Incentivizing Agentic Retrieval for Disease-Centric Clinical Case Search via Trajectory Memory

Jie Lin[1*] **Xiang Liu**[2*] **Lihao Liu**[3] **Liansheng Wang**[1,4†]
[1]Department of Computer Science, School of Informatics, Xiamen University
[2]Department of Artificial Intelligence, School of Informatics, Xiamen University
[3]AWS Agentic AI
[4]National Institute for Data Science in Health and Medicine, Xiamen University

https://github.com/jaylinio/PatientSearch

Abstract

Clinical case search aims to retrieve disease-consistent prior cases from electronic health record databases for a natural-language patient query under a retrieval objective. Patient queries often provide incomplete clinical evidence, requiring retrieval policies to adapt to both the available evidence and prior search trajectories. Existing clinical retrieval methods typically flatten the query into a single representation or use static retrieval pipelines, while explicit mechanisms for cross-case adaptation remain an emerging research direction. **PatientSearch-SE** is a self-evolving framework for disease-centric clinical case search. It treats each query as a partially observed patient state, matching reliable evidence directly, completing consequential gaps with evidence-grounded hypotheses, and distilling successful search trajectories into reusable memory. This accumulated experience allows the framework to adapt its retrieval behavior to the information available in each case and guide subsequent searches without updating model parameters. The evaluation combines disease-centric retrieval outcomes with trajectory-level process probes for retrieval behavior analysis. Experiments on PMC-Patients and MIMIC-IV show higher disease-centric retrieval metrics for PatientSearch-SE than recent retrieval baselines across both datasets. Its ablation and memory-accumulation results provide evidence for the contribution of external memory to retrieval, and its retrieved cases provide additional evidence for diagnosis-masked downstream few-shot RAG prediction under this protocol.

1 Introduction

Clinical case search supports clinical decision support, case-based reasoning, medical education, and cohort discovery because similar past cases can surface disease-consistent evidence and related clinical context [1–3]. In this paper, we study *disease-centric* case search, where retrieved cases should match the target query at the disease level, while the full patient state still informs matching and re-ranking. This setting has three practical difficulties. First, a clinical query is *multi-dimensional*: demographics, symptoms, laboratory findings, diagnoses, treatments, outcomes, and timelines appear in different formats and should not all be matched with one retrieval primitive. Second, real queries are often *underspecified*. A physician may describe symptoms and demographics but omit laboratory values, treatment history, or the diagnosis itself, so the search system must handle missing evidence rather than only reward literal overlap. Third, retrieval can benefit from *reusable search experience*. Completed trajectories reveal which hypotheses helped, which tools added little, and which execution

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

patterns recovered disease-consistent cases. Without cross-case adaptation, these signals cannot inform subsequent searches.

Together, these challenges motivate a retrieval framework that can represent heterogeneous clinical evidence, handle underspecified queries, and reuse search experience across cases. Traditional sparse retrieval, dense retrieval, and similar-patient cohorting methods often flatten the query into a largely uniform matching problem, which obscures the heterogeneous structure of clinical evidence [1, 4–8]. Biomedical encoders such as ClinicalBERT [9] and PubMedBERT [10] improve semantic matching, but they still apply mostly static retrieval logic and do not distinguish well-specified dimensions from underspecified ones. Agentic search systems have recently shown that tool use and trajectory optimization can improve complex search behavior [11–14], yet these methods target general web search rather than clinically grounded case retrieval. Conversely, clinical reasoning and external-memory medical systems [15–17] suggest the value of medical priors, memory, and trajectory feedback, but they do not explicitly model per-dimension information sufficiency or connect memory reuse to adaptive clinical retrieval under a disease-centric objective. Patient-level representation learning, contrastive similar-patient retrieval, clinical concept extraction, and graph-augmented medical search also preserve structure that flat document retrieval loses [5–8, 2, 4]. These methods are complementary to our setting, but many assume longitudinal visits, diagnosis-code sequences, or patient graphs as native input. These considerations point toward a clinical case search setting in which heterogeneous patient evidence is represented explicitly, underspecified dimensions are handled according to their evidential status, and retrieval policies can adapt across cases.

PatientSearch-SE is a clinically grounded framework for disease-centric clinical case search. It moves toward this setting through three design choices.

1. **Sufficiency-aware planning.** We convert a natural-language clinical query into a structured reasoning state by classifying each of seven clinical dimensions as information-sufficient or information-insufficient. Sufficient dimensions support direct evidence matching. Insufficient dimensions trigger targeted hypothesis completion.
2. **Hierarchical tool orchestration.** We organize retrieval into atomic tools, composite clinical tools, and adaptive tool chains, then use sufficiency states and retrieval intent to control candidate generation and re-ranking. Well-specified queries use direct evidence matching, while incomplete queries use hypothesis-guided retrieval.
3. **Trajectory memory consolidation.** We use a no-parameter-update external memory that evaluates search trajectories with a disease-centric objective and six retrieval-process indicators, then stores high-quality experience as disease cards and strategy templates.

Accordingly, the paper evaluates how explicit sufficiency modeling, hierarchical retrieval, and trajectory memory shape disease-centric search under heterogeneous and incomplete queries. Component ablations, memory-accumulation analyses, and diagnosis-masked downstream evaluation separate the contributions of structured representations, adaptive retrieval decisions, and memory reuse.

2 Method: PatientSearch-SE

Given a patient record database $\mathcal{R}$ and a natural-language clinical query q, PatientSearch-SE returns a Top-K list of disease-consistent prior cases. Relevance is defined at the disease or disease-family level using International Classification of Diseases (ICD) codes and Medical Subject Headings (MeSH) descriptors. Non-disease fields such as demographics, symptoms, tests, treatments, outcomes, and timelines influence matching and re-ranking, but they are not treated as claims of full clinical equivalence. The query-side target label is not exposed to the planner or searcher at inference time. Full evaluation definitions are provided in Appendix B.

PatientSearch-SE has three coupled modules: a *sufficiency-aware planner* (Section 2.1), a *hierarchical tool orchestration* system (Section 2.2), and a *trajectory memory consolidation* layer (Section 2.3). This decomposition mirrors the retrieval-control problem: decide which evidence is reliable, choose matching actions that fit that evidence, and retain reusable search behavior across cases. Given a free-form clinical query q, the method first constructs a structured patient state $\mathcal{P}(q) = \{x_k\}_{k\in\mathcal{K}}$ over the seven clinical dimensions defined in Eq. 1.

$$\mathcal{K} = \{\texttt{demo}, \texttt{sym}, \texttt{lab}, \texttt{diag}, \texttt{treat}, \texttt{out}, \texttt{time}\}, \tag{1}$$

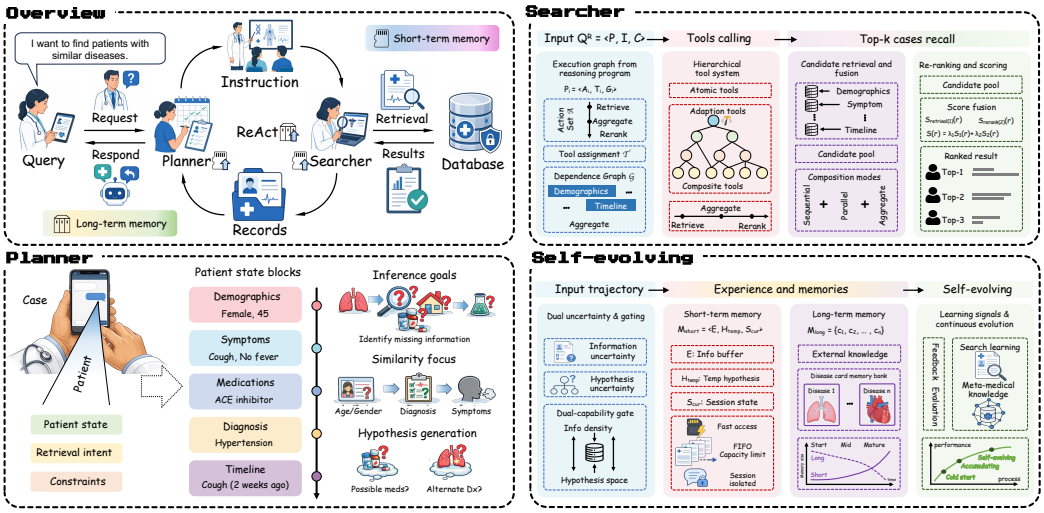


Figure 1: Overview of PatientSearch-SE. The planner converts a clinical query into a structured retrieval program with per-dimension sufficiency states. The tool orchestration layer selects and composes retrieval tools based on these state signals for retrospective disease-centric case search. Ground-truth disease labels are used only for evaluation and leakage-controlled memory consolidation, not for inference-time retrieval. The memory module scores trajectories and writes reusable abstractions to external memory without parameter updates.

where each x_k denotes the evidence extracted for dimension k. The planner then produces a sufficiency profile $\mathbf{z} \in \{0,1\}^{|\mathcal{K}|}$, an intent variable $\mathcal{I}$, and a dimension-wise constraint set $\mathcal{C} = \{c_k\}_{k\in\mathcal{K}}$. The orchestration layer maps $(\mathcal{P}, \mathbf{z}, \mathcal{I}, \mathcal{C})$ to an executable retrieval graph, and the learning layer consolidates high-quality trajectories into reusable external memory. The benchmark objective is intentionally disease-centric: non-disease dimensions affect retrieval and re-ranking, but success is still measured at the disease-family level rather than as full clinical equivalence. We defer the formal disease-level evaluation target to Appendix B, but keep the method-side notation explicit in the main text because the three modules share the same state variables. Each execution trajectory is written as:

$$\tau = \big(q, \mathcal{P}, \mathbf{z}, \mathcal{I}, \mathcal{C}, \mathcal{G}, \mathcal{L}_K\big), \tag{2}$$

where $\mathcal{G}$ is the execution graph and $\mathcal{L}_K$ is the final Top-K ranked case list. This trajectory representation is given in Eq. 2. Figure 1 summarizes the overall architecture.

Operationally, PatientSearch-SE follows five steps: it extracts the seven-dimensional patient state from the query, labels each dimension as sufficient or insufficient for direct matching, compiles these labels into retrieval constraints, retrieves and re-ranks cases with dimension-specific tools, and then consolidates only high-quality trajectories into external memory after evaluation-time leakage filtering. This summary is meant to make the execution path explicit before introducing the module-level details below.

2.1 Planner: Sufficiency-aware planning

The planner converts a natural language query q into a structured reasoning program with three outputs: the patient state $\mathcal{P}(q) = \{x_k\}_{k\in\mathcal{K}}$, the inferred retrieval intent $\mathcal{I}$, and a constraint set $\mathcal{C}$ that later instantiates the tool orchestration plan. This planning step matters because disease-centric clinical retrieval can fail before search begins if the system treats unreliable or missing query fields as firm evidence. The seven dimensions in $\mathcal{K}$ are the basic planning units because different parts of a clinical query are rarely specified with equal reliability. The appendix gives the full attribute inventory. The main method only uses the dimension-level abstraction.

For each clinical dimension $k \in \mathcal{K}$, the planner predicts a binary *information sufficiency state* according to Eq. 3.

$$z_k = \mathbb{I}[s_k(x_k, q) \geq \gamma_k] \in \{0,1\}, \tag{3}$$

where $s_k(\cdot)$ is a dimension-specific sufficiency assessor and γ_k is its decision threshold. Here, $z_k = 1$ means the observed evidence is sufficient for direct matching, while $z_k = 0$ marks the dimension as underspecified. The planner stacks these states into the sufficiency profile $\mathbf{z} = (z_k)_{k \in \mathcal{K}}$. The formulation captures an asymmetry in disease-centric case search: some dimensions can be matched directly, while others should only be completed hypothetically. PatientSearch-SE therefore treats sufficiency as an explicit control variable rather than a latent by-product of text encoding.

In our implementation, s_k is instantiated as a prompted model confidence score. The prompt receives the original clinical query, the extracted patient-state field x_k, and, when available, leakage-controlled memory priors whose applicability metadata match the current sufficiency profile and retrieval intent. It returns one confidence score in $[0, 1]$ and a short evidence-grounded justification for each dimension. The thresholds γ_k are fixed before evaluation and listed in Appendix J. Memory priors may affect whether observed evidence is judged discriminative, but they are not allowed to fill absent fields or reveal held-out target labels.

Conditioned on $\mathcal{P}(q)$ and $\mathbf{z}$, the planner infers the retrieval intent $\mathcal{I}$, which may correspond to diagnostic confirmation, differential diagnosis, treatment selection, prognosis estimation, or case similarity exploration. The inferred intent determines which missing dimensions should be completed and how retrieval should balance direct evidence against inferred evidence. For each dimension with $z_k = 0$, the planner generates a hypothesis set according to Eq. 4.

$$\mathcal{H}_k = \textsc{Hypothesize}_k(\{x_j : z_j = 1\}_{j \in \mathcal{K}}, \mathcal{I}), \tag{4}$$

so hypothesis generation is explicitly conditioned on the reliable dimensions rather than on the raw query alone. For example, when symptom and demographic evidence are sufficient but diagnosis is omitted, $\mathcal{H}_{\texttt{diag}}$ contains plausible diagnostic candidates that support disease-oriented retrieval without assuming the query explicitly states the target disease.

Finally, the planner compiles these signals into a dimension-wise constraint set $\mathcal{C} = \{c_k\}_{k \in \mathcal{K}}$, where each constraint follows the tuple defined in Eq. 5.

$$c_k = \big(m_k, \mathcal{V}_k, \eta_k\big), \qquad m_k = \begin{cases} \texttt{direct}, & z_k = 1, \\ \texttt{hyp}, & z_k = 0, \end{cases} \tag{5}$$

with matching mode m_k, candidate value set $\mathcal{V}_k$, and tolerance or weighting metadata η_k. We instantiate $\mathcal{V}_k = \{x_k\}$ under direct matching and $\mathcal{V}_k = \mathcal{H}_k$ under hypothesis-augmented matching. This compact notation makes the planner output directly executable: the searcher only needs to read $(m_k, \mathcal{V}_k, \eta_k)$ to know whether a field should be matched literally or searched through plausible completions. The complete planning algorithm is provided in Appendix F. As disease-card memory grows, prior disease patterns provide additional context for sufficiency estimation and hypothesis generation.

2.2 Searcher: Hierarchical tool orchestration

The retrieval infrastructure has three layers: atomic tools, composite clinical tools, and adaptive tool chains. Once sufficiency has been identified, the next question is which matching operations fit each dimension and how those operations should be combined. The core design choice is that heterogeneous clinical fields should not be forced into a single representation or matching primitive. We therefore introduce the representation selector in Eq. 6.

$$\rho : \mathcal{K} \to \{\texttt{num}, \texttt{med}, \texttt{short}, \texttt{long}\}, \tag{6}$$

which maps each dimension to a dominant representation type. Demographics, laboratories, and timeline primarily use numeric matching. Symptoms and diagnosis use medical-concept matching. Treatment mixes medical concepts and short text, and outcome uses long-text matching. Appendix C and Appendix D give the tool inventory and composition rules.

The searcher materializes planner outputs as an execution graph whose actions retrieve, aggregate, or re-rank candidates. Planner-side sufficiency states directly govern tool assignment: sufficient dimensions trigger direct-match composite tools, whereas insufficient dimensions trigger hypothesis-augmented tools that incorporate inferred candidates and cross-dimension context. Re-ranking is chosen from the overall sufficiency profile $\mathbf{z}$ and intent $\mathcal{I}$. When diagnosis is insufficient but symptoms or demographics are sufficient, the searcher uses symptom-anchored retrieval followed

by diagnosis-hypothesis re-ranking. When laboratory, treatment, or outcome fields are incomplete, it re-ranks candidates over the observed fields together with inferred values for the missing field. When timeline evidence is incomplete, temporal compatibility receives lower weight than symptom, diagnosis, and treatment compatibility.

Search then proceeds in two stages. Retrieval tools first gather candidate cases from multiple dimensions under $\mathcal{C}$. Specifically, Eq. 7 defines the Top-B candidate list and pooled candidate set for each query.

$$\mathcal{U}_k(q) = \mathrm{TopB}_{r \in \mathcal{R}} \; S_k(r \mid c_k), \qquad \mathcal{U}_{pool}(q) = \mathrm{Dedup}\Big(\bigcup_{k \in \mathcal{K}} \mathcal{U}_k(q) \Big), \tag{7}$$

where B is the per-dimension candidate budget and $\mathrm{Dedup}(\cdot)$ removes duplicate records across dimensions before re-ranking. Re-ranking tools then refine $\mathcal{U}_{pool}(q)$ using $\mathcal{I}$ and $\mathbf{z}$. The first stage determines which cases are worth considering, and the second determines which of those cases best match the disease-centric retrieval intent under the current sufficiency profile. We combine the two stages through the score in Eq. 8.

$$\mathrm{Score}(r \mid q) = \lambda_1(\mathbf{z}, \mathcal{I})\, S_{\mathrm{ret}}(r \mid q) + \lambda_2(\mathbf{z}, \mathcal{I})\, S_{\mathrm{rer}}(r \mid q) \tag{8}$$

where S_{ret} aggregates dimension-level evidence and S_{rer} captures cross-field compatibility through one or more re-rankers. The fusion weights sum to one and are conditioned on $\mathbf{z}$ and $\mathcal{I}$, so the system can emphasize direct evidence for well-specified queries and cross-field reasoning when important dimensions are incomplete. The same information profile controls both candidate generation and final ranking instead of applying one retrieval policy to all clinical cases. As long-term memory accumulates disease cards and trajectory templates, it provides priors for tool-chain selection and re-ranking policy in later queries.

2.3 Trajectory memory consolidation

The system adapts through external memory while preserving a *no-parameter-update* design. We add this memory layer because a one-shot retrieval agent cannot know in advance which hypotheses, tool combinations, and search strategies will work across diseases and query profiles. Instead of modifying model weights, PatientSearch-SE updates disease cards and retrieval strategy templates from trajectory feedback. This makes cross-query adaptation explicit while keeping the retrieval backbone fixed during evaluation. Our experiments use Qwen3 backbones throughout, so transfer to other backbones should be read as a design property rather than an empirical claim. The memory problem involves three uncertainties: missing query information, the hypothesis space used to complete underspecified dimensions, and the choice of tools and orchestration patterns for a given case. To support this loop, the system maintains hierarchical memory as defined in Eq. 9.

$$\mathcal{M}_t = \big(\mathcal{M}_t^{\mathrm{short}}, \mathcal{M}_t^{\mathrm{long}}\big), \tag{9}$$

where $\mathcal{M}_t^{\mathrm{short}}$ stores the session-local patient state, temporary hypotheses, and tool traces, while $\mathcal{M}_t^{\mathrm{long}}$ stores disease cards and reusable retrieval strategy templates. Each long-term item is paired with applicability metadata such as disease family, sufficiency profile, retrieval intent, and reliability, so reuse is conditioned on where a strategy appears to apply rather than on superficial case similarity alone. High-quality trajectories are abstracted from short-term memory into long-term memory, while long-term memory is queried when insufficient dimensions require additional hypotheses or orchestration priors. During retrieval, the planner reads applicable disease cards as hypothesis priors, and the searcher reads strategy templates as tool-selection and re-ranking priors. Disease-card injection is bounded by $J = 5$ and triggered when an important dimension is insufficient and the highest-scoring applicable card has relevance above 0.6. Detailed schema and migration protocols are deferred to Appendix E, Appendix G, and Appendix H.

High-quality trajectories are filtered by an experience consolidation agent, which does not copy raw sessions verbatim. After trajectory evaluation, it applies Add, Modify, Keep, and Delete operations to disease cards and strategy templates using the rules in Appendix H. It extracts reusable strategy templates, generalizes disease-specific experience, and applies leakage control so that stored content cannot trivially recover evaluation answers. Memory reuse affects score weighting as well as the retrieval policy encoded in strategy templates and tool-assignment behavior.

Leakage boundary. The planner and searcher do not access ground-truth disease labels, final relevance labels, or future-query outcomes during retrieval. For adaptive runs, each query is scored before any memory write. Only after scoring can disease-level relevance signals enter the consolidator, and then only to decide whether a trajectory is eligible for abstraction. Before a memory item is written, the consolidator removes raw query strings, patient identifiers, exact held-out labels, full ranked lists, and case-unique spans. Later queries receive only disease-family abstractions, sufficiency-profile metadata, and strategy-template preferences rather than exact answers or ranked examples.

Because the method adapts through memory and strategy updates rather than gradient-based optimization, trajectory quality must be evaluated explicitly. We use the final disease-level retrieval outcome as the primary result signal and pair it with the composite process score defined in Eq. 10 for each execution trajectory:

$$\Phi(\tau) = \alpha_0 R_{final} + \alpha_1 E_{plan} + \alpha_2 E_{coh} + \alpha_3 E_{cand} + \alpha_4 E_{rank} + \alpha_5 E_{orch} + \alpha_6 E_{cons} \quad (10)$$

where E_{plan}, E_{coh}, E_{cand}, E_{rank}, E_{orch}, and E_{cons} denote planning completeness, intent-constraint coherence, candidate coverage, re-ranking improvement, tool-orchestration appropriateness, and reasoning consistency, respectively. Their closed-form definitions are given in Appendix I. These indicators are retrieval-process probes rather than independently validated labels of clinical reasoning quality. Memory is then updated by consolidating only high-quality trajectories according to Eq. 11.

$$\mathcal{M}_{t+1} = \textsc{Consolidate}\big(\mathcal{M}_t, \mathcal{T}_t^+\big), \qquad \mathcal{T}_t^+ = \{\tau : \Phi(\tau) > \xi\}, \quad (11)$$

This update abstracts successful trajectories into leakage-controlled templates. The system can then reuse planning and retrieval behaviors that tend to work, rather than memorizing answers for individual queries. We therefore evaluate final retrieval metrics together with trajectory indicators and memory size across cases.

3 Experiments

3.1 Experimental setup

Datasets. We evaluate on *PMC-Patients* [1], a curated case-report benchmark grounded in MeSH disease categories, and *MIMIC-IV* [18], a retrospective EHR-derived benchmark grounded in ICD-based disease labels. These datasets test structured clinical narratives and noisier hospital records. Appendix N gives construction details, split protocol, coverage control, and dataset statistics. Each query is a fixed retrospective case description. Automatic completion uses the available evidence under a shared non-interactive query interface across methods.

Baselines and metrics. We compare against sparse retrievers, dense retrievers, reasoning-enhanced sparse/dense retrievers, and end-to-end reasoning retrieval systems, and we add two internal structured ablations: a patient-state-only variant and a structured field-aware retriever with a missingness heuristic. Baselines use the same query/corpus fields when their interface permits it; these ablations remove sufficiency-aware planning, hypothesis-guided search, trajectory memory, and adaptive orchestration. We report retrieval metrics, disease-level clinical consistency, trajectory indicators, memory growth, and diagnosis-masked downstream RAG metrics. Full baseline definitions, metric details, and implementation settings are in Appendices P, R, and S.

Validation scope. The trajectory indicators in Equation 10 are author-specified retrieval-process probes, not independent clinical-reasoning labels; we interpret them only after the primary disease-level metrics and audit them in Appendix U. Appendix T discusses same-interface coverage limits for graph, sequence, and contrastive patient-representation methods. The experiments are organized around four questions: whether PatientSearch-SE improves same-interface disease-centric retrieval, which components account for the gain, whether memory reuse contributes to retrieval gains across cases, and whether retrieved cases help a frozen diagnosis-masked downstream predictor.

3.2 Main results

This experiment tests whether sufficiency-aware retrieval control improves disease-centric case search over sparse, dense, biomedical, reasoning-enhanced, and agentic retrievers under the same query

interface. Table 1 reports the main retrieval results on PMC-Patients and MIMIC-IV. PatientSearch-SE reports higher metric values than the external same-interface baseline families on all columns at 30B; the 8B variant is also strongest except for MIMIC-IV P@10, where Search-o1 is slightly higher. Relative to the best external baseline, the 30B model changes MRR by 0.071 on PMC-Patients and 0.009 on MIMIC-IV, and changes Recall@1000 by 0.077 and 0.046.

These results support the paper's narrower thesis that, in retrospective disease-centric case retrieval, explicit information sufficiency is associated with better candidate generation and final ranking. The patient-state-only variant suggests that clinical-dimension structuring is useful by itself, and the structured field-aware retriever shows that deterministic field weighting and missingness handling further strengthen a non-agentic clinical retriever. PatientSearch-SE adds sufficiency states, hypothesis-guided retrieval, hierarchical orchestration, and trajectory memory on top of these structured variants.

The gains are larger on the curated PMC-Patients subset and more modest on MIMIC-IV, where the strongest reasoning baseline is already close on top-rank metrics. This pattern suggests that sufficiency-aware planning and hierarchical orchestration may improve candidate coverage before final re-ranking, with clearer gains in the case-report setting and smaller but still positive gains in discharge-summary retrieval. Table 1 reports mean $\pm$ standard deviation for non-deterministic reasoning and agentic rows over three prompt-order shuffling runs.

Method	Param.	PMC-Patients subset				MIMIC-IV subset			
		MRR↑	P@10↑	NDCG@10↑	Recall@1000↑	MRR↑	P@10↑	NDCG@10↑	Recall@1000↑
Sparse retriever									
BM25 [19]	-	0.358	0.206	0.220	0.438	0.860	0.740	0.749	0.672
Dense retriever									
ReasonIR [20]	8B	0.693	0.500	0.516	0.574	0.868	0.732	0.745	0.688
GTE [21]	1.5B	0.693	0.496	0.512	0.566	0.883	0.740	0.754	0.681
Qwen3-Embedding [22]	0.6B	0.707	0.522	0.538	0.604	0.893	0.773	0.784	0.708
BGE-M3 [23]	0.6B	0.650	0.436	0.453	0.512	0.847	0.678	0.695	0.612
BioLinkBERT [24]	0.34B	0.458	0.243	0.254	0.356	0.278	0.140	0.143	0.188
ClinicalBERT [9]	0.11B	0.445	0.243	0.252	0.341	0.365	0.195	0.201	0.244
PubMedBERT [10]	0.1B	0.413	0.213	0.222	0.318	0.192	0.093	0.092	0.156
Reasoning with sparse retriever (BM25)									
Deepseek [25]	671B	0.694±0.006	0.471±0.007	0.493±0.007	0.536±0.010	0.879±0.004	0.765±0.005	0.775±0.005	0.704±0.008
Kimi-K2 [26]	32B	0.666±0.007	0.449±0.008	0.469±0.008	0.522±0.011	0.875±0.004	0.765±0.005	0.774±0.005	0.701±0.008
QwQ [27]	32B	0.548±0.010	0.337±0.011	0.352±0.011	0.461±0.013	0.871±0.005	0.761±0.006	0.770±0.006	0.692±0.009
Qwen3 [28]	30B	0.591±0.009	0.391±0.010	0.407±0.010	0.487±0.012	0.860±0.005	0.740±0.006	0.750±0.006	0.672±0.009
Qwen3 [28]	8B	0.419±0.012	0.248±0.012	0.263±0.012	0.443±0.014	0.840±0.006	0.720±0.007	0.728±0.007	0.664±0.010
HuatuoGPT-o1 [29]	7B	0.483±0.011	0.287±0.012	0.303±0.012	0.451±0.014	0.869±0.005	0.753±0.006	0.763±0.006	0.681±0.009
Reasoning with dense retriever (Qwen3-Embedding)									
Deepseek [25]	671B	0.820±0.005	0.635±0.006	0.656±0.006	0.704±0.009	0.891±0.004	0.774±0.005	0.785±0.005	0.735±0.007
Kimi-K2 [26]	32B	0.809±0.006	0.620±0.006	0.641±0.006	0.692±0.009	0.885±0.004	0.770±0.005	0.780±0.005	0.730±0.007
QwQ [27]	32B	0.803±0.006	0.615±0.007	0.637±0.007	0.684±0.010	0.879±0.005	0.762±0.006	0.774±0.006	0.724±0.008
Qwen3 [28]	30B	0.790±0.007	0.604±0.007	0.622±0.007	0.668±0.010	0.860±0.005	0.738±0.006	0.749±0.006	0.706±0.008
Qwen3 [28]	8B	0.722±0.008	0.539±0.009	0.555±0.009	0.612±0.011	0.838±0.006	0.714±0.007	0.722±0.007	0.682±0.009
HuatuoGPT-o1 [29]	7B	0.775±0.007	0.590±0.008	0.609±0.008	0.651±0.010	0.866±0.005	0.743±0.006	0.753±0.006	0.710±0.008
Reasoning retrieval system									
Search-o1 [14]	32B	0.824±0.005	0.639±0.006	0.663±0.006	0.721±0.009	0.919±0.004	<u>0.815±0.005</u>	0.825±0.005	0.778±0.007
Search-r1 [11]	7B	0.802±0.006	0.613±0.007	0.634±0.007	0.694±0.010	0.905±0.004	0.792±0.005	0.802±0.005	0.746±0.008
PaSa [30]	7B	0.808±0.006	0.621±0.007	0.643±0.007	0.707±0.009	0.907±0.004	0.802±0.005	0.812±0.005	0.761±0.008
Deep-DxSearch [15]	7B	0.815±0.005	0.631±0.006	0.653±0.006	0.713±0.009	0.914±0.004	0.809±0.005	0.819±0.005	0.769±0.007
DeepRetrieval [31]	3B	0.761±0.007	0.572±0.008	0.591±0.008	0.642±0.011	0.884±0.005	0.762±0.006	0.772±0.006	0.704±0.009
Ours									
Patient-state only	-	0.742±0.006	0.575±0.007	0.596±0.007	0.635±0.009	0.899±0.004	0.798±0.005	0.809±0.005	0.724±0.008
Field-aware retriever	-	0.822±0.006	0.657±0.007	0.680±0.007	0.724±0.009	0.906±0.004	0.804±0.005	0.815±0.005	0.758±0.008
Full model	8B	<u>0.885±0.004</u>	<u>0.774±0.005</u>	<u>0.786±0.005</u>	<u>0.764±0.007</u>	<u>0.920±0.003</u>	0.813±0.004	<u>0.826±0.004</u>	<u>0.792±0.006</u>
Full model	30B	**0.895±0.003**	**0.786±0.004**	**0.799±0.004**	**0.798±0.006**	**0.928±0.003**	**0.824±0.004**	**0.836±0.004**	**0.824±0.006**

Table 1: Performance comparison on the PMC-Patients subset and MIMIC-IV subset used in our evaluation. The Ours block includes two internal structured variants and the full PatientSearch-SE model. Patient-state only uses fixed validation-set weights to fuse dimension-level scores. The field-aware retriever applies deterministic field weights and a missingness heuristic to structured patient fields, without sufficiency-aware planning, hypothesis-guided retrieval, adaptive orchestration, or trajectory memory. Full model denotes PatientSearch-SE; parameter counts indicate backbone size. MRR is mean reciprocal rank, P@10 is precision among the top 10 retrieved cases using $\mathrm{rel}(q, r) \geq 1$ as the relevance threshold, NDCG@10 is normalized discounted cumulative gain at rank 10 using the graded labels $\mathrm{rel}(q, r) \in \{0, 1, 2\}$, and Recall@1000 is the fraction of disease-consistent targets with $\mathrm{rel}(q, r) \geq 1$ recovered within the top 1000. Values with error bars are mean $\pm$ standard deviation over three prompt-order shuffling runs with fixed model weights and decoding parameters. Deterministic sparse and dense embedding baselines are reported without run variance. Arrows in the header indicate the preferred metric direction. Boldface marks the highest result in each column and underlining marks the second-highest result.

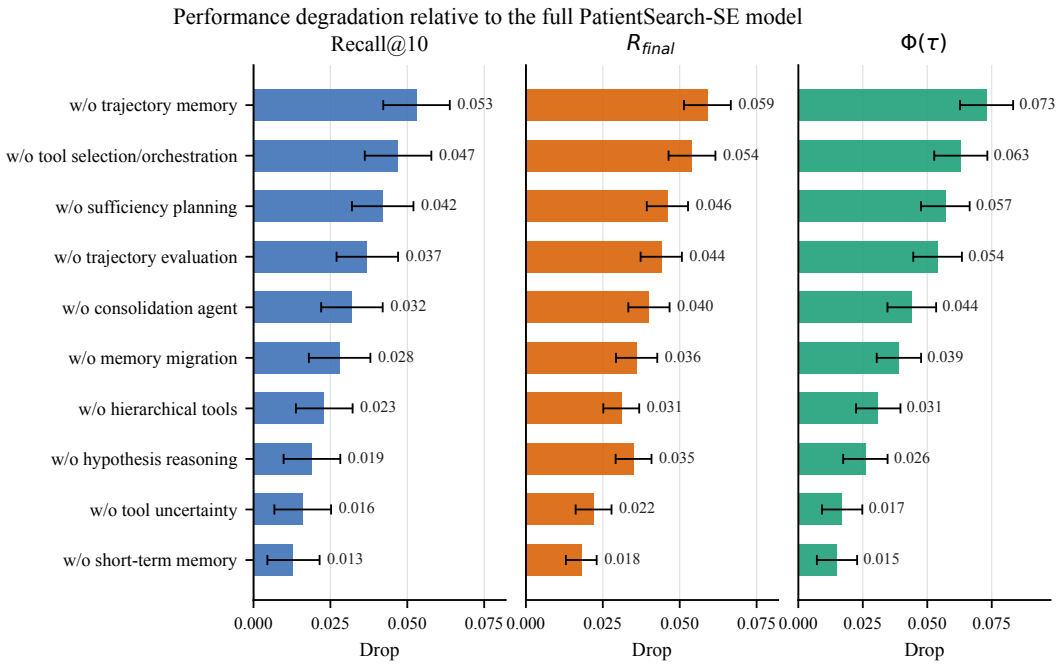


Figure 2: Component ablations on validation sets, shown as degradation relative to the full PatientSearch-SE model. The sufficiency ablation removes per-dimension sufficiency states while retaining structured patient-state extraction. Bars report the drop in Recall@10, R_{final}, and $\Phi(\tau)$; error bars propagate the reported standard deviations of the full model and ablated variant over three prompt-order shuffling runs. R_{final} and $\Phi(\tau)$ are defined in Sections B and I.

3.3 Ablation study

We ablate each major component to separate gains from query structuring, tool orchestration, and trajectory memory. Each variant removes one capability while preserving the same split, retrieval budget, and evaluation protocol as the full system.

Figure 2 supports three conclusions. First, sufficiency modeling contributes beyond structured patient-state retrieval: removing it lowers Recall@10 from 0.763 to 0.721 and reduces R_{final} and $\Phi(\tau)$ by 0.046 and 0.057. Second, structured execution matters beyond adding tools: flattening the hierarchy, removing tool uncertainty, and removing tool selection all degrade performance. Third, long-term trajectory memory contributes beyond one-shot retrieval, with the largest drop under "w/o trajectory memory" (Recall@10 0.710, R_{final} 0.836, $\Phi(\tau)$ 0.754). Because the patient-state-only variant already accounts for a substantial share of the improvement over flat retrieval, we interpret these ablations as incremental evidence for sufficiency-aware control and memory rather than as a claim that either component alone explains the full gain. Sensitivity tests in Appendix L show the same pattern under one-factor-at-a-time perturbations.

3.4 Memory accumulation analysis

We study retrieval performance as a function of accumulated memory. The analysis tracks the final disease-level signal, representative trajectory indicators, memory footprint, and lookup overhead to test whether disease cards and strategy templates provide useful adaptation beyond one-shot retrieval without uncontrolled memory growth.

From cold start to 1000 processed cases, R_{final} rises from 0.783 to 0.895, while E_{plan}, E_{coh}, E_{cand}, and E_{orch} also improve. The largest early gains appear in E_{orch} and E_{plan}, which is consistent with rapid acquisition of execution templates. These process-indicator trends are not external validation of better clinical reasoning; they show that trajectories increasingly satisfy the fixed operational rubric used for retrieval-trace analysis. Memory growth remains sublinear relative to processed cases: the long-term store grows to 403 disease cards and 128 strategy templates at 1000 cases, while

Table 2: Retrieval, trajectory, and memory-footprint indicators as memory accumulates. Values are mean ± standard deviation. Cards/Templates are counted after merge/delete operations. Overhead is the additional per-query memory-matching latency.

Phase	# Cases	R_{final}	E_{plan}	E_{coh}	E_{cand}	E_{orch}	# Cards	# Templates	Avg. Inj.	Overhead
Cold start (pre-filled)	0	0.783±0.008	0.684±0.010	0.717±0.009	0.825±0.007	0.653±0.011	124±5	38±3	2.1±0.3	18±3 ms
Early	100	0.836±0.006	0.750±0.008	0.770±0.008	0.863±0.006	0.725±0.009	171±7	57±4	3.0±0.4	24±4 ms
Mid	500	0.873±0.005	0.805±0.006	0.814±0.006	0.895±0.005	0.780±0.007	286±10	94±6	3.7±0.4	36±5 ms
Mature	1000	0.895±0.003	0.841±0.005	0.843±0.005	0.917±0.004	0.816±0.006	403±14	128±8	4.1±0.5	49±6 ms

Table 3: Diagnosis-masked downstream few-shot RAG prediction on the PMC-Patients subset and MIMIC-IV subset used in our evaluation. Values with error bars are mean ± standard deviation over three prompt-order shuffling runs. Top-1 and Top-3 FS Acc. are downstream prediction accuracies using the top one or top three retrieved cases as few-shot evidence, while MRR is the retrieval mean reciprocal rank of the first disease-consistent case. Vanilla reports zero-shot accuracy with no retrieved evidence, so Top-3 FS Acc. and MRR are not applicable.

	PMC-Patients subset			MIMIC-IV subset		
Method	Top-1 FS Acc.↑	Top-3 FS Acc.↑	Ret. MRR↑	Top-1 FS Acc.↑	Top-3 FS Acc.↑	Ret. MRR↑
Vanilla (zero-shot)	0.464±0.010	–	–	0.594±0.011	–	–
BM25-RAG	0.582±0.009	0.616±0.010	0.597±0.009	0.528±0.011	0.708±0.010	0.872±0.010
Dense-RAG	0.674±0.009	0.774±0.007	0.729±0.010	0.688±0.010	0.736±0.010	0.905±0.009
Reasoning-RAG	0.719±0.006	0.837±0.008	0.776±0.005	0.761±0.009	0.782±0.009	0.924±0.008
PatientSearch-SE-RAG	**0.759±0.006**	**0.893±0.005**	**0.824±0.006**	**0.782±0.008**	**0.863±0.008**	**0.941±0.007**

the average number of injected cards stays below the top-J bound and lookup overhead remains below 50 ms. Because this retrospective analysis cannot fully separate reusable strategy abstraction from benchmark-pattern reuse, we interpret it as evidence of memory-guided adaptation rather than unrestricted continual learning.

The write policy in Appendix H bounds memory growth: raw cases, full ranked lists, and exact held-out labels are removed before writes. Near-duplicate cards are merged, stale or contradicted cards can be deleted, and only the top-J matched cards are injected into a new session. These mechanisms are not a complete deployment-scale memory-compression solution, but they make memory bloat an evaluated system property rather than an implicit assumption.

3.5 Downstream diagnosis-masked few-shot RAG prediction

A useful case retriever should also surface evidence that helps a frozen downstream predictor infer disease labels from partially specified descriptions. We therefore add a retrospective *diagnosis-masked few-shot RAG prediction* evaluation on both datasets. For each query, explicit disease names, ICD/MeSH identifiers, and direct diagnostic statements are removed before retrieval and prediction. A frozen Qwen3-8B-Instruct predictor receives the masked query and, for retrieval-conditioned methods, the top retrieved cases. All methods share the predictor, prompt, decoding configuration, and output parser; query cases are excluded from the corpus, and test-time memory is frozen or updated only prequentially.

Table 3 shows that PatientSearch-SE-RAG gives the strongest retrieval-conditioned downstream accuracy on both datasets. Relative to REASONING-RAG, Top-1 few-shot accuracy rises from 0.719 to 0.759 on PMC-Patients and from 0.761 to 0.782 on MIMIC-IV; Top-3 accuracy rises from 0.837 to 0.893 and from 0.782 to 0.863. The retrieval MRR column shows the same strongest rank quality for PatientSearch-SE-RAG, while the lower MIMIC-IV Top-1 value for BM25-RAG than for zero-shot VANILLA indicates that weak retrieved evidence can hurt a frozen predictor. This auxiliary result supports the narrower claim that PatientSearch-SE retrieves evidence useful to the fixed predictor under this retrospective protocol, but it does not establish holistic clinical retrieval quality beyond the disease-centric prediction endpoint.

4 Limitations and Future Work

The evaluation target remains disease-centric retrieval rather than full clinical equivalence across treatment, outcome, or longitudinal management. The memory mechanism is leakage-controlled

external adaptation, not unrestricted continual learning, and the trajectory indicators are author-defined process probes rather than independently validated clinical-reasoning labels. Cold-start behavior, knowledge noise from external distillation, and long-term memory growth also remain open issues. Future work should test broader downstream tasks, validate trajectory indicators externally, sharpen sufficiency modeling, compress memory stores, and evaluate beyond PMC-Patients and MIMIC-IV.

5 Conclusion

We introduced PatientSearch-SE for disease-centric clinical case search. The framework uses sufficiency-aware planning to decide which patient dimensions can be matched directly, hierarchical tool orchestration to adapt retrieval and re-ranking, and trajectory memory to consolidate reusable search experience without parameter updates. Across PMC-Patients and MIMIC-IV, PatientSearch-SE improves disease-level retrieval accuracy over the compared same-interface baselines, shows a retrospective memory-accumulation trend, and retrieves evidence that helps a frozen diagnosis-masked few-shot predictor under the evaluated protocol.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62371409) and National High Level Hospital Clinical Research Funding (2025-PUMCH-D-002 and BJ-2025-138).

References

[1] Zhengyun Zhao, Qiao Jin, Fangyuan Chen, Tuorui Peng, and Sheng Yu. A large-scale dataset of patient summaries for retrieval-based clinical decision support systems. *Scientific data*, 10 (1):909, 2023.

[2] Shuheng Chen, Namratha Patil, Haonan Pan, Angel Hsing-Chi Hwang, Yao Du, Ruishan Liu, and Jieyu Zhao. Med-copilot: A medical assistant powered by graphrag and similar patient case retrieval. *arXiv preprint arXiv:2603.00460*, 2026.

[3] Qiao Jin, Zifeng Wang, Charalampos S Floudas, Fangyuan Chen, Changlin Gong, Dara Bracken-Clarke, Elisabetta Xue, Yifan Yang, Jimeng Sun, and Zhiyong Lu. Matching patients to clinical trials with large language models. *Nature communications*, 15(1):9074, 2024.

[4] Lang Cao, Qingyu Chen, and Yue Guo. Ehr-rag: Bridging long-horizon structured electronic health records and large language models via enhanced retrieval-augmented generation. *arXiv preprint arXiv:2601.21340*, 2026.

[5] Shichao Fang, Shenda Hong, Qing Li, Pengfei Li, Tim Coats, Beiji Zou, and Guilan Kong. Cross-modal similar clinical case retrieval using a modular model based on contrastive learning and k-nearest neighbor search. *International Journal of Medical Informatics*, 193:105680, 2025.

[6] Deyi Li, Aditi Shukla, Sravani Chandaka, Bradley Taylor, Jie Xu, and Mei Liu. Autoencoder-based representation learning for similar patients retrieval from electronic health records: Comparative study. *JMIR Medical Informatics*, 13:e68830, 2025.

[7] Nicolas Garcelon, Antoine Neuraz, Vincent Benoit, Rémi Salomon, Sven Kracker, Felipe Suarez, Nadia Bahi-Buisson, Smail Hadj-Rabia, Alain Fischer, Arnold Munnich, et al. Finding patients using similarity measures in a rare diseases-oriented clinical data warehouse: Dr. warehouse and the needle in the needle stack. *Journal of biomedical informatics*, 73:51–61, 2017.

[8] Christel Gérardin, Arthur Mageau, Arsène Mékinian, Xavier Tannier, and Fabrice Carrat. Construction of cohorts of similar patients from automatic extraction of medical concepts: Phenotype extraction study. *JMIR Medical Informatics*, 10(12):e42379, 2022.

[9] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. In *Proceedings of the 2nd clinical natural language processing workshop*, pages 72–78, 2019.

[10] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021.

[11] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

[12] Bowei He, Minda Hu, Zenan Xu, Hongru Wang, Licheng Zong, Yankai Chen, Chen Ma, Xue Liu, Pluto Zhou, and Irwin King. Search-r2: Enhancing search-integrated reasoning via actor-refiner collaboration. *arXiv preprint arXiv:2602.03647*, 2026.

[13] Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, et al. Verltool: Towards holistic agentic reinforcement learning with tool use. *arXiv preprint arXiv:2509.01055*, 2025.

[14] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5420–5438, 2025.

[15] Qiaoyu Zheng, Yuze Sun, Chaoyi Wu, Weike Zhao, Pengcheng Qiu, Yongguo Yu, Kun Sun, Jian Zhang, Yanfeng Wang, Ya Zhang, et al. End-to-end agentic rag system training for traceable diagnostic reasoning. *arXiv preprint arXiv:2508.15746*, 2025.

[16] Xiao Han, Yuzheng Fan, Sendong Zhao, Haochun Wang, and Bing Qin. Gsem: Graph-based self-evolving memory for experience augmented clinical reasoning. *arXiv preprint arXiv:2603.22096*, 2026.

[17] Ruofan Jin, Zaixi Zhang, Mengdi Wang, and Le Cong. Stella: Self-evolving llm agent for biomedical research. *arXiv preprint arXiv:2507.02004*, 2025.

[18] 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, 2023.

[19] Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc, 2009.

[20] Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen-tau Yih, Pang Wei Koh, et al. Reasonir: Training retrievers for reasoning tasks. *arXiv preprint arXiv:2504.20595*, 2025.

[21] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

[22] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

[23] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 4(5), 2024.

[24] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. Linkbert: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, 2022.

[25] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[26] Kimi Team, Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi k2: Open agentic intelligence, 2026. URL `https://arxiv.org/abs/2507.20534`.

[27] Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, 2025.

[28] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[29] Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. Huatuogpt-o1, towards medical complex reasoning with llms. *arXiv preprint arXiv:2412.18925*, 2024.

[30] Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, et al. Pasa: An llm agent for comprehensive academic paper search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11663–11679, 2025.

[31] Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, and Jiawei Han. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. *arXiv preprint arXiv:2503.00223*, 2025.

[32] Umid Suleymanov, Zaur Rajabov, Emil Mirzazada, and Murat Kantarcioglu. Beyond refusal: Probing the limits of agentic self-correction for semantic sensitive information. *arXiv preprint arXiv:2602.21496*, 2026.

[33] Tianle Xia, Ming Xu, Lingxiang Hu, Yiding Sun, Wenwei Li, Linfang Shang, Liqun Liu, Peng Shu, Huan Yu, and Jie Jiang. Search-p1: Path-centric reward shaping for stable and efficient agentic rag training. *arXiv preprint arXiv:2602.22576*, 2026.

[34] Tongyu Wen, Guanting Dong, and Zhicheng Dou. Smartsearch: Process reward-guided query refinement for search agents. *arXiv preprint arXiv:2601.04888*, 2026.

[35] Hongliang Lu, Yuhang Wen, Pengyu Cheng, Ruijin Ding, Jiaqi Guo, Haotian Xu, Chutian Wang, Haonan Chen, Xiaoxi Jiang, and Guanjun Jiang. Search self-play: Pushing the frontier of agent capability without supervision. *arXiv preprint arXiv:2510.18821*, 2025.

[36] Yi Yao, He Zhu, Piaohong Wang, Jincheng Ren, Xinlong Yang, Qianben Chen, Xiaowan Li, Dingfeng Shi, Jiaxian Li, Qiexiang Wang, et al. O-researcher: An open ended deep research model via multi-agent distillation and agentic rl. *arXiv preprint arXiv:2601.03743*, 2026.

[37] Jiaru Zou, Ling Yang, Yunzhe Qi, Sirui Chen, Mengting Ai, Ke Shen, Jingrui He, and Mengdi Wang. Autotool: Dynamic tool selection and integration for agentic reasoning. *arXiv preprint arXiv:2512.13278*, 2025.

[38] Yuqi Zhu, Shuofei Qiao, Yixin Ou, Shumin Deng, Shiwei Lyu, Yue Shen, Lei Liang, Jinjie Gu, Huajun Chen, and Ningyu Zhang. Knowagent: Knowledge-augmented planning for llm-based agents. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3709–3732, 2025.

[39] Yanming Liu, Xinyue Peng, Zixuan Yan, Yanxin Shen, Wenjie Xu, Yuefeng Huang, Xinyi Wang, Jiannan Cao, Jianwei Yin, and Xuhong Zhang. Dep-search: Learning dependency-aware reasoning traces with persistent memory. *arXiv preprint arXiv:2601.18771*, 2026.

[40] Wenhan Liu, Xinyu Ma, Weiwei Sun, Yutao Zhu, Yuchen Li, Dawei Yin, and Zhicheng Dou. Reasonrank: Empowering passage ranking with strong reasoning ability. *arXiv preprint arXiv:2508.07050*, 2025.

[41] Deniz Qian, Hung-Ting Chen, and Eunsol Choi. Rvr: Retrieve-verify-retrieve for comprehensive question answering. *arXiv preprint arXiv:2602.18425*, 2026.

[42] Wenhan Liu, Xinyu Ma, Yutao Zhu, Yuchen Li, Daiting Shi, Dawei Yin, and Zhicheng Dou. Agentic-r: Learning to retrieve for agentic search. *arXiv preprint arXiv:2601.11888*, 2026.

[43] Yuqi Zhou, Sunhao Dai, Changle Qu, Liang Pang, Jun Xu, and Ji-Rong Wen. Learning to retrieve from agent trajectories. *arXiv preprint arXiv:2604.04949*, 2026.

[44] Guangyuan Ma, Yongliang Ma, Xuanrui Gou, Zhenpeng Su, Ming Zhou, and Songlin Hu. Lightretriever: A llm-based text retrieval architecture with extremely faster query inference. *arXiv preprint arXiv:2505.12260*, 2025.

[45] Yuwen Du, Rui Ye, Shuo Tang, Xinyu Zhu, Yijun Lu, Yuzhu Cai, and Siheng Chen. Openseeker: Democratizing frontier search agents by fully open-sourcing training data. *arXiv preprint arXiv:2603.15594*, 2026.

[46] Yuzheng Cai, Siqi Cai, Yuchen Shi, Zihan Xu, Lichao Chen, Yulei Qin, Xiaoyu Tan, Gang Li, Zongyi Li, Haojia Lin, et al. Training-free group relative policy optimization. *arXiv preprint arXiv:2510.08191*, 2025.

[47] Rubing Chen, Jian Wang, Wenjie Li, Xiao-Yong Wei, and Qing Li. To retrieve or to think? an agentic approach for context evolution. *arXiv preprint arXiv:2601.08747*, 2026.

[48] Zhongxiang Sun, Qipeng Wang, Weijie Yu, Jingxuan Yang, Haolang Lu, and Jun Xu. Deep search with hierarchical meta-cognitive monitoring inspired by cognitive neuroscience. *arXiv preprint arXiv:2601.23188*, 2026.

[49] Yujiang Wu, Shanshan Zhong, Yubin Kim, and Chenyan Xiong. What generative search engines like and how to optimize web content cooperatively. *arXiv preprint arXiv:2510.11438*, 2025.

[50] Jiajie Zhang, Xin Lv, Ling Feng, Lei Hou, and Juanzi Li. Chaining the evidence: Robust reinforcement learning for deep search agents with citation-aware rubric rewards. *arXiv preprint arXiv:2601.06021*, 2026.

[51] Kaixuan Fan, Kaituo Feng, Manyuan Zhang, Tianshuo Peng, Zhixun Li, Yilei Jiang, Shuang Chen, Peng Pei, Xunliang Cai, and Xiangyu Yue. Exploring reasoning reward model for agents. *arXiv preprint arXiv:2601.22154*, 2026.

[52] Chuzhan Hao, Wenfeng Feng, Guochao Jiang, Guofeng Quan, Guohua Liu, and Yuewei Zhang. Beyond stochastic exploration: What makes training data valuable for agentic search. *arXiv preprint arXiv:2604.08124*, 2026.

[53] Shupeng Li, Weipeng Lu, Linyun Liu, Chen Lin, Shaofei Li, Zhendong Tan, Hanjun Zhong, Yucheng Zeng, Chenghao Zhu, Mengyue Liu, et al. Qianfanhuijin technical report: A novel multi-stage training paradigm for finance industrial llms. *arXiv preprint arXiv:2512.24314*, 2025.

[54] Yong Xien Chng, Tao Hu, Wenwen Tong, Xueheng Li, Jiandong Chen, Haojia Yu, Jiefan Lu, Hewei Guo, Hanming Deng, Chengjun Xie, et al. Sensenova-mars: Empowering multimodal agentic reasoning and search via reinforcement learning. *arXiv preprint arXiv:2512.24330*, 2025.

[55] Salaheddin Alzubi, Noah Provenzano, Jaydon Bingham, Weiyuan Chen, and Tu Vu. Evoskill: Automated skill discovery for multi-agent systems. *arXiv preprint arXiv:2603.02766*, 2026.

[56] Pengcheng Qiu, Chaoyi Wu, Shuyu Liu, Yanjie Fan, Weike Zhao, Zhuoxia Chen, Hongfei Gu, Chuanjin Peng, Ya Zhang, Yanfeng Wang, et al. Quantifying the reasoning abilities of llms on clinical cases. *Nature Communications*, 16(1):9799, 2025.

[57] Christopher Chiu, Silviu Pitis, and Mihaela van der Schaar. Simulating viva voce examinations to evaluate clinical reasoning in large language models. *arXiv preprint arXiv:2510.10278*, 2025.

[58] Saber Zerhoudi, Michael Granitzer, and Jelena Mitrovic. Agentsim: A platform for verifiable agent-trace simulation. *arXiv preprint arXiv:2604.26653*, 2026.

[59] DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, et al. Deepseek-v3.2: Pushing the frontier of open large language models, 2025. URL https://arxiv.org/abs/2512.02556.

A Related work

Clinical case retrieval and similar-patient search. Prior work on clinical retrieval has studied domain-specific representations, clinically grounded benchmarks, and structured matching over EHR-derived evidence [1, 3, 4, 2, 32]. Related similar-patient and cohort-retrieval systems use modular contrastive retrieval, autoencoder-based representation learning, medical-concept extraction, or rare-disease warehouse search to find clinically related patients [5–8]. These systems improve medical search quality relative to generic document retrieval and share the goal of patient-level retrieval. However, most still apply a fixed matching policy once the query representation is built: the same query processing and matching logic is used regardless of whether diagnoses are explicit, laboratory evidence is missing, or treatment history is underspecified. PatientSearch-SE differs by making information sufficiency an explicit planning variable and by using that variable to alter both candidate generation and re-ranking behavior.

Structured patient representations. Graph-based, sequential, and contrastive patient representations are a close neighboring family because they preserve patient structure more explicitly than flat text embeddings. In structured EHR settings, these methods can encode visit order, diagnosis/procedure co-occurrence, medication histories, laboratory trajectories, and patient–concept graphs before retrieval. Our work shares the view that patient similarity is multi-dimensional, but differs in the input contract and control target. The query in our benchmark is a heterogeneous natural-language case description, and the retrieval policy must decide at inference time which dimensions are reliable enough for literal matching and which dimensions should be completed hypothetically. A graph or sequence encoder could serve as a corpus-side or dimension-specific tool inside PatientSearch-SE, but it would not by itself solve the planner's sufficiency-control problem. This distinction is why our primary comparison uses same-interface sparse, dense, biomedical, reasoning, and agentic retrievers, while full graph/sequence integration remains future work rather than a claim resolved by Table 1.

Agentic search and tool-augmented retrieval. Recent agentic search systems show that complex retrieval can benefit from tool use, trajectory optimization, and iterative reasoning [11, 12, 33–35, 14, 36, 13]. Related work studies tool selection, dependency-aware decomposition, retrieve-verify loops, reasoning-intensive reranking, and agent-retriever co-optimization [37–45]. However, these methods are largely developed for web or general-domain search. They do not directly address clinical case search, where dimensions have different reliability levels and retrieval decisions depend on whether a dimension should be matched literally, completed hypothetically, or deferred to cross-field reasoning. PatientSearch-SE adapts the agentic-search perspective to this clinical setting through sufficiency-guided orchestration.

Clinical reasoning agents and memory-augmented systems. Clinical reasoning systems such as DeepDxSearch, GSEM, and broader memory-augmented medical agents use reflective trajectory feedback and externalized knowledge stores to improve decision making without relying solely on end-to-end retriever training [15–17]. Some of these systems emphasize applicability-aware experience reuse and structured memory updates rather than flat case accumulation [16]. More generally, work on training-free evolution, hierarchical experience design, and agent reward shaping has explored how agents can improve through trajectory selection, critique, and memory consolidation [46–55]. PatientSearch-SE builds on this line of work but targets disease-centric clinical retrieval rather than general reasoning. The main distinction is that memory-guided adaptation is coupled to retrieval control: per-dimension sufficiency states and trajectory indicators decide whether retrieval succeeds and how the retrieval policy should change across information-complete and information-incomplete queries.

Clinical reasoning evaluation. Recent clinical reasoning benchmarks argue that final-answer quality and reasoning-process quality should be assessed separately, with attention to process attributes such as efficiency, completeness, information seeking, confidence calibration, and adaptive hypothesis revision under sequential uncertainty [56, 57]. These studies also expose failure modes such as

anchoring, premature closure, and inadequate investigation selection [57]. Work on verifiable agent-trace simulation similarly emphasizes that trace-level metrics should be tied to observable execution states rather than treated as opaque preference scores [58]. PatientSearch-SE is not a diagnostic benchmark, so we do not claim to measure these constructs directly. Instead, we use trajectory indicators more narrowly as retrieval-oriented probes of planning, candidate generation, and orchestration behavior.

B Problem definition

Given a patient record database $\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$ and a natural language clinical query q, the goal is to produce the Top-K ranked list defined in Eq. 12 under the following retrospective evaluation target.

$$\mathcal{L}_K(q) = \mathrm{TopK}_{r \in \mathcal{R}} \ \mathrm{Score}(r \mid q), \tag{12}$$

where each query–case pair is associated with a disease-level relevance label after ICD/MeSH normalization and disease-family alignment. The query-side target label is *not* exposed to the planner or searcher at inference time. A retrieval is therefore considered successful when the returned cases have the same disease or a clinically similar disease type as the target query. This target intentionally prioritizes disease-level consistency and should not be read as a full measure of holistic case equivalence.

Formally, for a query q and retrieved case r_i, we use the graded disease-relevance label defined in Eq. 13.

$$\mathrm{rel}(q, r_i) \in \{0, 1, 2\}, \tag{13}$$

where 0 denotes an unrelated disease, 1 denotes a similar disease type, and 2 denotes the exact same disease. For metrics that require a normalized similarity signal, we use Eq. 14.

$$\mathrm{Sim}_{disease}(q, r_i) = \frac{\mathrm{rel}(q, r_i)}{2}. \tag{14}$$

This signal is used only for evaluation and for filtering trajectories during memory consolidation. Unless otherwise stated, binary retrieval metrics treat labels 1 and 2 as relevant, while NDCG uses the graded label directly. Let $\mathcal{L}_K(q) = \big(r_{(1)}, \ldots, r_{(K)}\big)$ denote the final ranked list. The per-query final result signal is defined in Eq. 15.

$$R_{final}(q) = \max_{1 \le i \le K} \frac{\mathrm{rel}(q, r_{(i)})}{2} \tag{15}$$

with $K = 1000$ for the final ranked list; reported table values are averages over evaluation queries.

C Clinical dimensions and tool inventory

Table 4 summarizes the clinical state fields, representation types, and default tools used by the planner and searcher.

Table 4: Clinical dimensions, representations, and default retrieval tools.

Dimension	Attributes	Representation	Default tool(s)
Demographics	Age, gender, ethnicity, occupation, medical/family/psychosocial history	Numeric / categorical / string	NumericRangeComparator, StringFuzzyMatch
Symptoms	Chief complaints, symptom descriptions, physical exam findings	Medical-concept text	MedicalConceptEmbeddingSimilarity, TokenOverlapMatch
Laboratory	Test names, values, units, diagnostic methods	Numeric / categorical	NumericRangeComparator
Diagnosis	ICD codes, disease names, differential diagnoses, diagnostic reasoning	Medical-concept text	MedicalConceptEmbeddingSimilarity
Treatment	Medications, procedures, dosages, intervention changes	Medical concepts + short text	MedicalConceptEmbeddingSimilarity, ShortTextEmbeddingSimilarity
Outcome	Follow-up results, clinical status, adherence, adverse events	Long text	LongDocumentEmbeddingSimilarity
Timeline	Event dates, temporal sequence	Numeric dates	NumericRangeComparator

Representation types. *Medical-concept text* denotes standardized symptoms, diagnoses, laboratory names, and drugs up to 128 tokens. It is encoded with biomedical embeddings to capture synonymy

and hierarchy. *Short text* denotes natural-language clinical descriptions up to 128 tokens and is compared with sentence embeddings. *Long text* denotes case reports, notes, and discharge summaries longer than 128 tokens. It is chunked, encoded hierarchically, and pooled at document level.

Atomic tools. NumericRangeComparator handles numeric ranges. StringFuzzyMatch handles spelling and name variants. TokenOverlapMatch provides rapid keyword filtering. MedicalConceptEmbeddingSimilarity compares standardized biomedical concepts. ShortTextEmbeddingSimilarity compares short clinical descriptions. LongDocumentEmbeddingSimilarity compares long notes. IntentAwareReRanker adjusts ranking by retrieval intent. MultiFieldCrossAttentionRanker re-ranks by cross-field compatibility over symptoms, diagnosis, treatment, and outcome.

D Tool composition and execution graph

We implement three composition operators over tool outputs. Let $\mathcal{T} = \{T_1, \ldots, T_n\}$ denote a set of atomic or composite tools, and let each tool return a score in $[0, 1]$ for record r.

Sequential composition. The sequential score is defined in Eq. 16.

$$S_{\text{seq}}(r \mid q) = T_n\Big(r \,\Big|\, q, \{T_j(r \mid q)\}_{j=1}^{n-1}\Big), \tag{16}$$

where downstream tools consume the filtered or reweighted outputs of upstream tools. We use this operator when an earlier tool narrows the candidate space for a later, more expensive tool. For example, a demographics-and-symptoms query first applies NumericRangeComparator to retain age-compatible candidates and then applies MedicalConceptEmbeddingSimilarity only within that reduced pool.

Parallel composition. The parallel score is defined in Eq. 17.

$$S_{\text{par}}(r \mid q) = \sum_{j=1}^{n} \mu_j \, T_j(r \mid q), \qquad \sum_{j=1}^{n} \mu_j = 1, \tag{17}$$

where all tools operate independently on the same candidate set and their outputs are fused by convex weights μ_j. We use this mode when several dimensions are simultaneously informative, such as symptom-, diagnosis-, and treatment-aware matching for a richly specified query.

Aggregation composition. The aggregation score is defined in Eq. 18.

$$S_{\text{agg}}(r \mid q) = g\Big(S_{\text{seq}}^{(1)}(r \mid q), \ldots, S_{\text{seq}}^{(m)}(r \mid q)\Big), \tag{18}$$

where each $S_{\text{seq}}^{(i)}$ may itself be a sequential or parallel composite, and $g(\cdot)$ is instantiated as weighted sum in retrieval mode and weighted max in hypothesis-expansion mode. This operator lets the searcher combine heterogeneous chains into a single candidate score while preserving the planner's field priorities.

Composition hierarchy. A tool chain is a rooted composition tree: leaves are atomic tools, internal nodes are sequential / parallel / aggregation operators, and the root score is passed to candidate generation or re-ranking. Table 5 summarizes the execution graph actions and the sufficiency-conditioned re-ranking policy.

Table 5: Execution graph actions and sufficiency-conditioned re-ranking.

Item	Role	Scope	Activated tools / policy
retrieve	Fetch candidates from database	Single dim. or global	Dimension-specific tool tree over $\mathcal{R}$
aggregate	Merge dimension-level candidates	Global	Deduplication and weighted/max aggregation over $\{\mathcal{U}_k\}$
rerank	Refine the candidate pool	Single dim. or global	IntentAwareReRanker and/or MultiFieldCrossAttentionRanker
$z_k = 1$	Direct evidence matching	Per dimension	Minimal or standard re-ranking, with intent-aware re-ranking when intent is explicit
$z_k = 0$	Hypothesis-augmented matching	Per dimension	MultiFieldCrossAttentionRanker + long/short-text similarity over inferred values

E Short-term memory schema

Each short-term memory instance has four fields. $\mathcal{E}$ stores original, hypothesized, and intermediate patient evidence. $\mathcal{H}_{temp}$ stores session hypotheses with confidence and evidence links. $\mathcal{S}_{current}$ stores $\mathbf{z}$, step count, and gating history. $\mathcal{T}_{trace}$ stores selected tools, composition modes, and per-step recall/ranking/cost signals for trajectory evaluation and template extraction.

F Complete algorithm

Execution protocol. Given query q, database $\mathcal{R}$, and long-term memory $\mathcal{M}^{\text{long}}$, PatientSearch-SE: (1) initializes short-term memory and extracts $\mathcal{P}$, (2) predicts sufficiency profile $\mathbf{z}$, intent $\mathcal{I}$, and hypotheses for dimensions with $z_k = 0$, (3) compiles constraints $\mathcal{C}$ and builds execution graph $\mathcal{G}$, (4) retrieves per-dimension candidates $\mathcal{U}_k = \text{TopB}_{r\in\mathcal{R}}\, S_k(r \mid c_k)$ and forms $\mathcal{U}_{pool} = \text{Dedup}(\cup_k \mathcal{U}_k)$, (5) re-ranks $\mathcal{U}_{pool}$ with a strategy selected from $(\mathbf{z}, \mathcal{I})$ and scores cases by Equation 8, (6) constructs τ, computes $\Phi(\tau) = \alpha_0 R_{final} + \sum_{j=1}^{6} \alpha_j E_j$, and consolidates τ into long-term memory only when $\Phi(\tau) > \xi$ after leakage filtering.

G Disease card schema

Each disease card $c_i \in \mathcal{M}^{\text{long}}$ stores identity fields (ICD-10 code, disease name, aliases, source), clinical features, treatment pathways, prognostic features, applicability conditions (supported sufficiency profiles, intents, and transfer boundaries), and metadata (usage, timestamp, confidence, and retrieval strategy summary).

H Memory migration protocol

Short→Long migration. Migration is triggered by $\Phi(\tau) > \xi$, confirmed feedback, or repeated information-hypothesis associations. The consolidator analyzes successful paths, abstracts them into disease-card fields and strategy templates, checks duplicates, and incrementally updates long-term memory.

Experience consolidation agent. To prevent evaluation leakage, a dedicated consolidator mediates all writes. It receives planner outputs, $\mathcal{T}_{trace}$, Top-K results with disease-code similarity signals, and trajectory metrics. It then performs trajectory filtering, experience abstraction, leakage control, and Add/Delete/Modify/Keep decisions. In the main experiments, the consolidator uses the same Qwen3 scale as the planner/searcher under a separate prompt, following a shared-frozen-model design.

The four memory operations are selected using Eq. 19.

$$o^\star(c, \tau) = \arg \max_{o\in\{\texttt{Add},\texttt{Delete},\texttt{Modify},\texttt{Keep}\}} U_o(c, \tau), \tag{19}$$

where c is the nearest existing disease card or strategy template, and U_o is an operation-specific utility. We use the following rules. *Add* is used when no existing card reaches cosine similarity 0.85 and $\Phi(\tau) > \xi$. *Modify* is used when a disease-family match improves E_{rank} or E_{orch} by at least 0.05. *Keep* is used when the nearest card already captures the pattern. *Delete* is used when a card is contradicted by two later high-quality trajectories or unused for 500 cases.

Leakage control is enforced before any memory write. The consolidator removes raw query strings, patient identifiers, exact held-out disease labels, exact ranked result lists, and any span that can be deterministically matched back to a unique evaluation case. The retained abstraction is limited to disease family, symptom pattern, sufficiency profile, tool-choice preference, and ranking-policy summary.

Long→Short injection. When sufficient session context exists and new hypotheses are needed, the system matches disease cards by patient state, ranks them by relevance and confidence, converts selected cards to temporary hypotheses, and transfers long-term confidence to hypothesis scores according to Eq. 20.

$$\mathcal{M}_t^{\text{inject}} = \text{TopJ}_{c\in\mathcal{M}_t^{\text{long}}}\, \text{Rel}\left(c, \mathcal{P}(q), \mathbf{z}, \mathcal{I}\right), \tag{20}$$

where $J = 5$ in the default setting. Injection is triggered only when at least one important dimension is insufficient and the highest-scoring matched long-term card has relevance above 0.6. Otherwise, the searcher proceeds without memory augmentation.

I Trajectory evaluation details

We decompose trajectory quality into six intermediate indicators in addition to the final disease-level result signal. These indicators are designed to probe planning completeness, information-seeking efficiency, and orchestration coherence in retrospective retrieval. They do not independently score medical factuality against external references or expert annotations.

Planning completeness. We compute this indicator using Eq. 21, with its structural and hypothesis components defined in Eq. 22.

$$E_{plan} = \delta_{\text{plan}} E_{\text{struct}} + (1 - \delta_{\text{plan}}) E_{\text{hyp}}, \tag{21}$$

where the component terms are defined as follows:

$$E_{\text{struct}} = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} e_k, \qquad E_{\text{hyp}} = \frac{1}{\max(1, |\{k : z_k = 0\}|)} \sum_{k : z_k = 0} \max_{h \in \mathcal{H}_k} p(h \mid \mathcal{P}, \mathcal{I}). \tag{22}$$

Here $e_k \in [0, 1]$ scores extraction into the structured patient state, and $p(h \mid \mathcal{P}, \mathcal{I})$ is the normalized confidence of hypothesis h. We set $\delta_{\text{plan}} = 0.6$, instantiate e_k by exact match for numeric/categorical fields and clipped cosine similarity for free text, and allow at most three hypotheses per insufficient dimension.

Intent-constraint coherence. We compute this indicator using Eq. 23, with its rule and execution components defined in Eq. 24.

$$E_{coh} = \delta_{\text{coh}} E_{\text{rule}} + (1 - \delta_{\text{coh}}) E_{\text{exec}}, \tag{23}$$

where the component terms are defined as follows:

$$E_{\text{rule}} = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \mathbb{I}\big[m_k = \bar{m}_k(\mathcal{I}, z_k)\big], \qquad E_{\text{exec}} = \frac{1}{|V|} \sum_{v \in V} \mathbb{I}\big[\pi(v) \in \bar{\Pi}_{\mathcal{I}}(\psi(v))\big]. \tag{24}$$

Here $\bar{m}_k(\mathcal{I}, z_k)$ is the planner-preferred matching mode, and $\bar{\Pi}_{\mathcal{I}}(\psi(v))$ is the set of operation types allowed for intent $\mathcal{I}$. We set $\delta_{\text{coh}} = 0.5$ and fix the intent rule library before evaluation: diagnostic confirmation emphasizes diagnosis/symptoms, differential diagnosis symptoms/labs, treatment selection treatment/outcome, prognosis outcome/timeline, and case similarity diagnosis/symptoms/timeline.

Candidate coverage. We define this indicator in Eq. 25.

$$E_{cand} = \max_{r \in \mathcal{U}_{pool}} \frac{\text{rel}(q, r)}{2}, \tag{25}$$

which measures whether the full union of dimension-level candidates contains the correct or clinically similar disease family before re-ranking and truncation. With seven dimensions and $B = 200$, this pool contains at most 1400 unique cases. In contrast, R_{final} is computed on the final Top-1000 list.

Re-ranking improvement. We define this indicator in Eq. 26.

$$E_{rank} = \sum_{i=1}^{K} \omega_i \cdot \frac{\text{rel}(q, r_{(i)})}{2}, \quad \omega_1 > \omega_2 > \cdots > \omega_K, \quad \sum_{i=1}^{K} \omega_i = 1 \tag{26}$$

which rewards ranking similar cases toward the top of the final list.

Tool-orchestration appropriateness. We compute this indicator using Eq. 27.

$$E_{orch} = \text{clip}_{[0,1]} \Big(\frac{\beta_1 E_{cand} + \beta_2 E_{rank}}{\beta_1 + \beta_2} - \beta_3 C_{\text{lat}}(\tau) - \beta_4 C_{\text{waste}}(\tau) \Big), \tag{27}$$

where C_{lat} is normalized latency/compute cost and C_{waste} is the fraction of tool calls whose removal changes neither E_{cand} nor E_{rank} by more than 0.01. We use $(\beta_1, \beta_2, \beta_3, \beta_4) =$

$(0.35, 0.35, 0.15, 0.15)$, normalizing the positive coverage and ranking terms by $\beta_1 + \beta_2$ before subtracting the cost penalties.

Reasoning consistency. We compute this indicator using Eq. 28.

$$E_{cons} = 1 - \frac{1}{\max(1, |E|)} \sum_{(u,v)\in E} \mathbb{I}\big[\psi(u) = \psi(v) \wedge \pi(u) = \texttt{rerank} \wedge \pi(v) = \texttt{retrieve}\big], \quad (28)$$

which penalizes transitions that contradict the planner's intended order. Only retrieve→aggregate, retrieve→rerank, and aggregate→rerank are penalty-free. Same-dimension rerank→retrieve is inconsistent unless a new constraint update is issued.

For the trajectory total score in Eq. 10, we use the weights specified in Eq. 29.

$$(\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6) = (0.35, 0.15, 0.10, 0.15, 0.10, 0.10, 0.05), \quad (29)$$

which gives the largest weight to the final disease-level result while still preserving substantial weight on intermediate reasoning signals. A trajectory is consolidated into long-term memory when $\Phi(\tau) > \xi$ with $\xi = 0.75$.

J Reproducibility details

We report four groups of reproducibility details for the final experiments: (1) planner configuration, including prompt structure or model settings for sufficiency assessment, intent inference, and hypothesis generation, (2) retrieval and re-ranking configuration, including candidate budgets, field weights, and threshold or fusion settings used in the execution graph, (3) memory and consolidation configuration, including update thresholds, template-selection rules, and leakage-control heuristics, and (4) system-level runtime settings, including hardware, batch size, and the number of repeated runs or seeds used for reporting. Unless otherwise stated, all methods use the same query/corpus splits and the same disease-label coverage control described in Section N. Table 6 summarizes the default settings.

Table 6: Default implementation settings.

Component	Default	Component	Default
Backbone	Shared Qwen3-8B or Qwen3-30B for planner/searcher/consolidator, with temperature 0.2 for planner and consolidator	Sufficiency thresholds γ_k	demo .80, sym .60, lab .70, diag .75, treat .60, out .50, time .60
Intent set	diagnostic confirmation, differential diagnosis, treatment selection, prognosis estimation, case similarity exploration	Max hypotheses	3 per insufficient dimension, with normalized planner confidence
Candidate budget B	200 per dimension before deduplication	Candidate pool / ranking depth	up to 1400 after union and deduplication; final Top-1000 after re-ranking, with $K = 1000$ for metrics
Fusion weights	cold start $\lambda_1 = 0.6$, $\lambda_2 = 0.4$, with query-adaptive adjustment within ± 0.1	Field weights w_k	uniform within active intent-specific fields and zero on inactive fields
Re-ranker mixture ν_m	IntentAwareReRanker for intent-explicit cases, otherwise $(0.4, 0.6)$ over intent-aware and cross-attention rerankers	Memory threshold	$\xi = 0.75$ on $\Phi(\tau)$
Leakage control	remove raw query strings, patient identifiers, exact held-out labels, and full Top-K lists before writes	Evaluation runs	3 prompt-order shuffles with fixed weights/decoding, deterministic retrievers without run variance

The planner uses a fixed three-part prompt format for all datasets: (i) structured patient-state extraction into the seven dimensions in Equation 1, (ii) per-dimension sufficiency prediction with one confidence score per dimension, and (iii) intent prediction plus up to three hypotheses for each insufficient dimension. The searcher receives only the structured planner outputs $(\mathcal{P}, \mathbf{z}, \mathcal{I}, \mathcal{C})$ and does not access ground-truth disease labels. The consolidator is implemented with the same underlying Qwen3 model instance or model scale as the planner and searcher, but under a separate consolidation prompt that performs trajectory filtering, abstraction, leakage control, and Add/Delete/Modify/Keep memory decisions.

K Prompt templates

Tables 7, 8, 9, and 10 provide the prompt templates used by the LLM-mediated steps in PatientSearch-SE. The templates are written as instruction rules rather than dataset-specific examples so that

they can be reused across PMC-Patients and MIMIC-IV without exposing ground-truth disease labels to the planner or searcher. The variable `query` denotes the clinical query (diagnosis-masked for the downstream evaluation in Appendix Q), `patient_state` denotes the seven-dimensional state in Equation 1, `sufficiency` denotes $\mathbf{z}$ with confidence scores, `constraints` denotes $\mathcal{C}$, and `memory_items` denotes the leakage-controlled long-term disease cards or strategy templates injected into the current session.

Table 7: Planner prompt templates.

Prompted step	Coupled skill	Prompt template
Patient-state extraction	Clinical evidence structuring	Read `query` and extract only evidence explicitly supported by the text into seven dimensions: demographics, symptoms, laboratory findings, diagnoses, treatments, outcomes, and timeline. Preserve normalized clinical terms when possible, keep absent fields empty rather than inferred, attach a short evidence span or paraphrase for every nonempty field, and do not add disease names, identifiers, or labels unsupported by the query. Return a structured patient state with one entry per dimension.
Sufficiency assessment	Dimension-wise sufficiency judging	Given `query`, `patient_state`, and applicable leakage-controlled disease-card priors from `memory_items`, decide for each dimension whether the observed evidence is sufficient for direct retrieval matching. Mark `sufficient` only when the field contains specific, query-supported evidence that can constrain retrieved cases; mark `insufficient` when the field is absent, vague, contradictory, or only indirectly implied. Use memory only as an applicability prior for judging whether the observed evidence is likely discriminative under the current disease family, sufficiency profile, and intent; do not let memory fill an absent field or reveal held-out target labels. Assign one confidence score in $[0, 1]$ per dimension and provide a one-sentence justification tied to observed evidence and, when used, the selected memory prior.
Intent inference	Retrieval-intent classification	Given `patient_state` and `sufficiency`, choose exactly one retrieval intent from diagnostic confirmation, differential diagnosis, treatment selection, prognosis estimation, and case similarity exploration. Prefer the intent best supported by explicit query language and active dimensions; if the query only asks for similar prior cases, use case similarity exploration. Output the selected intent, active dimensions, and dimensions that should receive higher ranking weight.
Hypothesis generation	Evidence-conditioned clinical completion	For each insufficient dimension, generate at most three clinically plausible hypotheses conditioned only on sufficient dimensions, selected leakage-controlled disease-card priors from `memory_items`, and the selected intent. Select memory priors only when their applicability metadata matches the current patient state, sufficiency profile, and retrieval intent, and transfer long-term confidence to hypothesis confidence only as a prior. Each hypothesis must include confidence, supporting observed dimensions, memory support if used, and a retrieval use case. Do not present hypotheses as facts, do not copy hidden labels, and omit any hypothesis that is not grounded by the available evidence or conflicts with observed query evidence.
Constraint compilation	Planner-to-searcher handoff	Convert `patient_state`, `sufficiency`, intent, hypotheses, and selected memory-derived hypothesis priors into dimension-wise retrieval constraints. Use direct matching for sufficient dimensions and hypothesis-augmented matching for insufficient dimensions with accepted hypotheses. For each dimension, output matching mode, value set, tolerance or weight metadata, memory provenance when a value came from an injected disease card, and a short note explaining how the constraint should affect candidate generation or re-ranking.

Table 8: Searcher and tool-orchestration prompt templates.

Prompted step	Coupled skill	Prompt template
Memory-to-session injection	Disease-card and strategy-template reuse	Given `patient_state`, `sufficiency`, intent, and candidate `memory_items`, select only memory items that are applicable to the current sufficiency profile and retrieval intent. Convert selected disease-card features into temporary hypotheses and selected strategy templates into tool-selection priors. Ignore cards whose disease family, transfer boundary, or reliability metadata conflicts with the observed query evidence.
Tool-chain planning	Hierarchical retrieval orchestration	Given `constraints`, intent, sufficiency profile, and selected strategy-template priors from injected `memory_items`, build an execution graph over retrieve, aggregate, and rerank actions. Use a strategy prior only when its supported sufficiency profiles, intents, transfer boundaries, and reliability metadata apply to the current session. Match numeric dimensions with range or temporal comparators, terminology-heavy dimensions with biomedical concept matching, and long-text fields with long-context comparison. Use direct-match tools for sufficient dimensions and hypothesis-augmented tools for insufficient dimensions. Keep the graph ordered so retrieval and aggregation precede reranking unless a new constraint update is issued.
Candidate-pool review	Coverage-oriented search control	After per-dimension retrieval, inspect the candidate-pool summary, active constraints, tool traces, and any injected strategy-template priors. Identify whether any important sufficient dimension is missing from the candidate pool, whether an insufficient dimension requires broader hypothesis search, and whether duplicate or low-yield tool calls should be removed. Reuse memory only to adjust tool breadth, composition mode, or follow-up retrieval priorities when the prior remains compatible with observed evidence and trace feedback. Output only execution adjustments; do not use ground-truth disease labels or final relevance labels.
Cross-field reranking	Intent-aware clinical compatibility judging	Given `query`, `patient_state`, intent, sufficiency profile, constraints, selected ranking-policy priors from injected `memory_items`, and a candidate case summary, score cross-field compatibility for disease-centric retrieval. Emphasize active intent-specific dimensions, discount insufficient fields unless supported by hypotheses, and separate observed-evidence matches from hypothesis or memory-prior matches. Use memory-derived ranking preferences only when their applicability metadata matches the current sufficiency profile and intent, and ignore priors that conflict with observed evidence. Return a normalized score, matched evidence fields, conflict fields, memory-prior contribution if used, and a concise ranking rationale.

L Sensitivity analysis

Sensitivity analysis tests whether the reported gains depend on a narrow hyperparameter setting. Figure 3 reports one-factor-at-a-time perturbations around the default configuration.

Table 9: Trajectory evaluation and memory-update prompt templates.

Prompted step	Coupled skill	Prompt template
Trajectory-process evaluation	Retrieval trace auditing	Given planner outputs, execution graph, candidate-pool statistics, pre-rerank and post-rerank summaries, and final disease-level evaluation signals, evaluate E_{plan}, E_{coh}, E_{cand}, E_{rank}, E_{orch}, and E_{cons} according to the fixed rules in Appendix I. Report each score with one evidence-based justification and flag unsupported hypotheses, intent-constraint mismatch, redundant tool calls, or inconsistent rerank-before-retrieve transitions.
Trajectory memory consolidation agent	Trajectory-guided external-memory writing	Act as the trajectory memory consolidation agent for PatientSearch-SE. Given `patient_state`, sufficiency profile, retrieval intent, hypotheses, constraints, execution graph, tool traces, trajectory scores, final Top-K summary, and nearest existing memory items, determine what reusable experience should be written to long-term memory. Extract only generalizable disease-family patterns, applicability conditions, successful or failed hypothesis types, tool-orchestration lessons, and ranking-policy preferences. Output two possible artifacts: a disease card update and a strategy-template update, each with confidence, supported sufficiency profiles, supported intents, transfer boundaries, and evidence links to trace-level observations. Do not store raw query text, patient identifiers, exact held-out disease labels, exact ranked lists, or case-unique spans.
Experience consolidation	Leakage-controlled abstraction	Given a completed trajectory and its process scores, decide whether it is eligible for long-term memory using $\Phi(\tau) > \xi$. If eligible, abstract reusable disease-family features, sufficiency conditions, successful hypotheses, tool-chain choices, and ranking-policy lessons. Remove raw query strings, patient identifiers, exact held-out labels, full ranked lists, and any span that uniquely identifies an evaluation case. Write only generalized disease-card fields or strategy-template fields.
Memory operation selection	Add/Delete/Modify/Keep decision	Given a proposed abstraction and nearest existing memory items, choose one operation from Add, Delete, Modify, and Keep. Add only when no reliable existing item covers the same disease family and sufficiency profile; Modify when the new trajectory improves or corrects an existing item; Delete contradicted, duplicated, or unused items; Keep when the proposed abstraction adds no reliable information. Provide operation utility and a short rationale.

Table 10: Downstream diagnosis-masked RAG prompt template.

Prompted step	Coupled skill	Prompt template
Diagnosis-masked prediction	Retrieval-conditioned clinical prediction	Given a disease-masked query and zero, one, or three retrieved case summaries, predict the most likely disease label or disease family supported by the visible evidence. First identify shared symptoms, laboratory findings, treatments, outcomes, timelines, and demographic context between the query and retrieved cases; then infer the disease family from the shared pattern. Do not copy masked disease names, identifiers, or labels, and do not use retrieval rank as the only reason for the prediction. Return the predicted label, confidence, supporting evidence, and any ambiguity caused by label granularity.

M Structured input reliability

Because PatientSearch-SE operates over structured patient states and corpus-side clinical fields, we also report a closely related human-audited structuring result on PMC case reports from the same PMC-Diseases data-construction pipeline used in our experiments. On 1,200 sampled cases, that extraction-and-categorization protocol achieved 94.67% note-span F1, 99.77% field-level accuracy on patient information, and category-level F1 scores of 95.84%, 95.21%, 96.02%, and 94.77% for clinical findings, diagnostic assessment, therapeutic interventions, and follow-up/outcomes, respectively. We treat these numbers as evidence that large-scale clinical case structuring can be made reliable enough for downstream retrieval, rather than as a direct gold-label evaluation of the online PatientSearch-SE planner itself.

N Experimental dataset construction details

We evaluate on two clinical case search benchmarks.

Table 11: Dataset statistics for PMC-Patients and MIMIC-IV after data construction and coverage control. Total Samples is the retained dataset size, Total Diseases is the number of unique disease labels after normalization, Query and Corpus are the split sizes, and Query Diseases and Corpus Diseases count unique disease labels in each split. Total diseases equal corpus diseases because the corpus fully covers query diseases.

Dataset	Total Samples	Total Diseases	Query	Corpus	Query Diseases	Corpus Diseases
PMC-Patients	55,912	4,198	8,885	47,027	2,989	4,198
MIMIC-IV	42,651	12,091	7,848	34,803	5,643	12,091

PMC-Patients [1] contains 167K patient case reports from PubMed Central, with 293K patient-patient similarity annotations. We interpret these case pairs through disease-centric supervision using MeSH disease categories as semantic anchors. Each patient record includes structured clinical information

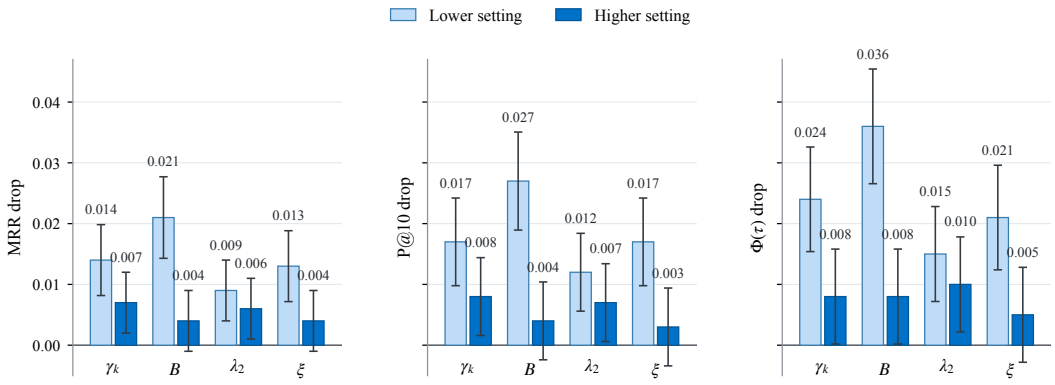


Figure 3: Sensitivity analysis with one-factor-at-a-time perturbations around the default configuration. Bars show the decrease in each metric relative to its default. Within each parameter group, lighter and darker bars indicate the lower and higher settings, respectively (small and large for B). Error bars propagate the standard deviations of the default and perturbed configurations over three prompt-order shuffling runs. MRR is mean reciprocal rank, P@10 is precision among the top 10 retrieved cases, and $\Phi(\tau)$ is the weighted trajectory-quality score.

spanning demographics, symptoms, diagnoses, treatments, and outcomes. To construct the evaluation split, we begin with the full PMC-Patients corpus (n=130K+ cases) and retain only cases annotated with official MeSH descriptors (n=55,912), extracting the disease (category C) portion of the MeSH hierarchy. The retained cases are then divided into a corpus split (n=47,027) and a query split (n=8,885) at an 8:2 ratio. To ensure that disease labels in the query split are fully covered by the corpus split, we perform a coverage check on the query split's MeSH disease fields: any query record whose disease label is absent from the corpus split is transferred into the corpus split. After this adjustment, the final corpus contains 4,198 unique disease labels and the query split contains 2,989 unique disease labels, with the corpus providing complete label coverage over the query split.

MIMIC-IV [18] is a large-scale real-world EHR database comprising over 299K patients and 43M clinical notes from the ICU and general hospital settings. To construct clinical case retrieval pairs from MIMIC-IV, we begin with all discharge summaries (n=340K+) and retain only patients with a single case by selecting `subject_id`s that appear exactly once, yielding 42,651 single-case records. This filtering keeps the benchmark scale above 40K cases and closer to the PMC-Patients setting. We then create an initial corpus/query split following the same 8:2 protocol as PMC-Patients and apply the same disease-label coverage control strategy by checking whether every disease label observed in the query split is covered by the corpus split and transferring uncovered query records into the corpus when needed. After this adjustment, the final corpus and query splits contain 34,803 and 7,848 records, respectively. The corpus contains 12,091 unique disease labels and the query split contains 5,643 unique disease labels, with full corpus coverage of query diseases. This dual-benchmark evaluation ensures that our approach generalizes across both curated case reports and raw clinical records.

O Qualitative case analysis

We include two representative cases to inspect how retrieved evidence is used when the query is either evidence-complete or evidence-incomplete under diagnosis-masked retrospective evaluation. Figure 4 covers a multisystem 22q11.2 deletion syndrome case and a tuberculosis-associated Guillain-Barré syndrome case. In the first, direct symptom/diagnosis/treatment matching improves the response score from 2.0 to 7.0 relative to DeepSeek-V3.2 [59]. In the second, retrieved cases bridge missing diagnostic trajectory and treatment information, improving the score from 0.0 to 9.0 by reducing unsupported hallucination.

Scores use a 0–10 rubric with diagnostic accuracy 40%, treatment appropriateness 30%, outcome realism 20%, and clinical specificity / hallucination avoidance 10%. The examples support the same interpretation as Table 3, where retrieved analogues can help a frozen reader under the retrospective setting.

Patient state — **Evidence-complete**

30-year-old Japanese female with surgically corrected congenital heart disease (CHD) in infancy, mild intellectual disability since childhood, and severe, treatment-resistant psychotic symptoms (persistent hallucinations, delusions, incoherent speech, explosive/impulsive/aggressive behavior) since adolescence despite maximal doses of ...

Retrieval intent

Exploration-focused search to bridge significant information gaps in adverse events and prognostic assessment, ...

Similarity constraints

"Did the patient develop agranulocytosis or other hematological complications during clozapine treatment given the required periodic monitoring?",
"Was there evidence of metabolic syndrome or weight gain as a common adverse effect of second-generation antipsychotic treatment?"

Clinical questions

Based on the provided partial medical record and imaging evolution, deduce:
1.Diagnostic Reasoning and most likely Diagnosis
2.Appropriate Treatment Type
3.Expected Clinical Outcome

Similar cases （Top 3）

PMC7383000-1: 25F with tetralogy of Fallot, mild ID, first-episode psychosis; confirmed 22q11.2DS; multiple antipsychotics failed → olanzapine + lamotrigine controlled symptoms; discharged seizure-free.
PMC10149101-1: 30F with ASD + mild ID + severe aggression/self-harm; long hospitalizations; clozapine + adjuncts achieved behavioral stabilization but remained dependent.
PMC6060754-1: 6M boy with autism, ID, ADHD-like features; ketogenic diet → significant improvement in behavior ...
Shared features: CHD/neurodevelopmental delay + early psychiatric/behavioral symptoms; variable response to antipsychotics or non-pharmacologic interventions; lifelong management required.

Model answer w / o see similar cases

CHD + mild ID + treatment-resistant psychosis + confirmed 22q11.2 deletion.
Likely Diagnosis: Psychotic disorder associated with 22q11.2 deletion syndrome.
Treatment: Clozapine (titrated to 300 mg/day) + fluvoxamine (CYP1A2 inhibition), biperiden for EPS, ...
Outcome: Substantial reduction in psychotic symptoms (PANSS ↓166→107) and EPS (DIEPSS ↓6→2);

Model answer w / see similar cases

Reference 22q11.2DS case shows CHD, mild ID, and refractory psychosis controlled by antipsychotic switch + adjuncts. Here, 22q11.2 deletion + CHD + treatment-resistant schizophrenia-like psychosis match the neuropsychiatric phenotype of 22q11.2DS.
Likely Diagnosis: Schizophrenia in the context of 22q11.2 deletion syndrome.
Treatment: Second-generation antipsychotics (olanzapine, risperidone, clozapine); clozapine titration ...
Outcome: Severe violent behavior disappeared; more peaceful days without restrictions; improved functioning but remained dependent on assistance.

Ground truth

Diagnostic Reasoning: diagnosed with schizophrenia and mild intellectual disability (IQ = 68); ... cavity; confirmed diagnosis with clinical symptoms
Treatment Type: second-generation antipsychotics; olanzapine; risperidone; clozapine; quetiapine; zotepine; lithium; ...
Clinical Outcome: level of functioning (interpersonal relations, self-care) significantly diminished; required constant assistance with ADLs (toileting, bathing, eating); exhibited undressing in public, excessive drinking, violent outbursts; necessitated isolation; unable to leave hospital for 8 years;
MeSH used: DiGeorge Syndrome

Evaluation report

Comparison	Score	Reason
Baseline	2.0	The AI answer fails to match the Ground Truth on all key dimensions.
RAG	7.0	The AI answer demonstrates good clinical reasoning but lacks specificity in several key areas.

(a) Evidence-complete case.

Patient state — **Evidence-incomplete**

32-year-old male with cough and hemoptysis for 18 days, followed by bilateral lower limb weakness (distal onset, progressing to hips) for 2 days. Examination: acute flaccid paraplegia, patchy sensory loss, left 7th LMN cranial nerve palsy; no autonomic/bladder/bowel involvement. Labs: CSF protein 158 mg/dL with 3 lymphocytes; sputum GeneXpert positive for tuberculosis; NCV performed.

Retrieval intent

Inference/Exploration strategy needed due to multiple information gaps across clinical dimensions. The case presents a complex neurological presentation with tuberculosis comorbidity, requiring exploration of diagnostic reasoning, treatment administration details, and comprehensive follow-up monitoring that is currently missing from the record.

Similarity constraints

Did the patient develop hepatotoxicity or other adverse effects from the ATT medication regimen given the tuberculosis diagnosis?",
"Was there evidence of cerebrospinal fluid analysis showing albuminocytological dissociation to support the AIDP (Guillain-Barré syndrome) diagnosis?"

Clinical questions

Based on the provided partial medical record and imaging evolution, deduce:
1.Diagnostic Reasoning and most likely Diagnosis
2.Appropriate Treatment Type
3.Expected Clinical Outcome

Similar cases (Top 3)

PMC7851665-1: 27M with preceding viral infection → acute ascending flaccid tetraparesis, areflexia, albuminocytological dissociation in CSF; diagnosed as acute motor axonal neuropathy (AMAN) variant of Guillain-Barré syndrome (GBS); treated with therapeutic plasma exchange (TPE); complete neurological recovery in 1 month.
PMC5337770-1: 9M boy with 6-month foot pain/swelling → talonavicular joint tuberculosis confirmed by culture; treated with standard 4-drug ATT (12 months total); full pain-free recovery and normal function at 3-year follow-up.
PMC3969641-1: 11F with 2-month thyroid swelling → tuberculous thyroiditis confirmed by FNAC (caseating granulomas + AFB); treated with ATT (duration not specified); gradual regression of swelling and lymphadenopathy.
Shared features: acute/subacute weakness or swelling; infection-triggered or granulomatous pathology; response to immunomodulation (GBS) or prolonged ATT (TB).

Model answer w /o similar cases

Confirmed pulmonary TB + acute flaccid paraplegia, patchy sensory loss, cranial neuropathy, and CSF albuminocytological dissociation suggest GBS triggered by TB infection.
Likely Diagnosis: Tuberculous Meningitis (TBM).
Treatment: IVIG + category-1 ATT.
Outcome: Excellent neurological recovery expected.

Model answer w / similar cases

Reference GBS case shows identical acute flaccid paralysis and CSF findings with full recovery after immunomodulation post-infection. Combined with confirmed TB, this is GBS triggered by tuberculosis.
Likely Diagnosis: Suggestive of AIDP / Guillain-Barré syndrome triggered by tuberculosis.
Treatment: Intravenous immunoglobulin; category-1 ATT.
Outcome: Completely recovered neurologically.

Ground truth

Diagnostic Reasoning: suggestive of AIDP
Treatment Type: intravenous immunoglobulin; CAT 1 ATT
Clinical Outcome: completely recovered neurologically
MeSH used: Guillain-Barre Syndrome; Tuberculosis

Evaluation report

Comparison	Score	Reason
Baseline	0.0	The AI's response is a complete hallucination and misdiagnosis.
RAG	9.0	The AI's diagnostic reasoning correctly identifies 'Guillain-Barre Syndrome' as the primary neurological condition, which matches the specific MeSH tag and the clinical note's terminology ('AIDP', a subtype of GBS).

(b) Evidence-incomplete case.

Figure 4: Qualitative case studies. Retrieved cases support disease-aligned, treatment-aware, and outcome-aware reasoning under the retrospective rubric used in the PMC-Diseases case analysis.

P Experimental baseline details

We compare against five categories of external baselines.

- **Sparse retriever**: BM25 [19].
- **Dense retriever**: ReasonIR [20], GTE [21], Qwen3-Embedding [22], BGE-M3 [23], BioLinkBERT [24], ClinicalBERT [9], PubMedBERT [10].
- **Reasoning with sparse retriever (BM25)**: Deepseek [25], Kimi-K2 [26], QwQ [27], Qwen3 [28], HuatuoGPT-o1 [29].
- **Reasoning with dense retriever (Qwen3-Embedding)**: Deepseek [25], Kimi-K2 [26], QwQ [27], Qwen3 [28], HuatuoGPT-o1 [29].
- **Reasoning retrieval system**: Search-o1 [14], Search-r1 [11], PaSa [30], DeepDxSearch [15], DeepRetrieval [31].

The patient-state-only row in Table 1 is an internal PatientSearch-SE variant rather than an external baseline. It extracts the seven clinical dimensions in Equation 1, encodes each active dimension with the same biomedical embedding backend as the dense retriever, and fuses dimension-level scores with fixed validation-set weights, but removes sufficiency-aware tool selection, hypothesis-guided search, and trajectory memory. The structured field-aware retriever keeps the same extracted dimensions but replaces the fixed fusion with deterministic field-aware weights and a missingness heuristic that down-weights absent or vague fields, assigns pre-fixed weights to observed high-specificity fields, and leaves missing fields uncompleted. This row therefore tests a stronger non-agentic structured retriever while excluding sufficiency scoring, hypothesis-guided retrieval, adaptive tool orchestration, and memory reuse.

Q Downstream diagnosis-masked few-shot RAG details

The protocol is as follows. For each query, we remove explicit disease names, ICD/MeSH identifiers, and direct diagnostic statements from the query before both retrieval and prediction. A frozen downstream predictor must predict the target disease label or disease family from the masked query alone for the zero-shot vanilla baseline, or from the masked query together with retrieved cases for retrieval-conditioned methods. For retrieval-conditioned methods, we evaluate two evidence budgets: Top-1 few-shot uses only the highest-ranked retrieved case, and Top-3 few-shot uses the three highest-ranked retrieved cases. We additionally report retrieval MRR, defined by the reciprocal rank of the first disease-consistent retrieved case, to separate retrieval rank quality from downstream prediction accuracy. All methods share the same downstream predictor, prompt, decoding configuration, and output parser, so the only changing factor is whether and which retrieved examples are supplied to the predictor. In our experiments, the downstream predictor is Qwen3-8B-Instruct with greedy decoding.

Leakage control is essential for this evaluation. Query cases must be excluded from the retrieval corpus, the downstream predictor must remain frozen, and test-time memory must either be frozen before evaluation or updated only under a strict prequential protocol in which each query is scored before any memory write. When feasible, retrieved cases should be reported under both diagnosis-hidden and diagnosis-visible settings, with the diagnosis-hidden setting treated as the primary evaluation because it better isolates evidence usefulness from answer copying.

The remaining errors for retrieval-conditioned methods are not attributable to a mismatch between retrieval Top-K and prediction Top-K: in this experiment, Top-1 and Top-3 refer to the number of retrieved cases supplied as few-shot evidence. Manual inspection suggests three main failure sources. First, the retrieved cases may be disease-consistent but contain weak or indirect diagnostic evidence after explicit disease names are masked. Second, disease labels can differ in granularity, so a retrieved case may support the correct disease family while the predictor is evaluated against a more specific target label. Third, even when relevant evidence appears in the retrieved examples, the frozen reader can still fail to abstract the shared clinical pattern from a small few-shot context, especially for heterogeneous MIMIC-IV records.

R Experimental metric details

We report: (1) *Retrieval metrics*: MRR, P@10, NDCG@10, Recall@1000, where MRR is the reciprocal rank of the first disease-consistent retrieved case, P@10 and Recall@1000 treat $\mathrm{rel}(q, r) \geq 1$ as relevant, and NDCG@10 uses the graded labels $\mathrm{rel}(q, r) \in \{0, 1, 2\}$, (2) *Clinical consistency*: normalized disease-level similarity $\mathrm{rel}(q, r)/2$, measured over the top-ranked retrieved cases and aligned with the disease-centric objective in Section B, (3) *Trajectory-level indicators*: planning completeness E_{plan}, intent-constraint coherence E_{coh}, candidate coverage E_{cand}, re-ranking improvement E_{rank}, tool-orchestration appropriateness E_{orch}, reasoning consistency E_{cons}, and trajectory total score $\Phi(\tau)$ (Section 2.3), (4) *System-level*: performance versus memory size (learning curve), reported with both final result signals and intermediate process indicators, and (5) *Diagnosis-masked downstream few-shot RAG metrics*: zero-shot Top-1 accuracy for the vanilla baseline, Top-1 few-shot accuracy and Top-3 few-shot accuracy for retrieval-conditioned prediction, and retrieval MRR for the corresponding retrieved case lists.

S Experimental implementation details

PatientSearch-SE is implemented as a three-stage pipeline consisting of a planner, a searcher, and an experience consolidator. For the main experiments, all three modules are instantiated with the same Qwen3 backbone, using two model scales: Qwen3-8B and Qwen3-30B. That is, the 8B setting uses Qwen3-8B for planner, searcher, and consolidator, and the 30B setting uses Qwen3-30B for all three roles. This shared-backbone design follows the same spirit as Training-Free GRPO, where experience summarization and update are carried out by the same underlying model family rather than by a separately trained consolidator. The planner extracts patient-state information, predicts per-dimension sufficiency labels, infers retrieval intent, and constructs a similarity constraint set. The searcher instantiates these outputs as an execution graph over retrieve, aggregate, and rerank actions. The consolidator evaluates completed trajectories with the objective in Section 2.3 and writes only high-quality, leakage-controlled abstractions into long-term memory.

At the retrieval layer, the implementation follows the representation and tool decomposition described in Appendices C and D. Numeric fields are handled by range-style matching, terminology-heavy fields by biomedical semantic matching, and long-form clinical text by long-context document comparison. Composite clinical tools aggregate these primitives at the field level, and the final ranking score fuses retrieval-stage and re-ranking-stage evidence according to Equation 8. All baselines are evaluated under the same split protocol and candidate pool construction rules unless the baseline's native formulation requires a different inference interface, in which case we preserve the baseline interface but keep the evaluation target identical.

The prompt structure, candidate-budget values, fusion weights, threshold choices, and memory-update hyperparameters are specified in Appendix J so that the full experimental pipeline can be reconstructed end to end.

T Structured baseline coverage

Several reviewer-facing questions concern why we do not report a full heterogeneous graph neural network or longitudinal sequence model as an additional row in Table 1. We treat this as a real limitation, but also as an interface issue rather than a simple missing implementation. The primary evaluation interface in this paper is natural-language case search: PMC-Patients contains case-report summaries and MIMIC-IV is converted into discharge-summary case records. Many graph or sequence patient-similarity models assume a different native input, such as visit-level event sequences, typed diagnosis/procedure/medication graphs, or longitudinal code tensors. Running them directly would either require discarding the free-text query setting or adding a separate corpus construction pipeline whose assumptions are not shared by the other baselines.

To ensure a matched-interface comparison, Table 1 uses external baselines that can consume the same query and corpus fields, including sparse retrieval, biomedical dense encoders, reasoning-augmented retrievers, and end-to-end agentic retrieval systems. The two internal rows in the Ours block separate structured representation from the full retrieval policy. The patient-state-only variant extracts the seven patient-state dimensions in Equation 1, encodes each dimension separately, and fuses dimension scores with fixed validation-set weights. The structured field-aware retriever adds deterministic field-aware weighting and missingness handling, but still uses a non-agentic retrieval policy without sufficiency-state planning, hypothesis completion, trajectory memory, or adaptive tool-chain selection. Together, these rows test whether explicit clinical-dimension encoding and a stronger heuristic structured retriever close the gap to full PatientSearch-SE.

This means our empirical claim should be read narrowly: PatientSearch-SE reports higher metric values than the same-interface sparse, dense, biomedical, reasoning, agentic, and internal structured retrievers tested here under retrospective disease-centric case search, while the two internal ablations show the contribution of explicit clinical-dimension structuring and heuristic missingness-aware weighting inside our own method family. We do not claim to dominate all graph or sequence patient-similarity models under their native structured-EHR settings. Incorporating those encoders as additional atomic tools is a natural extension of the hierarchy in Section 2.2.

U Trajectory-indicator audit protocol

The trajectory indicators in Section I are intended to make the retrieval process inspectable, not to serve as independent clinical-truth labels. Table 12 lists the audit protocol used when interpreting them.

For each reported run, the rule libraries for E_{coh} and E_{cons} are fixed before evaluation. Indicators that depend on disease-level labels, such as E_{cand} and E_{rank}, are computed only after retrieval and are not exposed to the planner or searcher. During manual trace inspection, we focus on disagreement cases where the final retrieval metric and the process score diverge: high final relevance with low process score often indicates a lucky retrieval path, while low final relevance with high process score often indicates insufficient corpus coverage or an incorrect disease-family mapping. These cases are the most useful for refining the metric definitions, and we leave external expert calibration as future work.

Table 12: Audit checklist for trajectory indicators. The checks are designed to make the process probes falsifiable from logged retrieval traces. They are not a substitute for external clinician validation.

Indicator	Logged evidence	Audit question	Common failure mode
E_{plan}	Extracted patient-state fields, sufficiency scores, hypothesis list	Are sufficient fields actually supported by query text, and are hypotheses conditioned on observed dimensions?	Unsupported hypothesis completion
E_{coh}	Intent label, matching modes, fixed intent rule library	Do constraints follow the declared intent and sufficiency state?	Intent–tool mismatch
E_{cand}	Candidate pool before re-ranking, disease-family relevance labels	Does candidate generation recover at least one disease-consistent case before re-ranking?	Over-narrow candidate pool
E_{rank}	Candidate order before and after re-ranking	Does re-ranking move disease-consistent cases upward rather than merely reshuffle irrelevant cases?	Re-ranking without useful gain
E_{orch}	Tool sequence, latency, wasted-call estimate	Are activated tools justified by dimension type and do they change coverage or ranking enough to justify cost?	Redundant tool calls
E_{cons}	Execution graph order and constraint updates	Does the trace preserve retrieve→aggregate→rerank ordering unless a new constraint update is issued?	Inconsistent rerank-before-retrieve loop

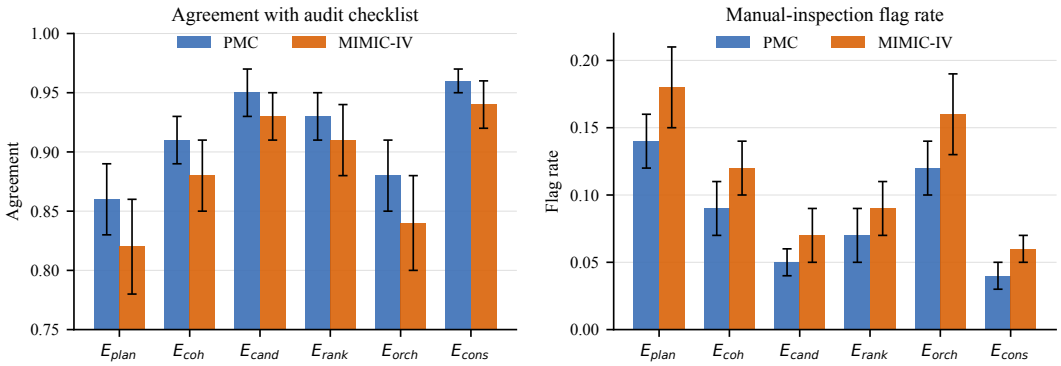


Figure 5: Trajectory-indicator audit results on 200 sampled trajectories per dataset. Agreement denotes consistency between the automatic indicator judgment and the checklist in Table 12. Flag rate is the fraction of traces marked for manual inspection.

Figure 5 reports the trajectory audit results on 200 sampled trajectories per dataset. Agreement rates are higher for trace-observable indicators such as execution order and candidate coverage, and lower for hypothesis-related indicators where independent clinical judgment would be more informative.

MIMIC-IV receives lower agreement and higher flag rates because discharge summaries are noisier than curated case reports. The two indicators most likely to require external validation are E_{plan} and E_{orch}, which depend on whether hypotheses and tool choices are clinically appropriate rather than merely trace-consistent.

V Memory scalability audit

External memory can improve adaptation but can also grow into an inefficient case cache. PatientSearch-SE therefore restricts writes and reads in four ways: (1) the consolidator removes raw query strings, identifiers, exact held-out labels, and full ranked lists, (2) new disease cards are added only when no existing card passes the similarity threshold and the trajectory passes $\Phi(\tau) > \xi$, (3)

duplicate, contradicted, or unused cards are merged or deleted, and (4) only the top-J matched cards are injected into a session. Table 2 reports the resulting card count, template count, injected-card count, and lookup overhead alongside retrieval quality. The curve is sublinear in the number of processed cases because many high-quality trajectories update existing disease cards or strategy templates rather than creating new entries. Lookup overhead remains below 50 ms at the mature phase because only card embeddings and metadata are matched before the top-J injection step.

We do not interpret monotonic memory growth as inherently good. A useful memory system should improve retrieval while keeping the number of injected cards and lookup overhead bounded. If the card store grows while R_{final} or E_{orch} plateaus, the appropriate response is more aggressive compression or pruning, not unbounded accumulation. This is why memory bloat is included in the limitations and why the current results should be read as retrospective evidence of memory-guided adaptation rather than proof of deployment-scale continual learning.

W Stepwise trajectory example

The qualitative examples in Appendix O show final retrieved evidence. Table 13 shows the complementary pathwise view used to inspect whether sufficiency states actually changed retrieval behavior.

Table 13: Stepwise trace template for an evidence-incomplete query. The concrete disease label is hidden from the planner and searcher. It is used only after retrieval for evaluation.

Stage	State	Action	Audit signal
Patient-state extraction	Symptoms and demographics observed, diagnosis and treatment absent	Build $\mathcal{P}(q)$ over seven dimensions	Check extracted fields against query spans
Sufficiency prediction	$z_{\texttt{sym}} = 1$, $z_{\texttt{demo}} = 1$, $z_{\texttt{diag}} = 0$, $z_{\texttt{treat}} = 0$	Mark symptoms/demographics for direct matching and diagnosis/treatment for hypothesis completion	E_{plan}, E_{coh}
Hypothesis generation	Plausible disease-family hypotheses from observed symptoms and demographics	Create $\mathcal{H}_{\texttt{diag}}$ and treatment-context hypotheses without copying held-out label	Unsupported-hypothesis check
Tool orchestration	Direct symptom tool, demographic filter, hypothesis-augmented diagnosis search, cross-field reranker	Build candidate pool and re-rank by intent-aware compatibility	E_{cand}, E_{orch}
Final ranking	Disease-consistent cases move upward after re-ranking	Return Top-K list for disease-level evaluation	E_{rank}, R_{final}

This pathwise presentation is useful because it exposes failures hidden by final accuracy. For example, a correct Top-1 result with unsupported diagnosis hypotheses would still be flagged by the audit, while a low final score with coherent planning may indicate that the corpus lacks disease-consistent candidates under the current query.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See the Abstract and Section 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 4.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not present theoretical results; see Section 2 for definitions.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Section 3 and Appendices J–S.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Code will be released after publication; see Appendix J.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 3 and Appendices N, P, J, and S.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See Table 1, Figure 3, and Table 3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix J and Appendix V.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: See Section 5.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Section 5.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No high-risk model or dataset is released; see Section 5.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: See Appendix N and Appendix P.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new asset is released at submission time; see Appendix N.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing or new human-subjects study is conducted; see Section 3.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No new human-subjects study is conducted; see Section 3.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Qwen3 models implement the planner, searcher, and trajectory memory consolidator, and a frozen Qwen3-8B-Instruct model serves as the downstream predictor. Their roles, configurations, and prompt templates are described in Section 2 and Appendices J, K, Q, and S.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.