# LEARNING TO EVOLVE CLINICAL HARNESSES THROUGH ENTITY-GRAPH-GUIDED ORCHESTRATION FOR CLINICAL DECISION MAKING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Large language model based agents show promise for clinical decision making but struggle to maintain coherent patient states and coordinate execution across long-horizon workflows. Harness optimization can improve this execution infrastructure without updating model parameters, yet loosely structured clinical evidence and fixed proposal strategies limit its effectiveness. Thus, we propose a framework that learns to evolve clinical harnesses by coupling entity-graph-guided harness with a self-evolving proposer. *To represent evolving clinical evidence coherently and explicitly*, a shared patient entity graph explicitly represents evolving clinical evidence, diagnostic hypotheses, and their dependencies, coordinating memory retrieval, clinical-skill selection, and tool execution through a consistent, stage-visible patient state. *To mitigate optimization collapse arising* from a fixed proposer, we introduce a *self-evolving proposer* that learns how to evolve the harness via Candidate-DAG memory and an evolving proposer skill library. Candidate-DAG memory preserves alternative harness designs, component inheritance, and case-level corrections and regressions, including useful interventions from lower-scoring candidates. An evolving proposer skill library distills this experience into reusable procedures for parent selection, cross-branch component reuse, avoidance of unsuccessful interventions, and targeted acquisition of training-derived medical knowledge. Extensive experiments on MedChain and ClinicalBench datasets demonstrate that our framework outperforms general-purpose and medical-specialized LLMs, multi-agent systems, and existing harness optimization methods in average performance.

## 1 INTRODUCTION

Clinical decision making (CDM) requires integrating evolving patient evidence with medical knowledge to guide diagnosis and treatment (Liu et al., 2025). Recent large language model (LLM) based agents support this process through collaborative reasoning that combines complementary medical perspectives (Kim et al., 2024), active information acquisition that reduces uncertainty (Li et al., 2024; Nori et al., 2025), and memory-augmented workflows that retain and reuse clinical context (Liu et al., 2025; Ge et al., 2026). Despite these advances, agents with strong medical knowledge and local reasoning capabilities still struggle to achieve reliable end-to-end CDM. As an interactive, long-horizon process spanning referral, history taking, examination, diagnosis, and treatment, CDM requires agents to continuously maintain patient state and coordinate decisions across interdependent stages. However, state management, process verification, and error recovery remain insufficiently coordinated in existing agent designs, leaving early errors to propagate and compound in subsequent decisions. These limitations suggest that the performance bottleneck lies not only in the medical capabilities of the underlying model, but also in the infrastructure supporting long-horizon agent execution.

Harness engineering offers a potential solution to the coordination challenges of long-horizon CDM by systematically organizing the memory, tools, verification, and control flow surrounding an LLM. Recent work further automates the improvement of this system while keeping model parameters fixed. One line of work treats harness design as *search over executable implementations*, generating and refining candidate programs using environment feedback or the accumulated code, scores, and



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.

execution traces of previous experiments (Lou et al., 2026; Lee et al., 2026). This approach allows information management and action control to be optimized beyond isolated prompt modifications. A complementary line emphasizes *diagnosis-guided refinement*, using execution traces and component-level observations to identify weaknesses, propose targeted interventions, and assess their effects through outcome analysis or regression testing (Lin et al., 2026; Zhang et al., 2026). Together, these studies demonstrate that substantial gains can be obtained while keeping model parameters fixed. However, these methods focus on coding and other environments with executable specifications, structured traces, and relatively direct evaluation feedback. This raises a central question: *how can we jointly design clinically grounded harnesses and their automatic optimization mechanisms to enable sustained improvement in long-horizon medical decision making?*

The first challenge is representing complex and evolving clinical evidence in a form that supports harness evolution. Many mathematical and coding benchmarks provide relatively explicit task specifications and correctness checks, such as reference answers or executable tests (Lee et al., 2026; Jimenez et al., 2024). In CDM, however, symptoms, laboratory findings, images, and diagnostic hypotheses become available progressively, and their interpretation depends on the surrounding evidence and clinical stage (Schmidgall et al., 2024; Liu et al., 2025). When a generic harness passes this information through loosely structured text, the relationships among observations, hypotheses, and missing evidence can remain implicit. Consequently, memory retrieval, skill selection, and tool use may rely on inconsistent interpretations of the patient state, while unsupported intermediate conclusions propagate into subsequent decisions. These ambiguities also obscure how modifications to individual components affect the downstream workflow, making clinically meaningful improvements difficult to identify and evaluate. Effective medical harness evolution therefore requires a clinically structured harness that preserves the context and dependencies of patient evidence, maintains a coherent patient state, and makes interactions across modules and stages explicit enough to support systematic optimization.

The second challenge is optimization collapse for harness evolution, and its root cause is that the harness search loop is bounded by a fixed proposer. Existing harness optimization typically follows a propose–evaluate–select loop: a proposer generates candidate edits to prompts, memory, tools, or orchestration, and greedy selection based on an aggregate score determines which candidate advances (Ursekar et al., 2026; Luo et al., 2026). This greedy, score-driven mechanism is unstable in two ways. First, when ambitious structural changes fail to yield immediate gains, the search retreats to conservative prompt edits, settling into local optima or even regressing from prior progress. Second, it discards useful components embedded in lower-scoring branches, obscuring case-level regressions that could otherwise guide future search. Archive-based approaches partially mitigate this by keeping alternative designs available for later reuse (Luo et al., 2026; Zhang et al., 2025), but merely retaining candidates does not guarantee that their successes and failures actively steer subsequent proposals. Both failure modes trace back to the same cause: the proposer remains fixed across iterations, so once its initial design priors are exhausted, selection has nothing new to choose among, and the loop stagnates or drifts backward instead of progressing. Thus, it is necessary to let the proposer itself evolve across iterations, rather than remain fixed.

To address these challenges, we propose a framework that *learns to evolve clinical harnesses* by coupling entity-graph-guided harness with a self-evolving proposer. As shown in Figure 1, *to represent evolving clinical evidence coherently and explicitly*, we initialize a clinically structured harness around a shared patient entity graph. This graph explicitly represents clinical evidence, diagnostic hypotheses, and their dependencies, coordinating memory retrieval, clinical-skill selection, and tool execution through a consistent, stage-visible patient state. By preserving evidence provenance and exposing cross-module dependencies, it provides a coherent foundation for targeted harness modifications and their evaluation. *To mitigate optimization collapse arising* from a fixed proposer, we introduce a *self-evolving proposer* that learns how to evolve the harness by converting the successes and failures of previous candidates into reusable proposal procedures through Candidate-DAG memory and an evolving proposer skill library. Specifically, the Candidate-DAG memory preserves alternative candidates, component inheritance, and case-level corrections and regressions, retaining useful interventions from lower-scoring branches and revealing effects obscured by aggregate scores. We then distill this accumulated experience into reusable proposal procedures that guide parent selection, cross-branch module reuse, avoidance of unsuccessful interventions, and targeted requests for training-derived medical knowledge. Evaluation feedback thus changes not only which harness components are retained, but also how subsequent candidates are proposed, supporting exploration

beyond the proposer's initial design priors. Extensive experiments on MedChain and ClinicalBench demonstrate that our framework outperforms general-purpose and medical-specialized LLMs, multi-agent systems, and harness optimization methods in average performance. Our contributions are summarized as follows:

- We propose an entity-graph-guided clinical harness that coordinates memory retrieval, clinical skills, and tool execution through a shared, structured patient state.
- We introduce a self-evolving proposer that learns to evolve clinical harnesses by distilling accumulated experimental experience into reusable proposal strategies through Candidate-DAG memory and an evolving skill library.
- Extensive experiments on MedChain and ClinicalBench demonstrate superior average performance over evaluated baselines.

## 2 Related Work

### 2.1 LLM Agents for Sequential Clinical Decision Making

Clinical agent research has shifted from static QA to interactive clinical reasoning: MediQ addresses active information acquisition under incomplete observations (Li et al., 2024); AgentClinic evaluates multimodal interaction and tool use (Schmidgall et al., 2024); MedChain extends evaluation across interdependent clinical stages (Liu et al., 2025); and the Sequential Diagnosis Benchmark considers diagnostic accuracy and cost (Nori et al., 2025)—together underscoring the need to maintain patient state and coordinate evidence-dependent decisions across stages. At the system level, MDAgents adapts collaboration to task complexity (Kim et al., 2024), ClinicalAgents combines working/experience memory with dynamic orchestration (Ge et al., 2026), and MeDxAgent examines prompt–workflow–agent design interactions (Sanghvi et al., 2026). Automated adaptation has further explored graph-structured diagnostic workflow search (Zhuang et al., 2025) and evolving reusable skill memories (Sun et al., 2026). However, these approaches mainly target workflows or skill memory alone. Thus, we introduce an entity-graph-based harness to jointly evolve memory, skills, tools, and orchestration around a shared patient state.

### 2.2 Automatic Agent and Harness Engineering

Automatic agent engineering has grown from tuning individual components to searching entire executable systems. DSPy optimizes instructions and demonstrations in language-model programs, while GPTSwarm optimizes node prompts and graph connectivity (Khattab et al., 2024; Zhuge et al., 2024). ADAS and the Darwin Gödel Machine extend this to code-defined agents, retaining prior designs for continued exploration (Hu et al., 2025; Zhang et al., 2025). Extending this line, AutoHarness and Meta-Harness search executable harness implementations via environment feedback and candidate histories, while Agentic Harness Engineering and Self-Harness focus on execution diagnosis, targeted edits, and validation (Lou et al., 2026; Lee et al., 2026; Lin et al., 2026; Zhang et al., 2026). Complementary work strengthens the search process itself: VeRO supplies versioned candidates, execution feedback, and budget-aware evaluation; Gated Semantic MAP-Elites (GSME) retains alternative edits in a gated archive; and recent evaluations stress matched optimization budgets and held-out testing for assessing generalization (Ursekar et al., 2026; Luo et al., 2026; Wang et al., 2026). However, current methods rely on stable proposers and unstructured state representations, limiting their ability to capture heterogeneous evidence, clinical dependencies, and case-level regressions. Thus, we introduce a framework that learns to evolve clinical harnesses by coupling entity-graph-guided harness with a self-evolving proposer.

## 3 Method

In Figure 1, we introduce a clinical harness evolution framework that iteratively refines an entity-graph-based harness through a self-evolving proposer. We first initialize a clinical harness in which a shared patient entity graph coordinates memory retrieval, clinical-skill selection, and tool execution across clinical stages. Starting from this initialization, the proposer selects previously evaluated candidates as parents and specifies which components to modify or combine. Each candidate

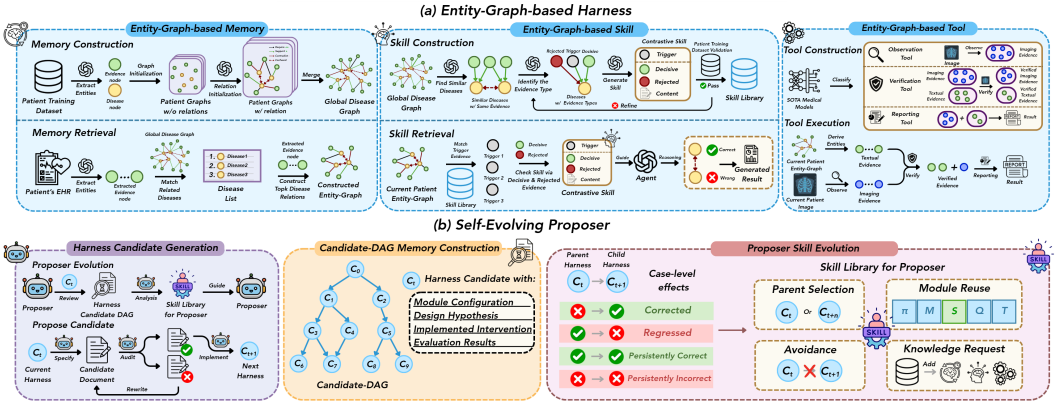


Figure 1: Overview of our framework: (a) an entity-graph-based medical harness coordinates memory, clinical skills, and tools through a shared patient state; (b) a self-evolving proposer uses Candidate-DAG memory and an evolving proposer skill library to guide subsequent harness modifications.

specification is audited before implementation, and the resulting harness is evaluated on a shared evolution-validation cohort. The proposer then compares the candidate with its parents, records its configuration and case-level effects in the Candidate-DAG, and uses the accumulated results to update its proposer skill library. The updated memory and skills guide the next round of candidate generation. This cycle repeats while the underlying LLM parameters remain fixed. We represent the $i$-th harness candidate as:

$$C_i = \left(\pi_i, \mathcal{M}_i, \mathcal{S}_i, \mathcal{Q}_i, \mathcal{T}_i\right), \tag{1}$$

where $\pi_i$ denotes clinical-task orchestration, $\mathcal{M}_i$ the memory system, $\mathcal{S}_i$ the clinical-skill library, $\mathcal{Q}_i$ the component selector, and $\mathcal{T}_i$ the available tools. The initialization defines $C_0$. Subsequent candidates modify a declared subset of these components and inherit reusable components from evaluated parent candidates.

### 3.1 Entity-Graph-Based Medical Harness Initialization

We initialize the harness with a shared representation of clinical evidence and a stage-wise execution procedure. This initialization provides a medically structured starting point for optimizing memory, skills, tools, and their coordination. For each patient, the benchmark specifies an ordered workflow $(f_1, \ldots, f_L)$, whose stages may include referral, examination planning, image interpretation, diagnosis, and treatment.

**Shared Patient State** Each patient $x$ is represented as a typed entity graph $\mathcal{G}_x = (\mathcal{V}_x, \mathcal{E}_x)$, with node set $\mathcal{V}_x = \mathcal{V}_x^E \cup \mathcal{V}_x^D \cup \mathcal{V}_x^O$ and edge set $\mathcal{E}_x = \mathcal{E}_x^C \cup \mathcal{E}_x^\Delta \cup \mathcal{E}_x^O$. Here, $\mathcal{V}_x^E$ contains clinical evidence, including symptoms, medical history, physical signs, laboratory results, and imaging findings. Each evidence node records its normalized concept, polarity, provenance, and the stage at which it becomes available. $\mathcal{V}_x^D$ contains confirmed, suspected, excluded, or differential disease hypotheses, while $\mathcal{V}_x^O$ contains operational information such as intermediate decisions, retrieved information, image references, and tool observations. Clinical edges $\mathcal{E}_x^C$ encode supporting, contradictory, and required evidence and differential edges $\mathcal{E}_x^\Delta$ connect confusable diseases and identify their shared or distinguishing evidence. Operational edges $\mathcal{E}_x^O$ link intermediate results to their sources and verification processes. The details of graph construction are described in Section 3.1.1.

**Stage-wise Orchestration** Each clinical stage $f_\ell$ proceeds through state reading, reasoning with modules, and state updating. The harness exposes only the currently available patient subgraph, which provides a shared state for memory retrieval, skill selection, and tool planning. The agent's stage prediction is stored as an intermediate decision or hypothesis, while newly available structured results and verified tool findings are added as evidence before the next stage begins. This procedure preserves evidence provenance and prevents earlier stages from accessing future observations. The same procedure repeats across stages until the final clinical output is produced. Training labels are used only for training-time memory construction and clinical-skill construction and validation.

### 3.1.1 ENTITY-GRAPH-BASED MEMORY

**Memory Construction** For each training case, we extract and normalize clinical evidence and initialize supporting, contradictory, required-evidence, and differential relations from the training-case information. The resulting relational patient graphs are aggregated into a global disease graph, which summarizes recurring evidence–disease associations and frequently observed differential diagnoses. Evidence and disease concepts with the same clinical meaning are merged across cases, while their occurrence statistics and source cases are retained. The global graph represents reusable medical relationships rather than directly encoding benchmark answers; benchmark-specific outputs remain associated through task-specific adapters. This separation allows the same clinical representation to support different stages without conflating medical concepts with benchmark labels.

**Memory Retrieval** At the first stage, entities extracted from the available patient record initialize the patient graph; later stages update it with newly available evidence. The memory module matches the stage-visible evidence entities against the global disease graph. The retrieved subgraph identifies diseases that are supported, contradicted, or insufficiently assessed by the available evidence. For a candidate disease $d$, we summarize its evidence using supporting evidence $S(d)$, contradictory evidence $C(d)$, satisfied requirements $Q^+(d)$, and missing requirements $Q^-(d)$. Candidate diseases are ranked using

$$R(d) = \alpha S(d) - \beta C(d) + \gamma Q^+(d) - \eta Q^-(d) \tag{2}$$

where $\alpha, \beta, \gamma, \eta \geq 0$ control the relative contributions of the evidence terms. The weighting strategy can be adapted to the target benchmark. The memory returns a ranked disease list, with each item containing supporting and contradictory evidence, unresolved requirements, related differential diagnoses, and representative training cases. The top-ranked disease hypotheses and their retrieved relations augment the current patient graph around the observed evidence, forming a shared patient state $\mathcal{G}_x = (\mathcal{V}_x, \mathcal{E}_x)$ for subsequent skill selection and tool planning.

### 3.1.2 ENTITY-GRAPH-BASED SKILL

**Skill Construction** We construct a library of reusable clinical reasoning instructions from training cases. Each skill specifies an applicable task and stage, triggering evidence, decisive evidence, rejected evidence, a reasoning principle, and provenance from the training data. Together, these elements describe when a clinical distinction should be considered and when the corresponding instruction should be withheld. The library contains general and contrastive skills. General skills capture broad clinical contexts, such as pediatric, emergency, pregnancy-related, oncological, or structural conditions. To construct contrastive skills, we first identify confused diseases with shared evidence in the global disease graph, then compare their associated training cases to find the clinical reasoning chain and distinguish triggering, decisive and rejected Evidence. Evidence that consistently distinguishes one alternative from another is treated as decisive, while evidence favoring the competing alternative is treated as rejected evidence. Before entering the library, each candidate skill is checked on training cases to verify that it triggers as intended and that accuracy on the cases where it is triggered reaches a predefined threshold. Skills that fail either check are returned for regeneration and checked again. We initialize the skill quality $Q(s)$ with this empirical post-trigger accuracy.

**Skill Retrieval** At each clinical stage, the harness matches currently visible patient evidence against skills applicable to that task and stage. For a skill $s$ and patient $x$, let $T_s(x)$, $D_s(x)$, and $B_s(x)$ denote the matched triggering, decisive, and rejected evidence, respectively. Skill eligibility is defined as

$$I_s(x) = \mathbf{1}\begin{bmatrix} Q(s) \geq \tau_s, & |T_s(x)| \geq m_T, \\ |D_s(x)| \geq m_D, & B_s(x) = \emptyset \end{bmatrix}, \tag{3}$$

where $Q(s)$ is the estimated skill quality, $\tau_s$ is a quality threshold, and $m_T$ and $m_D$ are the required numbers of triggering and decisive matches. A skill is eligible only when it is sufficiently reliable, supported by current-patient evidence, and not blocked by contradictory evidence. Eligible skills are ranked by evidence match and reliability. To limit conflicting instructions, the harness injects only the most relevant eligible skill into each task prompt. The selected skill guides reasoning without directly modifying the model output. During testing, skill performance is assessed using the test reference answers and the benchmark evaluator. The resulting feedback is passed to the proposer, which determines whether to update $Q(s)$.

#### 3.1.3 ENTITY-GRAPH-BASED TOOL

**Tool Construction** We construct a tool library from existing medical vision models, used without task-specific training, and organize it into three functional categories. *Observation tools* extract candidate imaging evidence from medical images. *Verification tools* first check whether these findings are visually supported, then jointly review the images and the current patient's textual evidence to assess their clinical relevance and consistency. They return verified imaging evidence and verified textual evidence, where textual verification denotes relevance and consistency with the image review. *Reporting tools* generate a clinical report from both verified evidence sets and original images.

**Tool Execution.** During inference, the selector jointly considers the current patient graph and available medical images to select tools matching the relevant clinical entities, imaging modality, and anatomical region. Textual evidence is drawn from the stage-visible patient graph, while observation tools extract candidate imaging findings. The selected tools execute in the order of observation, verification, and reporting. Candidate findings, textual evidence, and images enter verification; the resulting verified textual and imaging evidence enters report generation. Verification results update the patient graph while preserving evidence provenance, and the report is stored as an intermediate task result for subsequent stages.

### 3.2 SELF-EVOLVING PROPOSER

Starting from the initialized harness, the proposer generates and evaluates new candidates using the history of previous experiments. It evolves its optimization knowledge through two complementary components: a Candidate-DAG memory that preserves evaluated harnesses and their observed effects, and a proposer skill library that converts this evidence into reusable proposal procedures. After each evaluation, the new results update both components and guide subsequent candidate generation. This self-evolution changes the proposer's accessible experience and optimization strategies without updating the underlying LLM parameters.

#### 3.2.1 CANDIDATE-DAG MEMORY

We represent the evolution history as a directed acyclic graph $\mathcal{G}_C = (\mathcal{V}_C, \mathcal{E}_C)$. Each node represents a complete and reproducible harness candidate $C_i$, recording its module configuration, parent candidates, design hypothesis, implemented intervention, and evaluation result. An edge $(j, i) \in \mathcal{E}_C$ links a parent $C_j$ to its child $C_i$ and records the inherited or modified components. A child may combine beneficial interventions from two parent candidates, giving it multiple incoming edges. All parents precede their child in the evaluation history, preserving acyclicity.

The candidate memory compares candidate outcomes with those of their parents at the case level. Let $y_i(x) \in [0, 1]$ denote the score achieved by $C_i$ on patient $x$. For a parent–child pair $(C_j, C_i)$ evaluated on the same patient cohort $\mathcal{X}$, the average effect is

$$\Delta_{j \to i} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \big(y_i(x) - y_j(x)\big). \tag{4}$$

Case-level effect analysis uses the performance of patients affected by the harness modification as its unit of analysis. The proposer reviews their relevant task scores and execution outcomes to judge whether the intervention produces a clear improvement or deterioration. We categorize clear improvements as *corrected* and clear deteriorations as *regressed*. The proposer also distinguishes *persistently correct* cases, whose satisfactory performance is maintained, from *persistently incorrect* cases, whose failures remain unresolved. These categories summarize proposer-assessed effects on patient performance and help expose changes obscured by aggregate scores.

Candidates investigating related hypotheses are grouped into branch families, such as memory retrieval, skill selection, tool routing, or orchestration. A lower-scoring branch remains available if it contains a useful module or corrects a distinct group of cases. The memory also aggregates observations associated with module changes: repeatedly beneficial interventions can be reused, consistently regressive interventions are marked as harmful, and repeatedly neutral interventions are marked as ineffective. Retaining this history helps the proposer recover useful alternatives and avoid repeating unsuccessful modifications.

Table 1: Performance comparison on MedChain dataset.

| Framework | Methods | Specialty Referral | | History Taking | Examination | Diagnosis | Treatment | Average |
|---|---|---|---|---|---|---|---|---|
| | | Level 1 | Level 2 | | | | | |
| General-purpose LLMs | GPT-5.1 | 0.5157 | 0.3370 | 0.5059 | 0.5638 | 0.5199 | 0.5210 | 0.4939 |
| | GPT-o3 | 0.5495 | 0.2866 | 0.3493 | – | 0.4891 | 0.3343 | 0.4018 |
| | DeepSeek V3.2 | 0.5321 | 0.3530 | 0.4593 | – | 0.5126 | 0.5769 | 0.4868 |
| | Kimi-K2 | 0.5227 | 0.3288 | 0.3667 | 0.4717 | 0.4297 | 0.5235 | 0.4405 |
| | Qwen3-32B | 0.5432 | 0.3628 | 0.5171 | 0.4819 | 0.5144 | 0.5153 | 0.4891 |
| Medical LLMs | MedGemma-27B | 0.5571 | 0.3665 | 0.4901 | – | 0.4879 | 0.4496 | 0.4702 |
| | Baichuan-M2-32B | 0.5067 | 0.3337 | 0.4730 | 0.4914 | 0.5058 | 0.5882 | 0.4831 |
| | Lingshu-32B | 0.5469 | 0.3673 | 0.4217 | 0.5457 | 0.4944 | 0.4934 | 0.4782 |
| | HuatuoGPT-o1-8B | 0.4989 | 0.3333 | 0.3402 | 0.4522 | 0.4496 | 0.2445 | 0.3865 |
| Multi-agent systems | MedAgent | 0.3830 | 0.2039 | 0.4454 | 0.4456 | 0.4102 | 0.3673 | 0.3759 |
| | MDAgents | 0.4754 | 0.2063 | 0.4412 | 0.5346 | 0.4198 | 0.4371 | 0.4191 |
| | MedChain-Agent | 0.5873 | 0.3505 | 0.5836 | 0.6566 | 0.4807 | 0.4613 | 0.5200 |
| | MedAgent-Pro | 0.5891 | 0.3714 | 0.4607 | 0.5477 | 0.4520 | 0.5215 | 0.4904 |
| Harness methods | Meta-Harness | 0.6135 | 0.3017 | 0.6056 | 0.6538 | 0.5385 | 0.5845 | 0.5496 |
| | Self-Harness | 0.6126 | 0.2988 | 0.6023 | 0.6445 | 0.5398 | 0.6009 | 0.5498 |
| | **Ours** | **0.6740** | **0.3933** | **0.6130** | **0.6837** | **0.5753** | **0.6430** | **0.5971** |

#### 3.2.2 PROPOSER SKILL EVOLUTION

The proposer skill evolution distills reusable optimization knowledge from the Candidate-DAG. These skills guide harness modification and evaluation, while the clinical skills guide medical reasoning. The library includes parent-selection skills for identifying reliable implementation bases, module-reuse skills for transferring effective components, avoidance skills for recording harmful or ineffective interventions, and knowledge-request skills for identifying persistent failures that may require additional training-derived medical knowledge. During candidate generation, these skills guide parent selection and the combination of beneficial changes across branches, enabling reliable components to be retained and useful mechanisms to be transferred. For persistent failures associated with missing clinical coverage, a knowledge request identifies the evidence gap and the affected harness module. Relevant associations or examples are then retrieved from the training set to augment memory, refine clinical skills, or revise the selection and configuration of available tools. These additions enter a new harness candidate and are evaluated through the same optimization loop. Each proposer skill records its supporting evidence, applicable conditions, recommended or forbidden actions, and current confidence. After a candidate is evaluated, the observed corrections, regressions, and module-level changes are used to create, strengthen, revise, or retire these skills. The updated library then guides the next proposal, allowing experimental experience to change how the proposer searches.

#### 3.2.3 HARNESS EVOLUTION LOOP

Each harness evolution round contains five steps: **(1) Generate a candidate specification.** Using the current Candidate-DAG and proposer skills, the proposer writes a candidate document specifying its parent candidates, a falsifiable hypothesis, the modules to change or combine, the expected effect, and potential regression risks. **(2) Audit and implement.** The document is checked for consistency, reproducibility, and data leakage. A failed audit returns feedback for rewriting and re-auditing the specification; only a passing document proceeds to implementation. Inherited and combined modules retain their recorded parent sources. **(3) Evaluate the harness.** Candidates are evaluated on a shared evolution-validation cohort under matched model and task settings. **(4) Analyze and record outcomes.** The proposer compares the candidate with its parents, reviews score changes and execution outcomes for affected patients, assigns the four case-level effect categories, and updates the Candidate-DAG and branch history with the declared module interventions. **(5) Evolve the proposer.** The accumulated evidence is distilled into new or revised proposer skills, updating the optimization knowledge used to generate the next candidate. Repeating this cycle couples harness improvement with the evolution of proposal procedures. The entity-graph initialization provides a clinically structured starting point, while the Candidate-DAG and proposer skills preserve experimental experience and guide subsequent exploration.

Table 2: Performance comparison on ClinicalBench.

| Framework | Methods | Department Guide | | Clinical Diagnosis | | | | | | Imaging Diagnosis | Overall | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Acc | DIFR | PD | DB | DD | FD | PT | TP | | DWR | CDR | Acceptability | Avg. |
| General-purpose LLMs | GPT-5.1 | 0.7000 | **1.0000** | 0.4087 | 0.3779 | 0.2853 | 0.6304 | 0.3051 | 0.2169 | 0.4023 | 7.3478 | 0.5870 | 0.2233 | 0.4807 |
| | GPT-o3 | 0.7391 | 0.9993 | 0.4022 | 0.4213 | 0.3548 | 0.6065 | 0.3504 | 0.2290 | 0.3947 | 9.4348 | 0.5565 | 0.2256 | 0.4997 |
| | DeepSeek V3.2 | 0.7043 | 0.9926 | 0.4587 | 0.4508 | 0.3432 | 0.6370 | 0.3086 | 0.2780 | 0.5041 | 10.2174 | 0.5739 | 0.2477 | 0.5197 |
| | Kimi-K2 | 0.5804 | 0.9133 | 0.3022 | 0.3763 | 0.2500 | 0.5217 | 0.3428 | 0.3599 | 0.4075 | 4.9130 | 0.4543 | 0.1822 | 0.4504 |
| | Qwen3-32B | 0.6108 | **1.0000** | 0.3543 | 0.4379 | 0.3273 | 0.4848 | 0.3282 | 0.2579 | 0.4166 | 6.5217 | 0.4978 | 0.2034 | 0.4687 |
| Medical LLMs | MedGemma-27B | 0.6826 | 0.9847 | 0.3261 | 0.4597 | 0.3337 | 0.5587 | 0.3692 | 0.2851 | 0.3411 | 5.9130 | 0.4783 | 0.2002 | 0.4823 |
| | Baichuan-M2-32B | 0.6978 | 0.9953 | 0.4161 | 0.4628 | 0.3482 | 0.5643 | 0.3157 | 0.2656 | 0.4156 | 9.3478 | 0.5543 | 0.2274 | 0.4979 |
| | Lingshu-32B | 0.6695 | 0.9820 | 0.3870 | 0.4018 | **0.4289** | 0.5522 | **0.3873** | **0.4069** | 0.3795 | 12.0435 | 0.5717 | 0.2626 | 0.5105 |
| | HuatuoGPT-o1-8B | 0.5173 | 0.9960 | 0.1369 | 0.1929 | 0.1887 | 0.3344 | 0.2218 | 0.3891 | 0.1190 | 1.0000 | 0.2543 | 0.0689 | 0.3440 |
| Multi-agent systems | MedAgent | 0.6760 | 0.9935 | 0.3122 | 0.4371 | 0.3394 | 0.4163 | 0.2801 | 0.2815 | 0.4195 | 4.8696 | 0.4130 | 0.1664 | 0.4617 |
| | MDAgents | 0.6543 | 0.9978 | 0.3311 | 0.4259 | 0.3347 | 0.4561 | 0.3171 | 0.2835 | 0.3369 | 4.0000 | 0.4783 | 0.1853 | 0.4597 |
| | MedChain-Agent | 0.8239 | 0.9978 | 0.4783 | 0.4421 | 0.3423 | 0.6000 | 0.3541 | 0.2666 | 0.5043 | 12.9565 | **0.6065** | **0.2645** | 0.5344 |
| | MedAgent-Pro | **0.8586** | 0.9870 | 0.5065 | 0.3159 | 0.2761 | 0.5935 | 0.3195 | 0.2687 | 0.2997 | 5.4783 | 0.5717 | 0.1892 | 0.4917 |
| Harness methods | Meta-Harness | 0.7021 | 0.9989 | 0.8456 | 0.4617 | 0.3411 | 0.6391 | 0.3084 | 0.2902 | 0.4885 | 13.6522 | 0.4434 | 0.1676 | 0.5640 |
| | Self-Harness | 0.6526 | 0.9989 | 0.8401 | 0.4638 | 0.3343 | 0.6081 | 0.3101 | 0.2850 | 0.4908 | 13.5217 | 0.3846 | 0.1449 | 0.5537 |
| | **Ours** | 0.7978 | **1.0000** | **0.8956** | **0.4918** | 0.3867 | **0.7086** | 0.3304 | 0.3202 | **0.5581** | **14.7826** | 0.5608 | 0.2341 | **0.6099** |

# 4 EXPERIMENTS

## 4.1 DATASETS AND EXPERIMENTAL SETTINGS

**Datasets.** We evaluate our framework on two public clinical decision-making benchmarks: MedChain (Liu et al., 2025) and ClinicalBench (Yan et al., 2024) datasets. MedChain is a multimodal benchmark for sequential clinical workflows, whereas ClinicalBench evaluates clinical reasoning using text-based patient cases. For fair comparison, we follow the original evaluation protocol of each benchmark and report task-level scores and their overall averages. Further dataset and evaluation details are provided in Appendix A.

**Implementation details.** Our Entity-Graph-Based Medical Harness is built around Qwen3.5-9B, whose parameters remain fixed throughout harness optimization. We use GPT-5.6-Sol in Codex as the proposer to iteratively refine the surrounding entity-graph-based medical harness components, including memory, skills, tools, and stage-wise orchestration. Graph memory and clinical skills are initialized from training cases, while validation feedback guides candidate evaluation and updates to the Candidate-DAG memory and proposer skill library.

## 4.2 COMPARISON WITH THE STATE-OF-THE-ARTS

We compare our framework with four categories of approaches: general-purpose LLMs, medical LLMs, medical multi-agent systems, and automatic harness optimization methods. As shown in Tables 1 and 2, our framework achieves the highest average scores among the evaluated methods, reaching 0.5971 on MedChain and 0.6099 on ClinicalBench. Built around a frozen Qwen3.5-9B backbone, it outperforms the evaluated general-purpose and medical LLM baselines on both benchmark averages. These results highlight the potential of harness engineering to improve clinical agents without updating the base model's parameters.

Compared with medical multi-agent systems, our framework improves the average score over MedChain-Agent (Liu et al., 2025), the strongest baseline in this category, by 14.83% on MedChain and 14.13% on ClinicalBench. On MedChain, the advantage spans all six task metrics, covering both early referral decisions and downstream diagnosis and treatment. For example, our treatment score reaches 0.6430, compared with 0.5215 for MedAgent-Pro, the strongest multi-agent baseline on this stage. This pattern is consistent with the motivation for coordinating patient evidence and module execution throughout the clinical trajectory.

Compared with automatic harness optimization methods, our framework improves the average score by 8.60% over Self-Harness (Zhang et al., 2026) on MedChain and by 8.14% over Meta-Harness (Lee et al., 2026) on ClinicalBench, the strongest respective baselines. On ClinicalBench, it exceeds both optimizers across all nine task-level metrics; for example, final diagnosis improves from 0.6391 with Meta-Harness to 0.7086. These improvements suggest that clinically structured patient representations and accumulated optimization experience provide useful support for harness evolution in sequential medical decision making.

Table 3: Component ablations on MedChain dataset.

| Methods | Specialty Referral | | History Taking | Examination | Diagnosis | Treatment | Average |
|---|---|---|---|---|---|---|---|
| | Level 1 | Level 2 | | | | | |
| w/o Memory | 0.5800 | 0.3211 | 0.4872 | 0.6796 | 0.5329 | 0.5790 | 0.5300 |
| w/o Clinical skills | 0.6380 | 0.3337 | 0.6072 | 0.6814 | 0.5384 | 0.5798 | 0.5630 |
| w/o Tools | 0.6731 | 0.3927 | **0.6142** | 0.6459 | **0.5768** | 0.6392 | 0.5903 |
| w/o Self-Evolving Proposer | 0.6490 | 0.3839 | 0.6095 | 0.6565 | 0.5449 | 0.6016 | 0.5742 |
| **Full harness** | **0.6740** | **0.3933** | 0.6130 | **0.6837** | 0.5753 | **0.6430** | **0.5971** |

Table 4: Effect of entity-graph scale on MedChain dataset.

| Framework | Methods | Specialty Referral | | History Taking | Examination | Diagnosis | Treatment | Average |
|---|---|---|---|---|---|---|---|---|
| | | Level 1 | Level 2 | | | | | |
| Our framework | 25% scale | 0.6437 | 0.3548 | 0.5904 | 0.6582 | 0.5479 | 0.6187 | 0.5690 |
| | 50% scale | 0.6594 | 0.3746 | 0.6018 | 0.6707 | 0.5617 | 0.6311 | 0.5832 |
| | 75% scale | 0.6689 | 0.3857 | 0.6086 | 0.6794 | 0.5706 | 0.6381 | 0.5919 |
| | **Full graph** | **0.6740** | **0.3933** | **0.6130** | **0.6837** | **0.5753** | **0.6430** | **0.5971** |

### 4.3 Ablation Study

Table 3 summarizes component ablations on MedChain. Removing memory and clinical skills reduces the average from 0.5971 to 0.5300 and 0.5630, respectively. The largest decline occurs without memory, highlighting the value of patient-relevant disease associations and structured evidence. Clinical skills complement this information with reusable, stage-specific reasoning guidance. Removing tools lowers the average to 0.5903, with the main impact concentrated in examination, consistent with the observation–verification–reporting workflow's role in supplying reviewed imaging findings. We interpret smaller fluctuations in other stages primarily as LLM reasoning variability. Removing the self-evolving proposer reduces the average to 0.5742. This decline supports using Candidate-DAG memory and proposer skills to convert past interventions and case-level effects into guidance for module reuse, branch exploration, and failure avoidance.

### 4.4 Effect of Entity-Graph Scale

Table 4 shows that increasing entity-graph scale from 25% to 50%, 75%, and the full graph raises the average score on MedChain from 0.5690 to 0.5832, 0.5919, and 0.5971, respectively. At 25% scale, all six task metrics are lower than with the full graph. This trend is consistent with broader coverage of evidence–disease associations supporting hypothesis retrieval and the assessment of supporting, contradictory, and missing evidence. Marginal gains diminish as graph scale increases: the 75% configuration is only 0.0052 below the full graph, suggesting that much of the performance benefit is already obtained at this scale, while additional graph content still improves the overall score.

## 5 Conclusion

We presented a framework that learns to evolve clinical harnesses by coupling entity-graph-guided harness with a self-evolving proposer. The entity graph maintains a shared, stage-visible patient state that coordinates memory retrieval, skill selection, and tool execution while preserving evidence provenance. Complementing this structured harness, the self-evolving proposer uses Candidate-DAG memory and an evolving proposer skill library to transform accumulated experimental outcomes into reusable optimization procedures. This design allows evaluation feedback to improve not only the harness itself, but also how subsequent modifications are proposed. Extensive experiments show superior performance over the evaluated baselines, with ablations supporting the contributions of the proposed components. These findings highlight the potential of clinically grounded orchestration and experience-driven proposal refinement for improving long-horizon clinical agents.

### AI USE STATEMENT

Generative AI tools were used to assist with language editing and manuscript organization. Within the proposed framework, Qwen3.5-9B serves as the clinical reasoning backbone, while GPT-5.6-Sol in Codex serves as the proposer for generating and implementing harness modifications, analyzing evaluation feedback, and evolving proposer skills. These uses are part of the experimental method described in the paper.

### REPRODUCIBILITY STATEMENT

Section 3 describes the entity-graph representation, harness components, selection rules, and candidate evolution procedure. Section 4.1 identifies the runtime and proposer models, while Appendix A specifies the dataset partitions and evaluation metrics. Harness candidates are evaluated on shared validation cohorts drawn from the training partitions using department-proportional stratification. The code will be made publicly available on GitHub after the publication of this paper.

## REFERENCES

Zhuohan Ge, Haoyang Li, Yubo Wang, Nicole Hu, Chen Jason Zhang, and Qing Li. ClinicalAgents: Multi-agent orchestration for clinical decision making with dual-memory. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026. doi: 10.1145/3770855.3818931.

Shengran Hu, Cong Lu, and Jeff Clune. Automated design of agentic systems. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2408.08435.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *The Twelfth International Conference on Learning Representations*, 2024.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into self-improving pipelines. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.03714.

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik Siu Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae Won Park. MDAgents: An adaptive collaboration of LLMs for medical decision-making. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2522.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-Harness: End-to-end optimization of model harnesses. *arXiv preprint arXiv:2603.28052*, 2026.

Shuyue Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan S. Ilgen, Emma Pierson, Pang Wei Koh, and Yulia Tsvetkov. MediQ: Question-asking LLMs and a benchmark for reliable interactive clinical reasoning. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.00922.

Jiahang Lin, Shichun Liu, Chengjun Pan, Lizhi Lin, Shihan Dou, Zhiheng Xi, Xuanjing Huang, Hang Yan, Zhenhua Han, Tao Gui, and Yu-Gang Jiang. Agentic harness engineering: Observability-driven automatic evolution of coding-agent harnesses. *arXiv preprint arXiv:2604.25850*, 2026.

Jie Liu, Wenxuan Wang, Zizhan Ma, Guolin Huang, Yihang Su, Kao-Jung Chang, Haoliang Li, Linlin Shen, Michael R Lyu, and Wenting Chen. MedChain: Bridging the gap between LLM agents and clinical practice with interactive sequence. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2498.

Xinghua Lou, Miguel Lázaro-Gredilla, Antoine Dedieu, Carter Wendelken, Wolfgang Lehrach, and Kevin P. Murphy. AutoHarness: Improving LLM agents by automatically synthesizing a code harness. *arXiv preprint arXiv:2603.03329*, 2026.

Xiaotian Luo, Fengxingyu Wang, Chuanrui Hu, Dizhan Xue, and Yafeng Deng. Self-evolving agent harnesses via gated semantic quality-diversity. *arXiv preprint arXiv:2607.13683v1*, 2026. URL https://arxiv.org/abs/2607.13683v1.

Harsha Nori, Mayank Daswani, Christopher Kelly, Scott Lundberg, Marco Tulio Ribeiro, Marc Wilson, Xiaoxuan Liu, Viknesh Sounderajah, Jonathan Carlson, Matthew P. Lungren, Bay Gross, Peter Hames, Mustafa Suleyman, Dominic King, and Eric Horvitz. Sequential diagnosis with language models. *arXiv preprint arXiv:2506.22405*, 2025. URL https://arxiv.org/abs/2506.22405.

Akshat Sanghvi, Naren Akash, Raza Imam, Amit Sharma, and Mohit Jain. MeDxAgent: Multi-agent consultation for interactive medical diagnosis. *arXiv preprint arXiv:2606.03416*, 2026. URL https://arxiv.org/abs/2606.03416.

Samuel Schmidgall, Rojin Ziaei, Carl Harris, Eduardo Reis, Jeffrey Jopling, and Michael Moor. AgentClinic: A multimodal agent benchmark to evaluate AI in simulated clinical environments. *arXiv preprint arXiv:2405.07960v1*, 2024. URL https://arxiv.org/abs/2405.07960v1.

Haoran Sun, Wenjie Li, Yujie Zhang, Zekai Lin, Fanrui Zhang, Kaitao Chen, Xingqi He, Yichen Li, Mianxin Liu, Lei Liu, and Yankai Jiang. Experience makes skillful: Enabling generalizable medical agent reasoning via self-evolving skill memory. *arXiv preprint arXiv:2606.09365*, 2026. URL https://arxiv.org/abs/2606.09365.

Varun Ursekar, Apaar Shanker, Veronica Chatrath, Yuan Xue, and Samuel Marc Denton. VeRO: A harness for agents to optimize agents. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. arXiv:2602.22480.

Yike Wang, Huaisheng Zhu, Zhengyu Hu, Yige Yuan, Zhengyu Chen, Shakti Senthil, Hannaneh Hajishirzi, Yulia Tsvetkov, Pradeep Dasigi, and Teng Xiao. Rethinking the evaluation of harness evolution for agents. *arXiv preprint arXiv:2607.12227*, 2026. URL https://arxiv.org/abs/2607.12227.

Yiqing Xie, Sheng Zhang, Hao Cheng, Pengfei Liu, Zelalem Gero, Cliff Wong, Tristan Naumann, Hoifung Poon, and Carolyn Rose. DocLens: Multi-aspect fine-grained evaluation for medical text generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 649–679. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.39. URL https://aclanthology.org/2024.acl-long.39/.

Weixiang Yan, Haitian Liu, Tengxiao Wu, Qian Chen, Wen Wang, Haoyuan Chai, Jiayi Wang, Weishan Zhao, Yixin Zhang, Renjun Zhang, Li Zhu, and Xuandong Zhao. ClinicalLab: Aligning agents for multi-departmental clinical diagnostics in the real world. *arXiv preprint arXiv:2406.13890*, 2024. URL https://arxiv.org/abs/2406.13890.

Hangfan Zhang, Shao Zhang, Kangcong Li, Chen Zhang, Yang Chen, Yiqun Zhang, Lei Bai, and Shuyue Hu. Self-Harness: Harnesses that improve themselves. *arXiv preprint arXiv:2606.09498*, 2026.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin gödel machine: Open-ended evolution of self-improving agents. *arXiv preprint arXiv:2505.22954*, 2025.

Yangyang Zhuang, Wenjia Jiang, Jiayu Zhang, Ze Yang, Joey Tianyi Zhou, and Chi Zhang. Learning to be a doctor: Searching for effective medical agent architectures. In *Proceedings of the ACM International Conference on Multimedia*, 2025. URL https://arxiv.org/abs/2504.11301.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Language agents as optimizable graphs. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.16823.

# A Dataset and Evaluation Details

## A.1 MedChain

**Dataset composition.** MedChain (Liu et al., 2025) comprises 12,163 clinical cases spanning 19 medical specialties and 156 subspecialties, together with 7,338 medical images and corresponding reports. The cases originate from the Chinese medical platform iiYi and include patient characteristics, chief complaints, medical histories, examination findings, diagnoses, and treatments. Records undergo standardization, de-identification, and clinical quality assessment. From the original training partition of 9,724 cases, we reserve 1,000 cases for harness evolution using department-proportional stratification, leaving 8,724 cases for memory and skill construction and validation. The test partition contains 2,439 cases

**Clinical workflow.** Each case progresses through five stages: specialty referral, history taking, examination, diagnosis, and treatment. Agents initially receive limited patient information and acquire additional evidence through simulated consultations. Intermediate outputs inform subsequent stages, allowing errors in earlier decisions to affect downstream performance. The examination stage requires interpreting medical images and generating corresponding reports. This setting assesses whether an agent can maintain coherent decisions as patient information accumulates.

**Evaluation.** Specialty referral uses accuracy for first-level departments and intersection over union (IoU) for second-level departments. History taking evaluates examination selection using IoU. Examination evaluates report completeness using the claim recall metric from DocLens (Xie et al., 2024), while diagnosis uses a five-level clinical grading rubric. Treatment is evaluated through IoU between predicted and reference treatment items. Our main table reports both referral levels and the remaining four stage scores, together with their overall average.

## A.2 ClinicalBench

**Dataset composition.** ClinicalBench, introduced in ClinicalLab (Yan et al., 2024), contains 1,500 de-identified cases from Chinese Grade 3A hospitals, covering 24 departments and 150 diseases, with ten cases per disease. Records include patient demographics, chief complaints, medical histories, examination results, imaging reports, and clinician-provided diagnostic and treatment information. From the original training partition of 1,040 cases, we reserve 200 cases for harness evolution using department-proportional stratification, leaving 840 cases for memory and skill construction and validation. The test partition contains 460 cases.

**Clinical tasks.** The benchmark comprises eight tasks: department guidance, imaging diagnosis based on textual reports, and six clinical diagnosis tasks. These six tasks are preliminary diagnosis (PD), diagnostic basis (DB), differential diagnosis (DD), final diagnosis (FD), principle of treatment (PT), and treatment plan (TP). Together, they assess department selection, evidence-based reasoning, diagnostic differentiation, and treatment planning.

**Evaluation.** Department guidance uses accuracy and the department instruction following rate (DIFR), which evaluates compliance with the requested number and permitted names of departments. PD and FD use accuracy; the remaining generative tasks use the mean of BLEU, ROUGE, and BERTScore. The overall average combines these nine scores. Additional measures include department win rate (DWR), summarizing relative performance across departments; comprehensive diagnostic rate (CDR), measuring jointly correct department selection and diagnosis; and Acceptability, combining CDR with the average quality of generated diagnostic content. These three measures are reported separately from the overall average.

# B MedChain Cases Demonstration

This section presents entity-graph and entity-graph-based skill cases from MedChain. The patient entity-graph and clinical skill illustrate the entity-graph-based medical harness. We translate clinical concepts into English and omit redundant metadata, while preserving the stored node types, eligibility conditions, and module intervention.

Table 5: Evidence nodes in the compact patient entity graph.

| ID | Normalized evidence | First visible stage | Clinical role |
|---|---|---|---|
| $E_1$ | Persistent right-upper-quadrant pain | Task 1 | Focal biliary pain |
| $E_2$ | Fever | Task 1 | Inflammatory evidence |
| $E_3$ | Nausea with vomiting | Task 1 | Associated gastrointestinal symptom |
| $E_4$ | Acute onset, approximately three hours | Task 1 | Temporal acuity |
| $E_5$ | Poor response to antipyretic treatment | Task 1 | Persistent inflammatory course |
| $E_6$ | Upper-abdominal pain | Task 1 | Broader pain localization |
| $E_7$ | Positive Murphy sign | Task 2 | Biliary examination finding |
| $E_8$ | Abdominal ultrasonography performed | Task 2 | Imaging provenance |
| $E_9$ | Gallbladder-wall thickening | Task 2 | Biliary imaging finding |
| $E_{10}$ | Soft abdomen | Task 2 | Negative peritoneal finding |
| $E_{11}$ | Normal bowel sounds | Task 2 | Preserved bowel activity |
| $E_{12}$ | No palpable abdominal mass | Task 2 | Negative mass finding |
| $E_{13}$ | Gallstones, verified finding | Task 3 | Biliary etiology |
| $E_{14}$ | Liver and spleen not palpable | Task 2 | Negative organ enlargement |

Table 6: Disease nodes and retained relations in the compact patient entity graph.

| ID | Disease node | Status | Selected incoming or differential relations |
|---|---|---|---|
| $D_1$ | Acute cholecystitis | Suspected | $E_1, E_2, E_6, E_7, E_9, E_{13} \rightarrow D_1$ (support); $E_{10} \rightarrow D_1$ (contradict) |
| $D_2$ | Acute pancreatitis | Differential | $D_1 \leftrightarrow D_2$ and $D_3 \leftrightarrow D_2$ (confusable) |
| $D_3$ | Common-bile-duct stone | Suspected | $E_1, E_2, E_6, E_{13} \rightarrow D_3$ (support); $D_3 \leftrightarrow D_2$ (confusable) |
| $D_4$ | Gallstones | Differential | $D_1 \leftrightarrow D_4$ (confusable) |

### B.1 Patient Entity Graph

Tables 5 and 6 give a compact subgraph extracted from MedChain case `1052` ("Treatment of acute cholecystitis"). It contains 14 evidence nodes and four disease nodes. The evidence remains stage-indexed: history is available at Task 1, physical examination and ultrasound findings are added at Task 2, and the verified gallstone finding is written back at Task 3.

### B.2 Entity-Graph-Based Clinical Skill

Table 7 shows a Task 4 contrastive skill from the clinical-skill library constructed from training cases. The example is clinically meaningful because urinary frequency and urgency are nonspecific: the skill prevents the harness from inferring benign prostatic hyperplasia (BPH) from surface symptoms alone and requires disease-specific evidence before the instruction is injected. For better illustration, the conditions are translated to sentences from discrete evidence nodes.

Table 7: Entity-graph-based clinical skill for the BPH diagnostic boundary.

| Field | Skill content |
|---|---|
| Trigger condition | The stage-visible patient graph contains lower-urinary-tract irritative symptoms, such as urinary frequency or urgency. |
| Decisive condition | The graph additionally contains explicit evidence of BPH or prostatic enlargement; the decisive evidence must be present before the skill is eligible. |
| Rejected condition | Reject the skill when the graph instead supports a competing context such as positive urinary leukocytes, pediatric age, or female sex. |
| Skill content | Lower-urinary-tract symptoms occur in several conditions and are not diagnostic by themselves. When BPH-specific evidence is present and no rejected condition is active, prioritize BPH in the differential and explicitly contrast it with urinary infection, cystitis, diabetes-related urinary symptoms, and bladder stones. |