GUIDELINES AS CONTROL LOGIC FOR MULTIMODAL SEQUENTIAL CLINICAL DECISION-MAKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world clinical decision-making is dynamic, requiring clinicians to continuously acquire patient information and make sequential decisions according to Clinical Practice Guidelines (CPGs). Existing approaches typically incorporate CPG knowledge in two forms: external or parametric. External approaches commonly rely on retrieval-augmented generation (RAG), which injects large amounts of retrieved CPGs into the model context, following a "Guideline as Context" paradigm. Parametric approaches encode guideline knowledge into model parameters, yet still fail to transform guidelines into explicit, traceable, and easily updatable decision control logic. To address these limitations, we propose CPGAgent, a guideline-grounded framework that introduces a "Guidelines as Control Logic" paradigm for multimodal sequential clinical decision-making. Our main idea is to encode CPGs as latent guideline memory slots and dynamically activate relevant slots according to evolving patient states, enabling guideline-guided trajectory planning. For evaluating dynamic decision trajectories, we introduce CPGTrajBench, a multimodal executable benchmark in which evidence is progressively revealed through agent actions. We further develop a guideline-grounded evaluation framework with trajectory-planning rubrics, assessing guideline-consistent planning across the entire clinical trajectory. Experiments show that CPGAgent achieves the best trajectory performance across three cancer types, matching the strongest baseline on one cohort and outperforming it by 9.6 and 9.2 points on the other two. Anonymized code is provided in the reproducibility statement.

1 INTRODUCTION

Medical agents powered by large language models (LLMs) have shown increasing potential for complex clinical workflows. However, reliable clinical decision-making remains challenging because patient information is progressively revealed and states evolve over time. In real-world care, decisions must adapt to changing patient states while remaining aligned with Clinical Practice Guidelines (CPGs), which specify evidence requirements and recommended actions across diagnosis, staging, treatment, and follow-up. This highlights a fundamental challenge for medical agents: how to leverage CPGs to guide sequential decisions under evolving clinical states.

Existing approaches incorporate CPG knowledge primarily in two forms: *external knowledge* and *parametric knowledge*. External approaches access CPGs at inference time through RAG. For example, MedAgent-Pro(Wang et al., 2026) retrieves medical guidelines with a RAG agent to support diagnostic planning, while GARMLE-G Li et al. (2025b) retrieves relevant CPG snippets to ground model-generated clinical recommendations. However, these approaches largely follow a ***Guideline as Context*** paradigm: guidelines are provided as contextual evidence, while the model must still determine their applicability to the current patient state and decide the subsequent clinical action. In contrast, parametric approaches instead internalize guideline knowledge through training. For example, MedGuideX (Shen et al., 2026) explicitly transforms CPG recommendations into executable decision logic and converts the resulting logic into training data for post-training. Although such approaches explicitly extract decision logic from CPGs, they ultimately return to model training, where the extracted logic is absorbed into model parameters and loses its explicit, traceable, and easily updatable form at inference time. Consequently, neither paradigm maintains CPGs as an explicit, state-aware control mechanism that governs sequential clinical decision-making.

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.

Therefore, we propose **CPGAgent**, the first ***Guideline as Control Logic*** framework for multimodal sequential clinical decision-making. The central idea of CPGAgent is to encode CPGs as latent guideline memory slots and dynamically activate decision-relevant memories according to the evolving patient state. Figure 4 illustrates this state-dependent activation process, where guideline memories dynamically transition across clinical stages such as radiology, pathology, staging, biomarkers, and treatment planning as patient evidence accumulates. CPGAgent consists of three core components, as shown in the upper part of Figure 1, the **Guideline Planner** implements this control mechanism as the upper-level planning agent. It represents long and heterogeneous CPGs with compact Latent Guideline Memory Slots, applies a Soft Gate to activate decision-relevant memories, and uses Dynamic Anchor Attention to integrate the activated knowledge with the evolving patient state. The planner then identifies the current clinical pathway, determines unresolved information requirements, and generates guideline-consistent plans. As shown in the lower part of Figure 1, **Medical Executor** serves as the execution agent, translating these plans into concrete medical operations. The two agents interact through a shared **Patient State Machine**, which maintains confirmed evidence, unresolved clinical information, completed actions, and the current decision stage. Newly acquired observations are written back to the state machine, which in turn drives the next round of guideline-memory activation and planning. Through this closed-loop process, CPGAgent transforms clinical guidelines from passive reference context into explicit, state-dependent control logic for the evolving patient state.

To evaluate dynamic clinical decision trajectories under evolving patient states, we introduce **CPGTrajBench**, shown in Figure 2, a multimodal executable benchmark for longitudinal clinical decision-making. Rather than exposing a complete patient case at once, CPGTrajBench progressively reveals EHR, radiology, pathology, and molecular evidence through agent actions, forming an evolving clinical trajectory across diagnosis, staging, and treatment. We further develop a guideline-grounded evaluation framework with **trajectory-planning rubrics**, which assess whether planning remains guideline-consistent as the patient state evolves.

Our contributions are as follows:

- We propose CPGAgent, the first ***Guideline as Control Logic*** framework for multimodal sequential clinical decision-making. By encoding CPGs as latent guideline memory slots and dynamically activating relevant memories according to the evolving patient state, CPGAgent transforms clinical guidelines from passive reference context into explicit control logic for trajectory planning.
- We develop a state-conditioned guideline planning mechanism for efficient use of CPGs. It combines Latent Guideline Memory Slots for compact guideline representation, a Soft Gate for slot selection, and Dynamic Anchor Attention for state-conditioned evidence integration.
- We introduce CPGTrajBench, a multimodal interactive benchmark that progressively reveals heterogeneous clinical evidence through agent actions, enabling evaluation of sequential diagnosis, staging, and treatment under partial observability. We further develop guideline-grounded trajectory-planning rubrics that flexibly align variable-length agent trajectories to assess guideline consistency and clinical admissibility across the entire clinical trajectory.

2 Related Work

2.1 Guideline-Grounded Clinical Reasoning

Existing guideline-grounded reasoning works mainly follow three directions: formalizing guideline decision logic, aligning guidelines with patient-specific trajectories, and evaluating guideline adherence in dynamic clinical settings. Representative studies include MedGUIDE (Li et al., 2025a), MedGuideX (Shen et al., 2026), CPGPrompt (Deng et al., 2026), and MedDM (Li et al., 2023a) for formalizing guideline decision logic; CliCARE (Li et al., 2026) and CancerGUIDE (Unell et al., 2025) for patient-guideline alignment; and CPGBench (Tan et al., 2026) and MedSP1000 (Liang et al., 2026) for multi-turn or trajectory-level evaluation. Our framework further separates guideline-based planning from multimodal evidence acquisition throughout the evolving clinical trajectory.

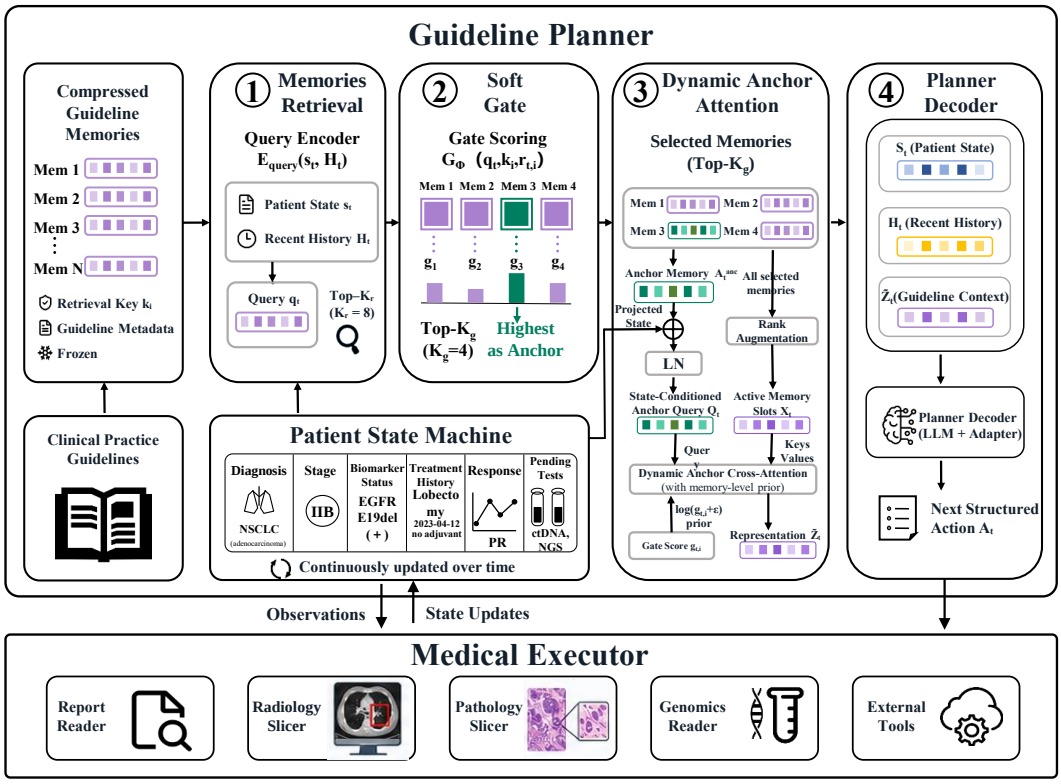


Figure 1: Overview of **CPGAgent**, the guideline-grounded agent system contains an upper-level Guideline Planner and a lower-level Medical Executor. A shared Patient State Machine connects guideline-based planning with multimodal evidence acquisition and execution.

2.2 Medical Agentic Systems

Medical agentic systems have evolved from tool-augmented foundation models toward more structured and adaptive clinical decision-making. Early systems such as MMedAgent (Li et al., 2024) orchestrate specialized medical models for heterogeneous tasks, while MedAgent-Pro (Wang et al., 2026) further introduces explicit planning and tool execution for multimodal diagnosis. More recent systems, including EvoClinician (He et al., 2026) and MedExAgent (Gao et al., 2026a), emphasize iterative and sequential decision-making over evolving patient information, while Meissa (Chen et al., 2026) distills agentic tool-use behaviors into a compact model. These works reflect a progression from tool augmentation to structured orchestration and adaptive clinical agents.

3 Methods

3.1 CPGAgent

We propose a guideline-grounded dual-agent system that separates high-level clinical planning from multimodal evidence acquisition and tool execution. As illustrated in Fig. 1, the **Guideline Planner** serves as the upper-level controller. Given the evolving patient state and recent clinical trajectory, it retrieves relevant guideline knowledge, identifies unresolved information requirements, and generates the next guideline-consistent clinical plans. **Medical Executor** acts as the lower-level executor, converting the plans into executable actions and returning the resulting evidence to support subsequent decision-making. The two agents communicate through a shared **Patient State Machine**, which maintains structured clinical evidence accumulated throughout the trajectory, including diagnosis, stage, biomarker status, completed examinations, treatment history, treatment response, and pending information. At decision step t, the Guideline Planner generates a structured action $\mathcal{A}_t$ based on the current patient state $\mathbf{s}_t$ and recent trajectory history $\mathcal{H}_t$. Medical Executor executes the

requested action or tool operation and obtains new multimodal observations $\mathbf{o}_t$. The patient state is subsequently updated as

$$\mathbf{s}_{t+1} = \mathcal{U}\left(\mathbf{s}_t, \mathcal{A}_t, \mathbf{o}_t\right). \tag{1}$$

Repeating this process produces a longitudinal diagnostic–treatment trajectory in which guideline reasoning, evidence acquisition, and patient-state evolution remain explicitly traceable.

3.2 Guideline Planner

The Guideline Planner provides state-aware access to clinical guidelines without repeatedly inserting full guideline passages into the context. It consists of **Latent Guideline Memory Slots**, constructed by a separately trained Memory Encoder and maintained in a frozen Memory Store, together with a **Soft Gate** for state-conditioned memory selection, **Dynamic Anchor Attention** for memory fusion, and an independently adapted **Planner Decoder**. At each decision step, the current Patient State and recent trajectory history are used to selectively activate and integrate relevant guideline memories into a fixed-length latent representation for next-action generation.

Latent Guideline Memory Slots. We divide each clinical guideline into decision-oriented chunks and encode every chunk offline into $N = 64$ continuous latent slots, $\mathbf{M}_i = [\mathbf{m}_{i,1}, \ldots, \mathbf{m}_{i,N}] \in \mathbb{R}^{N \times H_M}$, together with a retrieval key $\mathbf{k}_i$ and guideline metadata. The resulting Memory Store remains frozen during downstream planner training and inference, providing reusable latent representations of guideline knowledge.

State-conditioned retrieval and Soft Gate. At planning step t, the current Patient State $\mathbf{s}_t$ and recent trajectory history $\mathcal{H}_t$ are encoded into a retrieval query. Before retrieval, the Memory Store is restricted according to the cancer type and guideline version associated with the current case. Candidate memories are then retrieved according to their similarity to the query:

$$\mathbf{q}_t = E_{\text{query}}\left(\mathbf{s}_t, \mathcal{H}_t\right), \qquad r_{t,i} = \text{sim}\left(\mathbf{q}_t, \mathbf{k}_i\right). \tag{2}$$

Here, $r_{t,i}$ measures *semantic relevance* between the current clinical context and guideline memory i. We retain the Top-K_r memories ($K_r = 8$) as candidates.

However, semantic relevance alone does not determine whether a guideline memory is important for the current clinical decision. We therefore introduce a patient-state-conditioned Soft Gate that reranks the retrieved candidates according to *decision relevance*. For each candidate,

$$d_{t,i} = G_\phi\left(\mathbf{q}_t, \mathbf{k}_i, r_{t,i}\right), \qquad g_{t,i} = \frac{\exp(d_{t,i}/\tau)}{\sum_{j \in \mathcal{I}_t} \exp(d_{t,j}/\tau)}, \qquad i \in \mathcal{I}_t, \tag{3}$$

where $\mathcal{I}_t$ contains the Top-K_g memories ranked by $d_{t,i}$ ($K_g = 4$), and τ controls the sharpness of the gate distribution. Unselected memories receive zero weight. The retrieval stage therefore identifies semantically relevant guideline regions, while the Soft Gate further selects memories that are most relevant to the current decision.

Dynamic Anchor Attention (DAA). The active memories selected by the Soft Gate may all be clinically relevant, but they need not contribute equally to the current decision. We therefore introduce DAA for anchor-centered memory fusion. Inspired by query-based latent aggregation mechanisms such as Q-Former (Li et al., 2023b) and Perceiver-style resampling (Alayrac et al., 2022), DAA differs in that its queries are derived from a dynamically selected guideline memory rather than a fixed set of learnable query tokens.

Specifically, the memory with the highest gate weight is selected as the anchor $\mathbf{A}_t^{\text{anc}}$. The current Patient State is projected into the latent memory space and injected into the anchor slots to form state-conditioned anchor queries:

$$\mathbf{Q}_t = \text{LN}\left(\mathbf{A}_t^{\text{anc}} + \mathbf{W}_s^D \mathbf{s}_t\right). \tag{4}$$

Each active memory is augmented with a learnable rank embedding according to its gate ordering. Let $\mathbf{X}_{t,i,j}$ denote slot j of memory i after this augmentation. The state-conditioned anchor slots $\mathbf{Q}_t$ serve as queries, while all active slots serve as keys and values. For attention head h,

$$L_{h,t,n,i,j} = \frac{\left(\mathbf{W}_Q^h \mathbf{Q}_{t,n}\right)^\top \left(\mathbf{W}_K^h \mathbf{X}_{t,i,j}\right)}{\sqrt{d_h}} + \log\left(g_{t,i} + \epsilon\right). \tag{5}$$

The gate term acts as a memory-level prior, allowing highly ranked guideline memories to exert greater influence while still permitting each anchor slot to selectively gather complementary information from the other active memories. Multi-head attention followed by residual and feed-forward transformations produces a fixed-length fused guideline representation $\mathbf{Z}_t \in \mathbb{R}^{64\times H_M}$. Thus, the Soft Gate determines *which guideline memories* are active and their relative importance, whereas DAA determines *how their information is integrated* under the current Patient State.

Planner Decoder. The Planner Decoder is initialized from the trained Memory Encoder and subsequently adapted independently while the Memory Encoder remains frozen. The fused guideline representation is projected into the decoder hidden space through a learnable projection B_ψ, yielding $\widetilde{\mathbf{Z}}_t = B_\psi(\mathbf{Z}_t)$. The next clinical action is then generated as

$$\mathcal{A}_t = \mathrm{Planner}_\theta\left(\mathbf{s}_t, \mathcal{H}_t, \widetilde{\mathbf{Z}}_t\right). \tag{6}$$

The structured output specifies the next clinical step to be executed by Medical Executor, such as acquiring additional evidence, invoking a multimodal examination or external tool, establishing a clinical conclusion, or proposing a treatment-related decision.

Training Strategy. The Guideline Planner is trained in four stages that progressively separate latent guideline representation, action generation, dynamic memory selection, and joint fine-tuning. Detailed objectives and optimization settings are provided in Appendix A.

3.3 Medical Executor

Medical Executor serves as the execution layer of CPGAgent, translating structured actions from the Guideline Planner into operations over multimodal patient data and external clinical tools. Given a planner action, it invokes the corresponding tool for EHR retrieval, radiology or pathology analysis, laboratory and biomarker assessment, molecular-data processing, or other clinical operations, and converts the result into a standardized observation $\mathbf{o}_t$. Medical Executor does not directly modify the clinical state or determine subsequent decisions; instead, the observation is returned to the Patient State Machine for validation and integration before the next planning step. Detailed tools are provided in Appendix B.

3.4 Patient State Machine

The Patient State Machine provides a structured representation of the evolving clinical state and serves as the communication interface between the Guideline Planner and Medical Executor. It maintains validated clinical evidence, completed actions, and unresolved information requirements rather than repeatedly reconstructing the trajectory from free-form history. After each Medical Executor execution, newly acquired observations are validated and incorporated into the state while previously established information is preserved unless revised by new evidence. The resulting state history provides an auditable record of patient-state transitions for subsequent planning and step-level evaluation. The patient state template is provided in Appendix C.

4 CPGTrajBench

4.1 Overview of CPGTrajBench

To evaluate dynamic clinical decision trajectories under evolving patient states, we construct **CPG-TrajBench**, a multimodal and interactive benchmark for clinical agents. Existing clinical benchmarks have progressively expanded from medical knowledge assessment and case-level reasoning toward more realistic clinical workflows, as summarized in Table 1. Building on these advances, CPGTrajBench integrates multimodal evidence from Text, EHR, Radiology, Pathology, and Molecular data within an executable environment. Rather than focusing primarily on final outputs or fixed workflows, CPGTrajBench is designed to assess whether agents acquire relevant evidence, adapt decisions, and maintain guideline-consistent trajectories as patient states evolve. To achieve this goal, we develop **trajectory-planning rubrics** that evaluate whether sequential clinical actions remain guideline-consistent and state-appropriate throughout the entire trajectory.

Table 1: Comparison of clinical benchmarks. Clinical modalities are restricted to {Text, EHR, Radiology, Pathology, Molecular}. Interaction settings are categorized as Static QA, Dialogue/Simulation, or Executable Environment. Workflow stages follow healthcare agent benchmark conventions. Decision granularity denotes the evaluation unit. Guideline integration: × / Partial / ✓.

Benchmark	Clinical Modality	Interaction	Workflow Stage	Decision Granularity	Guideline
MedQA (Jin et al., 2020)	Text	Static QA	Diagnosis	Question	×
MedGUIDE (Li et al., 2025a)	Text	Static QA	Diagnosis, Treatment	Question, Case	✓
CPGPrompt (Deng et al., 2026)	Text	Static QA	Diagnosis, Treatment	Question, Workflow	✓
OncoCoT (Yang et al., 2026)	Text	Static QA	Diagnosis, Treatment	Case, Trajectory	Partial
MedGuideX (Shen et al., 2026)	Text	Static QA	Diagnosis, Treatment	Question, Workflow	✓
ClinicalAgent Bench (Liao et al., 2024)	Text, EHR, Pathology	Static QA	Diagnosis, Data	Question	Partial
MedAgentBoard (Zhu et al., 2025b)	Text, EHR, Pathology	Static QA	Diagnosis, Communication, Research, Data	Question, Workflow	×
AI Hospital (Fan et al., 2024)	Text	Dialogue/Sim.	Diagnosis, Communication	Case, Trajectory	Partial
HealthBench (Arora et al., 2025)	Text	Dialogue/Sim.	Diagnosis, Communication	Conversation	Partial
MAI-DxO (Nori et al., 2025)	Text	Dialogue/Sim.	Diagnosis	Case, Trajectory	×
CP-Env (Zhu et al., 2025a)	Text	Dialogue/Sim.	Diagnosis, Treatment, Communication	Case, Trajectory	Partial
CPGBench (Tan et al., 2026)	Text	Dialogue/Sim.	Diagnosis, Treatment	Conversation, Trajectory	✓
HealthBench Professional (Hicks et al., 2026)	Text	Dialogue/Sim.	Diagnosis, Treatment, Communication, Research	Conversation	Partial
MedSP1000 (Liang et al., 2026)	Text	Dialogue/Sim.	Diagnosis, Treatment, Communication	Case, Trajectory	Partial
MedAgentSim (Almansoori et al., 2025)	Text, EHR	Dialogue/Sim.	Diagnosis, Communication	Case, Trajectory	Partial
AgentClinic (Schmidgall et al., 2024)	Text, EHR, Pathology	Dialogue/Sim.	Diagnosis, Communication	Case, Trajectory	Partial
MedAgentBench (Jiang et al., 2025)	EHR	Exec. Env.	Treatment, Data	Case, Workflow	Partial
HealthAdminBench (Bedi et al., 2026)	EHR	Exec. Env.	Administration	Workflow	×
PhysicianBench (Liu et al., 2026c)	EHR, Text	Exec. Env.	Diagnosis, Treatment, Data	Case, Workflow	×
CamylaBench (Gao et al., 2026b)	Radiology, Pathology	Exec. Env.	Research	Workflow	×
AutoMedBench (Liu et al., 2026a)	Radiology, Pathology	Exec. Env.	Research	Workflow	×
MedAgentGym (Xu et al., 2025)	EHR, Text, Molecular	Exec. Env.	Data, Research	Workflow	×
MTBBench (Vasilev et al., 2025)	Text, Pathology, Molecular	Exec. Env.	Diagnosis, Treatment	Case, Trajectory	×
HealthAgentBench (Liu et al., 2026b)	Text, EHR, Radiology, Pathology	Exec. Env.	Diagnosis, Treatment, Data, Research	Case, Workflow	×
CPGTrajBench (Ours)	**Text, EHR, Radiology, Pathology, Molecular**	**Exec. Env.**	**Diagnosis, Staging, Treatment**	**Case, Trajectory**	✓

4.2 Trajectory-Planning Rubric Construction

To capture how clinical decisions should evolve with the patient state, we construct trajectory-planning rubrics through a multi-round process from an initially partial state S_0. As illustrated in Figure 2, hidden evidence is progressively revealed through clinical actions as the trajectory evolves. At step i, guideline knowledge relevant to the current patient state is retrieved as

$$g_i = \Gamma(S_i, \mathcal{G}). \tag{7}$$

The trajectory generator π_{traj} then proposes a guideline-supported action set and provisional next state, which are verified against the held-out complete patient record:

$$(A_i, \tilde{S}_{i+1}) = \pi_{\text{traj}}(S_i, g_i) \quad V_i = \mathcal{V}\left(A_i, \tilde{S}_{i+1}, g_i, R_{\text{c}}\right) \quad S_{i+1} = \text{Revise}\left(\tilde{S}_{i+1}, V_i\right). \tag{8}$$

Here, V_i identifies unsupported, conflicting, or missing evidence and required state corrections. The complete record R_{c} is used only for verification and is unavailable to the trajectory generator. The resulting state-aware trajectory-planning rubric is

$$\mathcal{R}_c^{\text{traj}} = \{(S_i, A_i)\}_{i=0}^{T}, \tag{9}$$

where T denotes the final decision step. All generated trajectories are reviewed by clinical experts for guideline consistency and clinical plausibility.

Rather than prescribing a single gold action, each A_i contains four categories:

$$A_i = \{A_i^{\text{req}}, A_i^{\text{acc}}, A_i^{\text{cond}}, A_i^{\text{unsafe}}\}, \tag{10}$$

corresponding to required, acceptable, conditional, and unsafe actions. These respectively capture expected actions, valid alternatives, actions requiring unmet prerequisites, and actions that violate the current patient state or guideline constraints.

4.3 Trajectory-Planning Evaluation

Figure 2 demonstrates our trajectory-planning evaluation pipeline. We first align the rubric trajectory with the agent trajectory, then score the matched decision steps while penalizing missed rubric steps and redundant agent actions. For rubric step $D_i = (S_i, A_i)$ and agent step $Z_j = (S_j^{\text{agent}}, \hat{A}_j)$, their compatibility is defined as

$$M(D_i, Z_j) = \alpha \cos\big(S_i, S_j^{\text{agent}}\big) + \beta \cos\Big(A_i, \hat{A}_j\Big), \tag{11}$$

where both terms are cosine similarities computed between the corresponding state and action representations. The resulting optimal one-to-one alignment is denoted by $\mathcal{M}^*$, with details of the alignment procedure provided in Appendix D.1.

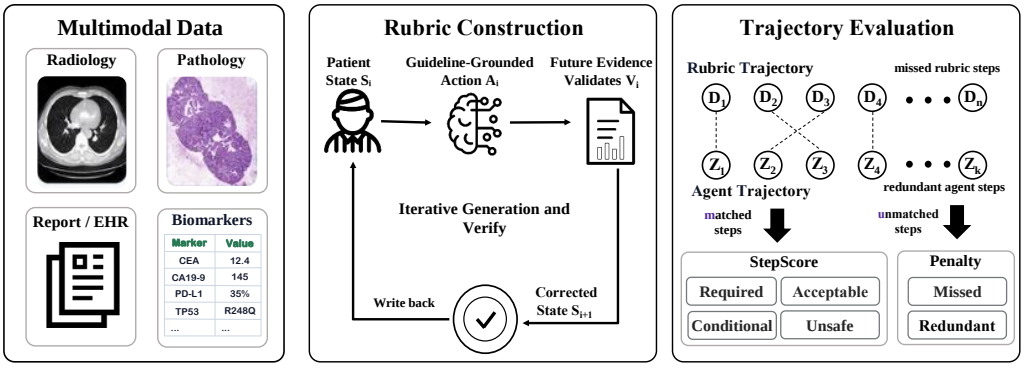


Figure 2: Overview of CPGTrajBench multimodal data, rubric construction and trajectory evaluation. Rubrics are iteratively constructed from evolving patient states through guideline-grounded planning and held-out verification, while agent trajectories are aligned with rubric trajectories for step-wise and trajectory-level scoring.

Each aligned decision is scored by $\mathrm{StepScore}_{ij}$(detailed formulation is provided in Appendix D.2), which accounts for required-action coverage, acceptable-action credit, conditional-action awareness, and unsafe-action penalties. The final trajectory score is

$$S_{\mathrm{traj}} = \frac{1}{|\mathcal{M}^*|} \sum_{(i,j)\in\mathcal{M}^*} \mathrm{StepScore}_{ij} - \lambda_{\mathrm{miss}} r_{\mathrm{miss}} - \lambda_{\mathrm{red}} r_{\mathrm{red}}, \tag{12}$$

where r_{miss} measures missed required rubric steps and r_{red} measures redundant agent actions. Their detailed definitions are provided in Appendix D.3.

5 Experiments

Implementation details. CPGAgent uses a Qwen3.5-9B Guideline Planner and GPT5.6-sol as the Medical Executor execution model. All methods are evaluated under the same tool interface and interaction budget. Detailed system configurations, runtime settings, and evaluation protocols are provided in Appendix E.

5.1 Main Results

We compare CPGAgent with a range of strong baselines, including general-purpose agents, existing medical agents (all using GPT-5.6-sol as the backbone LLM), and Executor Only, which removes the Guideline Planner and directly performs clinical execution. Results are reported across three cancer cohorts: lung adenocarcinoma (LUAD; 30 cases from a public TCGA cohort), uterine corpus endometrial carcinoma (UCEC; 40 cases), and nasopharyngeal carcinoma (NPC; 39 cases). We evaluate trajectory-level performance, which measures whether the agent maintains guideline-consistent decision making as patient states evolve throughout the clinical trajectory.

Table 2 shows that CPGAgent achieves the best trajectory performance across all three cancer types, with scores of 47.957, 49.663, and 55.519 on LUAD, UCEC, and NPC. On LUAD, CPGAgent performs comparably to the strongest external baseline, MEISSA, with a marginal improvement of 0.021 points. The advantage becomes substantially larger on UCEC and NPC, where CPGAgent outperforms the strongest external baselines by 9.560 and 9.217 points respectively. Across all cohorts, CPGAgent achieves a weighted trajectory score of 51.289, exceeding the strongest external baseline by 7.939 points. These results highlight the benefit of guideline-grounded planning for complex clinical trajectories.

The contribution of the Guideline Planner is further demonstrated by comparison with Executor Only, which shares the same clinical execution backbone but performs no explicit guideline-grounded planning. CPGAgent improves trajectory performance over Executor Only by 6.889

Table 2: Trajectory performance comparison on LUAD, UCEC, and NPC tasks.

Model	LUAD	UCEC	NPC	All Cohorts
General-purpose Agents				
Claude-Opus-5 Anthropic (2026)	47.921	<u>40.103</u>	43.163	<u>43.350</u>
Codex GPT-5.5 OpenAI (2026a)	41.679	30.068	43.651	38.124
Codex GPT-5.6-sol OpenAI (2026b)	47.819	35.466	35.964	39.044
Medical Agents				
MedAgent-Pro Wang et al. (2026)	47.288	37.029	38.868	40.511
MMedAgent Li et al. (2024)	38.947	34.929	45.756	39.909
MedExAgent Gao et al. (2026a)	44.045	35.831	44.689	41.261
EvoClinician He et al. (2026)	45.741	35.594	<u>46.302</u>	42.218
MEISSA Chen et al. (2026)	<u>47.936</u>	33.863	43.300	41.113
MMed-RAG Xia et al. (2025)	41.218	37.160	39.769	39.210
UniRAG Sharifymoghaddam et al. (2025)	39.933	33.718	35.977	36.237
Ours				
Executor Only	41.068	35.267	28.693	34.511
CPGAgent	**47.957**	**49.663**	**55.519**	**51.289**

points on LUAD, 14.396 points on UCEC, and 26.826 points on NPC. The consistently larger gains on UCEC and NPC indicate that explicitly operationalizing CPGs as control logic is particularly beneficial for longer or more complex decision trajectories, where locally plausible actions alone may fail to maintain guideline-consistent clinical progression. Overall, these results demonstrate that CPGAgent improves sequential clinical decision making primarily by providing state-aware, guideline-grounded control over the evolving trajectory.

5.2 Ablation Experiments and Trajectory Evaluation

Table 3: Ablation study of the Planner.

Configuration	Trajectory
Full Planner	**51.289**
w/o Learned Gate	48.909
w/o DAA	48.858
Random Memory	48.558

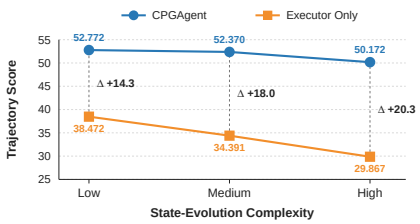


Figure 3: Trajectory performance under increasing state-evolution complexity.

We conduct component-wise ablations to examine the contribution of the Guideline Planner. As shown in Table 3, ablating or replacing key planner components consistently degrades trajectory-planning performance. Relative to the Full Planner (51.289), removing the learned Soft Gate or DAA reduces the Trajectory score to 48.909 and 48.858, respectively, while replacing learned memory selection with Random Memory further decreases performance to 48.558. These results indicate that effective trajectory planning depends jointly on learned memory selection and adaptive multi-memory fusion, rather than simply introducing additional latent memory tokens.

We further examine whether the benefit of guideline planning changes with patient-state evolution complexity. As shown in Figure 3, Executor Only degrades substantially as complexity increases, with its Trajectory score decreasing from 38.472 to 29.867, whereas CPGAgent remains comparatively stable, decreasing only from 52.772 to 50.172. Consequently, the performance gap widens from 14.300 points on Low-complexity cases to 17.979 on Medium-complexity cases and 20.305 on High-complexity cases. This trend suggests that the Guideline Planner becomes increasingly valuable when downstream decisions depend on more complex state evolution, supporting its role in maintaining coherent guideline-grounded trajectories.

5.3 How CPGs Act as Control Logic

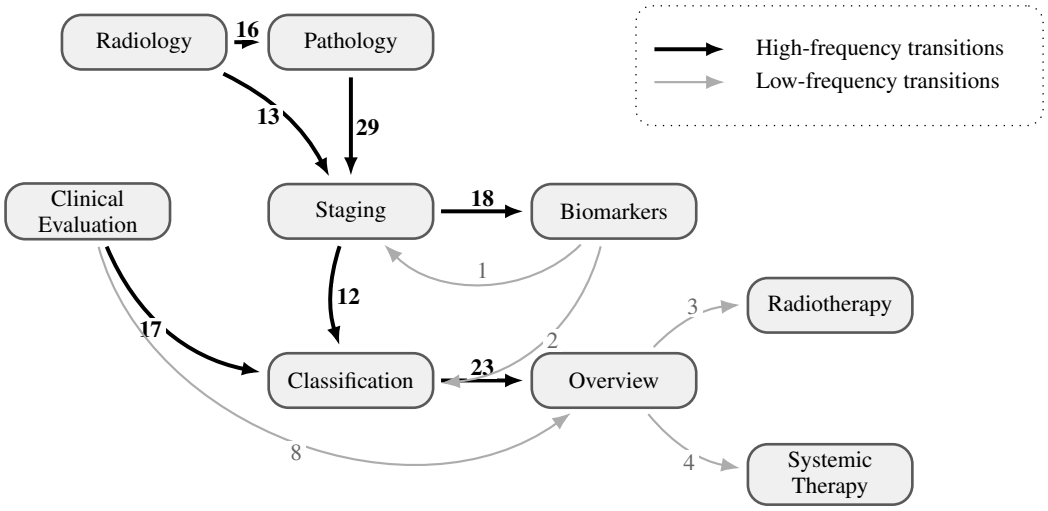


Figure 4: Memory transitions in the LUAD cohort.

CPGAgent operationalizes CPGs as control logic by dynamically activating guideline memories according to the evolving patient state. As shown in Figure 4, memory activation transitions across clinical stages such as radiology, pathology, staging, biomarkers, classification, and treatment planning, with frequent transitions reflecting common clinical pathways and less frequent ones capturing case-specific branches. In this way, activated guideline memories serve as state-dependent control signals that steer the agent through the evolving clinical trajectory.

6 Conclusion

In this work, we introduced CPGAgent, a guideline-grounded framework that shifts clinical guidelines from passive reference context to explicit control logic for multimodal sequential clinical decision-making. CPGAgent represents CPGs as latent guideline memories and dynamically activates decision-relevant knowledge according to evolving patient states, enabling guideline-consistent planning throughout the clinical trajectory. We also introduced CPGTrajBench, a multimodal executable benchmark with progressively revealed clinical evidence, together with trajectory-planning rubrics for evaluating guideline consistency across evolving patient states. Experiments across three cancer cohorts show that CPGAgent improves trajectory performance, achieving a 7.9-point overall improvement over the strongest baseline. Ablation and state-evolution analyses further demonstrate the importance of dynamic guideline-memory selection and integration, with the benefit of guideline planning becoming more pronounced as trajectory complexity increases. These results support *Guidelines as Control Logic* as a promising paradigm for clinical agents. The paradigm enables agents to adapt to evolving patient states while maintaining guideline-consistent trajectories.

AI use statement

In this work, we used generative AI tools for generate synthetic datasets, implement methods, translation, and clean / reformat datasets. We have not used generative AI tools for propose or refine hypotheses, design or provide feedback on research methodology or experiments, qualitative / thematic data analysis, and interpret results, and develop theoretical models / conceptual frameworks, formulate mathematical claims, provide key ingredients for proofs, assist in writing proofs are not applicable to this work. Additionally, we used generative AI tools for create or edit software code, sourcing/searching for information, edit a research paper to improve readability. We have reviewed all AI-assisted work. All AI-assisted code and implementations were reviewed and validated by two authors. Synthetic medical data generated with the assistance of LLMs were reviewed by two medi-

cal experts for quality assurance. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

This work uses de-identified clinical data and clinical practice guidelines for research purposes. All data handling follows applicable institutional and regulatory requirements.

Reproducibility statement

Code and data are available at https://anonymous.4open.science/r/cpgagent_anonymous-EE3D/. Implementation details for the dual-agent system, benchmark construction, and evaluation rubrics are provided in the main paper and appendix.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/960a172bc7fbf0177ccccbb411a7d800-Abstract-Conference.html.

Mohammad Almansoori, Komal Kumar, and Hisham Cholakkal. MedAgentSim: Self-evolving multi-agent simulations for realistic clinical interactions, 2025. URL https://arxiv.org/abs/2503.22678.

Anthropic. Claude opus 5. Large language model, 2026. URL https://www.anthropic.com/claude. Accessed: 2026-09-24.

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, 2025. URL https://arxiv.org/abs/2505.08775.

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, 2026. URL https://arxiv.org/abs/2604.09937.

Yixiong Chen, Xinyi Bai, Yue Pan, Zongwei Zhou, and Alan Yuille. Meissa: Multi-modal medical agentic intelligence, 2026. URL https://arxiv.org/abs/2603.09018.

Ruiqi Deng, Geoffrey Martin, Tony Wang, Gongbo Zhang, Yi Liu, Chunhua Weng, Yanshan Wang, Justin F Rousseau, and Yifan Peng. Cpgprompt: Translating clinical guidelines into llm-executable decision support, 2026. URL https://arxiv.org/abs/2601.03475.

Zhihao Fan, Jialong Tang, Wei Chen, Siyuan Wang, Zhongyu Wei, Jun Xi, Fei Huang, and Jingren Zhou. AI hospital: Benchmarking large language models in a multi-agent medical interaction simulator, 2024. URL https://arxiv.org/abs/2402.09742.

Vemund Fredriksen, Svein Ole M. Sevle, André Pedersen, Thomas Langø, Gabriel Kiss, and Frank Lindseth. Teacher-student approach for lung tumor segmentation from mixed-supervised datasets. *PLOS ONE*, 17(4):e0266147, 2022. doi: 10.1371/journal.pone.0266147. URL https://doi.org/10.1371/journal.pone.0266147.

Yicheng Gao, Xiaolin Zhou, Yahan Li, Yue Zhao, and Ruishan Liu. MedExAgent: Training llm agents to ask, examine, and diagnose in noisy clinical environments, 2026a. URL https://arxiv.org/abs/2605.07058.

Yifan Gao, Haoyue Li, Feng Yuan, Xin Gao, Weiran Huang, and Xiaosong Wang. Camyla: Scaling autonomous research in medical image segmentation, 2026b. URL https://arxiv.org/abs/2604.10696. Introduces CamylaBench.

Adam Goode, Benjamin Gilbert, Jan Harkes, Drazen Jukic, and Mahadev Satyanarayanan. OpenSlide: A vendor-neutral software foundation for digital pathology. *Journal of Pathology Informatics*, 4:27, 2013. doi: 10.4103/2153-3539.119005. URL https://doi.org/10.4103/2153-3539.119005.

Yufei He, Juncheng Liu, Zhiyuan Hu, Yulin Chen, Yue Liu, Yuan Sui, Yibo Li, Nuo Chen, Jun Hu, Bryan Hooi, Xinxing Xu, and Jiang Bian. EvoClinician: A self-evolving agent for multi-turn medical diagnosis via test-time evolutionary learning. *arXiv preprint arXiv:2601.22964*, 2026. URL https://arxiv.org/abs/2601.22964.

Rebecca Soskin Hicks, Mikhail Trofimov, Dominick Lim, Rahul K. Arora, Foivos Tsimpourlas, Preston Bowman, Michael Sharman, Chi Tong, Kavin Karthik, Arnav Dugar, Akshay Jagadeesh, Khaled Saab, Johannes Heidecke, Ashley Alexander, Nate Gross, and Karan Singhal. HealthBench professional: Evaluating large language models on real clinician chats, 2026. URL https://arxiv.org/abs/2604.27470.

Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, 2021. doi: 10.1038/s41592-020-01008-z. URL https://doi.org/10.1038/s41592-020-01008-z.

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, 2025. URL https://arxiv.org/abs/2501.14654.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams, 2020. URL https://arxiv.org/abs/2009.13081.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. doi: 10.1109/TBDATA.2019.2921572. URL https://doi.org/10.1109/TBDATA.2019.2921572.

Li et al. Medguide: Benchmarking clinical decision-making in large language models, 2025a. URL https://arxiv.org/abs/2505.11613.

Binbin Li, Tianxin Meng, Xiaoming Shi, Jie Zhai, and Tong Ruan. Meddm:llm-executable clinical guidance tree for clinical decision-making, 2023a. URL https://arxiv.org/abs/2312.02441.

Binxu Li, Tiankai Yan, Yuanting Pan, Jie Luo, Ruiyang Ji, Jiayuan Ding, Zhe Xu, Shilong Liu, Haoyu Dong, Zihao Lin, and Yixin Wang. MMedAgent: Learning to use medical tools with multi-modal agent. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 8745–8760. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.510. URL https://aclanthology.org/2024.findings-emnlp.510/.

Dongchen Li, Jitao Liang, Wei Li, Xiaoyu Wang, Longbing Cao, and Kun Yu. Clicare: Grounding large language models in clinical guidelines for decision support over longitudinal cancer electronic health records. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31554–31562, 2026. doi: 10.1609/aaai.v40i37.40421. URL https://doi.org/10.1609/aaai.v40i37.40421.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023b. URL https://proceedings.mlr.press/v202/li23q.html.

Wenhao Li, Hongkuan Zhang, Hongwei Zhang, Zhengxu Li, Zengjie Dong, Yafan Chen, Niranjan Bidargaddi, and Hong Liu. Refine medical diagnosis using generation augmented retrieval and clinical practice guidelines. *IEEE Journal of Biomedical and Health Informatics*, 2025b. doi: 10.1109/JBHI.2025.3641931. URL https://doi.org/10.1109/JBHI.2025.3641931.

Liang et al. Evaluating large language models in dynamic clinical decision-making with standardized patient cases, 2026. URL https://arxiv.org/abs/2606.05112.

Yusheng Liao, Shuyang Jiang, Yanfeng Wang, and Yu Wang. Reflectool: Towards reflection-aware tool-augmented clinical agents, 2024. URL https://arxiv.org/abs/2410.17657. Introduces ClinicalAgent Bench (CAB).

Junqi Liu, Selena Song, Yuhan Wang, Jiawei Mao, Hardy Chen, Xiaoke Huang, Tianhao Qi, Pengfei Guo, Yucheng Tang, Yufan He, Can Zhao, Andriy Myronenko, Dong Yang, Daguang Xu, and Yuyin Zhou. AutoMedBench: Towards medical AutoResearch with agentic AI models, 2026a. URL https://arxiv.org/abs/2606.01961.

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, 2026b. URL https://arxiv.org/abs/2606.31179.

Ruoqi 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, 2026c. URL https://arxiv.org/abs/2605.02240.

Ming Y. Lu, Bowen Chen, Drew F. K. Williamson, Richard J. Chen, Ivy Liang, Tong Ding, Guillaume Jaume, Igor Odintsov, Long Phi Le, Georg Gerber, Anil V. Parwani, Andrew Zhang, and Faisal Mahmood. A visual-language foundation model for computational pathology. *Nature Medicine*, 30(3):863–874, 2024. doi: 10.1038/s41591-024-02856-4. URL https://doi.org/10.1038/s41591-024-02856-4.

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

OpenAI. GPT-5.5. https://developers.openai.com/api/docs/models/gpt-5.5, 2026a. URL https://developers.openai.com/api/docs/models/gpt-5.5. Accessed: 2026-09-19.

OpenAI. GPT-5.6 Sol. https://developers.openai.com/api/docs/models/gpt-5.6-sol, 2026b. URL https://developers.openai.com/api/docs/models/gpt-5.6-sol. Accessed: 2026-09-19.

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, 2024. URL https://arxiv.org/abs/2405.07960.

Sahel Sharifymoghaddam, Shivani Upadhyay, Wenhu Chen, and Jimmy Lin. Unirag: Universal retrieval augmentation for large vision language models, 2025. URL https://arxiv.org/abs/2405.10311.

Shen et al. Medguidex: Internalizing decision logic from executable guidelines into large language models for clinical reasoning, 2026. URL https://arxiv.org/abs/2605.26567.

Tan et al. A decade-scale benchmark evaluating LLMs' clinical practice guidelines detection and adherence in multi-turn conversations, 2026. URL https://arxiv.org/abs/2603.25196.

Alyssa Unell, Noel C. F. Codella, Sam Preston, Peniel Argaw, Wen wai Yim, Zelalem Gero, Cliff Wong, Rajesh Jena, Eric Horvitz, Amanda K. Hall, Ruican Rachel Zhong, Jiachen Li, Shrey Jain, Mu Wei, Matthew Lungren, and Hoifung Poon. Cancerguide: Cancer guideline understanding via internal disagreement estimation, 2025. URL https://arxiv.org/abs/2509.07325.

Kiril Vasilev, Alexandre Misrahi, Eeshaan Jain, Phil F. Cheng, Petros Liakopoulos, Olivier Michielin, Michael Moor, and Charlotte Bunne. MTBBench: A multimodal sequential clinical decision-making benchmark in oncology, 2025. URL https://arxiv.org/abs/2511.20490.

Ziyue Wang, Junde Wu, Linghan Cai, Chang Han Low, Xihong Yang, Qiaxuan Li, and Yueming Jin. MedAgent-Pro: Towards evidence-based multi-modal medical diagnosis via reasoning agentic workflow. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZOuU0udyA4.

Peng Xia, Kangyu Zhu, Haoran Li, Tianze Wang, Weijia Shi, Sheng Wang, Linjun Zhang, James Y Zou, and Huaxiu Yao. Mmed-rag: Versatile multimodal rag system for medical vision language models. In *International Conference on Learning Representations*, volume 2025, pp. 66188–66217, 2025. URL https://openreview.net/forum?id=sSC7ADZhaX.

Ran Xu, Yuchen Zhuang, Yishan Zhong, Yue Yu, Zifeng Wang, Xiangru Tang, Hang Wu, May D. Wang, Peifeng Ruan, Donghan Yang, Tao Wang, Guanghua Xiao, Xin Liu, Carl Yang, Yang Xie, and Wenqi Shi. MedAgentGym: A scalable agentic training environment for code-centric reasoning in biomedical data science, 2025. URL https://arxiv.org/abs/2506.04405.

Peiru Yang, Yudong Li, Shiting Wang, Xin-Yi Liu, Haotian Gan, Xin Li, Qingyu Gao, and Yongfeng Huang. OncoCoT: A temporal-causal chain-of-thought dataset for oncologic decision-making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34277–34285, 2026. doi: 10.1609/aaai.v40i40.40724. URL https://doi.org/10.1609/aaai.v40i40.40724.

Yakun Zhu, Zhongzhen Huang, Qianhan Feng, Linjie Mu, Yannian Gu, Shaoting Zhang, Qi Dou, and Xiaofan Zhang. CP-Env: Evaluating large language models on clinical pathways in a controllable hospital environment, 2025a. URL https://arxiv.org/abs/2512.10206.

Yinghao Zhu, Ziyi He, Haoran Hu, Xiaochen Zheng, Xichen Zhang, Zixiang Wang, Junyi Gao, Liantao Ma, and Lequan Yu. MedAgentBoard: Benchmarking multi-agent collaboration with conventional methods for diverse medical tasks, 2025b. URL https://arxiv.org/abs/2505.12371.

A Guideline Planner Training Details

The Guideline Planner is trained in multiple stages to decouple *guideline representation learning*, *action generation*, and *dynamic memory selection*. This staged training avoids directly coupling the latent guideline memory space to downstream action-generation errors and allows each component to learn its intended functionality before joint optimization.

Stage 1: Latent guideline memory learning. We first train the Memory Encoder independently from the downstream Planner Decoder. The encoder is optimized using three memory-oriented objectives: `AE`, `RETRIEVE`, and `CONTINUE`, sampled with a ratio of 5:5:3. The Memory Encoder uses a dedicated LoRA adapter and does not receive downstream planning losses during this stage. These objectives encourage the encoder to learn reusable latent representations of guideline content rather than adapting the memory space directly to a specific downstream action-generation objective. After training, all decision-oriented guideline chunks are encoded offline into latent memory slots and retrieval keys. The resulting Memory Store and Memory Encoder are then frozen during subsequent planner training.

Stage 2: Planner Decoder training. We next train the Planner Decoder using annotated positive guideline memories rather than memories predicted by the Retriever and Soft Gate. For each trajectory transition, strong and weak positive guideline memories are aggregated into a fixed 64-slot latent representation and supplied to the Decoder together with the current Patient State and recent trajectory history. Under teacher forcing, the Decoder is trained to generate the clinically accepted next action from the correct guideline evidence. This stage isolates action-generation learning from errors introduced by dynamic retrieval and memory selection. The Decoder is optimized with

$$\begin{aligned}\mathcal{L}_{\text{decoder}} = 1.00\,\mathcal{L}_{\text{SFT}} + 0.30\,\mathcal{L}_{\text{pref}} + 0.20\,\mathcal{L}_{\text{progress}} \\ + 0.10\,\mathcal{L}_{\text{phase}} + 0.20\,\mathcal{L}_{\text{prov}}^{\text{dec}}.\end{aligned} \tag{13}$$

Here, $\mathcal{L}_{\text{SFT}}$ supervises structured clinical action generation; $\mathcal{L}_{\text{pref}}$ encourages preference for accepted actions over constructed negative alternatives; $\mathcal{L}_{\text{progress}}$ supervises prediction of patient-state progression; $\mathcal{L}_{\text{phase}}$ supervises the current clinical phase; and $\mathcal{L}_{\text{prov}}^{\text{dec}}$ encourages the Decoder representation to preserve the provenance of supporting guideline evidence.

Stage 3: Dynamic memory selection and fusion. After the Decoder has learned action generation from positive guideline evidence, we train the Retriever, Soft Gate, and Dynamic Anchor Attention modules to recover and fuse appropriate guideline memories from the current Patient State and recent trajectory history. Memory selection is supervised using guideline provenance annotations from human-reviewed clinical trajectories. Positive memories correspond to guideline evidence supporting the accepted clinical action, while negative memories provide supervision for distinguishing semantically related but decision-inappropriate guideline content. The memory-selection modules are optimized with

$$\begin{aligned}\mathcal{L}_{\text{gate}} = 1.00\,\mathcal{L}_{\text{action}} + 0.30\,\mathcal{L}_{\text{retrieval}} + 0.20\,\mathcal{L}_{\text{prov}}^{\text{gate}} \\ + 0.10\,\mathcal{L}_{\text{margin}} + 0.01\,\mathcal{L}_{\text{sparse}}.\end{aligned} \tag{14}$$

Here, $\mathcal{L}_{\text{action}}$ measures whether the selected and fused guideline memories support the correct downstream clinical action; $\mathcal{L}_{\text{retrieval}}$ encourages retrieval of guideline memories relevant to the current patient state; $\mathcal{L}_{\text{prov}}^{\text{gate}}$ aligns memory-selection predictions with annotated guideline provenance; $\mathcal{L}_{\text{margin}}$ encourages provenance-supported memories to receive higher gate scores than negative candidates; and $\mathcal{L}_{\text{sparse}}$ regularizes memory activation to discourage diffuse selection over unrelated guideline memories.

Stage 4: Joint fine-tuning. Finally, the Planner Decoder and dynamic memory-selection modules are jointly fine-tuned at a reduced learning rate. During this stage, the Memory Encoder and precomputed guideline Memory Store remain frozen, while the Decoder, Retriever, Soft Gate, and DAA are allowed to adapt jointly. This final stage exposes the Decoder to dynamically retrieved and fused guideline memories, allowing action generation to adapt to the distribution of evidence produced by the learned memory-selection mechanism while preserving the latent guideline representation learned in the first stage.

B Clinical Tools

Medical Executor accesses heterogeneous clinical evidence through standardized tools rather than directly reading raw case files. These tools expose structured clinical findings and selected multimodal artifacts while retaining source provenance, allowing evidence acquired during execution to be incorporated into the Patient State in a controlled manner.

Radiology tools. For lung adenocarcinoma (LUAD), the Lung-CT tool operates on preprocessed CT volumes resampled to approximately 1-mm isotropic spacing. Lung and tumor regions are segmented using LungTumorMask v1.3.1 (Fredriksen et al., 2022), and representative tumor-centered slices, contextual images, and segmentation overlays are returned to the agent. For endometrial carcinoma (UCEC), the UCEC-MRI tool processes T2-weighted and diffusion-weighted MRI, with nnU-Net v2 (Isensee et al., 2021) used offline to generate standardized tumor regions for downstream review. For nasopharyngeal carcinoma (NPC), the NPC-MRI tool extracts representative contrast-enhanced T1 MRI regions based on available primary-tumor or lymph-node masks. Segmentation and volume-level preprocessing are performed offline, while the benchmark runtime exposes only clinically relevant review images and structured spatial information to the agent.

Pathology tools. The pathology tools use CONCH v1. (Lu et al., 2024) for prompt-guided retrieval of representative regions from whole-slide images accessed through OpenSlide (Goode et al., 2013). Patch embeddings and prompt similarities are computed offline, and the runtime returns top-ranked pathology patches together with slide context and metadata. Cancer-specific wrappers use morphology prompts tailored to lung cancer, UCEC, and NPC while sharing the same retrieval workflow.

Structured clinical and guideline tools. Additional tools expose bounded views of clinical summaries, pathology reports, treatment history, follow-up records, imaging manifests, and molecular biomarkers. Guideline retrieval searches version-matched guideline sections using cached text representations and retrieval scores, with FAISS-based indexing where applicable (Johnson et al., 2019). All retrieved evidence is returned with source provenance and can subsequently update the Patient State.

C Patient State Machine Template

The Patient State Machine is represented as a structured object that maintains the current clinical state, guideline context, acquired evidence, completed actions, and unresolved information required for subsequent planning. The patient state machine template is shown below.

Patient State Machine Template

```
{
  case_id: TEMPLATE_CASE_ID,
  cancer_type: nsclc,
  cancer_family: lung,
  disease_subtype: nsclc,
  current_phase: diagnostic_workup,
  guideline_context: {
    guidelines: [
      {
        guideline_id: NSCLC_2010,
        version: 2010,
        effective_from: 2010-01-01,
        effective_to: 2010-12-31
      }
    ]
  },
  known_diagnosis: null,
  known_stage: null,
  known_biomarkers: {},
  risk_stratification: {},
  radiology_summary: null,
  pathology_summary: null,
  molecular_summary: null,
  guideline_summary: null,
  available_modalities: {
    clinical: true,
    reports: true,
    radiology: true,
    pathology: true,
    molecular: true,
    guideline: true,
    evaluation: false
  },
  completed_skills: [],
  completed_actions: [],
  previous_actions: [],
  pending_actions: [],
  blocked_actions: [],
  blocked_pathways: [],
  treatment_history: [],
  current_treatment_line: null,
  selected_treatment: null,
  evidence_ledger: [],
  evidence_refs: [],
  confirmed_evidence: {},
  missing_information: [],
  unresolved_information: [],
  warnings: [],
```

```
    last_transition: null,
    state_update_summary: {},
    free_text_summary: null
  }
```

D CPGTRAJBENCH TRAJECTORY-PLANNING EVALUATION DETAILS

D.1 RUBRIC-GROUNDED TRAJECTORY ALIGNMENT

Agent trajectories may differ from the trajectory-planning rubric in the number and positions of decision steps, making direct step-wise comparison inappropriate. We therefore align rubric step $D_i = (S_i, A_i)$ and agent step $Z_j = (S_j^{\text{agent}}, \hat{A}_j)$ using the compatibility function defined in Eq. 11.

The alignment jointly considers patient-state compatibility and clinical action correspondence. We use an order-preserving one-to-one alignment, where each rubric step and each agent step can be matched at most once while maintaining their temporal order. The optimal alignment is obtained as

$$\mathcal{M}^* = \arg\max_{\mathcal{M}} \left[\sum_{(i,j)\in\mathcal{M}} M(D_i, Z_j) - \lambda_{\text{gap}} \, \text{Gap}(\mathcal{M}) \right], \tag{15}$$

where $\mathcal{M}$ denotes a set of order-preserving one-to-one rubric–agent correspondences. The alignment gap is defined as

$$\text{Gap}(\mathcal{M}) = \frac{|\mathcal{U}_{\text{miss}}(\mathcal{M})|}{N_R} + \frac{|\mathcal{U}_{\text{red}}(\mathcal{M})|}{N_A}, \tag{16}$$

where $\mathcal{U}_{\text{miss}}(\mathcal{M})$ denotes missed rubric steps and $\mathcal{U}_{\text{red}}(\mathcal{M})$ denotes redundant agent steps, while N_R and N_A are the total numbers of rubric and agent steps, respectively. The gap term serves only as an alignment regularizer, discouraging alignments with excessive missed rubric steps or redundant agent steps. The resulting $\mathcal{M}^*$ is used for subsequent per-step and trajectory-level scoring.

D.2 ACTION-SET BASED SCORING

For each aligned pair $(i, j) \in \mathcal{M}^*$, the agent action $\hat{A}_j$ is evaluated against the guideline-supported action set A_i defined in Eq. 10. The per-step decision score is

$$\begin{aligned} \text{StepScore}_{ij} = w_1 \ \text{Cover}\Big(\hat{A}_j, A_i^{\text{req}}\Big) + w_2 \ \text{Accept}\Big(\hat{A}_j, A_i^{\text{acc}}\Big) \\ + w_3 \ \text{CondAware}\Big(\hat{A}_j, A_i^{\text{cond}}\Big) - w_4 \ \text{UnsafeHit}\Big(\hat{A}_j, A_i^{\text{unsafe}}\Big). \end{aligned} \tag{17}$$

Required-action coverage measures whether actions expected under the current patient state are adequately addressed. Acceptable-action credit captures clinically valid but non-mandatory alternatives. Conditional-action awareness evaluates whether an action is proposed together with its required clinical prerequisites. An action that is valid only conditionally but is recommended without satisfying or stating those prerequisites is treated as a state violation. UnsafeHit penalizes actions that contradict guideline constraints, ignore required evidence, or are otherwise inappropriate under the current patient state.

D.3 TRAJECTORY-LEVEL SCORING

The trajectory-level score in Eq. 12 aggregates the per-step scores over the optimal alignment $\mathcal{M}^*$ together with trajectory-level penalties.

Let $\mathcal{U}_{\text{miss}}^{\text{req}}(\mathcal{M}^*)$ denote missed rubric steps containing required actions, and let $\mathcal{U}_{\text{red}}(\mathcal{M}^*)$ denote redundant agent steps. We define

$$r_{\text{miss}} = \frac{|\mathcal{U}_{\text{miss}}^{\text{req}}(\mathcal{M}^*)|}{N_R^{\text{req}}}, \qquad r_{\text{red}} = \frac{|\mathcal{U}_{\text{red}}(\mathcal{M}^*)|}{N_A}, \tag{18}$$

where N_R^{req} is the total number of rubric steps containing required actions and N_A is the total number of agent steps. Thus, r_{miss} measures the proportion of missed required rubric steps, while r_{red} measures the proportion of redundant agent steps.

This formulation evaluates whether an agent produces a clinically admissible and guideline-consistent trajectory rather than whether it exactly reproduces a single reference sequence.

E Implementation Details

System configuration. CPGAgent consists of a Qwen3.5-9B Guideline Planner and a Medical Executor execution agent instantiated with GPT-5.6-sol. The planner receives the current Patient State and guideline memories and outputs a structured plan, while Medical Executor performs multimodal tool use and updates the observable state. We allow at most four planner rounds per case and 12 tool-interaction rounds within each planner round. The Executor Only baseline uses the same GPT-5.6-sol model, tools, case inputs, and interaction budget but removes the Guideline Planner.

Guideline memory. The guideline corpus contains 79 chunks from four versioned guidelines: NSCLC 2010, SCLC 2010, CSCO Endometrial Cancer 2023, and CSCO Nasopharyngeal Carcinoma 2022. A Qwen3.5-9B Memory Encoder is trained for 5,000 steps using autoencoding, retrieval, and continuation objectives with a 5:5:3 sampling ratio, a learning rate of 2×10^{-4}, BF16 precision, and gradient accumulation of four. The trained encoder is frozen and exports 64 latent slots for each guideline memory.

Runtime and ablations. During execution, tool observations are incorporated into the Patient State and passed back to the planner for the next decision round. All ablation variants share the same frozen Memory Encoder, guideline memory store, execution agent, and tool interface, and differ only in the planner-side memory-selection or fusion mechanism. Training is performed on a single 80-GB A100 GPU.

Evaluation. All methods use the same case rubrics, reference trajectories, and evaluation protocol. All outputs are scored by the same Codex GPT-5.6-sol judge under the high reasoning setting.

F Trajectory Generator Prompt

The trajectory-planning rubrics in CPGTrajBench are constructed using large language models. Both the trajectory generator π_{traj} and the verifier V are implemented with `deepseek-chat` using a temperature of 0.2. They are invoked as independent LLM calls and do not share conversation history. At trajectory step t, π_{traj} receives only the current patient state S_t and the guideline slices associated with the current decision stage, and generates the next structured action set.

F.1 Generator System Prompt

System Prompt

```
You are AI-1 (Generator) in a dual-LM guideline-grounded trajectory construction
system.

Hard constraints:
- You ONLY see the current patient state S_t and the provided guideline slices (NCCN
Discussion or CSCO modules).
- You must NOT assume access to the full patient report.
- Do NOT invent lab values, stages, biomarkers, or treatments that are not present in
S_t.
- Prefer action sets over a single sentence.
- Separate mandatory, acceptable, conditional, and unsafe actions.
- If key evidence is missing in S_t, put dependent therapy decisions into conditional
actions and list open questions.
- Ground recommendations in the provided guideline slices. If slices are placeholders,
still propose clinically sensible next steps and mark uncertainty.

Return ONLY a JSON object with this schema:
```

```
{
  "phase": "string",
  "reasoning_summary": "short",
  "action_set": {
    "required": [
      {
        "action_id": "a0",
        "action_type": "staging|pathology|molecular_testing|treatment_planning|fitnes⌋
        s|surveillance|communication",
        "action": "concrete next clinical action",
        "priority": "high|medium|low",
        "guideline_basis": "which Discussion idea supports this",
        "required_evidence": ["..."],
        "rationale": "why now"
      }
    ],
    "acceptable": [],
    "conditional": [
      {
        "action_id": "c0",
        "action_type": "...",
        "action": "...",
        "condition": "if ...",
        "priority": "medium",
        "guideline_basis": "...",
        "required_evidence": ["..."],
        "rationale": "..."
      }
    ],
    "unsafe": [
      {
        "action_id": "u0",
        "action_type": "...",
        "action": "action that must NOT be done now",
        "priority": "high",
        "guideline_basis": "...",
        "rationale": "why premature or contraindicated"
      }
    ]
  },
  "open_questions": ["..."],
  "stop": false
}

Set stop=true only when S_t already supports a complete guideline-aligned management
plan and surveillance recommendation, and no further workup is needed.
```

F.2 Generator User Message

At each step, the Generator user message provides the current patient state S_t together with the guideline slice corresponding to the current decision phase.

User Message

```
case_id: CASE_ID
step: STEP
suggested_phase: PHASE

## Current patient state S_t (ONLY information available to you)
STATE_JSON

## Guideline slices for this phase
GUIDELINE_SLICE

Generate the next structured action_set for this step.
```

F.3 Verifier Prompt

The verifier V is invoked independently after trajectory generation. It receives the proposed action set and the patient-report slice visible at the current decision stage. It verifies the proposed actions and extracts only report-supported information into `state_update`, which is subsequently used to revise the patient state.

F.3.1 VERIFIER SYSTEM PROMPT

System Prompt

```
You are AI-2 (Verifier) in a dual-LM guideline-grounded trajectory construction system.

Your job:
- Compare Planner next actions against the patient report slice visible at the current
phase.
- Decide for each important action whether it is supported, unobserved, uncertain, or
contradicted/premature.
- Extract ONLY facts that are explicitly supported by the visible report slice into
state_update.
- Never fabricate missing tests or results.
- If the report slice does not mention something the guideline recommends, classify as
guideline_supported_but_unobserved and do NOT invent it into state_update.
- Do NOT release pathology, staging, molecular, treatment, or recurrence facts before
the report slice for this phase contains them.
- You do NOT replan the whole pathway; you verify and extract.

Return ONLY a JSON object:
{
  "verifications": [
    {
      "action_id": "a0",
      "action": "...",
      "classification": "supported_by_report_and_guideline|guideline_supported_but_un⌋
      observed|observed_but_guideline_uncertain|contradicted_or_premature",
      "report_support": "supported|not_observed|contradicted",
      "guideline_support": "supported|uncertain|contradicted",
      "evidence_spans": ["exact or near-exact quotes from report"],
      "notes": "short"
    }
  ],
  "state_update": {
    "histology": null,
    "pathologic_stage": null,
    "tnm": null,
    "surgery": null,
    "margin_or_residual_status": null,
    "ECOG": null,
    "molecular_status": null,
    "treatment_plan": null,
    "observed_treatment": null,
    "recurrence_or_metastasis": null
  },
  "completed_actions": [
    {
      "action": "...",
      "status": "completed|partially_supported|unobserved",
      "evidence": "quote or empty"
    }
  ],
  "open_questions": ["remaining missing items"],
  "step_summary": "one sentence describing what was verified this step",
  "continue": true
}

Rules for state_update:
- Include a field only when the report explicitly supports it.
- Use null for unknown fields (they will be ignored by the merger).
- Prefer concise normalized values (e.g., "lung adenocarcinoma", "IB, pT2a pN0", "R0").

Set continue=false only when the current phase is surveillance/recurrence or final
recommendation AND the visible report slice already supports a complete management
plan for that phase.
```

F.3.2 VERIFIER USER MESSAGE

At each step, the Verifier receives the Planner-generated action set and only the patient-report slice visible at the current decision stage. This ensures that verification and state updates are restricted to temporally available evidence.

User Message

```
case_id: CASE_ID
step: STEP
phase: PHASE
visible_report_sections: VISIBLE_SECTIONS

## Planner proposed next actions A_t
PLAN_JSON

## Patient report slice visible at this phase (for verification ONLY)
REPORT_SLICE

Verify actions against ONLY this report slice. Do not use facts from later timepoints
that are not present here. Extract only report-supported facts appropriate for the
current phase into state_update.
```