. The first three are E1 at 2162-02-03 12:31 (Administered), E2 at 18:32 (Not Given per Sliding Scale), and E3 at 23:18 (Administered).
Returned dose details	The second overall event, E2, has no dose given. Event E3 has dosegiven = 2 and dosegivenunit = Units. Event IDs keep the event list distinct from its multiple detail rows.
Model response	
Model Answer	2 Units

Table 21 (continued)

Reasoning excerpt	"Among the insulin entries with administrationevent recorded as "Administered," the second such record is administrationid E3 at 2162-02-03 23:18. In the linked medicationadministrationdetails record for that administration, the documented dose given is 2 with unit "Units.""
Reference and observed error	
Noise reference	NULL. The question identifies the second overall entry, whose observed state is Not Given per Sliding Scale and whose administered dose is absent.
Observed error	The original question explicitly says "second overall." The model instead counts only Administered events and returns the dose of the third overall event. This changes the selected event after the complete event list and dose details have been returned.

Table 22: A result from a later hospitalization replaces the requested encounter.

Paired requests	
Clean request	Patient P3's second hospitalization included a recorded diagnosis of atrial fibrillation. What was the earliest numeric Potassium result in blood during that hospitalization?
Clean reference	4.3 mEq/L at 2186-06-30 06:40 in the second hospitalization.
Noise request	Patient P3's first hospitalization included a recorded diagnosis of atrial fibrillation. What was the earliest numeric Potassium result in blood during that hospitalization?
Retrieved evidence	
Returned admissions	Tool call 1 lists two hospitalizations: A3 (admitted 2179-11-28) and A4 (2186-06-29). None of A3's nine diagnoses is atrial fibrillation.
Returned laboratory values	Tool call 1 returns blood potassium of 4.0 mEq/L at 2179-11-28 23:20 for A3. Tool call 2 finds code 42731 (Atrial fibrillation) in A4 and its earliest blood potassium, 4.3 mEq/L at 2186-06-30 06:40.
Model response	
Model Answer	4.3 mEq/L at 2186-06-30 06:40 during admission A4.
Reasoning excerpt	"The patient's first overall hospitalization (admission A3) did not have a recorded atrial fibrillation diagnosis. The first hospitalization that did include atrial fibrillation was admission A4, where ICD-9 code 42731 was recorded with description "Atrial fibrillation.""
Reference and observed error	
Noise reference	NULL. No hospitalization satisfies both the requested first-hospitalization scope and the recorded atrial-fibrillation condition.
Observed error	The model recognizes the mismatch but reports a real value from the second hospitalization, which does not satisfy the original request.

In all three cases, the reported values are present in the database but do not satisfy the original request. The cases cover Record-, Value-, and Query-level Noise, respectively, while all three trajectory errors are annotated as C2.3 (Temporal and Logical Constraint Error). Noise types describe unsupported request conditions; error categories describe how the agent uses evidence.

H Prompts

H.1 Agent System Prompt and Response Format

The template below shows the shared agent instructions and final-response format. At runtime, `[DATABASE SCHEMA]` is replaced with the exposed database's longitudinal schema description and table–column list. The harness saves the expanded system prompt with the run configuration and trajectories. The saved prompt, not a later rendering of this template, defines the run.

Base agent system prompt

You are an EHR database agent. Use the provided structured EHR database to answer the clinical request.
`[DATABASE SCHEMA]`
POPULATION REPORTING POLICY:
For population-level requests, calculate an aggregate only when at least five distinct population units satisfy every condition in the request. When no units qualify, state that no matching records were found. When one to four units qualify, state that the population is below the minimum reportable size and do not disclose the exact count or aggregate. This policy does not apply to patient-specific requests.
DATABASE ACCESS:
Every code execution already has a task-scoped read-only DuckDB connection named `conn`. Query that connection directly; do not open or attach another database.
Query it with Python, for example:

```
df = conn.execute("SELECT * FROM hospitaladmissions LIMIT 5").df()
print(df)
```

You are operating in a sequential tool-use environment.

Available actions:
`execute_code`: Run Python against the task-scoped read-only EHR connection.
- params: code (str)
- the environment provides a DuckDB connection named `conn`
- print query results needed for the final response
- do not open another database connection

To call the tool, return exactly one JSON object and no surrounding text:

```
{
  "action": "execute_code",
  "params": {
    "code": "<Python code using the provided conn variable>"
  }
}
```

The environment executes one tool call before returning its result. Inspect the schema as needed, query the database before answering, and continue until the requested rows and columns are complete. Do not place a tool call and a final answer in the same response.

After observing the complete result, answer the question clearly. Do not combine a tool call and a final answer in the same response.

Final response format and examples

FINAL RESPONSE:
Use the tools to inspect the database. When you are ready to submit, place your complete and concise response to the user's request inside exactly one <Final_answer>...</Final_answer> tag, with no content outside the tag.

Use the following format for your final response:
`<Final_answer>`
Answer: <direct answer (e.g., the requested value(s)) or NULL>
Reasoning: <a detailed and complete explanation of the brief answer above>
`</Final_answer>`

FORMAT EXAMPLES (fictional and unrelated to the current task):
These examples illustrate the response format; they are not evidence for the current question.

Example 1
Question: For patient DEMO-P01, what was the serum magnesium concentration during encounter DEMO-E01 on 12 April 2042?
Tool observation: The laboratory results for patient DEMO-P01 during encounter DEMO-E01 on 12 April 2042 contain no recorded serum magnesium measurement.
`<Final_answer>`
Answer: NULL
Reasoning: The requested value is the serum magnesium concentration for patient DEMO-P01 during encounter DEMO-E01 on 12 April 2042. The laboratory information returned for this patient, encounter and date contains no recorded result for that measurement. It therefore provides no concentration to

report for the requested serum magnesium test. The answer is NULL because the requested value is unavailable in these records; this statement concerns the available record, not whether the test was performed.
</Final_answer>

Example 2
Question: For patient DEMO-P02 during encounter DEMO-E02 on 9 June 2043, report the recorded blood glucose concentration at 08:30 and the insulin dose administered at 09:00.
Tool observation: The blood glucose measurement at 08:30 is 176 mg/dL. The insulin administration at 09:00 records a dose of 4 Units. Both records belong to patient DEMO-P02 and encounter DEMO-E02 on 9 June 2043.
<Final_answer>
Answer: Glucose: 176 mg/dL; Insulin dose: 4 Units
Reasoning: The two requested quantities are the blood glucose concentration and the administered insulin dose. The laboratory record gives a glucose concentration of 176 mg/dL at 08:30, and the medication administration record gives an insulin dose of 4 Units at 09:00. Both observations concern patient DEMO-P02 during encounter DEMO-E02 on 9 June 2043. The Answer associates each value with its corresponding measurement and recorded unit. The insulin entry describes an administered dose, not a dose recommendation or a concentration.
</Final_answer>

H.2 Evidence-Check Instruction

PE is a prompt-only baseline for the five local backbones and seven API-based models in Table 10. It appends this sentence to the Base system message without changing tools or interaction limits.

Evidence-check instruction (PE)

Please carefully check whether the EHR contains evidence supporting the requested information before answering.

I Limitations and Scope

EHR-RobustGym is built on MIMIC-IV v3.1, a large-scale de-identified EHR database available for research under credentialed access (Johnson et al., 2024). Its tasks therefore reflect the schema, coding conventions, patient population, and clinical practices of a single hospital system. While training gains extend to five external EHR benchmarks, broader generalization across institutions and clinical settings requires further evaluation. The Clean-Noise design supports controlled and reproducible evaluation of accurate answering and evidence-grounded abstention over structured EHRs. Free-text notes, imaging data, real-time clinician interaction, and prospective clinical deployment remain outside the current scope.