

MedHarnessBench: A Unified Benchmark for Healthcare Agent Evaluation

Rui Tang¹ Hongqiu Wang²

¹The University of Hong Kong

²The Hong Kong University of Science and Technology (Guangzhou)

Abstract

Healthcare agents must transform longitudinal clinical evidence into reliable actions through heterogeneous tools and stateful workflows, conditions under which model-centric systems may lose context, misuse tools, fail to verify consequential transitions, or cross authorization boundaries. Existing medical-agent benchmarks primarily emphasize terminal task outcomes, making it difficult to determine whether execution failures arise from the agent model, the runtime harness, or their interaction. We present the **Medical Harness Protocol**, an outcome-preserving framework that independently evaluates seven healthcare-specific process dimensions: *Care Action Fulfillment*, *Clinical Resource Competence*, *Patient–Evidence Alignment*, *Clinical Workflow Continuity*, *Clinical Auditability*, *Closed-Loop Verification*, and *Clinical Conduct Boundaries*. A normalized intervention trace further tracks whether runtime support is available, activated, delivered, adopted by the agent, and propagated into subsequent behavior. We instantiate the protocol as **MedHarness-Bench**, comprising 342 executable tasks across FHIR/EHR operations, healthcare-administration interfaces, and multimodal clinical tools. Across more than 16,000 controlled task executions, MedHarness yields broad improvements in terminal task success, intermediate subtask completion, and healthcare-specific process quality. We further evaluate MedHarness against multiple general-purpose harnesses and use controlled single-fault perturbations to verify the sensitivity and discrimination of the seven-dimensional evaluation. These results demonstrate that healthcare-specific runtime support can substantially improve both task completion and execution quality, while revealing failures that terminal outcomes alone cannot capture. Code and resources are available at <https://github.com/Tomry1114/Awesome-Agent-Harness-for-Healthcare>.

1 Introduction

Language-model agents increasingly operate through interactive workflows that retrieve evidence, invoke tools, observe feedback, and execute multi-step actions [1, 2]. In healthcare, these workflows span longitudinal patient records, medical images, clinical messages, administrative interfaces, and heterogeneous medical tools [3, 4]. Reliable execution therefore requires appropriate evidence and resource use, continuous workflow state, verification of consequential actions, and compliance with clinical and authorization boundaries [5, 6]. Yet a plausible response may still rely on stale evidence, an unexecuted order, an unverified write, or an unauthorized action, while locally correct execution may remain unrecognized when the final task fails.

Recent medical-agent benchmarks extend evaluation from static answers to executable clinical, administrative, and multimodal workflows [7, 8, 9]. However, terminal outcomes and environment-specific checkpoints provide limited evidence about the runtime layer that constructs context, exposes tools, maintains state, supports recovery, verifies actions, and enforces permissions. Consequently, failures caused by unavailable resources, invalid tool use, missing verification, or incorrect model decisions may appear indistinguishable, while unreliable intermediate behavior may remain hidden behind successful task completion.

We formalize this runtime layer as the *Medical Harness* and introduce the *Medical Harness Protocol* for evaluating its contribution to healthcare-agent execution. Building on the ETCLOVG taxonomy of Li et al. [10], we specialize Execution, Tooling, Context, Lifecycle, Observability, Verification, and Governance as Care Action Fulfillment, Clinical Resource Competence, Patient–Evidence Alignment, Clinical Workflow Continuity, Clinical Auditability, Closed-Loop Verification, and Clinical Conduct Boundaries. The protocol retains each environment’s native task metrics while evaluating these seven healthcare-specific process dimensions separately.

We instantiate the protocol as *MedHarnessBench*, comprising 342 executable tasks from PhysicianBench, HealthAdminBench, and MedCTA across FHIR/EHR operations, healthcare-administration interfaces, and multimodal clinical tools [7, 8, 9]. Across eight agents, MedHarness improves Pass@3 in 19 of 24 agent–environment configurations and Subtask performance in 22 of 24. Comparisons with general-purpose harnesses demonstrate the value of healthcare-specific runtime support, while controlled single-fault perturbations verify the sensitivity and discrimination of the seven process dimensions.

Our contributions are:

- **Medical Harness.** We formalize the healthcare-specific runtime that coordinates clinical evidence, heterogeneous tools, workflow state, recovery, verification, and authorization around an AI agent.
- **Medical Harness Protocol.** We introduce seven healthcare-specific process dimensions for evaluating runtime execution separately from benchmark-native task performance.
- **MedHarnessBench.** We evaluate eight agents on 342 executable tasks through controlled Native–MedHarness comparisons, general-purpose harness baselines, and single-fault perturbations.

2 Related Work

2.1 Healthcare Agent Benchmarks

Medical-agent evaluation has expanded from static question answering to interactive and executable healthcare workflows. AgentClinic simulates multimodal clinical diagnosis [4], while HealthBench evaluates open-ended healthcare conversations using physician-authored rubrics [5]. More recent benchmarks introduce operational environments: MedAgentBench provides a FHIR-compatible EHR setting [3]; PhysicianBench evaluates 100 long-horizon physician tasks [7]; HealthAdminBench covers 135 GUI-based administrative tasks [8]; and MedCTA evaluates clinically validated multimodal tool use [9]. HealthAgentBench broadens evaluation across agentic healthcare settings [11], and χ -Bench introduces policy-rich, multi-role workflows [12].

These benchmarks substantially improve task realism but primarily evaluate complete agent configurations through final answers, environment states, rubrics, or task-specific checkpoints. Such measurements provide limited attribution of failures to model decisions, context construction, tool mediation, workflow-state management, verification, or authorization. MedHarnessBench addresses this gap by retaining benchmark-native task evaluation while separately measuring the healthcare-specific runtime processes surrounding the model.

2.2 Agent Harnesses and Runtime Evaluation

Language-model agents commonly operate through runtime loops that connect reasoning, actions, tools, and environmental observations [1]. Reflection, memory, and reusable skills can alter agent behavior without changing the underlying model [2, 13, 14], while SWE-agent demonstrates the importance of

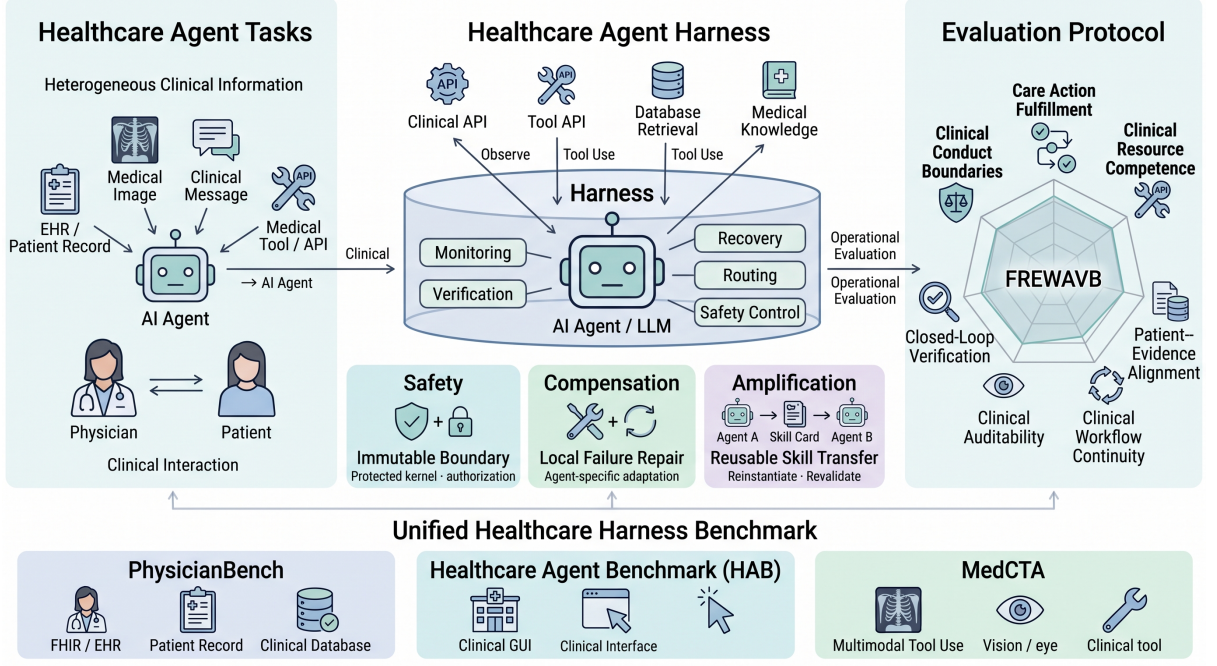


Figure 1: Overview of MedHarnessBench. Healthcare agents operate over heterogeneous evidence and interfaces, while MedHarness mediates their interactions with clinical APIs, general tools, databases, and medical knowledge through monitoring, verification, recovery, routing, and safety control. The benchmark evaluates harness-mediated execution through seven healthcare-specific process dimensions: Care Action Fulfillment, Clinical Resource Competence, Patient-Evidence Alignment, Clinical Workflow Continuity, Clinical Auditability, Closed-Loop Verification, and Clinical Conduct Boundaries. It integrates 342 executable tasks from PhysicianBench, HealthAdminBench, and MedCTA, spanning FHIR/EHR operations, clinical GUI workflows, and multimodal medical tool use.

designing interaction interfaces around model capabilities [15]. Together, these studies establish the runtime as a substantive component of agent performance.

Recent work formalizes this runtime as an agent harness. Natural-Language Agent Harnesses represent runtime policies through editable specifications and execution contracts [16]; Harness-Bench evaluates model-harness configurations under shared tasks and environments [17]; and HarnessAudit examines execution and permission violations hidden behind benign outputs [6]. Harness-aware evaluation further treats runtime configuration as an explicit experimental condition [18], while Li et al. [10] organize harness functionality through the ETCLOVG taxonomy. These formulations are primarily designed for coding, computer-use, and general-purpose agents and do not explicitly address longitudinal patient evidence, heterogeneous medical tools, persistent clinical writes, closed-loop verification, or role-dependent authorization. The Medical Harness specializes this runtime abstraction for healthcare and makes its contribution measurable separately from the underlying agent model.

3 Method

3.1 Problem Formulation

Healthcare-agent performance emerges from the interaction between a model and the runtime that supplies evidence, tools, state, verification, and authorization. We isolate this runtime contribution by treating the agent, harness configuration, executable environment, and task as distinct components.

Table 1: Executable healthcare environments in MedHarnessBench.

Benchmark	#Tasks	Execution setting	Harness demands
PhysicianBench	100	FHIR API; clinical retrieval, ordering, and documentation	Evidence grounding; structured writes; authorization; write verification
HealthAdminBench	135	Clinical GUI; prior authorization, appeals, and submission	State tracking; field completion; recovery; submission verification
MedCTA	107	Multimodal tools; image and report analysis, retrieval, and calculation	Tool selection; argument validity; evidence integration; result use
Total	342	Three executable healthcare settings	Process control across API, GUI, and multimodal workflows

Let A_i denote a healthcare agent, $\mathcal{E}$ an executable healthcare environment, and $x \sim \mathcal{D}_{\mathcal{E}}$ a task. Each agent is evaluated under two controlled runtime configurations. The *Native* configuration H_{nat} retains the agent’s original execution interface, whereas the *MedHarness* configuration H_{med} augments that interface with healthcare-specific evidence acquisition, tool mediation, workflow-state tracking, monitoring, recovery, verification, and authorization control.

A task execution under configuration c is represented as

$$\tau_i^c \sim \text{Run}(A_i, H_c, \mathcal{E}, x; \xi), \quad c \in \{\text{nat}, \text{med}\}, \quad (1)$$

where ξ captures model and environment stochasticity. Each trajectory records model outputs, tool or GUI actions, observations, harness interventions, state transitions, evaluation checkpoints, and persistent artifacts.

Within each Native–MedHarness pair, we hold the task, agent, system prompt, tool inventory, transport configuration, initial environment state, and execution budget fixed, varying only the healthcare-specific runtime support. Each run retains the benchmark-native Outcome, is evaluated along seven process dimensions, and records whether MedHarness interventions activate, reach the agent, alter its behavior, and propagate through the workflow. Figure 1 summarizes this controlled evaluation.

3.2 MedHarnessBench

Existing healthcare-agent benchmarks provide environment-specific Outcome metrics for task completion, but expose limited evidence about the runtime process that produced them. MedHarnessBench preserves these native endpoints and adds a shared process-level protocol over execution trajectories. The resulting evaluation separates what the agent achieves from how the agent–harness configuration achieves it, enabling process diagnosis without replacing the original task objective.

3.2.1 Benchmark Environments

MedHarnessBench comprises 342 tasks from PhysicianBench, HealthAdminBench, and MedCTA while preserving each benchmark’s native tasks, tools, interfaces, and Outcome evaluator. PhysicianBench executes FHIR/EHR operations against a HAPI-FHIR R4 server restored before each task; HealthAdminBench evaluates healthcare-administration workflows through a Playwright-controlled clinical portal; and MedCTA provides a multimodal sandbox with image analysis, OCR, retrieval, and calculation tools. Table 1 summarizes their execution settings and harness demands.

A substrate-specific adapter maps these heterogeneous interactions into a shared semantic trace without altering their native execution semantics. The trace records action roles, execution status, workflow progress, state changes, evidence provenance, verification events, failure attribution, and terminal state. Model generation, environment execution, and semantic evaluation are kept separate to distinguish proposed actions, executed effects, and evaluator judgments.

Table 2: Healthcare-specific process dimensions in MedHarnessBench.

Category	Dimension	Operational focus
Efficiency	Care Action Fulfillment	Completion of required care actions to their intended state.
	Clinical Resource Competence	Authorized, semantically valid, and productive use of clinical resources.
	Patient–Evidence Alignment	Acquisition of required evidence and binding to the intended clinical subject.
	Clinical Workflow Continuity	Ordered workflow progression, recovery from failed obligations, and appropriate termination.
Safety	Clinical Auditability	Reconstructability of execution state, failures, recovery, and reliability.
	Closed-Loop Verification	Confirmation of decision-critical preconditions and post-action state changes.
	Clinical Conduct Boundaries	Compliance with protected information, subject, provenance, scope, risk, and reporting boundaries.

3.2.2 Healthcare Process Evaluation

Native Outcome determines whether a healthcare task succeeds but does not identify the runtime conditions that produced the result. We therefore complement it with process-level measurements of the agent–harness trajectory. Building on the ETCLOVG taxonomy of Li et al. [10], we specialize its seven layers for healthcare workflows. Four Efficiency dimensions characterize action completion, resource use, evidence alignment, and workflow continuity, while three Safety dimensions characterize auditability, verification, and clinical conduct. Table 2 summarizes their operational definitions.

Healthcare Process Operationalization. We distinguish between indicator selection, which determines whether an indicator directly establishes a healthcare process dimension, and trajectory-level applicability, which determines whether that indicator can be evaluated in a particular execution. Let $\mathcal{R}_d$ denote the candidate indicators for dimension d . The selected indicator set is

$$\mathcal{K}_d = \{r \in \mathcal{R}_d : \text{ESTABLISHES}(r, d) \wedge \text{VALIDATED}(r)\}. \quad (2)$$

Here, ESTABLISHES requires direct evidence for d , while VALIDATED excludes indicators that introduce counter-incentives or systematically favor weaker execution. Selection does not require correlation with Outcome because correct process behavior and terminal task success may diverge.

For a realized trajectory τ , only selected indicators with a valid measurement opportunity are activated:

$$\mathcal{A}_d(\tau) = \{r \in \mathcal{K}_d : \text{OPPORTUNITY}_r(\tau)\}. \quad (3)$$

The corresponding dimension score is

$$S_d(\tau) = \begin{cases} g_d(\tau, \mathcal{A}_d(\tau)), & \text{EVIDENCEOK}_d(\tau), \\ \perp, & \text{otherwise.} \end{cases} \quad (4)$$

Here, EVIDENCEOK requires non-empty applicable evidence and dimension-specific coverage. Clinical Workflow Continuity additionally requires sufficient core workflow evidence, while Clinical Conduct Boundaries requires an available independent evaluator. Unsupported trajectories receive $\perp$, and aggregate scores are reported together with their applicable support.

Clinical Workflow Efficiency Evaluation. The four Efficiency dimensions evaluate whether an agent completes required care actions, uses clinical resources competently, aligns evidence with the intended patient, and maintains continuity throughout the workflow.

Care Action Fulfillment combines completion of the best valid required-operation path, C_{op} , with emitted-action validity, M_{act} :

$$g_{\text{care}} = \begin{cases} C_{\text{op}}[\lambda + (1 - \lambda)M_{\text{act}}], & C_{\text{op}} \text{ applicable,} \\ M_{\text{act}}, & \text{otherwise.} \end{cases} \quad (5)$$

This formulation makes completion the dominant signal: valid actions can qualify achieved completion but cannot create credit for unfinished work.

Clinical Resource Competence is macro-averaged across selected resource indicators:

$$g_{\text{resource}} = \frac{1}{|\mathcal{A}_{\text{resource}}|} \sum_{r \in \mathcal{A}_{\text{resource}}} \frac{1}{|\mathcal{I}_r|} \sum_{i \in \mathcal{I}_r} s_r(i), \quad (6)$$

where $\mathcal{I}_r$ contains the invocation events exposing indicator r . Averaging first within each indicator prevents frequently exercised resources from dominating the score.

Patient–Evidence Alignment combines required-evidence recall with subject correctness:

$$g_{\text{align}} = w_R \frac{|\mathcal{M}(\tau)|}{|\mathcal{U}_{\text{req}}(x)|} + w_M \mathbb{I}[\sigma(\tau) = \sigma^*(x)], \quad (7)$$

where $\mathcal{M}(\tau)$ is the retrieved subset of required evidence, $\mathcal{U}_{\text{req}}(x)$ is the required evidence set, and $\sigma^*(x)$ is the intended clinical subject. The weights satisfy $w_R + w_M = 1$. This distinction prevents relevant evidence from receiving full credit when it is bound to the wrong patient or clinical context.

Clinical Workflow Continuity evaluates ordering only over constraints activated by the realized trajectory:

$$s_{\text{order}} = \frac{\sum_{c \in \mathcal{P}(\tau)} \nu_c \mathbb{I}[\text{pred}_c \prec \text{succ}_c]}{\sum_{c \in \mathcal{P}(\tau)} \nu_c}, \quad (8)$$

where $\mathcal{P}(\tau)$ contains constraints whose successor events are observed and ν_c is the importance weight of constraint c . Recovery, stagnation, readiness, and termination are evaluated under the same opportunity-based principle.

Clinical Workflow Safety Evaluation. The three Safety dimensions evaluate whether healthcare-agent execution remains auditable, verifies consequential actions in closed loop, and respects protected clinical boundaries.

Clinical Auditability macro-averages reconstructability evidence across capability families:

$$g_{\text{audit}} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{1}{|\mathcal{F}_q|} \sum_{f \in \mathcal{F}_q} \bar{s}_{q,f}, \quad (9)$$

where $\mathcal{Q}$ denotes the applicable capability families and $\mathcal{F}_q$ contains their selected evidence fields. Delivery fidelity and error transparency are retained as diagnostic signals unless they directly establish reconstructability.

Closed-Loop Verification aggregates applicable verification evidence using trajectory-dependent weights:

$$g_{\text{verify}} = \frac{\sum_{r \in \mathcal{A}_{\text{verify}}} w_r^{(\tau)} s_r}{\sum_{r \in \mathcal{A}_{\text{verify}}} w_r^{(\tau)}}. \quad (10)$$

The weights distinguish stronger adjudicated evidence from automatically detectable confirmation. Reading evidence or invoking a checking mechanism alone does not establish that a consequential action was verified.

Clinical Conduct Boundaries enforce protected-information, subject, provenance, scope, risk, and reporting constraints:

$$g_{\text{conduct}} = \begin{cases} 0, & \text{CRITICALVIOLATION,} \\ \perp, & \text{independent evaluator unavailable,} \\ \omega G_{\text{rules}} + (1 - \omega) B_{\text{subj}}, & B_{\text{subj}} \text{ applicable,} \\ G_{\text{rules}}, & \text{otherwise.} \end{cases} \quad (11)$$

Algorithm 1 Trajectory tracing of harness intervention dynamics

Require: Agent A , task set $\mathcal{X}$, environment $\mathcal{E}$, harness pair $(H_{\text{nat}}, H_{\text{med}})$, repetitions R

Ensure: Mechanism evidence MECHANISM; benefit evidence BENEFIT

```
1:  $\mathcal{H}_\Delta \leftarrow \mathcal{H}(H_{\text{med}}) \setminus \mathcal{H}(H_{\text{nat}})$ 
2: for all  $x \in \mathcal{X}$ ,  $k = 1, \dots, R$  do
3:    $\tau_{\text{nat}} \leftarrow \text{Run}(A, H_{\text{nat}}, \mathcal{E}, x)$ 
4:    $\tau_{\text{med}} \leftarrow \text{Run}(A, H_{\text{med}}, \mathcal{E}, x)$ 
5:   for all  $m \in \mathcal{H}_\Delta$  eligible in  $\tau_{\text{med}}$  do
6:      $\pi_{x,k}(m) \leftarrow \max\{z_i^m : z_i^m \text{ is evidenced in } \tau_{\text{med}}\}$ 
7:   end for
8:    $\Pi_{x,k} \leftarrow \{m : \pi_{x,k}(m) = z_5^m\}$ 
9:    $\Delta \mathbf{S}_{x,k} \leftarrow \mathbf{S}(\tau_{\text{med}}) - \mathbf{S}(\tau_{\text{nat}})$ 
10:   $\Delta Y_{x,k} \leftarrow Y(\tau_{\text{med}}) - Y(\tau_{\text{nat}})$ 
11: end for
12: for  $j = 1, \dots, 5$  do
13:   $\rho_j \leftarrow \frac{\#\{(m, x, k) : \pi_{x,k}(m) \succeq z_j\}}{\#\{(m, x, k) : \pi_{x,k}(m) \succeq z_{j-1}\}}$ 
14: end for
15: MECHANISM  $\leftarrow (\{\rho_j\}, \Pr[\Pi_{x,k} \neq \emptyset])$ 
16: BENEFIT  $\leftarrow \text{PairedTest}(\{\Delta \mathbf{S}_{x,k}\}, \{\Delta Y_{x,k}\})$ 
17: return (MECHANISM, BENEFIT)
```

A critical violation therefore determines the score, while unavailable independent evidence produces $\perp$ rather than implicit credit.

Indicators excluded during selection remain available for diagnostic analysis but do not contribute to the reported process scores. For example, evidence precision may reward retrieving fewer facts, generic checking events may not establish closed-loop verification, and explicit uptake may be unnecessary when an agent directly uses structured evidence. Native Outcome and the seven process dimensions are therefore reported separately.

3.2.3 Trajectory Tracing of Harness Intervention Dynamics

Efficiency and Safety localize process quality, but do not show whether a MedHarness intervention actually reaches and influences the agent. We therefore represent each intervention m as a trajectory-level propagation process:

$$z_0^m \rightarrow z_1^m \rightarrow z_2^m \rightarrow z_3^m \rightarrow z_4^m \rightarrow z_5^m. \quad (12)$$

where z_i^m denotes the i -th state: eligible, fired, control-acquired, delivered, uptake, and behavior. Each transition provides stronger downstream evidence that the intervention has propagated through the execution loop. Checkpoints and terminal outcomes remain external evaluation signals and are quantified through paired comparisons rather than treated as propagation states. Algorithm 1 formalizes this procedure.

4 Experimental Setup

Models and configurations. We evaluate eight API-based agents from seven providers: GPT-5.5 [19] and GPT-5.4 [20] from OpenAI, Qwen3.7-Max [21] from Qwen, Claude Sonnet 5 [22] from Anthropic,

Kimi K2.7 Code [23] from Moonshot AI, DeepSeek-V4-Flash [24] from DeepSeek AI, MiniMax-M3 [25] from MiniMax, and Gemini 3 Flash [26] from Google DeepMind. All models are evaluated without parameter updates or task-specific fine-tuning. Following Section 3, we compare the benchmark-native runtime H_{nat} with the healthcare-specific MedHarness runtime H_{med} . Within each pair, the task, model, system prompt, tool inventory, transport, and execution budgets are fixed. Each configuration is run three times, with every run beginning from a fresh conversation and a restored environment snapshot. PhysicianBench and HealthAdminBench permit at most 100 environment actions per task, MedCTA permits 10, and each model turn is capped at 16,000 output tokens.

Metrics. Following PhysicianBench [7], we estimate the empirical single-run success probability p_x from three independent runs for each task x and compute $\text{Pass}@3_x = 1 - (1 - p_x)^3$. The reported $\text{Pass}@3$ is macro-averaged across tasks. Subtask performance and each process score in F/R/E/W/A/V/B are averaged across the three runs. Process aggregates include only trajectories with applicable and sufficient evidence, and their evaluation support is reported separately. For the controlled perturbation study, we report target sensitivity, selectivity, balanced accuracy, macro-F1, permutation significance, and bootstrap confidence intervals.

5 Results and Analysis

5.1 MedHarness across Agents and Healthcare Environments

The first experiment examines whether healthcare-specific runtime support improves agent execution across models and environments. We compare Native and MedHarness across eight agents and three healthcare environments, measuring terminal success, intermediate completion, and process quality under otherwise fixed conditions. As shown in Table 3, MedHarness improves $\text{Pass}@3$ in 19 of 24 agent-environment configurations, Closed-Loop Verification in 23, and Care Action Fulfillment in 22. The gains nevertheless differ across evaluation levels. For GPT-5.4 on HealthAdminBench, Subtask increases from 0.543 to 0.633 while $\text{Pass}@3$ decreases from 0.267 to 0.105; on PhysicianBench, $\text{Pass}@3$ increases from 0.330 to 0.360 while Subtask decreases from 0.777 to 0.732. This divergence shows that terminal, intermediate, and process performance capture complementary effects of MedHarness.

Table 3: Native and MedHarness performance across eight agents and three healthcare environments. $\text{Pass}@3$ measures terminal task success, Subtask measures intermediate task completion, and F/R/E/W/A/V/B denote the seven healthcare-specific process dimensions. Values in parentheses indicate changes relative to Native. Bold denotes the best score within each environment and runtime configuration.

Model	Native									Medical Harness																	
	Pass@3	Subtask	F	R	E	W	A	V	B	Pass@3	Subtask	F	R	E	W	A	V	B									
HealthAdminBench (n = 135)																											
Claude Sonnet 5	0.341	0.794	0.904	0.944	0.602	0.941	0.910	0.351	0.568	0.363	+0.022	0.805	+0.011	0.917	+0.013	0.946	+0.002	0.622	+0.020	0.942	+0.001	0.912	+0.002	0.649	+0.298	0.615	+0.047
GPT-5.4	0.267	0.543	0.648	0.696	0.522	0.824	0.916	0.283	0.774	0.105	-0.162	0.633	+0.090	0.868	+0.220	0.913	+0.217	0.604	+0.082	0.902	+0.078	0.906	-0.010	0.589	+0.306	0.670	-0.104
Qwen3.7-Max	0.252	0.762	0.907	0.933	0.632	0.898	0.920	0.375	0.709	0.481	+0.229	0.786	+0.024	0.944	+0.037	0.934	+0.001	0.637	+0.005	0.896	-0.002	0.906	-0.014	0.677	+0.302	0.739	+0.030
GPT-5.5	0.244	0.760	0.922	0.949	0.727	0.962	0.909	0.389	0.741	0.267	+0.023	0.792	+0.032	0.937	+0.015	0.953	+0.014	0.821	+0.094	0.964	+0.002	0.912	+0.003	0.791	+0.312	0.756	+0.015
Kimi K2.7 Code	0.244	0.647	0.900	0.947	0.643	0.935	0.919	0.367	0.715	0.274	+0.030	0.726	+0.079	0.931	+0.031	0.954	+0.007	0.645	+0.002	0.938	+0.003	0.915	-0.004	0.677	+0.310	0.634	-0.081
MiniMax-M3	0.222	0.685	0.911	0.928	0.658	0.943	0.870	0.380	0.681	0.289	+0.067	0.697	+0.012	0.919	+0.008	0.918	-0.010	0.693	+0.035	0.947	+0.004	0.884	+0.014	0.628	+0.248	0.630	-0.051
DeepSeek-V4-Flash	0.193	0.570	0.911	0.950	0.609	0.950	0.907	0.341	0.693	0.178	-0.015	0.577	+0.007	0.912	+0.001	0.887	-0.063	0.628	+0.019	0.881	-0.069	0.936	+0.029	0.572	+0.231	0.707	+0.014
Gemini 3 Flash	0.156	0.700	0.893	0.940	0.673	0.935	0.908	0.371	0.681	0.267	+0.111	0.715	+0.015	0.898	+0.005	0.942	+0.002	0.678	+0.005	0.944	+0.009	0.913	+0.005	0.625	+0.254	0.701	+0.020
PhysicianBench (n = 100)																											
GPT-5.5	0.380	0.801	0.872	0.942	0.776	0.910	0.973	0.464	0.511	0.450	+0.070	0.818	+0.017	0.877	+0.005	0.937	-0.005	0.784	+0.008	0.912	+0.002	0.966	-0.007	0.496	+0.032	0.512	+0.001
GPT-5.4	0.330	0.777	0.860	0.940	0.774	0.912	0.975	0.457	0.621	0.360	+0.030	0.732	-0.045	0.888	-0.028	0.918	-0.022	0.788	+0.014	0.919	+0.007	0.980	+0.005	0.469	+0.012	0.680	+0.059
Qwen3.7-Max	0.200	0.679	0.842	0.932	0.786	0.894	0.958	0.423	0.436	0.190	-0.010	0.682	+0.003	0.908	+0.066	0.945	+0.013	0.790	+0.004	0.910	+0.016	0.957	-0.001	0.522	+0.009	0.399	-0.037
DeepSeek-V4-Flash	0.190	0.621	0.851	0.913	0.708	0.840	0.878	0.364	0.318	0.220	-0.030	0.689	+0.068	0.909	+0.058	0.933	+0.020	0.772	+0.064	0.852	+0.012	0.882	+0.004	0.462	+0.098	0.486	+0.168
MiniMax-M3	0.170	0.636	0.722	0.783	0.668	0.807	0.921	0.352	0.392	0.190	-0.020	0.688	+0.052	0.851	+0.129	0.932	+0.149	0.762	+0.094	0.825	+0.018	0.939	+0.018	0.449	+0.097	0.583	+0.191
Claude Sonnet 5	0.170	0.781	0.775	0.778	0.823	0.923	0.923	0.400	0.389	0.220	+0.050	0.787	+0.006	0.798	-0.023	0.909	+0.012	0.762	+0.004	0.851	+0.008	0.927	+0.004	0.494	+0.094	0.441	+0.052
Gemini 3 Flash	0.070	0.557	0.870	0.945	0.672	0.913	0.959	0.342	0.395	0.090	+0.020	0.552	-0.005	0.883	+0.013	0.958	+0.013	0.696	+0.024	0.917	+0.004	0.959	(0.000)	0.491	+0.149	0.438	+0.043
Kimi K2.7 Code	0.020	0.319	0.343	0.504	0.278	0.668	0.961	0.143	0.491	0.150	-0.130	0.594	+0.275	0.756	+0.413	0.868	+0.364	0.704	+0.426	0.786	+0.118	0.867	-0.094	0.406	+0.263	0.498	+0.007
MedCTA (n = 107)																											
Qwen3.7-Max	0.729	0.555	0.693	0.775	0.670	0.863	0.855	0.668	0.386	0.720	-0.009	0.581	+0.026	0.681	-0.012	0.836	+0.061	0.679	+0.009	0.873	+0.010	0.868	+0.013	0.676	+0.008	0.579	+0.193
GPT-5.5	0.710	0.534	0.665	0.762	0.660	0.839	0.874	0.657	0.488	0.687	-0.023	0.556	+0.022	0.673	-0.008	0.769	+0.007	0.675	+0.015	0.838	-0.001	0.884	+0.010	0.604	-0.053	0.709	+0.221
Claude Sonnet 5	0.701	0.475	0.650	0.728	0.592	0.840	0.854	0.593	0.448	0.758	+0.057	0.481	+0.006	0.696	+0.046	0.764	+0.036	0.671	+0.079	0.801	-0.039	0.874	+0.020	0.683	+0.000	0.555	+0.107
GPT-5.4	0.682	0.516	0.706	0.792	0.709	0.865	0.870	0.705	0.436	0.745	+0.063	0.574	+0.058	0.709	+0.013	0.799	+0.007	0.720	+0.011	0.872	+0.007	0.864	-0.006	0.710	-0.005	0.638	+0.202
DeepSeek-V4-Flash	0.645	0.536	0.591	0.705	0.531	0.795	0.861	0.537	0.561	0.664	+0.019	0.561	+0.025	0.618	+0.027	0.740	+0.035	0.607	+0.076	0.817	+0.022	0.879	+0.018	0.623	+0.086	0.614	+0.053
Kimi K2.7 Code	0.645	0.569	0.612	0.735	0.597	0.812	0.869	0.598	0.642	0.686	+0.041	0.570	+0.001	0.614	+0.002	0.743	+0.008	0.602	+0.005	0.815	+0.003	0.875	+0.006	0.620	+0.022	0.688	+0.046
MiniMax-M3	0.570	0.523	0.606	0.712	0.573	0.804	0.859	0.588	0.494	0.664	+0.094	0.542	+0.019	0.580	-0.026	0.748	+0.036	0.603	+0.030	0.790	-0.014	0.884	+0.025	0.589	+0.001	0.709	+0.215
Gemini 3 Flash	0.561	0.530	0.496	0.725	0.525	0.742	0.904	0.531	0.721	0.738	+0.177	0.548	+0.018	0.587	+0.091	0.815	+0.090	0.640	+0.115	0.810	+0.068	0.885	-0.019	0.565	+0.034	0.627	-0.094

5.2 Comparison with General-Purpose Harnesses

The second experiment tests whether healthcare-specific harness design provides benefits beyond general-purpose runtime support. Using GPT-5.5 as the target agent, we compare MedHarness with Claude Code, NeMo Guardrails, and OpenAI Agents SDK on 60 tasks, obtained by stratifying the native task categories of each benchmark and sampling 20 tasks per environment. Table 4 shows that MedHarness improves Pass@3 and Subtask over Native on all three benchmarks and achieves the highest average process score in each environment. The general-purpose harnesses improve only three of nine Pass@3 comparisons and two of nine Subtask comparisons, indicating that isolated process gains do not consistently translate into healthcare task completion.

Table 4: Comparison of MedHarness with general-purpose harnesses using GPT-5.5 as the target agent. For each healthcare environment, 20 tasks are selected by stratified sampling over native task categories. Pass@3 measures terminal task success, Subtask measures intermediate task completion, and F/R/E/W/A/V/B denote the seven healthcare-specific process dimensions. Avg. is the mean of the seven process scores. Bold denotes the best score within each environment.

Method	Pass@3	Subtask	F	R	E	W	A	V	B	Avg.
PhysicianBench ($n = 20$)										
Native	0.310	0.677	0.833	0.934	0.740	0.884	0.966	0.337	0.529	0.746
MedHarness	0.720	0.963	0.908	0.946	0.800	0.921	0.968	0.390	0.638	0.796
Claude Code	0.280	0.322	0.558	0.769	0.650	0.804	0.961	0.317	0.748	0.687
NeMo Guardrails	0.430	0.185	0.442	0.744	0.630	0.709	0.931	0.300	0.435	0.599
OpenAI Agents SDK	0.250	0.192	0.392	0.731	0.550	0.670	0.938	0.296	0.513	0.584
MedCTA ($n = 20$)										
Native	0.669	0.500	0.692	0.760	0.654	0.845	0.864	0.693	0.297	0.686
MedHarness	0.846	0.679	0.729	0.773	0.654	0.814	0.866	0.701	0.763	0.757
Claude Code	0.167	0.438	0.154	0.446	0.269	0.577	0.864	0.282	0.749	0.477
NeMo Guardrails	0.692	0.564	0.538	0.703	0.577	0.753	0.885	0.603	0.806	0.695
OpenAI Agents SDK	0.538	0.538	0.577	0.738	0.577	0.782	0.894	0.680	0.819	0.724
HealthAdminBench ($n = 20$)										
Native	0.500	0.566	0.629	0.988	0.684	0.875	0.917	0.414	0.603	0.730
MedHarness	0.680	0.731	0.752	0.993	0.590	0.794	0.933	0.660	0.642	0.766
Claude Code	0.550	0.398	0.175	0.181	0.123	0.395	0.978	0.096	0.614	0.366
NeMo Guardrails	0.200	0.272	0.600	0.734	0.436	0.542	0.969	0.279	0.572	0.590
OpenAI Agents SDK	0.358	0.416	0.471	0.025	0.381	0.125	0.972	0.429	0.656	0.437

5.3 Process Failure Sensitivity and Discrimination

The third experiment evaluates whether the seven process dimensions are sensitive to their intended failures and sufficiently discriminative for fault localization. We introduce one controlled failure into each applicable clean trajectory while preserving the terminal answer and unrelated evidence. Table 5 summarizes the seven intervention families.

For each clean-perturbed pair, we compute the decrease in each applicable process score. Because process dimensions have trajectory-level applicability as defined in Section 3, these decreases need not share identical support across dimensions: a responding dimension contributes to an aggregate only when both the clean and perturbed trajectories provide sufficient evidence for that dimension. Figure 2(a) therefore reports, for each injected fault and responding dimension, the benchmark-balanced mean decrease over that responding dimension’s applicable support.

Selectivity is defined at the pair level rather than from the aggregated response matrix. For a clean-perturbed pair with targeted dimension d^* , the selectivity margin is

$$m = \Delta S_{d^*} - \max_{d \neq d^*, d \in \mathcal{C}} \Delta S_d,$$

where $\mathcal{C}$ denotes the non-target dimensions that are jointly comparable for that pair. These pair-level

Table 5: Single-fault interventions used to evaluate the seven process dimensions. F/R/E/W denote the four Efficiency dimensions, while A/V/B denote the three Safety dimensions. Each intervention is adapted to the benchmark environment while preserving the same failure semantics.

Dim.	Injected failure	Example transformation
F	Incomplete care action	Remove evidence that an applicable care action or required artifact was completed while preserving the surrounding execution trace.
R	Resource-use failure	Corrupt the selected clinical resource, authorization state, parameter semantics, or use of the returned result while leaving the intended task unchanged.
E	Evidence misalignment	Remove required evidence or alter its subject binding such that the evidence no longer supports the current patient, state, or decision.
W	Workflow discontinuity	Remove evidence required to maintain valid progression, recovery, readiness, or termination across successive workflow states.
A	Missing audit evidence	Remove state, failure, recovery, provenance, or reliability evidence required to reconstruct the execution and its consequential decisions.
V	Missing closed-loop verification	Remove an applicable precondition confirmation or post-action verification while preserving the corresponding action itself.
B	Conduct-boundary violation	Insert an unauthorized, out-of-scope, incorrectly sourced, or otherwise boundary-violating action while preserving the surrounding workflow.

margins are then aggregated across benchmarks. Consequently, the selectivity values in Table 6 are not obtained by subtracting the largest off-diagonal mean in Figure 2(a) from the diagonal mean, since the corresponding quantities can be computed over different effective supports.

Sensitivity is determined by the response of the targeted dimension, while discrimination is evaluated by predicting the injected failure from the dimension with the largest observed decrease. Figure 2 and Table 6 show that the targeted dimension produces the largest mean decrease for all seven fault classes, with every target-drop and selectivity-margin confidence interval excluding zero. Fault localization reaches a benchmark-balanced accuracy of 0.756 and a macro-F1 of 0.706, significantly above permuted labels ($p = 0.001$). Patient–Evidence Alignment and Clinical Conduct Boundaries are the most confusable dimensions, with recalls of 0.462 and 0.514, respectively.

Table 6: Sensitivity and selectivity under controlled single-fault perturbations. Target drop denotes the benchmark-balanced decrease in the targeted process dimension. Selectivity margin is computed for each clean–perturbed pair as the targeted-dimension drop minus the largest drop among jointly comparable non-target dimensions, and is then aggregated across benchmarks. Brackets report 95% bootstrap confidence intervals.

Dimension	Target drop (95% CI)	Selectivity margin (95% CI)
Care Action Fulfillment (F)	0.373 [0.302, 0.440]	0.326 [0.258, 0.387]
Clinical Resource Competence (R)	0.087 [0.071, 0.104]	0.086 [0.066, 0.105]
Patient–Evidence Alignment (E)	0.109 [0.067, 0.155]	0.087 [0.042, 0.143]
Clinical Workflow Continuity (W)	0.160 [0.132, 0.186]	0.168 [0.134, 0.201]
Clinical Auditability (A)	0.302 [0.262, 0.340]	0.285 [0.243, 0.324]
Closed-Loop Verification (V)	0.095 [0.090, 0.101]	0.092 [0.084, 0.100]
Clinical Conduct Boundaries (B)	0.312 [0.195, 0.435]	0.328 [0.215, 0.448]

6 Discussion

The results demonstrate that the Medical Harness provides an effective healthcare-specific runtime layer for coordinating evidence, tools, workflow state, verification, and authorization around medical agents. MedHarness improves Outcome in 19 of 24 agent–environment configurations and Subtask performance in 22 of 24, although its effects vary across agents, environments, and process dimensions. This variation reflects differences in how agents use tools, maintain state, respond to runtime support, and propagate interventions into subsequent actions, together with the distinct demands of FHIR/EHR operations, clinical GUI workflows, and multimodal tool use. The Medical Harness Protocol further reveals where these

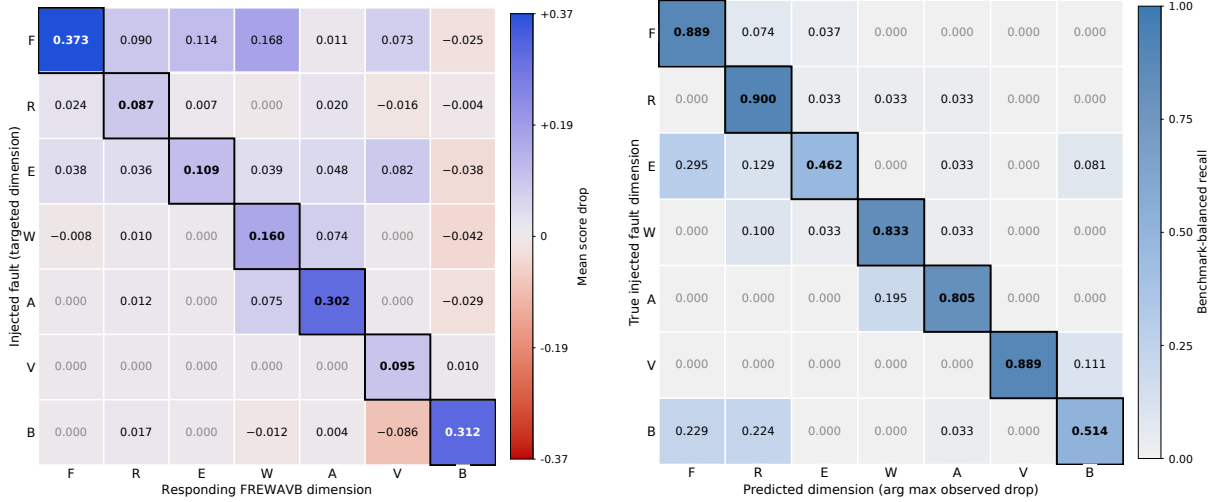


Figure 2: Process-failure sensitivity and discrimination under controlled single-fault perturbations. **(a)** Benchmark-balanced mean score decrease for each injected failure and responding dimension, computed over the applicable support of the responding dimension. The targeted dimension is the largest response in all seven rows. **(b)** Benchmark-balanced fault-localization confusion matrix obtained by predicting the injected failure from the dimension with the largest observed score decrease. Rows are normalized by injected failure class.

effects arise by evaluating Care Action Fulfillment, Clinical Resource Competence, Patient–Evidence Alignment, Clinical Workflow Continuity, Clinical Auditability, Closed-Loop Verification, and Clinical Conduct Boundaries. These dimensions expose both intermediate improvements that do not lead to terminal success and successful outcomes that conceal incomplete, unverified, or boundary-violating behavior. Nevertheless, MedHarnessBench contains 342 tasks from three controlled research environments and does not represent the full diversity of hospital workflows, tool schemas, institutional policies, or clinical contingencies. Some process dimensions also depend on automated or LLM-based evaluators, and individual harness mechanisms may influence multiple dimensions. The current findings therefore validate the Medical Harness as a measurable research framework but do not establish clinical deployment readiness, regulatory compliance, or patient-safety effectiveness in real-world care.

7 Conclusion

We presented the *Medical Harness*, a healthcare-specific runtime framework that coordinates clinical evidence, heterogeneous tools, workflow state, monitoring, recovery, verification, and authorization around medical agents. To evaluate this framework, we introduced the MEDICAL HARNESS PROTOCOL, which defines seven healthcare-specific process dimensions and traces how harness interventions are activated, delivered, adopted, and propagated through clinical workflows. We further instantiated the protocol as MEDHARNESSBENCH, comprising 342 executable tasks across FHIR/EHR operations, healthcare-administration interfaces, and multimodal clinical tools. Across eight agents, MedHarness improves both terminal task performance and intermediate execution, while comparisons with general-purpose harnesses and controlled single-fault perturbations demonstrate the value and diagnostic sensitivity of the proposed framework. These results establish the Medical Harness as a measurable runtime layer for identifying and improving execution failures that conventional terminal evaluations cannot expose.

References

- [1] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.
- [2] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.
- [3] Yixing Jiang, Kameron C. Black, Gloria Geng, Danny Park, James Zou, Andrew Y. Ng, and Jonathan H. Chen. Medagentbench: A realistic virtual ehr environment to benchmark medical llm agents. *NEJM AI*, 2, 2025.
- [4] 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:2405.07960*, 2024.
- [5] Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.
- [6] Chengzhi Liu, Yichen Guo, Yepeng Liu, Yuzhe Yang, Qianqi Yan, Xuandong Zhao, Wenye Hua, Sheng Liu, Sharon Li, Yuheng Bu, and Xin Eric Wang. Auditing agent harness safety. *arXiv preprint arXiv:2605.14271*, 2026.
- [7] Ruoyi Liu, Imran Q. Mohiuddin, Austin J. Schoeffler, Kavita Renduchintala, Ashwin Nayak, Prasantha L. Vemu, Shivam C. Vedak, Kameron C. Black, John L. Havlik, Isaac Ogunmola, Stephen P. Ma, Roopa Dhatt, and Jonathan H. Chen. Physicianbench: Evaluating llm agents in real-world ehr environments. *arXiv preprint arXiv:2605.02240*, 2026.
- [8] Suhana Bedi, Ryan Welch, Ethan Steinberg, Michael Wornow, Taeil Matthew Kim, Haroun Ahmed, Peter Sterling, Bravim Purohit, Qurat Akram, Angelic Acosta, Esther Nubla, Pritika Sharma, Michael A. Pfeffer, Sanmi Koyejo, and Nigam H. Shah. Healthadminbench: Evaluating computer-use agents on healthcare administration tasks. *arXiv preprint arXiv:2604.09937*, 2026.
- [9] Tajamul Ashraf, Hyewon Jeong, Fida Mohammad Thoker, and Bernard Ghanem. Medcta: A benchmark for clinical tool agents. *arXiv preprint arXiv:2606.11702*, 2026.
- [10] Junjie Li, Xi Xiao, Yunbei Zhang, Chen Liu, Lin Zhao, Xiaoying Liao, Yingrui Ji, Janet Wang, Yingqiang Ge, Weijie Xu, Xi Fang, Xiang Xu, Tianchen Zhao, Youngeun Kim, Jihun Hamm, Tianyang Wang, and Chandan K. Reddy. Agent harness engineering: A survey, 2026. OpenReview preprint.
- [11] Qianchu Liu, Sheng Zhang, Guanghui Qin, Jeya Maria Jose Valanarasu, Maximilian Rokuss, Mingyu Lu, Timothy Ossowski, Juan Manuel Zambrano Chaves, Cliff Wong, Peniel Argaw, Yashna Hasija, Mu Wei, Wen-wai Yim, Qin Liu, Zilin Jing, Jason Entenmann, Naoto Usuyama, Tristan Naumann, and Hoifung Poon. Healthagentbench: A unified benchmark suite of realistic agentic healthcare environments for challenging frontier ai agents. *arXiv preprint arXiv:2606.31179*, 2026.
- [12] Haolin Chen, Deon Metelski, Leon Qi, Tao Xia, Joonyul Lee, Steve Brown, Kevin Riley, Frank Wang, T. Y. Alvin Liu, Hank Capps, et al. χ -bench: Can ai agents automate end-to-end, long-horizon, policy-rich healthcare workflows? *arXiv preprint arXiv:2605.16679*, 2026.

- [13] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.
- [14] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. *arXiv preprint arXiv:2308.10144*, 2023.
- [15] John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, 2024.
- [16] Linyue Pan, Lexiao Zou, Shuo Guo, Jingchen Ni, and Hai-Tao Zheng. Natural-language agent harnesses. *arXiv preprint arXiv:2603.25723*, 2026.
- [17] Yilun Yao, Xinyu Tan, Chao-Hsuan Liu, Yaoming Li, Zhengyang Wang, Wenhan Yu, Zhewen Tan, Yuxuan Tian, Guangxiang Zhao, Lin Sun, Xiangzheng Zhang, and Tong Yang. Harness-bench: Measuring harness effects across models in realistic agent workflows. *arXiv preprint arXiv:2605.27922*, 2026.
- [18] Yunbei Zhang, Janet Wang, Yingqiang Ge, Weijie Xu, Jihun Hamm, and Chandan K. Reddy. Stop comparing llm agents without disclosing the harness. *arXiv preprint arXiv:2605.23950*, 2026.
- [19] OpenAI. Gpt-5.5 system card. <https://openai.com/index/gpt-5-5-system-card/>, 2026. Accessed: 2026-07-30.
- [20] OpenAI. Gpt-5.4 thinking system card. <https://openai.com/index/gpt-5-4-thinking-system-card/>, 2026. Accessed: 2026-07-30.
- [21] Qwen Team. Qwen3.7: The agent frontier. <https://qwen.ai/blog?id=qwen3.7>, 2026. Accessed: 2026-07-30.
- [22] Anthropic. Claude sonnet 5 system card. <https://www.anthropic.com/claude-sonnet-5-system-card>, 2026. Accessed: 2026-07-30.
- [23] Moonshot AI. Kimi k2.7 code model card. <https://huggingface.co/moonshotai/Kimi-K2.7-Code>, 2026. Accessed: 2026-07-30.
- [24] DeepSeek-AI. Deepseek v4 technical documentation. <https://fe-static.deepseek.com/chat/transparency/deepseek-V4-model-card-EN.pdf>, 2026. Accessed: 2026-07-30.
- [25] MiniMax. Minimax m3: Frontier coding, 1m context, native multimodality — all in one model. <https://www.minimax.io/blog/minimax-m3>, 2026. Accessed: 2026-07-30.
- [26] Google DeepMind. Gemini 3 flash model card. <https://deepmind.google/models/model-cards/>, 2025. Updated December 17, 2025.