CLINLENS: TOWARDS LONG-HORIZON LLM AGENTS FOR LONGITUDINAL MULTIMODAL CLINICAL DATA SCIENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Automating clinical data science requires agents to translate patient-centric objectives into executable workflows grounded in longitudinal multimodal evidence. Existing benchmarks provide limited coverage of two essential capabilities: grounding analyses in the correct patient episode and harmonizing multimodal observations. We introduce **ClinLens**, a patient-centric benchmark for evaluating large language model (LLM) agents on long-horizon clinical data science workflows. **ClinLens** comprises 126 executable tasks spanning five data modalities, four analytical scopes, and five analytical capabilities. The tasks assess agents' ability to maintain a coherent patient-specific analytical context across interdependent stages of evidence grounding, multimodal integration, and analytical pipeline generation and execution. We evaluate general-purpose, coding, and biomedical agents, with task runs averaging 36.93 steps and 8.06 minutes. The strongest evaluated agent achieves an execution success rate of 86.4% and a final-answer accuracy of 54.0%, but a strict pass rate of only 29.4%. Failure analysis identifies patient-episode grounding as the most frequently recorded failure category under the evaluation procedure. By jointly evaluating analytical outcomes and the validity of the supporting clinical context, **ClinLens** provides a testbed for advancing reliable agents for longitudinal clinical data science.

1 INTRODUCTION

Clinical data science often requires computational analysis for individual patients to characterize disease progression and treatment response. Such analysis draws on longitudinal clinical data collected across admissions and care settings, e.g., structured records, physiological time series, and medical images (Johnson et al., 2023c). To conduct analysis, researchers typically construct a long-horizon data science workflow, including identifying relevant clinical episodes, integrating multimodal data, and implementing statistical or predictive methods. Recent advances in large language models (LLM), particularly in reasoning (Guo et al., 2025) and code generation (Novikov et al., 2025), offer new opportunities for agents to automate these workflows (Jing et al., 2025; Zhang et al., 2026). However, reliably translating patient-specific analytical objectives into executable workflows grounded in longitudinal clinical evidence remains a central challenge.

Evaluating this capability requires benchmarks centered on longitudinal, multimodal patient trajectories. Existing benchmarks mainly focus on evaluating agents on automating dataset-centric analytical workflows (Xu et al., 2026a; Wang et al., 2026; Xu et al., 2026b), but provide limited coverage of patient-centric evidence integration across clinical episodes and modalities. Specifically, two key capabilities remain insufficiently evaluated: (1) *patient-episode grounding*, which links evidence to the correct patient and clinical episode across admissions, ICU stays, and repeated studies; and (2) *asynchronous multimodal integration*, which aligns multimodal observations collected at different times and sampling rates within task-specific temporal windows.

To address this gap, we introduce **ClinLens**, a patient-centric benchmark for evaluating LLM agents on longitudinal clinical data science. Our contributions are threefold:

- **Problem framing**. We formulate longitudinal clinical analysis as an end-to-end workflow connecting patient-episode grounding and asynchronous multimodal integration with analytical pipeline

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

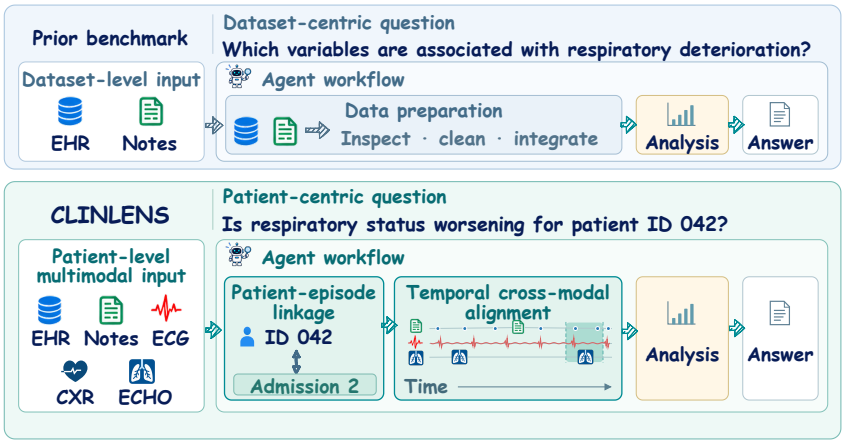


Figure 1: **Prior benchmarks vs. `ClinLens`.** Left: Existing benchmarks mainly focus on dataset-centric tasks with limited modality coverage. Right: **`ClinLens`** evaluates patient-centric clinical analysis across five modalities, requiring agents to link evidence to the correct patient and episode, align asynchronous multimodal observations before generating and executing analytical workflows.

generation, execution, and verification. This formulation makes constructing a valid patient-centric analytical context an explicit task requirement.

- **Benchmark**. We instantiate this formulation in 126 executable tasks spanning five data modalities, i.e., structured electronic health records (EHR), clinical notes, electrocardiogram (ECG), chest X-ray (CXR), and echocardiogram (ECHO). A 4×5 task taxonomy crosses four patient-episode levels (patient, admission, ICU stay, and study) with five clinical analysis capabilities, i.e., course profiling, correlation analysis, treatment effect estimation, risk prediction, and phenotype discovery. In practice, each task runs averaging 36.93 steps and 8.06 minutes.
- **Empirical findings**. We evaluate a broad range of general-purpose, coding, and biomedical agents, revealing a substantial gap between execution, answer correctness, and workflow validity. The strongest evaluated agent achieves 86.4% execution success and 54.0% final-answer accuracy, but only a 29.4% strict pass rate. Failure analysis further identifies patient-episode grounding as the most frequently observed failure under the evaluation procedure.

2 Benchmark Construction

`ClinLens` evaluates whether agents can reconstruct patient-episode relationships, align asynchronous multimodal evidence, and produce an auditable clinical analysis. Its construction connects clinical questions, executable reference workflows, and artifact-level evaluation (Figure 2).

2.1 Task Definition and Evaluation

Task instructions. Each task asks an agent to answer a clinical data-analysis question for a named target patient within a specified longitudinal setting. The instruction defines the analysis population, temporal constraints, relevant modalities, required computation, and output contract. Depending on the question, the analysis may concern the target patient's course or involve estimation and modeling using a reference cohort. The agent must implement the analysis and submit a final answer together with required artifacts, such as cohort tables, temporal alignments, statistics, or model outputs. These artifacts allow the evaluator to inspect how the answer was obtained.

Longitudinal multimodal patient packages. Each task provides a bounded package containing the target patient and a reference cohort, with structured EHR records, notes, ECGs, CXRs, and ECHOs as relevant to the task. The package preserves native patient and episode identifiers, timestamps, repeated observations, and modality missingness rather than providing a pre-aligned feature table. Agents must link records to the correct patient and clinical episode, align asynchronously recorded modalities within the specified windows, and perform the necessary preprocessing and analysis. Thus,

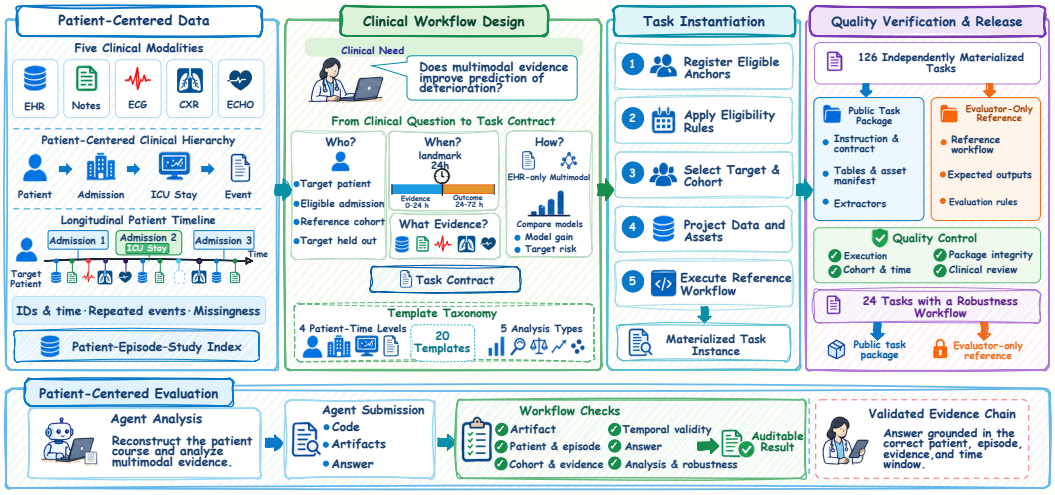


Figure 2: **Construction and evaluation pipeline of `ClinLens`.** Five clinical data sources are organized into patient-centric packages. Workflow templates and executable references define the task instances, while submitted artifacts and final answers support deterministic evaluation.

patient-centric grounding establishes whose evidence is used and which episode it belongs to, while temporal alignment establishes how observations from different modalities can be combined.

Evaluation. Each task is paired with an evaluator-private executable reference that produces the expected artifacts and answer under fixed analytical and temporal specifications. Evaluation first checks artifact completeness and parseability, then validates patient, cohort, temporal, and analytical semantics, and finally compares the submitted answer with the reference under task-specific matching rules. Final-answer correctness and full workflow validity are assessed separately: a matching answer alone does not establish that the underlying analysis satisfies the task contract. Operational details and metric definitions appear in Appendices A and B.

2.2 Construction Pipeline and Coverage

Patient-centric data curation. We integrate MIMIC-IV structured records (Johnson et al., 2024a; 2023b), MIMIC-IV-Note (Johnson et al., 2023a), MIMIC-IV-ECG (Gow et al., 2023), MIMIC-CXR-JPG (Johnson et al., 2024b), and MIMIC-IV-ECHO (Gow et al., 2026). Shared patient identifiers assemble bounded packages retaining native admission, ICU-stay, study, and event identifiers. Source timestamps, repeated examinations, one-to-many relations, and modality missingness remain visible, as do modality-specific representations: structured tables, narrative notes, physiological waveforms, radiographs, and echocardiographic studies.

We standardize schemas and file paths and register accessible raw assets, but do not precompute task-specific episode assignments, cross-modal joins, temporal filtering, repeated-measure aggregation, or analytical features. In particular, linking a study without an encounter identifier to the appropriate episode remains an agent responsibility, as does handling missing observations. This boundary standardizes data access while retaining the patient-grounding and temporal-alignment work being evaluated.

Clinical design and task instantiation. Construction begins with a clinical question. A workflow template fixes the eligible population, clinical anchor, evidence and outcome windows, relevant modalities, analytical objective, and expected outputs. For example, T09 compares EHR-only and multimodal models for subsequent deterioration using evidence available within the first 24 hours of an admission. Answering it requires both correct admission linkage and temporally eligible evidence before modeling. Fixed selection and sampling rules bind each template to eligible targets and independently materialized packages, yielding 126 tasks across the 20 templates. For each instance, an executable reference implements the question under prespecified preprocessing, analysis, and

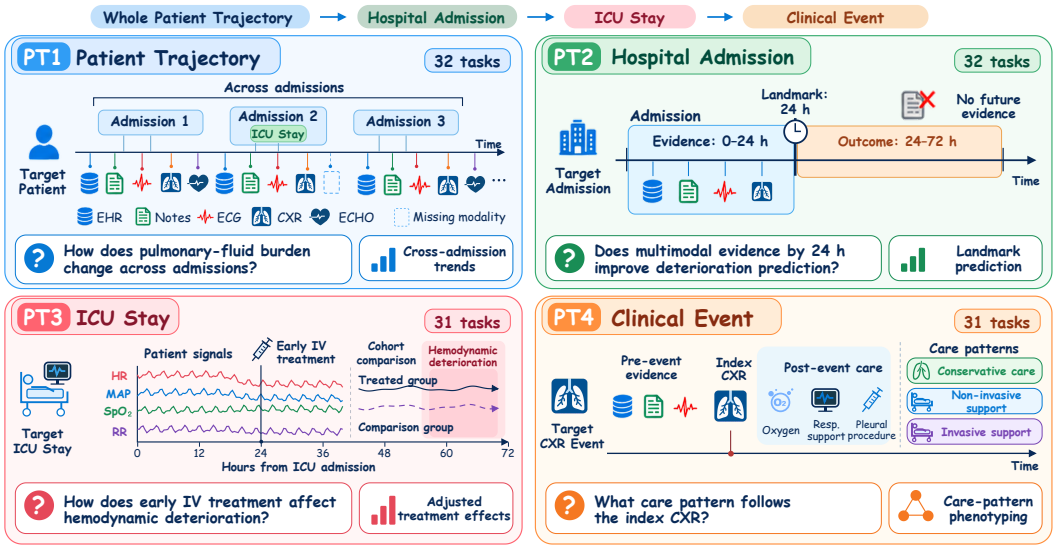


Figure 3: **Representative tasks across four patient-time scopes.** PT1 follows changes across admissions; PT2 uses admission-landmark evidence for prediction; PT3 estimates effects within an ICU stay; and PT4 analyzes patterns around a clinical event or study. Examples illustrate individual tasks, not all 20 templates. Timelines and physiological traces are schematic.

fallback rules. Its execution checks feasibility and generates the expected artifacts and answer; the instruction, artifact requirements, and evaluation rules are derived from this validated protocol.

Quality control and reference isolation. Template-level clinical and statistical review examines cohort definitions, temporal directionality, modality use, variable definitions, and analytical procedures. Each materialized instance is rerun in a clean environment to check package boundaries, raw-asset accessibility, identifier and timestamp consistency, artifact schemas, row-level and temporal semantics, and reference-artifact–answer agreement. Only instances whose packages, reference workflows, and expected submissions pass the complete evaluator are retained. Public task packages are separated from evaluator-private executable references and expected artifacts and answers. This separation permits reproducible grading without exposing the reference solution to agents. Appendix A provides the operational details.

2.3 Benchmark Statistics

Code-grounded annotation of the reference workflows identifies a mean of 36.93 analytical operations per task (range: 17–63). These are logical analysis steps, not model interaction turns: they cover patient identification, episode construction, multimodal temporal alignment, statistical analysis, and validation.
Figure 4 reports task-level mean solving times, obtained by averaging available model–agent runtimes within each task. The overall mean is 8.06 minutes; capability means range from 5.34 minutes for course profiling to 10.02 minutes for association analysis. Runtimes include unsuccessful attempts and depend on the evaluated systems. They describe observed execution cost, whereas operation counts characterize the structure of the reference workflows.

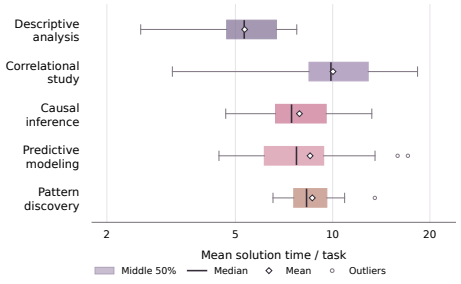


Figure 4: **Task-level mean solving time by clinical capability.**

Workflow taxonomy. Four patient-time scopes define the analysis unit: whole-patient trajectories, hospital admissions, ICU stays, and clinical event- or study-centered windows, containing 32, 32, 31, and 31 tasks, respectively. Figure 3 illustrates a representative task at each scope. Five clinical capabilities define the analytical objective: *clinical course profiling* (descriptive analysis), *clinical association analysis* (correlational analysis), *treatment effect estimation* (causal inference), *clinical risk prediction* (predictive modeling), and *clinical phenotype discovery* (pattern discovery). Crossing the two axes produces 20 workflow templates. The suite contains 26 profiling tasks and 25 tasks in each other capability (Figure 5). Each task uses a clinically relevant subset of the five modalities; the complete template index and modality coverage remain in Appendix H and Table 2.

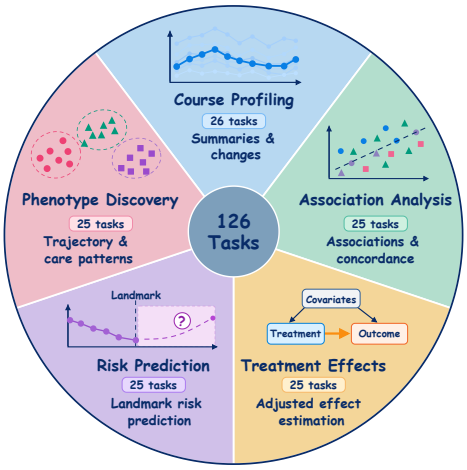


Figure 5: **Clinical capability taxonomy and task counts.**

3 Experiments

We organize the evaluation around four research questions:

- **RQ1 (Capability):** How does performance vary across capabilities and patient-time levels?
- **RQ2 (Failure localization):** Where do workflows first violate the clinical analysis contract?
- **RQ3 (Answer validity):** How often is a correct answer supported by a fully valid workflow?
- **RQ4 (Efficiency):** How do attempts and tokens relate to answer and workflow validity?

3.1 Experimental Setup

We evaluate a fixed suite of 126 clinical data-analysis tasks. Its selection emphasizes historically difficult workflows, so scores characterize this stress suite rather than an IID sample of clinical analyses. Standard configurations use zero-shot prompting, temperature 0, and a one-hour execution limit. Access is restricted to assigned packages and listed raw assets; private references and previous submissions are withheld. Frameworks, iteration limits, and metric thresholds are detailed in Appendix B.

The system comparison includes standardized scaffolds, coding harnesses, and biomedical-agent adapters (Table 1). DeepSeek-V4-Pro and GLM-5.2 retain evaluator-compatible historical results; Qwen-3.8-Max counts unevaluated tasks as failures. The diagnostic analyses use ten complete data-grounded configurations: GPT-5.6-Luna and Gemini-3.7-Flash, each paired with ReAct, DataVoyager, Reflexion, SelfDebug, and RAG-ReAct. NoDataGuess, coding harnesses, biomedical-agent adapters, and configurations with historical results or incomplete coverage are excluded. These analyses therefore characterize the selected standard data-grounded configurations rather than the full system comparison.

We report FinalAcc for final-result agreement, StrictPass for agreement plus all required artifact, schema, row-semantic, and validation gates, and ExecSuccess for completed execution with an accepted submission. The 24 robustness tasks require both base and additional analyses. *Artifact-ready* checks completeness and schema from evaluator reports and differs from ExecSuccess. System and efficiency tables macro-average patient-time levels; diagnostics pool 1,260 records for RQ1/RQ3 and 1,010 non-prediction records for RQ2 (Table 3). An exploratory analysis of semantic burden on 600 balanced-core records is reported in Appendix C.3.

Table 1: **Agent performance on `ClinLens` (%).** A: FINALACC; S: STRICTPASS; E: EXECSUCCESS. Overall is the macro-average across four patient-time scopes. Bold marks each group's highest displayed overall A and S, including ties.

	Overall			Whole patient			Admission			ICU stay			Event / study		
Agent / harness	A	S	E	A	S	E	A	S	E	A	S	E	A	S	E
GPT-5.6-Sol															Coding harness
Codex	**54.0**	**29.4**	86.4	40.6	18.8	90.6	62.5	40.6	100.0	61.3	19.4	96.8	51.6	38.7	58.1
DeepSeek-V4-Pro															Historical results†
NoDataGuess	3.1	—	—	0.0	—	—	12.5	—	—	0.0	—	—	0.0	—	—
ReAct	1.6	0.0	48.5	0.0	0.0	43.8	3.1	0.0	46.9	0.0	0.0	41.9	3.2	0.0	61.3
DataVoyager	**6.3**	**1.6**	60.3	6.2	3.1	81.2	9.4	0.0	43.8	0.0	0.0	45.2	9.7	3.2	71.0
Reflexion	**6.3**	0.0	85.0	6.2	0.0	87.5	15.6	0.0	78.1	0.0	0.0	77.4	3.2	0.0	96.8
SelfDebug	1.6	0.0	45.3	3.1	0.0	34.4	3.1	0.0	46.9	0.0	0.0	32.3	0.0	0.0	67.7
RAG-ReAct	3.1	0.8	44.6	6.2	3.1	40.6	6.2	0.0	31.2	0.0	0.0	38.7	0.0	0.0	67.7
GLM-5.2															Historical results†
NoDataGuess	3.9	—	—	0.0	—	—	15.6	—	—	0.0	—	—	0.0	—	—
ReAct	13.4	0.8	88.1	15.6	3.1	84.4	21.9	0.0	87.5	0.0	0.0	90.3	16.1	0.0	90.3
DataVoyager	11.8	2.4	94.4	12.5	3.1	100.0	18.8	3.1	90.6	0.0	0.0	93.5	16.1	3.2	93.5
Reflexion	**15.0**	**3.2**	98.4	12.5	3.1	96.9	28.1	3.1	100.0	0.0	0.0	96.8	19.4	6.5	100.0
SelfDebug	13.4	0.8	92.8	12.5	3.1	96.9	25.0	0.0	96.9	0.0	0.0	83.9	16.1	0.0	93.5
RAG-ReAct	10.2	0.8	88.1	12.5	3.1	90.6	18.8	0.0	87.5	0.0	0.0	83.9	9.7	0.0	90.3
GPT-5.6-Luna															Standard agents and coding harnesses
NoDataGuess	3.9	—	—	0.0	—	—	15.6	—	—	0.0	—	—	0.0	—	—
ReAct	13.4	5.5	93.7	12.5	6.2	90.6	21.9	9.4	87.5	0.0	0.0	100.0	19.4	6.5	96.8
DataVoyager	11.9	3.2	72.3	9.4	3.1	68.8	18.8	3.1	65.6	0.0	0.0	83.9	19.4	6.5	71.0
Reflexion	15.0	**7.9**	86.6	15.6	9.4	84.4	25.0	9.4	75.0	0.0	0.0	93.5	19.4	12.9	93.5
SelfDebug	15.0	**7.9**	92.1	18.8	12.5	84.4	21.9	3.1	93.8	0.0	0.0	96.8	19.4	16.1	93.5
RAG-ReAct	14.2	**7.9**	92.9	15.6	12.5	90.6	25.0	6.2	93.8	0.0	0.0	93.5	16.1	12.9	93.5
Codex	15.0	5.5	100.0	18.8	15.6	100.0	21.9	6.2	100.0	0.0	0.0	100.0	19.4	0.0	100.0
Claude Code	**21.3**	**7.9**	69.9	34.4	12.5	56.2	28.1	3.1	71.9	12.9	9.7	80.6	9.7	6.5	71.0
Gemini-3.7-Flash															Standard agents
NoDataGuess	4.7	—	—	0.0	—	—	18.8	—	—	0.0	—	—	0.0	—	—
ReAct	16.6	4.7	98.4	18.8	9.4	96.9	28.1	3.1	100.0	0.0	0.0	100.0	19.4	6.5	96.8
DataVoyager	16.6	**7.9**	100.0	18.8	15.6	100.0	28.1	9.4	100.0	0.0	0.0	100.0	19.4	6.5	100.0
Reflexion	**18.1**	7.1	99.2	18.8	12.5	96.9	34.4	9.4	100.0	0.0	0.0	100.0	19.4	6.5	100.0
SelfDebug	16.6	7.1	99.2	15.6	12.5	96.9	31.2	9.4	100.0	0.0	0.0	100.0	19.4	6.5	100.0
RAG-ReAct	15.8	6.3	100.0	15.6	15.6	100.0	28.1	3.1	100.0	0.0	0.0	100.0	19.4	6.5	100.0
Qwen-3.8-Max															Incomplete coverage‡
NoDataGuess	0.0	—	—	0.0	—	—	0.0	—	—	0.0	—	—	0.0	—	—
ReAct	**15.0**	**8.0**	61.3	12.5	6.2	100.0	28.1	6.2	100.0	0.0	0.0	22.6	19.4	19.4	22.6
DataVoyager	13.4	6.3	50.3	18.8	15.6	93.8	18.8	0.0	71.9	0.0	0.0	19.4	16.1	9.7	16.1
Reflexion	10.3	4.8	58.9	9.4	6.2	96.9	18.8	0.0	100.0	0.0	0.0	19.4	12.9	12.9	19.4
SelfDebug	12.7	6.4	59.7	9.4	6.2	100.0	21.9	3.1	100.0	0.0	0.0	19.4	19.4	16.1	19.4
RAG-ReAct	12.6	7.2	56.6	12.5	9.4	93.8	21.9	3.1	93.8	0.0	0.0	19.4	16.1	16.1	19.4
Biomedical-agent transfers															GPT-5.6-Luna backbone
BioMedAgent	10.3	3.1	52.4	15.6	12.5	59.4	12.5	0.0	40.6	0.0	0.0	45.2	12.9	0.0	64.5
BixBench	0.8	0.8	7.1	0.0	0.0	9.4	3.1	3.1	9.4	0.0	0.0	3.2	0.0	0.0	6.5
BioMaster	0.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.2
EHRAgent	**11.9**	**6.4**	87.4	15.6	12.5	87.5	12.5	0.0	78.1	0.0	0.0	96.8	19.4	12.9	87.1
Biomni	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

†DeepSeek/GLM retain evaluator-compatible historical results. ‡Qwen counts unevaluated tasks as failures. These groups are excluded from V6 diagnostics; unexecuted projections are omitted.

3.2 RQ1: CAPABILITY ACROSS CLINICAL WORKFLOWS

GPT-5.6-Sol with Codex is the strongest observed combined system, reaching 54.0% FINALACC, 29.4% STRICTPASS, and 86.4% EXECSUCCESS (68 correct answers and 37 strict passes). This comparison does not isolate a harness effect. Complete Luna/Gemini standard configurations reach at most 7.9% strict accuracy, and scaffold rankings vary by backbone and metric. Among biomedical adapters, EHRAgent leads with 11.9% answer accuracy and 6.4% strict accuracy; these are shared-interface transfer results, not rankings of unmodified native systems (Appendix C).

Figure 6 summarizes template-level performance across the ten complete configurations (1,260 task–configuration records). In panel (a), DS1–DS5 follow the clinical capability taxonomy; gray

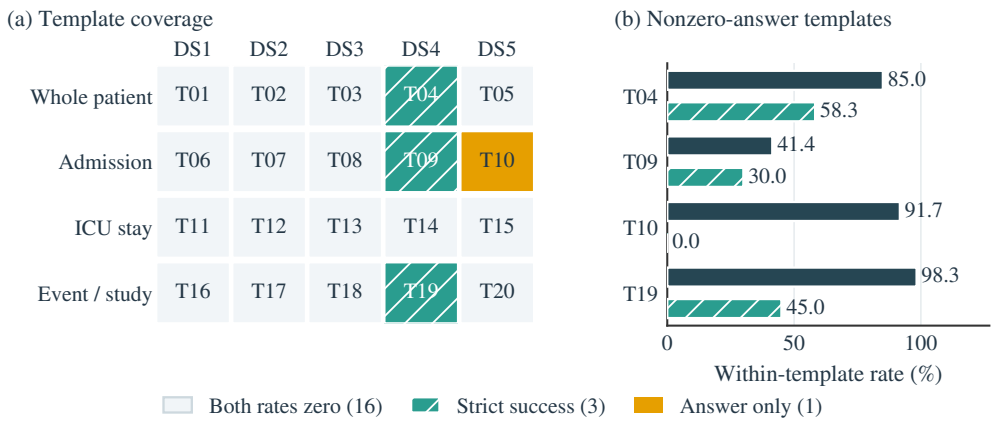


Figure 6: **Performance across patient-time scopes and clinical capabilities.** (a) Capability matrix and (b) template-level FINALACC and STRICTPASS, pooled across ten complete configurations.

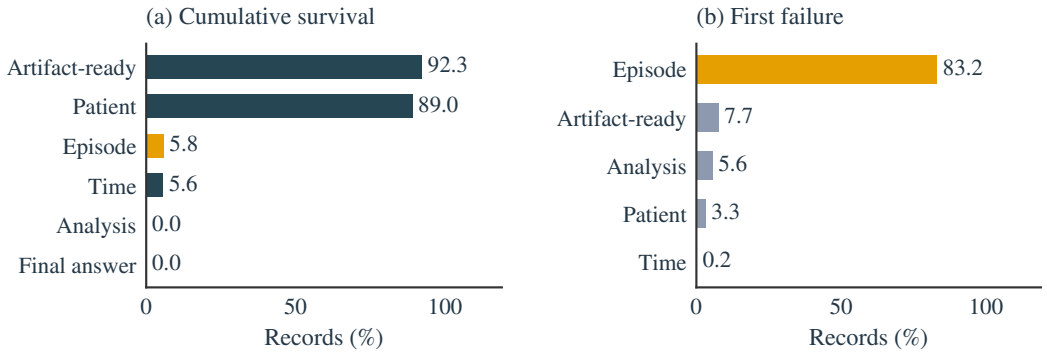


Figure 7: **Workflow validation and first-failure distribution.** (a) Cumulative gate survival and (b) mutually exclusive first failures across 1,010 non-prediction task–configuration records. Gates follow evaluator order.

cells indicate zero answer accuracy, hatching indicates nonzero strict success, and amber cells indicate answer-only success. Panel (b) plots percentage FINALACC and STRICTPASS in navy and teal.

Success is concentrated in four templates. T04, T09, T10, and T19 achieve answer/strict rates of 85.0/58.3%, 41.4/30.0%, 91.7/0.0%, and 98.3/45.0%; the other 16 templates have no correct answers. T14 has no correct predictions; T10 yields high answer accuracy without a valid workflow.

3.3 RQ2: WHERE CLINICAL WORKFLOWS FIRST FAIL

We localize failures through an ordered cumulative funnel: Artifact-ready → Patient → Episode → Time → Analysis → Final answer. It covers 1,010 non-prediction records because exported prediction reports omit the full grounding sequence. Figure 7 shows cumulative gate survival in panel (a) and mutually exclusive first failures in panel (b). The first gate uses Artifact-ready, not EXECSUCCESS. First failures follow evaluator order, not independent causes.

Episode grounding is the largest observed bottleneck. Of 1,010 records, 932 (92.3%) are artifact-ready and 899 (89.0%) pass patient grounding, but only 59 (5.8%) also pass episode grounding. Episode selection accounts for 840 first failures (83.2%), an 83.2-percentage-point drop in cumulative survival. No record passes the analysis stage, yielding zero cumulative final-answer success despite 5.4% marginal answer accuracy. Inspected errors include wrong ECHO adjacency and duplicate CXR-to-admission links (Appendix C.2). Such linkage errors change the analytical population before estimation. Few first failures at later gates imply neither correct temporal alignment nor valid analysis: earlier failures can mask additional downstream defects in the ordered first-failure summary.

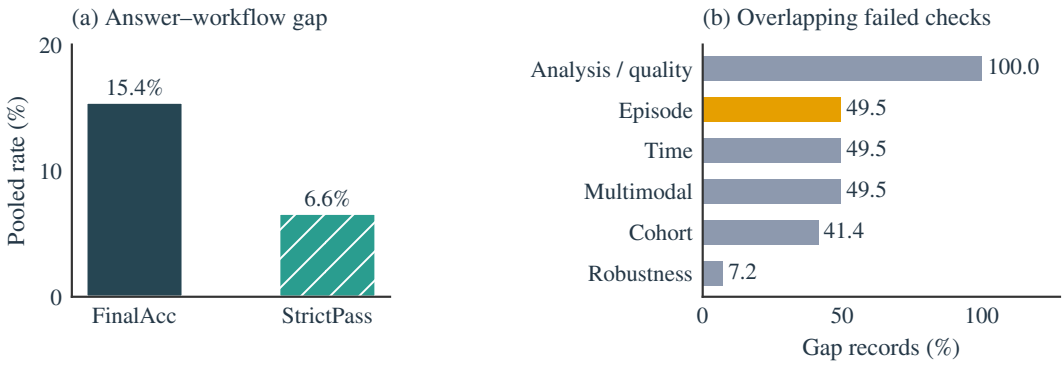


Figure 8: **Final-answer correctness and workflow validity.** (a) FINALACC versus STRICTPASS across 1,260 task–configuration records. (b) Overlapping failed checks among 111 answer-correct but workflow-invalid submissions.

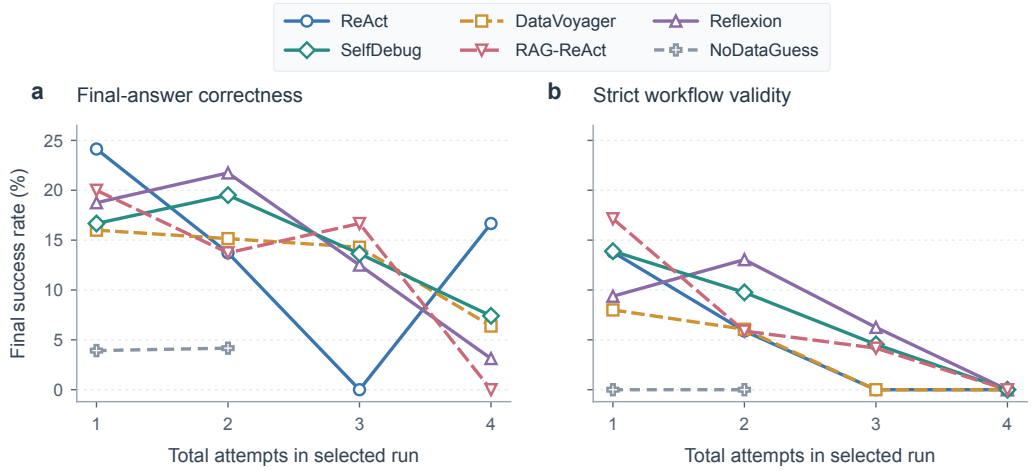


Figure 9: **Performance by observed attempt count.** GPT-5.6-Luna FINALACC (left) and STRICTPASS (right), grouped by scaffold and total attempts. Lines connect distinct task groups, not successive attempts on the same tasks.

3.4 RQ3: CORRECT ANSWERS VERSUS VALID WORKFLOWS

Only 83 of 194 correct answers (42.8%) satisfy the full workflow contract. Across 1,260 records, pooled answer accuracy is 15.4% and strict success is 6.6%, leaving 111 answer-correct but invalid submissions (Figure 8, panel (a)). Their mutually exclusive first failures divide into 55 episode-grounding failures, 51 prediction-quality failures, and 5 prediction-protocol failures.

Panel (b) reports overlapping failed checks among these 111 submissions: submissions can fail multiple checks, so percentages cannot be summed or read as independent defects. A target answer can match despite an incorrect event population or inadequate supporting model; the checks do not distinguish chance agreement from an answer insensitive to the upstream error (Appendix C.4).

3.5 RQ4: ATTEMPT COUNTS AND PERFORMANCE–EFFICIENCY TRADE-OFFS

We group GPT-5.6-Luna runs by their observed total attempts, using 126 tasks for each of six standard scaffolds (756 records). Each selected task–scaffold run enters one group. Figure 9 shows pooled final-answer correctness (left) and strict workflow validity (right) within each scaffold–attempt group. Lines connect distinct task groups, not successive scores for the same tasks; missing groups are left unplotted. Claude Code lacks comparable generation-and-repair attempt counts and is excluded.

More observed attempts do not consistently imply higher final success. Reflexion and SelfDebug have non-monotonic answer rates, and none of the five data-grounded scaffolds has a strict success in its four-attempt group. Task difficulty and stopping behavior can shape group composition. Without intermediate scores, the curves cannot establish whether revision improves or harms the same task.

Recorded computation likewise does not consistently accompany greater validity. SelfDebug reaches 14.99% answer and 7.94% strict accuracy at 46.02k tokens per task, close to ReAct's 45.68k. DataVoyager records 108.03k tokens with lower accuracy than ReAct. Claude Code reaches 21.27% answer accuracy at 738.72k tokens but matches SelfDebug's strict rate. These are descriptive framework comparisons; missing auxiliary calls make some token counts lower bounds. Appendix D provides the accounting protocol, per-group results, and the full framework efficiency table.

4 Related Work

General-purpose data science agents. General-purpose data science agents organize analytical work as an iterative process of planning, tool use, execution, and revision. ReAct and Reflexion provide general mechanisms for action-feedback interaction and self-reflection, while DataVoyager applies agent-based reasoning to data-driven hypothesis generation and verification (Yao et al., 2023; Shinn et al., 2023; Majumder et al., 2024a). Benchmarks examine complementary aspects of this process: DiscoveryBench evaluates data-driven discovery, ScienceAgentBench tests executable analyses derived from published studies, and BLADE assesses choices of variables, transformations, and statistical models (Majumder et al., 2024b; Chen et al., 2024; Gu et al., 2024). Evaluation also extends to multimodal scientific evidence in MoSciBench and longer research-replication workflows in PaperBench (Liu et al., 2026; Starace et al., 2025). These efforts establish important foundations for evaluating analytical execution. **ClinLens** focuses on patient-centric clinical requirement: an executable analysis must use evidence belonging to the intended patient and episode, with asynchronously acquired modalities aligned to the task's temporal windows.

Medical data science agents. Medical data-science research adapts agentic analysis to domain-specific records, tools, and scientific workflows, with benchmarks assessing the resulting capabilities. For clinical records, EHRAgent combines multi-table EHR reasoning with iterative code execution (Shi et al., 2024). For broader biomedical analysis, BioMedAgent and BioMaster coordinate specialized agents and workflow stages, while Biomni combines retrieval with programming and STELLA develops reusable reasoning templates and expandable tools (Bu et al., 2026; Su et al., 2025; Huang et al., 2025; Jin et al., 2025). Evaluation accompanies these methodological advances: BioDS-Bench and BixBench test biomedical programming and computational analysis, and MedAgentGym supports both evaluation and training through executable task environments (Xu et al., 2026a). Other clinical benchmarks emphasize modality breadth, as in MultiMedBench, or explicit temporal evidence grounding, as in CLIR-Bench (Tu et al., 2023; Nie et al., 2026). Building on these complementary directions, **ClinLens** jointly evaluates two requirements of longitudinal clinical analysis: patient-centric grounding across admissions, ICU stays, and repeated studies, and temporal alignment of task-relevant EHRs, notes, ECGs, CXRs, and ECHOs. Its artifact-level checks distinguish answer correctness from the validity of the underlying workflow. A comparison of clinical modalities and patient-centric workflow coverage is provided in Appendix G, Table 5.

5 Conclusion

We introduced **ClinLens**, a patient-centric benchmark for evaluating long-horizon LLM agents on longitudinal, multimodal clinical data science. Its 126 executable tasks make patient-episode grounding and asynchronous multimodal integration explicit requirements. Our experiments reveal substantial gaps between successful execution and valid clinical analysis. The strongest evaluated agent achieves 54.0% final-answer accuracy but only a 29.4% strict pass rate. Diagnostic analyses identify patient-episode grounding as the principal observed bottleneck. These findings highlight the construction and preservation of a valid patient-specific analytical context as a central challenge for clinical data science agents.

REFERENCES

Dechao Bu, Jingbo Sun, Kun Li, Zihao He, Wei Huang, Jinlin Hu, Shanshan Zhang, Shuangshuang Lei, Peipei Huo, Zhihao Wang, et al. Empowering AI data scientists using a multi-agent LLM framework with self-evolving capabilities for autonomous, tool-aware biomedical data analyses. *Nature Biomedical Engineering*, 2026. doi: 10.1038/s41551-026-01634-6.

Xinyun Chen, Maxwell Lin, Nathanael Sch"arli, and Denny Zhou. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*, 2023.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. *arXiv preprint arXiv:2410.05080*, 2024.

Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000. doi: 10.1161/01.CIR.101.23.e215.

Brian Gow, Tom Pollard, Larry A. Nathanson, Alistair Johnson, Benjamin Moody, Chrystinne Fernandes, Nathaniel Greenbaum, Jonathan W. Waks, Parastou Eslami, Tanner Carbonati, Ashish Chaudhari, Elizabeth Herbst, Dana Moukheiber, Seth Berkowitz, Roger Mark, and Steven Horng. MIMIC-IV-ECG: Diagnostic electrocardiogram matched subset. *PhysioNet*, 2023. doi: 10.13026/4nqg-sb35. Version 1.0.

Brian Gow, Tom Pollard, Nathaniel Greenbaum, Benjamin Moody, Ahram Han, Jonathan W. Waks, Alistair Johnson, Elizabeth Herbst, Parastou Eslami, Ashish Chaudhari, Tanner Carbonati, Seth Berkowitz, Roger Mark, and Steven Horng. MIMIC-IV-ECHO: Echocardiogram matched subset. *PhysioNet*, 2026. doi: 10.13026/nrjh-5r77. Version 1.0.

Ken Gu, Ruoxi Shang, Ruien Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, et al. BLADE: Benchmarking language model agents for data-driven science. *arXiv preprint arXiv:2408.09667*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Yusuf Roohani, Ryan Li, Lin Qiu, Gavin Li, Junze Zhang, et al. Biomni: A general-purpose biomedical AI agent. *bioRxiv*, 2025. doi: 10.1101/2025.05.30.656746.

Ruofan Jin, Zaixi Zhang, Mengdi Wang, and Le Cong. STELLA: Self-evolving LLM agent for biomedical research. *arXiv preprint arXiv:2507.02004*, 2025. doi: 10.48550/arXiv.2507.02004. URL https://arxiv.org/abs/2507.02004.

Liqiang Jing, Zhehui Huang, Xiaoyang Wang, Wenlin Yao, Wenhao Yu, Kaixin Ma, Hongming Zhang, Xinya Du, and Dong Yu. Dsbench: How far are data science agents from becoming data science experts? In *International Conference on Learning Representations*, volume 2025, pp. 32597–32649, 2025.

Alistair Johnson, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. MIMIC-IV-Note: Deidentified free-text clinical notes. *PhysioNet*, 2023a. doi: 10.13026/0p14-t007. Version 2.1.

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

Alistair Johnson, Matthew Lungren, Yifan Peng, Zhiyong Lu, Roger Mark, Seth Berkowitz, and Steven Horng. MIMIC-CXR-JPG: Chest radiographs with structured labels. *PhysioNet*, 2024b. doi: 10.13026/jsn5-t979. Version 2.1.0.

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

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

Fan Liu, Xiaozhao Zeng, and Hao Liu. Towards multimodal data-driven scientific discovery powered by LLM agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kZHSvETWdi.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Sanchaita Hazra, Ashish Sabharwal, and Peter Clark. Position: Data-driven discovery with large generative models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34350–34382. PMLR, 2024a. URL https://proceedings.mlr.press/v235/majumder24a.html.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. DiscoveryBench: Towards data-driven discovery with large language models. *arXiv preprint arXiv:2407.01725*, 2024b.

Ludovico Mitchener, Jon M. Laurent, Benjamin Tenmann, Siddharth Narayanan, Geemi P. Wellawatte, Andrew White, Lorenzo Sani, and Samuel G. Rodriques. BixBench: A comprehensive benchmark for LLM-based agents in computational biology. *arXiv preprint arXiv:2503.00096*, 2025.

Frank Nie, Ethan B Liu, Yuan Zhu, Loe Yan, Wei Fan, and Jindong Han. Clir-bench: Benchmarking multimodal question answering over irregular clinical time series. *arXiv preprint arXiv:2607.09880*, 2026.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce C. Ho, Carl Yang, and May Dongmei Wang. EHRAgent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22315–22339, 2024. doi: 10.18653/v1/2024.emnlp-main.1245.

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.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI's ability to replicate AI research. *arXiv preprint arXiv:2504.01848*, 2025.

Houcheng Su, Weicai Long, and Yanlin Zhang. BioMaster: Multi-agent system for automated bioinformatics analysis workflow. *bioRxiv*, 2025. doi: 10.1101/2025.01.23.634608.

Tao Tu, Shekoofeh Azizi, Danny Driess, Mike Schaekermann, Mohamed Amin, Pi-Chuan Chang, Andrew Carroll, Chuck Lau, Ryutaro Tanno, Ira Ktena, et al. Towards generalist biomedical AI. *arXiv preprint arXiv:2307.14334*, 2023.

Zifeng Wang, Benjamin Danek, Ziwei Yang, Zheng Chen, and Jimeng Sun. Making large language models reliable data science programming copilots for biomedical research. *Nature biomedical engineering*, pp. 1–15, 2026.

Ran Xu, Yuchen Zhuang, Yishan Zhong, Yue Yu, Zifeng Wang, Xiangru Tang, Hang Wu, May Dongmei Wang, Peifeng Ruan, Donghan Yang, et al. Medagentgym: A scalable agentic training environment for code-centric reasoning in biomedical data science. In *International Conference on Learning Representations*, volume 2026, pp. 89688–89730, 2026a.

Zihan Xu, Yanzhen Chen, Xiaocheng Zhang, Zhiting Fan, Weiqi Zhai, Hongxia Xu, and Zuozhu Liu. Longmedbench: Benchmarking medical agents for long-horizon clinical decision-making. *arXiv preprint arXiv:2607.09322*, 2026b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Shaolei Zhang, Ju Fan, Meihao Fan, Yizhe Liu, Yuxin Zhang, and Xiaoyong Du. Deepanalyze: Agentic large language models for autonomous data science. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=tcPZvRXQhX.

A Task Construction and Operational Details

The following operational descriptions supplement the construction overview in the main text.

Task instructions. Each task is framed as a scientific question with three elements: (i) a patient-centered clinical context identifying the target patient and longitudinal setting; (ii) an analysis request specifying the clinical question, analysis population, temporal constraints, relevant modalities, and required computation; and (iii) an output contract defining the expected answer format and required artifacts. The clinical context and analysis request determine which patient episodes, time windows, and cross-modal evidence are relevant, while the output contract provides the anchor for evaluation. Final answers may be numerical values, categorical labels, short strings, or structured results, and the accompanying artifacts expose intermediate cohorts, temporal alignments, statistics, or model outputs needed to verify the analysis. Agents independently determine the necessary data exploration, preprocessing, cross-modal alignment, statistical analysis, and modeling steps.

Longitudinal multimodal patient packages. Each task provides a bounded, patient-first package that preserves both the longitudinal organization of a clinical trajectory and the native diversity of multimodal evidence. Records are organized by `subject_id` and retain repeated hospital admissions (`hadm_id`), ICU stays (`stay_id`), and modality-specific events or studies (`study_id`), together with their native timestamps. This hierarchy allows agents to reconstruct how a target patient's structured EHR events, clinical notes, ECGs, CXRs, and ECHOs evolve across episodes and time windows. The modalities also retain their original forms, including multi-table records, narrative text, physiological waveforms, radiographs, and echocardiographic studies. Modality availability, repeated examinations, and missing observations are preserved as properties of the source data. Schemas and file paths are standardized for access, while episode assignment for studies without encounter identifiers, cross-modal alignment, temporal filtering, repeated-measure aggregation, feature construction, and missing-data handling remain agent responsibilities rather than precomputed package properties.

Evaluation. **`ClinLens`** adopts a patient-centered evaluation protocol: an agent is judged by whether its submission correctly reconstructs the target patient's longitudinal clinical context and produces an analytically valid answer. Each task is paired with an evaluator-private executable reference workflow that generates the expected artifacts and answer under fixed analytical and temporal specifications. Evaluation proceeds in three stages. First, the evaluator checks the existence, schema, and parseability of all required artifacts. Second, task-specific semantic checks validate target-patient identification, cohort construction, episode linkage, temporal boundaries, cross-modal alignment, and required statistical or model outputs. Third, the reported answer is compared with the reference result using declared tolerances for numerical outputs and task-specific matching rules for categorical or structured outputs. The deterministic references and task-specific matching rules make the operational evaluation automatic and reproducible across submissions to the same frozen task.

Patient-centered indexing and packaging. We integrate five PhysioNet resources: MIMIC-IV structured records (Johnson et al., 2024a; 2023b), MIMIC-IV-Note (Johnson et al., 2023a), MIMIC-IV-ECG (Gow et al., 2023), MIMIC-CXR-JPG (Johnson et al., 2024b), and MIMIC-IV-ECHO (Gow et al., 2026). Shared patient identifiers are used to assemble a bounded, multi-table package for each task, containing one named target patient and a reference cohort. Native patient, admission, ICU-stay, study, and event identifiers and timestamps are preserved, together with repeated examinations, one-to-many relationships, modality availability, and missing observations. Structured EHR events, clinical notes, ECG waveforms, chest radiographs, and echocardiographic studies retain their modality-specific representations and source clocks. We standardize schemas and file paths and register accessible raw assets, but do not precompute episode assignments, cross-modal joins, temporal alignments, features, or aggregate measurements. Agents use these packages to reconstruct the target patient's course and integrate multimodal evidence for the requested analysis.

Clinically grounded task design and instantiation. Task construction begins with realistic clinical data-analysis needs, such as characterizing a patient's longitudinal course, estimating associations or treatment effects, predicting subsequent outcomes from evidence available at a clinical landmark, and identifying recurring patterns in multimodal records. Each clinical need is translated into a patient-centered workflow that specifies the target-patient context, analysis population, eligibility criteria, temporal boundaries, relevant modalities, analytical objective, and expected outputs. We then instantiate six or seven eligible target patients per workflow template using fixed selection and sampling rules, independently materializing a patient package for each instance. An executable reference workflow operationalizes the clinical question under prespecified preprocessing, analytical, and fallback rules, confirming that the task is feasible and produces a reproducible result. The natural-language instruction, required artifacts, and evaluator are subsequently derived from this validated task protocol to maintain consistency between the intended clinical question and its evaluation.

Validation and quality control. Quality control is applied at both the template and instance levels. Clinical and statistical review examines the validity of cohort definitions, temporal directionality, modality use, variable definitions, and analytical procedures. Each materialized instance is then rerun in a clean environment, where automated checks verify package boundaries, modality and raw-asset accessibility, identifier and timestamp consistency, artifact schemas, row-level and temporal semantics, and agreement between the reference artifacts and final answer. Public task packages are separated from evaluator-private references to prevent access to gold artifacts or answers. Only instances whose packages, reference workflows, and expected submissions pass the complete evaluator are retained.

Illustrative workflows. The taxonomy requires agents to construct clinically and temporally valid analyses rather than retrieve static labels. For example, T09 constructs a 24-hour admission landmark and compares EHR-only and multimodal models for subsequent clinical deterioration. T18 uses a pretreatment CXR event to emulate a target trial comparing alternative pleural-drainage windows. T20 aligns CXR-derived pneumothorax evidence with respiratory support and procedures to discover reproducible acute management patterns. Section 3.2 reports results across all 20 templates; Appendix C.2 reports inspected failures showing how incorrect linkage changes the analysis units.

Reported allocation. The five analysis types in **`ClinLens`** capture complementary capabilities required for longitudinal clinical data science: (1) *descriptive analysis*, which characterizes patient states and changes over time; (2) *correlational analysis*, which quantifies relationships between clinical variables or modalities; (3) *causal inference*, which estimates contrasts between clinically defined interventions or exposure strategies; (4) *predictive modeling*, which predicts future outcomes using only evidence available before a specified landmark; and (5) *pattern discovery*, which identifies reproducible trajectories, phenotypes, or clinical management patterns. Crossing these analysis types with the four patient-time levels produces 20 workflow templates; Figure 3 illustrates one representative task at each patient-time level. The benchmark contains 26 descriptive tasks and 25 tasks in each of the other four categories, with six or seven independently materialized instances per template. This near-uniform allocation supports comparisons across both longitudinal scopes and analytical capabilities. Figure 5 summarizes the five clinical analysis capabilities and their task counts, complementing the breakdown of longitudinal scopes and represented modalities in Table 2.

Table 2: **Summary of patient-time levels, represented modalities, and task counts in `ClinLens`.** Modalities are aggregated across the workflow templates within each level; individual tasks use clinically relevant subsets.

Patient-time level	Represented modalities	Tasks
Patient trajectory	EHR, NOTE, CXR, ECG, ECHO	32
Hospital admission	EHR, NOTE, CXR, ECG	32
ICU stay	EHR, NOTE, CXR, ECG	31
Clinical event/study	EHR, CXR, ECG, ECHO	31
Total		**126**

B Evaluation Protocol and Analysis Populations

Task suite and execution protocol. We evaluate the frozen V6 challenge suite of 126 tasks spanning 20 templates. The four patient-time levels contain 32, 32, 31, and 31 tasks. The suite emphasizes historically difficult workflows, so aggregate performance describes this stress suite rather than an IID distribution of clinical analyses. Standardized configurations use zero-shot prompting and temperature 0, with up to three generation or repair iterations and a maximum code-execution time of one hour per task; NoDataGuess receives one formatting iteration. Agents can access only their assigned package and manifest-listed raw assets. Private references, evaluator outputs, other packages, and previous submissions are withheld, restricting each run to its assigned task and permitted evidence.

Backbones and evidence coverage. The existing system comparison covers DeepSeek-V4-Pro, GLM-5.2, GPT-5.6-Luna, Gemini-3.7-Flash, and Qwen-3.8-Max. DeepSeek and GLM retain previously completed evaluator-compatible results because compatible sealed V6 Stage-B outputs are unavailable. Qwen uses the full 126-task denominator, counting tasks without evaluated submissions as unsuccessful. We therefore distinguish these descriptive comparisons from the complete V6 diagnostic cohort below. Unexecuted projected harness results are excluded from the empirical table.

Agent frameworks. We pair each backbone with six standardized agent settings. **NoDataGuess** receives the task instruction and answer schema without inspecting the patient package, providing an answer-prior control. **ReAct** alternates between reasoning, code execution, and observation (Yao et al., 2023). **DataVoyager** first profiles the package and then invokes a planner–coder–critic workflow. **Reflexion** revises its analysis through self-reflection without reference feedback (Shinn et al., 2023). **SelfDebug** repairs generated programs using execution and submission-check traces (Chen et al., 2023). **RAG-ReAct** retrieves task-relevant instructions and schema information from the assigned package before executing the reasoning, code-generation, and observation loop of a ReAct workflow.

We additionally evaluate Codex and Claude Code as interactive coding harnesses using GPT-5.6-Luna. These systems receive the same task instruction and patient package but use their native file-inspection, code-generation, execution, and revision loops. GPT-5.6-Sol with Codex is evaluated separately and reported as a model–harness combination, not an isolated harness comparison.

To examine the transfer of existing biomedical-agent designs, we adapt BioMedAgent (Bu et al., 2026), the BixBench agent environment (Mitchener et al., 2025), BioMaster (Su et al., 2025), EHRAgent (Shi et al., 2024), and Biomni (Huang et al., 2025) to the same package and submission interface. The adapters preserve each system's planning, retrieval, programming, debugging, and checking components while connecting them to the **`ClinLens`** data and artifact contract. All five biomedical agents use GPT-5.6-Luna as their backbone under this shared interface.

Evaluation metrics. We report three complementary metrics. FinalAcc (A) checks the designated final result using the task-specific categorical or numerical matching rule. StrictPass (S) additionally requires all applicable workflow gates:

$$\text{StrictPass}_i = \mathbb{1}\left[C_i^{\text{art}} \wedge C_i^{\text{parse}} \wedge C_i^{\text{rows}} \wedge C_i^{\text{val}} \wedge C_i^{\text{answer}}\right], \tag{1}$$

where the gates check artifact completeness, schema, cohort and row semantics, validation invariants, and final-answer correctness. The 24 robustness tasks must pass both base and additional analyses. For measured V6 prediction workflows, both EHR-only and multimodal models must satisfy

$$\text{AUROC} \geq \max(0.50, \text{AUROC}_{\text{ref}} - 0.10) \tag{2}$$

and achieve AUPRC at least as high as hidden-test outcome prevalence. EXECSUCCESS (E) records completed execution with a runner-accepted submission. Separately, *Artifact-ready* checks completeness and schema from evaluator reports and differs from EXECSUCCESS.

Analysis populations and aggregation. The new analyses reuse frozen task configurations, submissions, and evaluator reports without rerunning models or reference workflows. They cover ten complete data-grounded configurations: GPT-5.6-Luna and Gemini-3.7-Flash, each with ReAct, DataVoyager, Reflexion, SelfDebug, and RAG-ReAct. Qwen's incomplete runs, historical results, NoDataGuess, coding harnesses, and biomedical transfers are excluded from this diagnostic cohort. Table 3 specifies the analysis denominators. The main system and efficiency tables report unweighted macro-averages across four patient-time levels; diagnostic percentages pool the stated task–configuration records and should not be compared as identical aggregations.

Table 3: Analysis populations for the diagnostic evaluations. All four analyses use the same ten complete configurations; A1, A2, A3, and A6 are source-report identifiers.

Analysis	Location	Included tasks	Records
A1: capability matrix	RQ1	All 126	1,260
A2: semantic funnel	RQ2	101 non-prediction tasks	1,010
A3: semantic burden	Appendix C.3	I01–I03 of each template	600
A6: answer–workflow gap	RQ3	All 126	1,260

C SUPPLEMENTARY DIAGNOSTIC RESULTS

C.1 SYSTEM COMPARISONS AND TRANSFER

End-to-end performance remains limited. Table 1 reports system-level results. GPT-5.6-Sol with Codex is the strongest observed system, with 54.0% FINALACC, 29.4% STRICTPASS, and 86.4% EXECSUCCESS under patient-time macro averaging. This corresponds to 68 answer-correct tasks and 37 strict passes out of 126. These results describe the combined model and harness, rather than an isolated harness effect. In the standardized grid, Gemini-3.7-Flash with Reflexion attains the highest FINALACC (18.1%). The highest reported STRICTPASS is 8.0% for Qwen with ReAct, subject to the coverage caveat above; the complete Luna and Gemini configurations reach 7.9%.

Data access helps, but scaffold rankings depend on the metric. NoDataGuess obtains 0.0–4.7% FINALACC and no strict passes. Within the complete V6 cohort, data-grounded scaffolds improve answer accuracy, but none consistently dominates across backbones and metrics. Gemini with Reflexion leads on FINALACC, whereas DataVoyager leads on STRICTPASS for that backbone. Under Luna, Claude Code attains 21.3% FINALACC and 7.9% STRICTPASS, compared with Codex's 15.0% and 5.5%, even though execution success is lower (69.9% versus 100%). Thus, successful execution alone does not establish a valid clinical workflow or preserve strict-validity rankings.

The capability breakdown localizes success to four templates. Figure 6 crosses patient-time levels with five analysis capabilities for the ten complete configurations. Only T04, T09, T10, and T19 have nonzero FINALACC, at 85.0%, 41.4%, 91.7%, and 98.3%, respectively; their strict rates are 58.3%, 30.0%, 0.0%, and 45.0%. All remaining 16 templates have zero final-answer accuracy. Prediction is not uniformly tractable: T14, the ICU-stay prediction template, has no successful answer. T10 exhibits the opposite mismatch, with high pattern-label accuracy but no validated workflow. Capability and patient-time level jointly distinguish failures hidden by aggregate accuracy.

Biomedical-agent transfer remains selective. Among five adapted systems using Luna, EHRAgent performs best, reaching 11.9% FINALACC, 6.4% STRICTPASS, and 87.4% EXECSUCCESS. BioMedAgent obtains 10.3% and 3.1% under the two correctness metrics, BixBench obtains 0.8% under each, and BioMaster and Biomni obtain no correct answers. None answers an ICU-stay task correctly. These results describe interface transfer by the evaluated adapters, not rankings of the original systems under their native data, tools, and evaluation protocols.

C.2 FAILURE LOCALIZATION

An ordered funnel separates usable artifacts from valid analysis units. We construct the cumulative sequence Artifact-ready → Patient → Episode → Time → Analysis → Final answer. A record survives a stage only if it passes that stage and all preceding ones. Analysis aggregates downstream cohort, raw-evidence, and analytical requirements. We restrict the funnel to 1,010 non-prediction records because exported prediction Stage-B reports do not expose the full patient/episode/time gate sequence. First failures follow evaluator order and do not independently count underlying errors.

The largest drop occurs at episode grounding. Of 1,010 records, 932 (92.3%) are artifact-ready and 899 (89.0%) pass patient grounding, but only 59 (5.8%) also pass episode grounding (Figure 7). Episode selection is the first failure in 840 records (83.2%), compared with 78 artifact failures, 57 analysis failures, 33 patient failures, and two time-window failures. Only 57 records reach the analysis stage, and none passes it. The 83.2-percentage-point drop from patient to episode survival identifies the principal observed bottleneck as reconstructing admissions, ICU stays, studies, and eligible event units. The small number of first time-window failures does not imply that temporal alignment is easy: most records have already failed an earlier gate and cannot first fail here.

Template-level evidence identifies concrete reconstruction errors. Among 950 records from the 15 zero-accuracy non-prediction templates, 785 (82.6%) first fail episode grounding. The report's inspected examples include choosing non-adjacent ECHO studies instead of applying the specified timestamp and study-ID tie-break (T02), linking the same CXR study to multiple admissions (T03), and duplicating episode identities in 33/60 T06 submissions. These errors change the analytical population or exposure window before a statistical estimate is computed. T14 requires separate interpretation: 59/60 prediction records first fail exact private-test-unit coverage and one fails Stage-B execution. These are inspected failures, not a complete per-template diagnostic report.

C.3 SEMANTIC BURDEN AND WORKFLOW SUCCESS

Motivation and analysis population. We examine whether semantic grounding burden is associated with workflow success after adjustment for observed task characteristics. This exploratory analysis uses I01–I03 from each of the 20 templates, yielding 60 tasks and 600 task–configuration records. The ten complete configurations pair GPT-5.6-Luna and Gemini-3.7-Flash with ReAct, DataVoyager, Reflexion, SelfDebug, and RAG-ReAct.

Burden annotations and estimation. Contract-based annotations cover patient burden (repeated units, reference fitting, and patient-level splits or clustering), episode burden (admission/stay/study joins and multiple episodes), and time burden (windows, ordering, landmarks, future outcomes, tie-breaks, availability constraints, and resampling). We estimate linear probability models controlling for agent configuration, patient-time level, capability, package bytes, and file count, with standard errors clustered by the 20 templates. Exact CSV row counts were unavailable and are not included as task-level controls. Figure 10 reports adjusted coefficients and 95% confidence intervals in percentage points per one-standard-deviation increase in burden. Squares denote STRICTPASS and circles denote Artifact-ready, which checks artifact completeness and schema rather than execution success.

Episode burden is negatively associated with strict success. The episode-burden coefficient for STRICTPASS is -19.78 percentage points per standard deviation (95% CI $[-29.86, -9.70]$, $p = 0.0001$). Patient and time estimates are $+5.45$ and -4.40 points, respectively, with both confidence intervals including zero. The episode-specific association is consistent with the failure-localization results in Section 3.3.

Composite burden does not establish an independent association. The composite-burden coefficient for STRICTPASS is -6.84 percentage points (95% CI $[-17.85, 4.16]$, $p = 0.2229$). Its coefficient for Artifact-ready is -7.44 points (95% CI $[-15.57, 0.69]$, $p = 0.0729$). Both intervals include zero, so these estimates do not establish an independent association between composite burden and either outcome after the stated adjustment.

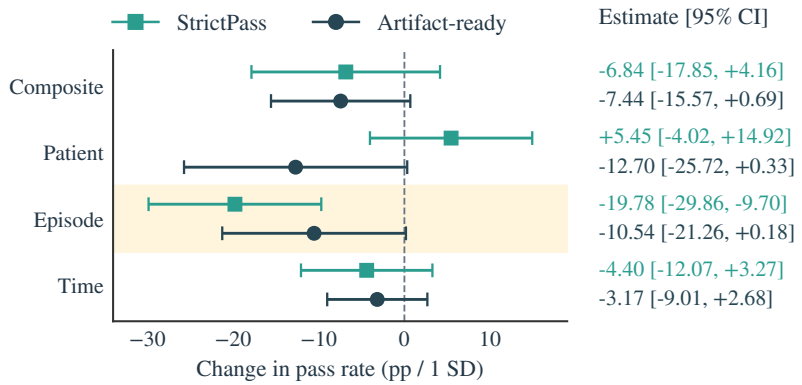


Figure 10: **Episode burden is negatively associated with strict success.** Adjusted coefficients with 95% confidence intervals for StrictPass and Artifact-ready.

Interpretation and limitations. These associations are exploratory rather than causal: burden varies across templates, not through a randomized intervention. Unobserved template difficulty, only 20 clusters, and annotations without an independent second rater limit interpretation. Matched task variants that alter episode-selection demands while preserving the analytical objective would provide a more controlled test of the episode-burden hypothesis.

C.4 Answer–workflow gap

Two distinct mechanisms explain the gap. The earliest failures include 55 episode-grounding failures and 56 analysis/model-quality failures: 51 fail `PredictionQualityFloorPass` and five fail `PredictionProtocolPass`: the designated target prediction can match within tolerance even when model quality or the prediction protocol is inadequate. The 55 episode failures match the answer-correct count for T10, whose strict rate is zero. Agreement on a final label can therefore coexist with an incorrect event population; the checks alone cannot distinguish chance agreement from a label insensitive to upstream errors under the specified matching rule.

Overlapping checks expose coupled defects. All 111 gap records fail an analysis/model-quality check; 55 also fail each of episode grounding, time alignment, and multimodal evidence, 46 fail cohort construction, and eight fail robustness checks. These categories overlap and differ from the mutually exclusive first-failure partition. They show why summing failed gates exaggerates the number of independent mistakes: one incorrect event set can invalidate several subsequent checks.

D Attempt Counts and Recorded Resource Use

Comparison protocol. We compare the six standard scaffolds and Claude Code under GPT-5.6-Luna on the same 126 tasks. For each final submission, we aggregate the recorded input and output tokens of the selected run, including its recorded repair calls, and pair them with patient-time macro FinalAcc, StrictPass, and elapsed time. Table 4 reports these framework-level performance and efficiency values. Because some auxiliary calls are absent from the saved logs, the token values for DataVoyager, Reflexion, and Claude Code are recorded lower bounds; they support descriptive comparisons of recorded token use, not estimates of complete billing cost or total computation.

Attempt-count analysis. To examine how repeated attempts relate to final outcomes, we group the six standard scaffolds by the total recorded attempts in each selected run (Figure 9). The analysis includes 756 task–framework records, with 126 tasks per scaffold. Within each framework, each task contributes once to the group matching its observed attempt count, from one to four. Each plotted rate is the fraction of tasks in that group whose final submission passes the corresponding metric; these pooled group rates differ from the patient-time macro rates in Table 4. NoDataGuess has observations

only at one and two attempts. Claude Code is excluded because its reported interaction turns do not provide an attempt count comparable to the generation and repair attempts of standard scaffolds.

Table 4: GPT-5.6-Luna framework efficiency on the same 126 tasks. FINALACC and STRICTPASS are patient-time macro rates; token and time values are task means. * marks incomplete token accounting.

Framework	FINALACC (%)	STRICTPASS (%)	Tokens (k)	Time (min)
NoDataGuess	3.91	0.00	16.09	1.02
ReAct	13.43	5.52	45.68	3.52
DataVoyager	11.87	3.18	108.03*	5.65
Reflexion	14.99	7.91	48.47*	4.53
SelfDebug	14.99	7.94	46.02	3.54
RAG-ReAct	14.19	7.91	45.17	6.42
Claude Code	21.27	7.94	738.72*	16.35

More attempts do not consistently correspond to higher final success. Final-answer correctness varies non-monotonically across attempt groups for several frameworks. For example, Reflexion reaches 18.75% in the one-attempt group and 21.74% in the two-attempt group, followed by 12.50% and 3.13% in the three- and four-attempt groups. SelfDebug shows a similar initial difference, from 16.67% to 19.51%, followed by 13.64% and 7.41%. Across all five data-grounded scaffolds, the four-attempt groups contain no strict successes (0/18 for ReAct, 0/47 for DataVoyager, 0/32 for Reflexion, 0/27 for SelfDebug, and 0/16 for RAG-ReAct). Thus, runs with more observed attempts do not consistently end with higher answer correctness or workflow validity.

What the attempt-count comparison establishes. The curves describe final outcomes conditional on observed attempt counts. Task difficulty and framework stopping behavior can affect which tasks enter each group, and the saved data contain no intermediate scores. Consequently, these results cannot establish whether an additional revision improves or harms the same task. Measuring revision gains requires scoring intermediate submissions on a fixed task cohort with controlled revision budgets and stopping rules, rather than comparing groups with different task compositions.

Additional orchestration is not consistently efficient. DataVoyager records 108.03k tokens per task, 2.37× ReAct's 45.68k, but obtains lower final-answer accuracy (11.87% versus 13.43%) and lower strict accuracy (3.18% versus 5.52%). SelfDebug records 46.02k tokens per task, nearly identical to ReAct, while improving final-answer accuracy to 14.99% and strict accuracy to 7.94%. In micro efficiency terms, SelfDebug produces 3.28 answer-correct tasks and 1.72 strict passes per million recorded tokens, compared with 2.95 and 1.22 for ReAct and 1.10 and 0.29 for DataVoyager. Reflexion and RAG-ReAct reach similar accuracy to SelfDebug, but require more elapsed time: 4.53 and 6.42 minutes per task, respectively, compared with 3.54 minutes for SelfDebug.

The coding harness improves answers at substantially higher cost. Claude Code reaches the highest GPT-5.6-Luna final-answer accuracy, 21.27%, but records 738.72k tokens and 16.35 minutes per task. This is 16.17× ReAct's recorded token use and 4.64× its elapsed time. Its STRICTPASS accuracy is 7.94%, equal to SelfDebug at the reported precision despite the large resource difference. In this comparison, the harness obtains more answer-correct outputs at higher recorded cost, without a corresponding increase in fully validated workflows. Framework differences and incomplete token accounting prevent attributing the observed performance differences to additional computation alone.

Most recorded computation does not end in a successful submission. Across the twelve complete standard configurations of GPT-5.6-Luna and Gemini-3.7-Flash, the selected runs contain 91.79 million recorded tokens. Final-answer failures account for 85.80% of these tokens, and strict failures account for 94.46%. This distribution mirrors RQ2: agents can spend substantial computation producing and repairing artifacts while retaining an incorrect episode, cohort, or analytical result.

E EVALUATION DESIGN AND AGENT-DESIGN IMPLICATIONS

What the benchmark currently measures. The present release stresses multimodal data engineering and analysis more than end-to-end perceptual model training. Packages expose raw CXR, ECG,

and ECHO assets for audit and optional analysis, but also include structured labels, measurements, and metadata so that a task is feasible under bounded compute. The observed failures therefore cannot be attributed mainly to image or waveform recognition. They instead reveal difficulty with linking modality events to encounters, enforcing cutoffs, constructing cohorts, selecting statistical methods, and extracting the requested answer. Future benchmark tracks can isolate raw-perception dependence by withholding derived labels and measurements but preserving raw-asset access.

Why deterministic answers remain useful. Open clinical analyses can admit multiple defensible workflows, but unrestricted judging is expensive and difficult to reproduce. We retain open implementation choices while fixing the operational definitions that determine the scored quantity. Intermediate artifacts make disagreements diagnosable. This design is intentionally stricter than a narrative answer judge: a plausible clinical interpretation cannot compensate for a leaked target window or a wrong AUROC difference. A calibrated hierarchical rubric could later award credit for valid intermediate steps while retaining the deterministic endpoint for complete workflows.

Implications for agent design. The results support three priorities. First, agents need explicit representations of patient, admission, ICU, and event-time relations rather than treating all CSV files as interchangeable tables. Second, validation should recompute invariants, cohort counts, and temporal boundaries, not only inspect whether code exits. Third, agents should trace the final scalar to a specific artifact and row. The high cost and lower performance of DataVoyager suggest that additional planning layers should be conditional on detected uncertainty rather than always invoked.

F LIMITATIONS

The benchmark is derived from a single health system and inherits the demographic, clinical-practice, documentation, and missingness biases of MIMIC-IV. It evaluates retrospective data analysis, not bedside diagnosis or treatment. The 126-task challenge suite deliberately oversamples difficult templates and is not an IID test set; results should be compared only on the same frozen manifest. Most experiments use one rollout per task, so Pass^3, variance estimates, and statistical significance remain to be measured. We have completed automated reference validation and manual realism review, but have not yet reported a blinded multi-clinician adjudication study or human data-scientist baseline. The medical-agent adapters necessarily modify I/O and sandboxes, and therefore test transfer rather than exact native-system performance. Proprietary snapshots and an API aggregation layer also limit rerun stability; releases must include request parameters, logs, and raw outputs.

G COMPARISON WITH RELATED BENCHMARKS

Table 5 compares the clinical inputs and workflow requirements emphasized by related benchmarks. We distinguish modality coverage from evidence handling: providing multiple data sources does not necessarily require agents to link observations to the correct patient, episode, and time window. In **`ClinLens`**, agents work with structured EHRs, clinical notes, CXRs, ECGs, and echocardiograms within bounded patient-centered packages. The comparison therefore considers not only which modalities are available, but also whether temporal alignment and the handling of missing or conflicting evidence are explicit task requirements. Coverage labels describe benchmark emphasis rather than exhaustive system capabilities.

H TEMPLATE CATALOG

The 20 templates span four patient-time scopes and five analytical categories. Each template defines a reusable analysis specification, while individual task instances bind it to a target patient and a bounded data package. Table 6 lists representative scored outputs; full workflow validity additionally requires satisfying the artifact, temporal, and semantic requirements specified in the corresponding task cards.

Table 5: **Clinical modality and workflow emphasis in related work.** The comparison focuses on clinical inputs and patient-centric evidence handling.

Work	EHR	Notes	CXR	ECG	ECHO	Temporal alignment	Missingness/ conflicts
CliniDSBench / BioDSBench	P	✗	✗	✗	✗	P	P
STELLA	P	✗	✗	✗	✗	P	P
MedAgentGym	✓	✓	✗	✗	✗	P	P
ClinLens (ours)	✓	✓	✓	✓	✓	✓	✓

✓ primary or explicit coverage; P partial, limited, or indirect coverage; ✗ non-primary emphasis, not demonstrated absence. Temporal alignment refers to cross-modal patient-time alignment. *Draft comparison; work identities, citations, and coverage labels require source verification.*

Table 6: The 20 benchmark templates and their scored outputs. Full task cards additionally specify temporal windows, package scope, required artifacts, validation rules, and tolerance.

ID	Time scale	Analysis	Representative scored output
T01	Patient	Profiling	trajectory enrichment
T02	Patient	Association	adjusted lag coefficient
T03	Patient	Temporal effect	difference-in-differences effect
T04	Patient	Prediction	multimodal AUROC gain
T05	Patient	Phenotyping	transition enrichment
T06	Admission	Profiling	onset-cascade separation
T07	Admission	Association	adjusted association
T08	Admission	Temporal effect	event-study effect
T09	Admission	Prediction	landmark AUROC gain
T10	Admission	Phenotyping	cluster-outcome enrichment
T11	ICU stay	Profiling	early-burden enrichment
T12	ICU stay	Association	distributed-lag odds ratio
T13	ICU stay	Temporal effect	intervention response effect
T14	ICU stay	Prediction	best modality combination
T15	ICU stay	Phenotyping	phenotype enrichment
T16	Event/study	Profiling	evidence-age degradation
T17	Event/study	Association	CXR-anchored adjusted odds ratio
T18	Event/study	Temporal effect	pre/post event effect
T19	Event/study	Prediction	ECG-anchored AUROC gain
T20	Event/study	Pattern mining	CXR-note discordance pattern

I Ethics, Data Governance, and Release

All source datasets are deidentified PhysioNet resources (Goldberger et al., 2000) governed by credentialed-access agreements. The benchmark does not attempt reidentification and must not be used for direct patient care. Raw MIMIC records cannot be redistributed through an unrestricted repository. We will release construction code, task schemas, evaluator code, and non-sensitive metadata publicly, while distributing derived patient packages and selected raw assets through PhysioNet under the applicable credentialed health-data license. Development-only absolute paths and full-repository locators will be removed from release manifests. The evaluator blocks network access, manifest-external raw assets, reference and gold files, and scans outside the task package. Agent outputs that resemble diagnosis or treatment advice are flagged, because the intended task is retrospective computation over research records, not a clinical recommendation for patient care.