SSE-Bio: A Structured Self-Evolving Agent with Agentic Retrieval Policy for Multi-Hop Biomedical Reasoning

Anonymous ACL submission

Abstract

Biomedical multi-hop question answering (QA) requires models to connect evidence across intermediate entities such as diseases, drugs, proteins, and phenotypes. Existing agents typically rely on static retrieval workflows or coarse-grained prompt rewriting, which can lead to instruction drift when reasoning procedures need to be updated. We propose **SSE-Bio**, a structured self-evolving agent with an agentic retrieval policy for multi-hop biomedical reasoning. Instead of globally rewriting agent instructions, SSE-Bio maintains a structured state, selectively retrieves knowledge triplets and prior templates through a trainable proxy policy, and improves its reasoning memory through fine-grained template editing. To optimise retrieval decisions, we introduce a proxy-training strategy based on group relative policy optimization, where the proxy is improved through decision-contrastive groups over alternative retrieval choices. Experiments on three biomedical multi-hop QA benchmarks show that SSE-Bio consistently outperforms existing baselines, achieving an improvement of 6.56 absolute points over the strongest self-evolving baseline on BioHopR.[1]

1 Introduction

Biomedical multi-hop question answering (QA) task is important for evidence-grounded biomedical discovery, such as disease understanding and drug repurposing (Xiang et al., 2025). It requires models to answer questions by reasoning across multiple entities, such as diseases, drugs, and proteins, connected through complex biomedical relations (Valizadeh and Parde, 2022; Wang et al., 2025). For example, a disease-oriented biomedical question may require a system to first identify the bridge protein associated with the disease and then infer the drug linked to that protein. It is challenging because biomedical evidence is highly specialised and relationally structured, so errors in resolving the intermediate bridge entity can mislead the subsequent reasoning chain, while in multi-answer settings the system must avoid prematurely stopping after finding only a subset of valid answers (Li and Du, 2023; Chen et al., 2025).

Recent retrieval-augmented generation (RAG) models have improved the performance of multi-hop biomedical QA by incorporating external knowledge into the reasoning process (He et al., 2024; Cao et al., 2024; Rezaei et al., 2025). However, without explicit verification and feedback over the reasoning path, they may leave intermediate bridge entities unresolved, causing the reasoning chain to drift and potentially returning incomplete answers in multi-answer scenarios. Interactive multi-agent systems make this process more flexible by enabling step-wise planning, execution, and feedback, while self-evolving agents further introduce long-term memory to adapt reasoning strategies across questions (Srivastava et al., 2025; Gao et al., 2025). Yet these agentic systems still face two limitations. First, they often rely on static retrieval workflows rather than explicitly learned retrieval control, which can lead to sub-optimal performance (Huang et al., 2025a). Second, their memory updates are often coarse-grained, typically through prompt-level rewriting, so that a local failure may alter the entire reasoning scaffold (Jin et al., 2025; Feng et al., 2026a). These limitations motivate a more controllable framework that combines explicit retrieval control with fine-grained memory evolution.

To address these limitations, we propose **SSE-Bio**, a structured self-evolving agent with an agentic retrieval policy for biomedical multi-hop QA. As shown in Fig. 1, SSE-Bio combines structured reasoning memory with explicit retrieval management in a unified agentic framework. During inference, the **Manager** converts the structured state

[1]The source code for SSE-Bio is publicly available at https://anonymous.4open.science/r/SSE-Bio-A67A.

into a query-specific plan, which the **Dev** agent executes with the retrieved triplets, while the **Critic** provides structured feedback for iterative refinement. The **Manager** maintains two forms of memory: a structured state that captures the current reasoning status and serves as the agent's short-term memory, and a long-term memory of reusable templates distilled from prior successful cases. Crucially, instead of evolving memory through global rewriting, SSE-Bio updates templates through fine-grained editing, enabling local and auditable memory evolution that is designed to mitigate hallucination and instruction-drift risks from unconstrained template rewriting. In addition, SSE-Bio introduces a trainable **Proxy** model to manage retrieval explicitly by deciding whether knowledge triplets and/or prior templates should be retrieved at each reasoning step. Knowledge triplets are beneficial in the biomedical domain since they organize complex biomedical information into a clear structure, giving the Manager structured bridge entities and relation paths to inspect during multi-hop verification. In this way, SSE-Bio supports both dynamic evolution of reasoning memory over time and adaptive, selective retrieval from the current reasoning state, rather than relying on static retrieval pipelines or coarse-grained agent rewriting.

To optimise the retrieval behaviour of the **Proxy**, we adopt a two-stage training strategy. We first use supervised fine-tuning (SFT) to initialise the Proxy with retrieval decision labels that indicate whether *knowledge triplets* and/or *prior templates* should be retrieved under a given structured reasoning state. These labels are constructed by comparing alternative retrieval branches and selecting the decisions that lead to better downstream reasoning outcomes. We then further refine the Proxy with Group Relative Policy Optimization (GRPO) (Shao et al., 2024). In particular, we introduce *decision-contrastive trajectory generation*, which expands alternative retrieval actions from the same structured reasoning state into contrastive rollout branches. The optimisation is driven by an answer-grounded reward that combines final answer correctness and evidence-supported reasoning behaviour, allowing GRPO to favour retrieval decisions that are both effective and well grounded. We evaluate SSE-Bio on three biomedical reasoning benchmarks, namely BioHopR, MedHop, and Humanity's Last Exam: Biomedicine, and show that it consistently outperforms strong baselines. Our contributions are summarised as follows:

- We propose **SSE-Bio**, a structured self-evolving agent for multi-hop biomedical reasoning, which unifies short-term structured state tracking, long-term template memory, and explicit retrieval management within a single agentic framework.
- We introduce a **fine-grained template evolution** mechanism that updates retrieved templates locally from successful trajectories, reducing unnecessary changes from the whole-template rewriting used in existing self-evolving agents.
- We design a **GRPO-based optimisation strategy** for the Proxy, which couples decision-contrastive trajectory generation with answer-grounded reward signals to compare retrieval branches under matched reasoning contexts and improve retrieval decisions through grounded supervision.

2 Related Work

Biomedical RAG Models. They improve biomedical QA by incorporating external knowledge into the reasoning process (Tran et al., 2025; Fang et al., 2025). In particular, i-MedRAG (Xiong et al., 2024) strengthens biomedical QA by retrieving medical evidence to support answer generation. Similarly, MedGraphRAG (Wu et al., 2025) organises medical knowledge into graph structure so that retrieval can better reflect biomedical relations. Building on this direction, KRAGEN (Matsumoto et al., 2024) combines knowledge graphs with retrieval-augmented reasoning for biomedical problem solving. More recently, AMG-RAG (Rezaei et al., 2025) further extends this line through agentic medical graph retrieval for evidence-grounded reasoning. Together, these methods show that external knowledge can improve biomedical multi-hop QA, but they still lack explicit verification and feedback over the evolving reasoning path, making them vulnerable to incorrect bridge-entity resolution and incomplete answers in multi-answer settings.

Biomedical Multi-agent Systems. They improve complex biomedical reasoning by enabling step-wise planning, execution, and feedback through specialised agent roles (Almansoori et al., 2025; Chen et al., 2025). MDAgents (Kim et al., 2024) improves medical reasoning through collaborative specialist agents. KGARevion (Su

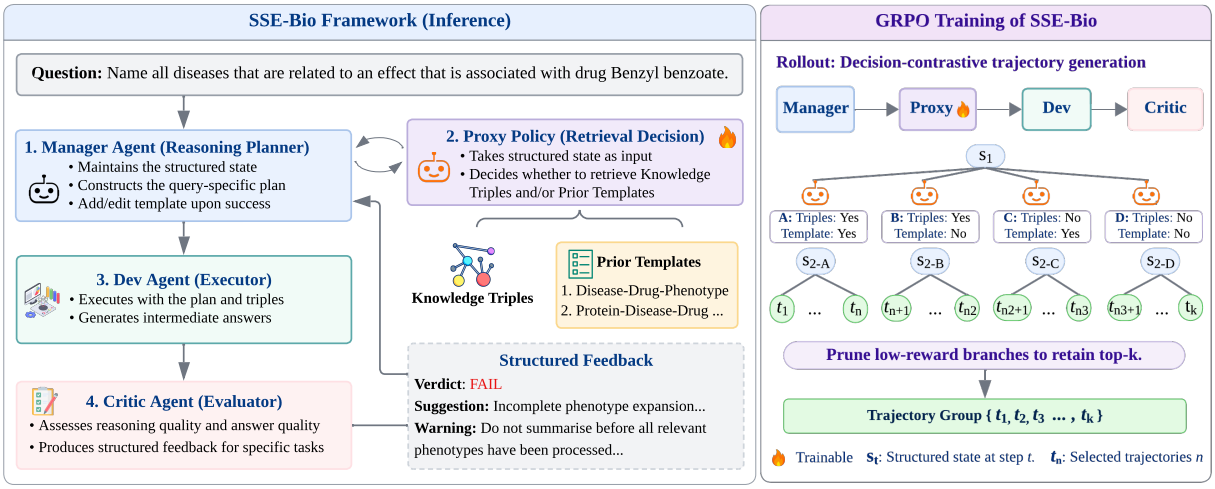

Figure 1: **Left:** The **Manager** maintains a structured state and template memory. Based on the current state, the **Proxy** decides whether to retrieve knowledge triplets and/or prior templates. The **Manager** then constructs a plan for the **Dev** agent, while the **Critic** returns structured feedback for iterative refinement. **Right:** Decision-contrastive trajectory generation explores alternative retrieval actions from the same state, and GRPO updates only the **Proxy** with an answer-grounded reward combining correctness and evidence support.

et al., 2024) introduces a knowledge-graph-based agent for knowledge-intensive biomedical QA. Biomni (Huang et al., 2025a) further extends this direction through broader biomedical tool use. Beyond such multi-agent coordination, self-evolving systems aim to improve reasoning across questions by accumulating experience from previous trajectories. STELLA (Jin et al., 2025) is a self-evolving biomedical agent that improves over time through an evolving Template Library and a dynamic Tool Ocean. Although it shows that long-term adaptation can improve biomedical reasoning, it provides limited control over when and what evidence should be retrieved, and its self-evolution updates templates by rewriting them from newly successful cases. In contrast, SSE-Bio explicitly couples state-aware retrieval control with structured long-term memory that evolves through fine-grained local editing, making both evidence use and memory evolution more bounded and controllable.

3 Methodology

We formulate SSE-Bio as an iterative agentic process in Sec. 3.1, then introduce its structured memory in Sec. 3.2, agentic retrieval policy in Sec. 3.3, and **Proxy** policy training in Sec. 3.4.

3.1 Agentic Formulation

We formulate biomedical multi-hop reasoning in SSE-Bio as an iterative agentic process involving four components: a **Manager**, a **Proxy**, a **Dev** agent, and a **Critic**. Given a biomedical question q, the goal of the agent is to resolve intermediate entities, verify the reasoning chain, and produce the answer. At reasoning round t, the **Manager** maintains a structured state u_t as the short-term memory of the current question. Conditioned on u_t, the **Proxy** selects a retrieval action a_t, which determines whether external knowledge should be introduced at the current step. The selected evidence is returned to the **Manager**, which combines it with u_t to construct a query-specific plan π_t for the **Dev** agent. SSE-Bio also maintains a long-term memory $\mathcal{E}$ of reusable templates distilled from prior successful trajectories. The **Dev** agent executes the current plan and produces a reasoning trajectory τ_t and answer $\hat{y}_t$. The **Critic**, without access to the gold answer, assesses whether the trajectory and answer are coherent and sufficiently supported, and returns structured feedback f_t for refinement. If the current reasoning fails, the **Manager** updates u_t and replans the next round. If it succeeds, SSE-Bio updates $\mathcal{E}$ through fine-grained template editing or new-template distillation. Thus, SSE-Bio evolves through short-term structured state tracking within each question and long-term memory evolution across questions.

3.2 Structured Memory

Short-term Memory. At each reasoning step, the **Manager** maintains a structured state u_t as the short-term memory of the current question. Concretely, *task type* denotes the relation pattern of the current question, *information gap* specifies the

missing evidence needed for the next reasoning step, *feedback* records the Critic's structured comments on the current trajectory, and *retrieval state* summarises what external evidence has already been retrieved. This state compactly represents the current reasoning state and makes retrieval decisions, planning, and refinement all grounded in the same explicit structure rather than scattered across free-form intermediate text. The structured state is updated across rounds to reflect reasoning progress and the structured feedback returned by the **Critic**. An example is provided in Fig. 3.

Long-term Memory Evolution. Beyond the current question, SSE-Bio maintains a long-term memory of reusable templates distilled from prior successful cases. Each template stores reusable reasoning experience through a fixed set of fields: *task type* identifies the question pattern to which the template applies, *reasoning flow* records a reusable high-level solution procedure, *verification criteria* specifies what conditions must be checked before returning an answer, *tool-use policy* specifies what external tools can be used to support the reasoning process, and *failure warning* highlights common mistakes to avoid. These templates are retrieved as planning priors for new questions. If the trajectory is aligned with a retrieved template, the **Manager** performs fine-grained template editing, updating only the local fields that require refinement. Otherwise, SSE-Bio performs new-template distillation, adding a new reusable template to memory. This design enables controllable memory evolution and constrains the kinds of broad template changes that can lead to instruction drift under coarse-grained rewriting. An example is provided in Fig. 4.

3.3 Agentic Retrieval Policy

The **Proxy** is responsible for explicitly controlling whether external knowledge should be introduced at each reasoning round. At reasoning round t, the Proxy selects a retrieval action:

$$a_t \sim \pi_\theta(\cdot \mid u_t), \qquad a_t \in \mathcal{A} = 2^{\{\mathrm{KG},\mathrm{TPL}\}}, \quad (1)$$

where KG and TPL denote knowledge-triplet retrieval and template retrieval, respectively. This yields four actions: retrieving neither source, retrieving only knowledge triplets, retrieving only templates, or retrieving both. When $\mathrm{KG} \in a_t$, SSE-Bio uses the *information gap* field in u_t as the retrieval query, encodes it together with candidate biomedical triplets using BiCA (Sinha et al., 2026), and returns the top-K triplets after schema and duplicate filtering. When $\mathrm{TPL} \in a_t$, SSE-Bio uses the *task type* encoded in u_t as the retrieval query, applies the same BiCA-based retrieval procedure to the long-term memory $\mathcal{E}$, and returns the top-K compatible template. In this way, knowledge-triplet retrieval provides step-specific factual support, while template retrieval provides a reusable reasoning prior. Appendix A.2 and A.5 report the memory schema and the retrieval budgets.

3.4 Training of the Proxy Policy

The **Proxy** is the only trainable component in SSE-Bio; the Manager, Dev, Critic, retrievers, and reasoning environment remain fixed. The goal of training is to learn a retrieval policy $\pi_\theta(a_t \mid u_t)$ that makes better retrieval decisions under the current structured reasoning state.

SFT Initialisation. We first use SFT (Wei et al., 2022; Huang et al., 2025b) to initialise the Proxy with retrieval decision pseudo-labels. For a given structured state u_t, we compare alternative retrieval branches and assign as pseudo-label the action whose rollout achieves the highest reward defined below. This stage provides a stable initialisation for the policy before reinforcement learning.

Decision Contrastive Trajectory Generation. We then refine the Proxy with GRPO (Shao et al., 2024; Park et al., 2025). SSE-Bio constructs *decision-contrastive trajectory groups* by expanding alternative retrieval actions from the same reasoning state, forming matched rollout branches as shown in Fig. 1. Branching is applied only in early reasoning rounds. Partial trajectories are pruned if they violate the required relation schema, leave the current information gap unresolved, or produce an unsupported bridge entity under the required schema; the remaining branches are ranked by intermediate answer-grounded reward signals, and only the top-K are retained for further rollout. This keeps exploration focused while avoiding exponential growth. Appendix A.6 provides the full pruning criteria and rollout-group statistics.

Reward Shaping. The Proxy is trained with an answer-grounded composite reward:

$$R(\tau) = \lambda_o R_{\mathrm{out}}(\tau) + \lambda_b R_{\mathrm{beh}}(\tau), \qquad (2)$$

where $\lambda_o + \lambda_b = 1$.

Outcome reward. The first term rewards final answer correctness:

$$R_{\text{out}}(\tau) = \mathbf{1}\{\text{corr}(\tau)\}, \tag{3}$$

where $\text{corr}(\tau)$ indicates whether the final answer generated by trajectory τ matches the gold answer of the current training question.

Behavioural reward. Final answer correctness alone is too sparse to distinguish trajectories with similar outcomes but different grounding quality. Therefore, we introduce a behavioural reward that measures whether key reasoning steps are supported by the retrieved evidence:

$$R_{\text{beh}}(\tau) = \frac{1}{|\mathcal{S}_\tau|} \sum_{s \in \mathcal{S}_\tau} \mathbf{1}\{s \text{ is evidence-supported}\}, \tag{4}$$

where $\mathcal{S}_\tau$ denotes key steps such as bridge resolution and answer verification. A step is counted as supported only when it is grounded by a retrieved triplet or is consistent with the verification criteria specified by the retrieved template: bridge resolution requires a retrieved query–bridge relation, while answer verification requires retrieved or template-supported bridge–answer evidence. This term rewards grounded use of evidence rather than additional retrieval calls, with full support rules in Appendix 8.

Group-relative Optimisation. For a retained group $\{\tau_i\}_{i=1}^K$ with rewards $\{r_i\}_{i=1}^K$, we compute the group-relative advantage as follows:

$$A_i = \frac{r_i - \mu(\mathbf{r})}{\sigma(\mathbf{r}) + \epsilon}, \tag{5}$$

where $\mathbf{r}$ denotes the rewards within the same trajectory group. Let (u_i, a_i) denote the Proxy state and retrieval action associated with trajectory τ_i. Since the Proxy outputs discrete retrieval actions rather than tokens, we apply GRPO at the action level and compare the current policy with the old sampling policy:

$$\rho_i(\theta) = \frac{\pi_\theta(a_i \mid u_i)}{\pi_{\theta_{\text{old}}}(a_i \mid u_i)}. \tag{6}$$

With $\bar{\rho}_i = \text{clip}(\rho_i, 1 - \epsilon_c, 1 + \epsilon_c)$, the Proxy is trained by maximising the action-level GRPO objective:

$$\mathcal{J}_{\text{GRPO}} = \mathbb{E}_i\left[\min(\rho_i A_i, \bar{\rho}_i A_i) - \beta_{\text{KL}} D^i_{\text{KL}}\right], \tag{7}$$

where D^i_{KL} regularises the retrieval-action distribution against the frozen SFT reference policy. Appendix A.6 reports the pruning rules and rollout statistics used for this optimisation.

4 Experimental Setup

Datasets. We evaluate SSE-Bio on three biomedical reasoning benchmarks. BioHopR (Kim et al., 2025) contains 7.63K paired instances constructed from complex biomedical relations, and we use a 7:3 train/test split, yielding approximately 5.34K training pairs and 2.29K test pairs. Each instance consists of a linked 1-hop and 2-hop question pair, and the benchmark supports both single-answer and multi-answer evaluation. We further assess cross-benchmark generalisation in a zero-shot setting on MedHop (Welbl et al., 2018) and Humanity's Last Exam: Biomedicine (HLE) (Phan et al., 2025). Appendix A.7 provides further details.

Baselines. We compare SSE-Bio against representative baselines from five categories: general-purpose LLMs (Grattafiori et al., 2024; OpenAI, 2024), medical LLMs (Chen et al., 2024), retrieval-augmented generation (Xiong et al., 2024; Rezaei et al., 2025) methods, agent-based systems (Su et al., 2024; Feng et al., 2026b; Kim et al., 2024; Huang et al., 2025a), and self-evolving agents (Jin et al., 2025), which together cover non-agentic, retrieval-augmented, and self-evolving reasoning settings. For a fair comparison, all agentic baselines use the same backbone configuration as SSE-Bio: Gemini-2.5-Pro for Manager/Critic-style roles and Claude-4.5-Sonnet for execution. Appendix A.9 provides details of each baseline.

Evaluation. For BioHopR, we follow the benchmark protocol and evaluate both single-answer and multi-answer instances using four metrics: **Prec**$_{H1}$, **Prec**$_{H2}$, **Both**$_{cor}$, and **Both**$_{wr}$. Prec$_{H1}$ and Prec$_{H2}$ measure precision on the linked 1-hop and 2-hop questions, while Both$_{cor}$ and Both$_{wr}$ measure the proportions of paired instances where both questions are answered correctly or incorrectly. Correctness is computed by embedding-based precision with cosine-similarity matching between predictions and gold answers. Higher values are better for Prec$_{H1}$, Prec$_{H2}$, and Both$_{cor}$, whereas lower values are better for Both$_{wr}$. For MedHop and HLE, we report zero-shot test accuracy. Appendix A.8 provides further details.

Implementation details. We fix Gemini-2.5-Pro as the Manager and Critic and Claude-4.5-Sonnet as the Dev agent, so performance differences can be attributed to retrieval-policy learning rather than changes in the reasoning backbones. SSE-Bio is trained on BioHopR and then transferred directly to

Family	Method	Single Answer				Multi Answer			
		Prec_{H1}	Prec_{H2}	Both_{cor} ↑	Both_{wr} ↓	Prec_{H1}	Prec_{H2}	Both_{cor} ↑	Both_{wr} ↓
LLM	Llama-3.1-8B	0.13	0.04	0.00	99.83	0.07	0.01	0.00	99.92
LLM	Llama-3.1-70B	26.41	9.45	4.94	69.06	19.82	6.94	3.24	76.51
LLM	GPT-4o	32.86	14.59	7.84	60.43	25.48	10.71	5.25	69.03
Medical LLM	HuatuoGPT-70B	0.15	0.00	0.00	99.84	0.09	0.00	0.00	99.91
Medical LLM	HuatuoGPT-8B	0.21	0.03	0.00	99.76	0.11	0.01	0.00	99.88
Medical LLM	UltraMedical-8B	13.77	5.19	2.29	83.30	9.44	3.29	1.36	88.60
RAG Model	i-MedRAG	25.20	7.48	4.43	71.75	16.86	5.73	2.49	79.90
RAG Model	AMG-RAG	28.13	8.94	5.61	68.47	19.26	6.62	3.38	77.21
Biomedical Agent	KGARevion	30.84	10.62	6.78	65.32	21.47	7.82	4.41	75.12
Biomedical Agent	DoctorAgent-RL	34.86	12.74	8.12	60.52	26.18	9.36	5.72	70.18
Biomedical Agent	MedAgents	36.74	13.62	8.89	58.53	28.05	10.12	6.18	68.01
Biomedical Agent	Biomni	39.12	15.30	10.45	56.03	30.64	11.41	7.28	65.23
Self-evolving Agent	STELLA	38.94	14.89	9.96	56.13	30.12	10.98	6.91	65.81
Self-evolving Agent	**SSE-Bio**	**47.21**†	**21.38**†	**16.52**†	**47.93**†	**38.36**†	**16.24**†	**11.73**†	**57.13**†

Table 1: Performance (%) comparison on **BioHopR**. Prec_{H1} and Prec_{H2} evaluate the linked 1-hop and 2-hop questions, while Both_{cor} and Both_{wr} measure whether both questions in a pair are jointly correct or jointly wrong. † indicates a significant improvement over STELLA (paired t-test, $p < 0.05$).

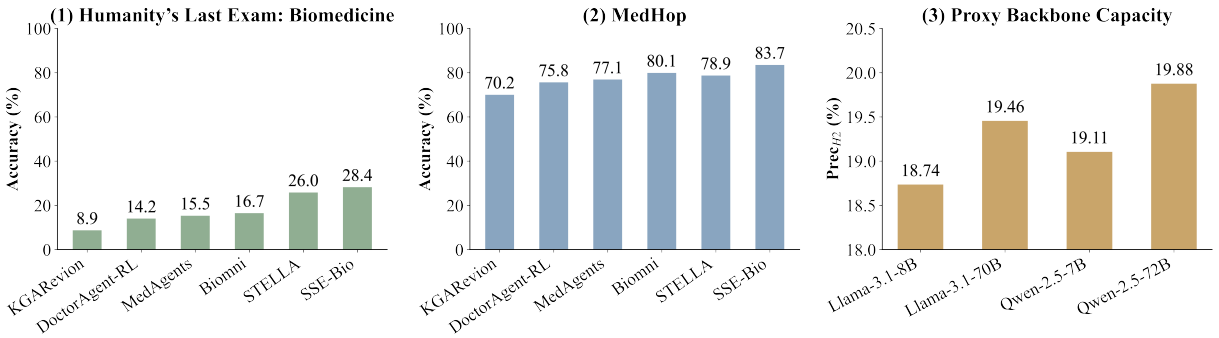

Figure 2: **(1)** Zero-shot accuracy (%) comparison with agentic baselines on HLE. **(2)** Zero-shot accuracy (%) comparison with agentic baselines on MedHop. **(3)** Performance (%) of different proxy backbones on BioHopR.

MedHop and HLE for zero-shot evaluation. During evaluation, the template memory is frozen, and no instance from the BioHopR test split, MedHop, or HLE is used to create or edit templates. Additional details are provided in Appendix A.4.

5 Results and Analysis

In this section, we answer five Research Questions (RQ) about SSE-Bio: overall performance (Sec. 5.1), the effect of the Proxy (Sec. 5.2), structured self-evolution (Sec. 5.3), retrieval headroom (Sec. 5.4), and robustness across biomedical relation types (Sec. 5.5).

5.1 RQ1: How does SSE-Bio compare with baselines?

We compare SSE-Bio with strong baselines in Table 1 and evaluate its cross-benchmark transfer in Fig. 2. Table 1 shows that SSE-Bio achieves the best performance across all four BioHopR metrics under both single-answer and multi-answer settings. In the single-answer setting, SSE-Bio significantly outperforms STELLA, on all four metrics (paired t-test, $p < 0.05$), improving from 38.94 to 47.21 on Prec_{H1}, from 14.89 to 21.38 on Prec_{H2}, and from 9.96 to 16.52 on Both_{cor}, while reducing Both_{wr} from 56.13 to 47.93. This shows that its gains extend beyond isolated hop-level prediction to paired multi-hop correctness.

In the more challenging multi-answer setting, SSE-Bio again ranks first on all four metrics, reaching 38.36 on Prec_{H1}, 16.24 on Prec_{H2}, and 11.73 on Both_{cor}, showing that its advantage is preserved under the harder answer-coverage requirement. Compared with i-MedRAG and Biomni, SSE-Bio further improves single-answer Both_{cor} by 12.09 and 6.07 points, indicating that state-aware retrieval and controllable memory evolution are more effective than fixed retrieval or generic biomedical agents. Fig. 2 further shows that SSE-Bio transfers strongly to MedHop and reaches 28.4% accuracy on HLE. These results show that SSE-Bio

Model / Setting	Backbone	Single Answer				Multi Answer			
		Prec_{H1}	Prec_{H2}	Both_{cor} ↑	Both_{wr} ↓	Prec_{H1}	Prec_{H2}	Both_{cor} ↑	Both_{wr} ↓
Effect of Structured State Tracking and Fine-grained Template Evolution									
Free-form state + rewriting	Qwen-72B	41.58	16.93	11.72	53.21	32.94	12.58	8.04	62.52
Structured state + rewriting	Qwen-72B	44.12	18.92	13.86	50.82	35.07	14.18	9.66	60.41
Free-form state + editing	Qwen-72B	45.03	19.47	14.52	50.02	35.88	14.86	10.28	59.54
Structured state + editing	Qwen-72B	**47.21**	**21.38**	**16.52**	**47.93**	**38.36**	**16.24**	**11.73**	**57.13**
Component Ablation									
w/o proxy	Qwen-72B	39.74	15.88	10.62	55.00	30.81	11.92	7.05	64.32
w/o triplets	Qwen-72B	43.91	18.64	13.64	51.09	34.72	14.02	9.41	60.67
w/o prior template	Qwen-72B	44.63	19.02	14.26	50.61	35.45	14.43	9.95	60.07
Effect of Training and Boundary Analysis									
SSE-Bio (SFT)	Qwen-8B	44.67	19.11	14.10	50.32	35.68	14.52	9.84	59.64
SSE-Bio (SFT)	Qwen-72B	45.72	19.88	15.08	49.48	36.81	15.19	10.62	58.62
SSE-Bio (GRPO)	Qwen-8B	46.08	20.64	15.62	48.90	37.24	15.71	11.08	58.13
SSE-Bio (GRPO)	Qwen-72B	**47.21**	**21.38**	**16.52**	**47.93**	**38.36**	**16.24**	**11.73**	**57.13**
SSE-Bio	Qwen-72B	47.21	21.38	16.52	47.93	38.36	16.24	11.73	57.13
+ Oracle Triplets	Qwen-72B	47.86	22.31	17.34	47.17	39.02	17.04	12.36	56.30
+ Oracle Template	Qwen-72B	47.63	21.98	17.02	47.41	38.75	16.81	12.14	56.58
+ Oracle Triplets & Template	Qwen-72B	**48.34**	**22.94**	**18.06**	**46.78**	**39.63**	**17.64**	**13.05**	**55.78**

Table 2: Ablation study and boundary analysis on **BioHopR**.

consistently outperforms strong baselines while also exhibiting generalisable reasoning ability. The token-costs are given in Appendix A.3.

5.2 RQ2: How does the Proxy affect SSE-Bio?

We evaluate whether the Proxy is necessary, whether it improves over fixed retrieval schedules, and whether GRPO further improves retrieval policy learning. In Table 2, the w/o Proxy setting, which retrieves neither knowledge triplets nor prior templates, reduces single-answer Both_{cor} from 16.52 to 10.62 and increases Both_{wr} from 47.93 to 55.00, indicating that retrieval control is important. Appendix A.11 further shows that the learned Proxy outperforms fixed retrieval schedules under the same backbone and retrieval budget, confirming that the gain is not simply due to retrieving more context. The training block in Table 2 shows that GRPO improves over SFT. With the Qwen-72B Proxy, single-answer Both_{cor} rises from 15.08 to 16.52 and Both_{wr} falls from 49.48 to 47.93, with the same trend in the multi-answer setting. Fig. 2 further shows that larger Proxy backbones yield stronger performance. Overall, the Proxy acts as an effective trainable control point for state-dependent retrieval decisions.

5.3 RQ3: What are the effects of structured self-evolution?

We evaluate structured self-evolution by comparing free-form and structured variants of short-term state tracking, together with coarse-grained rewriting and fine-grained template editing for long-term memory evolution, in Table 2. Replacing both structured state and fine-grained editing with free-form state and coarse-grained rewriting reduces single-answer Both_{cor} from 16.52 to 11.72 and increases Both_{wr} from 47.93 to 53.21. Introducing only structured state or only fine-grained editing recovers part of this gap, yielding Both_{cor} values of 13.86 and 14.52, respectively, but neither matches the full structured design. This pattern suggests that structured state tracking helps maintain an explicit reasoning status for retrieval and refinement, while fine-grained template editing updates reusable memory locally rather than rewriting the whole reasoning scaffold. Together, they provide a more controlled form of self-evolution. Appendix A.11 further supports this interpretation by showing that fine-grained editing changes fewer unrelated fields and introduces fewer unsupported or contradictory instructions than coarse rewriting.

5.4 RQ4: How much headroom remains under stronger retrieval?

We evaluate the remaining headroom of SSE-Bio from two perspectives in Table 2: the *Component Ablation* block tests whether triplets and prior templates each contribute to performance, and the *Boundary Analysis* block estimates how much further the system could improve with stronger evidence and template selection. In the component ablation block, removing triplets or prior templates reduces single-answer Both_{cor} from 16.52 to 13.64

Relation Type	STELLA				SSE-Bio			
	Single Answer		Multi Answer		Single Answer		Multi Answer	
	Prec_{H1}	Prec_{H2}	Prec_{H1}	Prec_{H2}	Prec_{H1}	Prec_{H2}	Prec_{H1}	Prec_{H2}
Same Query and Bridge								
Disease:Drug:Phenotype	48.27	22.83	38.64	10.58	59.12†	34.26†	47.03†	16.82†
Disease:Drug:Protein	46.75	10.36	37.42	4.37	52.18†	12.41†	41.02†	5.06†
Drug:Disease:Phenotype	50.18	20.67	33.15	8.74	58.74†	27.35†	37.91†	11.96†
Drug:Disease:Protein	48.86	11.94	31.83	3.89	50.67†	13.18†	33.24†	4.52†
Same Query and Target								
Disease:Phenotype:Drug	27.74	16.13	15.68	6.47	30.12†	18.42†	17.08†	7.31†
Disease:Protein:Drug	32.96	11.38	14.25	7.54	41.76†	18.93†	19.34†	12.84†
Drug:Phenotype:Disease	25.97	8.04	13.76	4.48	27.18†	9.07†	14.91†	5.18†
Drug:Protein:Disease	30.43	17.35	20.87	7.49	38.84†	25.96†	26.72†	11.83†
Phenotype:Disease:Drug	10.37	13.58	8.26	11.27	11.54†	15.06†	8.97†	15.47†
Phenotype:Drug:Disease	23.95	6.54	14.78	4.96	31.86†	11.74†	19.52†	8.36†
Protein:Disease:Drug	34.28	7.46	15.84	4.17	36.11†	8.53†	16.92†	4.82†
Protein:Drug:Disease	29.16	5.38	21.85	4.69	37.92†	9.86†	28.04†	7.74†

Table 3: Relation-type breakdown on **BioHopR**. † indicates a significant improvement (paired t-test, $p < 0.05$).

and 14.26, respectively, showing that factual evidence and reusable reasoning priors are complementary rather than redundant. In the boundary analysis block of Table 2, oracle triplets improve single-answer Both_{cor} to 17.34, oracle templates improve it to 17.02, and combining both reaches 18.06 while reducing Both_{wr} to 46.78. The multi-answer setting follows the same pattern, with combined oracle retrieval increasing Both_{cor} from 11.73 to 13.05 and reducing Both_{wr} from 57.13 to 55.78. These results show that SSE-Bio already learns useful retrieval behaviour, but still has measurable headroom under stronger evidence and template retrieval.

5.5 RQ5: How does SSE-Bio perform across different relation types?

To assess whether SSE-Bio's gains generalise across biomedical reasoning patterns, we report a relation-type breakdown on BioHopR and compare SSE-Bio with STELLA. As shown in Table 3, SSE-Bio improves over STELLA across all listed relation types, suggesting that its advantage is not confined to a specific query or bridge configuration. The gains are especially clear for relations involving stronger second-hop reasoning. For example, Disease:Drug:Phenotype improves from 22.83 to 34.26 on single-answer Prec_{H2} and from 10.58 to 16.82 on multi-answer Prec_{H2}. By contrast, Drug:Disease:Protein shows more modest gains, from 11.94 to 13.18 and from 3.89 to 4.52, indicating that some relation structures remain challenging. Multi-answer scores are consistently lower than single-answer scores, reflecting the stricter requirement of retrieving multiple correct answers. Overall, SSE-Bio provides robust relation-level improvements, with larger gains on relation types requiring effective bridge tracking and second-hop reasoning. Appendix A.12 provides a concrete failure case showing how relation-specific retrieval and local verification-field editing correct an unsupported bridge-entity error.

6 Conclusions

We presented SSE-Bio, a structured self-evolving agent for biomedical multi-hop reasoning that combines state-aware retrieval control through a trainable proxy with controllable memory evolution through structured templates and fine-grained editing. This design enables adaptive retrieval and reasoning without fixed retrieval pipelines or unconstrained self-rewriting. Experiments on three benchmarks show that SSE-Bio consistently outperforms strong baselines. On BioHopR, SSE-Bio improves over the self-evolving baseline, STELLA, by 6.56 points on single-answer Both_{cor}, while also achieving the best performance under multi-answer settings. Ablation and relation-type analyses further show that these gains are supported by effective retrieval control and structured self-evolution. Overall, our results suggest that combining structured retrieval control with fine-grained self-evolution is a promising step towards more controllable biomedical reasoning agents.

7 Limitations

We identify the following limitations of our work. (1) SSE-Bio relies on retrieved biomedical triplets and prior templates to support multi-hop reasoning. Although our retrieval design is effective, the oracle analysis shows that stronger evidence and template matching can still improve downstream performance. In future work, we plan to explore more advanced biomedical retrieval and template matching strategies. (2) SSE-Bio depends on long-term template memory to support structured self-evolution. While fine-grained editing makes memory updates more controllable than unconstrained rewriting, memory growth may still introduce redundant or outdated templates over time. Future work should therefore investigate memory consolidation, pruning, and provenance tracking. (3) Our proxy analysis shows that stronger proxy backbones consistently improve retrieval control. Although SSE-Bio already performs well with the current proxy settings, it remains an open question how far proxy scaling can further improve performance. We leave the exploration of larger and more efficient proxy architectures to future work. (4) SSE-Bio trains only the Proxy while relying on fixed commercial Manager, Dev, and Critic agents, so reproduction may be affected by prompt details and API behaviour. Moreover, SFT labels and GRPO rewards are induced from rollouts within the same fixed agent environment; future work should test whether the learned retrieval policy transfers across different agent backbones.

References

Mohammad Almansoori, Komal Kumar, and Hisham Cholakkal. 2025. Self-evolving multi-agent simulations for realistic clinical interactions. *arXiv preprint arXiv:2503.22678*.

Jiahang Cao, Jinyuan Fang, Zaiqiao Meng, and Shangsong Liang. 2024. Knowledge graph embedding: A survey from the perspective of representation spaces. *ACM Computing Surveys*, 56(6):1–42.

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

Kai Chen, Xinfeng Li, Tianpei Yang, Hewei Wang, Wei Dong, and Yang Gao. 2025. Mdteamgpt: A self-evolving llm-based multi-agent framework for multi-disciplinary team medical consultation. *arXiv preprint arXiv:2503.13856*.

Jinyuan Fang, Zaiqiao Meng, and Craig Macdonald. 2025. Kirag: Knowledge-driven iterative retriever for enhancing retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18969–18985.

Xinshun Feng, Xinhao Song, Lijun Li, Gongshen Liu, and Jing Shao. 2026a. Searl: Joint optimization of policy and tool graph memory for self-evolving agents. *arXiv preprint arXiv:2604.07791*.

Yichun Feng, Jiawei Wang, Lu Zhou, Zhen Lei, and Yixue Li. 2026b. Doctoragent-rl: A multi-agent collaborative reinforcement learning system for multi-turn clinical dialogue. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16952–16956. IEEE.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, and 1 others. 2025. A survey of self-evolving agents: On path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 1.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907.

Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Yusuf Roohani, Ryan Li, Lin Qiu, Gavin Li, Junze Zhang, and 1 others. 2025a. Biomni: A general-purpose biomedical ai agent. *bioRxiv*.

Ziyang Huang, Wangtao Sun, Jun Zhao, and Kang Liu. 2025b. Improve rule retrieval and reasoning with self-induction and relevance reestimate. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5473–5488.

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

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik S Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae W Park. 2024. Mdagents: An adaptive collaboration of llms for medical decision-making. *Advances in Neural Information Processing Systems*, 37:79410–79452.

Yunsoo Kim, Yusuf Abdulle, and Honghan Wu. 2025. BioHopR: A benchmark for multi-hop, multi-answer reasoning in biomedical domain. In *Findings of the Association for Computational Linguistics: ACL*

2025, pages 12894–12908. Association for Computational Linguistics.

Ruosen Li and Xinya Du. 2023. Leveraging structured information for explainable multi-hop question answering and reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6779–6789.

Nicholas Matsumoto, Jay Moran, Hyunjun Choi, Miguel E Hernandez, Mythreye Venkatesan, Paul Wang, and Jason H Moore. 2024. Kragen: a knowledge graph-enhanced rag framework for biomedical problem solving using large language models. *Bioinformatics*, 40(6):btae353.

OpenAI. 2024. Gpt-4o system card. https://openai.com/index/gpt-4o-system-card/. Accessed: 2026-05-14.

Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman E Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. 2025. Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30215–30248.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, and 1 others. 2025. Humanity's last exam. *Preprint*, arXiv:2501.14249.

Mohammad Reza Rezaei, Reza Saadati Fard, Jayson Lee Parker, Rahul G Krishnan, and Milad Lankarany. 2025. Agentic medical knowledge graphs enhance medical question answering: Bridging the gap between LLMs and evolving medical knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12682–12701, Suzhou, China. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Aarush Sinha, Roshan Balaji, Nirav Pravinbhai Bhatt, and 1 others. 2026. Bica: Effective biomedical dense retrieval with citation-aware hard negatives. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33010–33018.

Gaurav Srivastava, Zhenyu Bi, Meng Lu, and Xuan Wang. 2025. Debate, train, evolve: Self-evolution of language model reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32752–32798.

Xiaorui Su, Yibo Wang, Shanghua Gao, Xiaolong Liu, Valentina Giunchiglia, Djork-Arné Clevert, and Marinka Zitnik. 2024. Kgarevion: an ai agent for knowledge-intensive biomedical qa. *The Fourteenth International Conference on Learning Representations*.

Hieu Tran, Zonghai Yao, Zhichao Yang, Junda Wang, Yifan Zhang, Shuo Han, Feiyun Ouyang, and Hong Yu. 2025. Rare: Retrieval-augmented reasoning enhancement for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18305–18330.

Mina Valizadeh and Natalie Parde. 2022. The ai doctor is in: A survey of task-oriented dialogue systems for healthcare applications. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6638–6660.

Yinuo Wang, Baiyang Wang, Robert Mercer, Frank Rudzicz, Sudipta Singha Roy, Pengjie Ren, Zhumin Chen, and Xindi Wang. 2025. Trustworthy medical question answering: An evaluation-centric survey. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27477–27490.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, Yueming Jin, and Vicente Grau. 2025. Medical graph RAG: Evidence-based medical large language model via graph retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28443–28467, Vienna, Austria. Association for Computational Linguistics.

Chengrui Xiang, Tengfei Ma, Xiangzheng Fu, Yiping Liu, Bosheng Song, and Xiangxiang Zeng. 2025. From knowledge to treatment: Large language model assisted biomedical concept representation for drug repurposing. *arXiv preprint arXiv:2510.12181*.

Guangzhi Xiong, Qiao Jin, Xiao Wang, Minjia Zhang, Zhiyong Lu, and Aidong Zhang. 2024. Improving retrieval-augmented generation in medicine with iterative follow-up questions. In *Biocomputing 2025: Proceedings of the Pacific Symposium*, pages 199–214. World Scientific.

A Appendix

A.1 Computing Hardware

All experiments were conducted on a compute node equipped with six **NVIDIA GH200 Grace Hopper superchips**. This hardware provided sufficient memory and computational capacity for the training and inference settings used in this work.

A.2 Examples of Structured Memory

To make the structured memory design in Sec. 3.2 more concrete, we provide the field schema and two illustrative examples. Table 4 defines the fields used in SSE-Bio's short-term structured state and long-term template memory. The short-term state captures the current reasoning status for retrieval control and replanning, while the long-term template stores reusable reasoning priors that can be retrieved and locally edited for future questions. Figures 3 and 4 instantiate these definitions with concrete examples of a structured state and a fine-grained template update.

Figure 3 shows a representative structured state used as short-term memory during iterative reasoning. It illustrates how the **Manager** organises the task type, information gap, Critic feedback, and retrieval state into a compact representation for retrieval control and next-step planning.

Figure 4 shows an example of long-term template memory and its fine-grained update after a successful trajectory. The example highlights how SSE-Bio keeps the task type fixed while locally editing fields such as the reasoning flow, verification criteria, tool-use policy, and failure warning, rather than rewriting the entire template.

Memory	Field	Role
State u_t	Task type	template retrieval query
State u_t	Information gap	triplet retrieval query
State u_t	Feedback	Critic guidance for refinement
State u_t	Retrieval state	retrieved evidence summary
Template x	Task type	template retrieval key
Template x	Reasoning flow	reusable reasoning steps
Template x	Verification criteria	answer-checking requirements
Template x	Tool-use policy	external tools available for supporting the reasoning process
Template x	Failure warning	common failure pattern

Table 4: Field schema of the structured state and template memory in SSE-Bio.

Example: structured state

Task type	Disease–Protein–Drug (multi answer)
Information gap	The disease is known, but the bridge proteins and the candidate drugs must still be verified exhaustively.
Feedback	The Critic requires explicit disease–protein and protein–drug support before returning the final drug set.
Retrieval state	Knowledge triplets retrieved for disease–protein; one compatible template retrieved for multi-step verification.

Figure 3: Example of the structured state used as short-term memory for retrieval control and planning.

Example: Long-term Template Memory and Fine-grained Update

Task type	Disease–Protein–Drug (multi answer)
Reasoning flow	Identify candidate proteins associated with the disease; retrieve candidate drugs linked to those proteins...
Verification criteria	Ensure each predicted drug is linked to at least one protein associated with the disease...
Tool-use policy	Python for aggregation or deduplication...
Failure warning (before)	Avoid returning a partial drug set after verifying only one plausible disease-associated protein.
Failure warning (after)	Avoid finalising the answer before checking whether alternative disease-associated proteins change the verified drug set.

Figure 4: Example of long-term template memory and its fine-grained update after a successful case. The grey row shows the original field implicated by the failure, and the blue row shows the updated field. Other fields remain unchanged.

A.3 Efficiency Details (Token Cost)

Table 5 compares SSE-Bio with the strongest agentic baselines on the BioHopR test split in terms of total token usage and single-answer Both_{cor}. SSE-Bio achieves the best paired-correctness performance, reaching 16.52 on Both_{cor}, compared with 10.45 for Biomni and 9.96 for STELLA. This gain, however, comes with higher token usage: SSE-Bio uses 28.3M tokens, compared with 23.6M for Biomni and 21.2M for STELLA. These results indi-

cate that SSE-Bio improves effectiveness substantially, while incurring additional token cost relative to the compared baselines.

Model	Tokens (M)	Both_{cor}
Biomni	23.6	10.45
STELLA	21.2	9.96
SSE-Bio	28.3	16.52

Table 5: Efficiency comparison on the BioHopR test split.

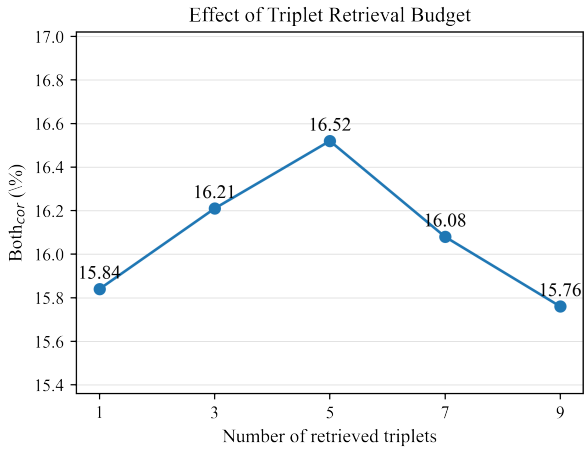

Figure 5: Effect of the triplet retrieval budget on BioHopR.

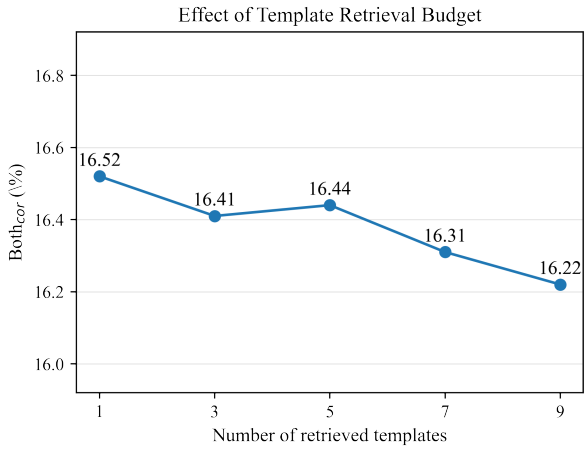

Figure 6: Effect of the template retrieval budget on BioHopR.

A.4 Implementation Details

SSE-Bio uses a four-agent architecture consisting of a Manager, a Proxy, a Dev agent, and a Critic. The Manager and Critic are implemented with Gemini-2.5-Pro, and the Dev agent uses Claude-4.5-Sonnet. These fixed commercial agents execute candidate rollouts and provide critic feedback for both SFT data construction and GRPO trajectory groups. The Proxy is the only trainable component and is instantiated with open-weight backbones, including Llama-3.1-8B/70B and Qwen-2.5-7B/72B. For compactness, result tables abbreviate Qwen-2.5-7B/72B as Qwen-8B/72B. We perform full-parameter fine-tuning for all Proxy backbones, including Qwen-2.5-72B, using the hardware described in Appendix A.1. Unless otherwise stated, reported SSE-Bio results use the Qwen-2.5-72B Proxy.

Hyperparameter	Value
Proxy backbone	Qwen-2.5-72B
SFT learning rate	1×10^{-5}
SFT batch size	64
SFT max length	2048
SFT # epochs	1
GRPO learning rate	5×10^{-7}
GRPO KL coefficient	0.002
GRPO batch size	1024
GRPO max length	2048
GRPO LR scheduler	Cosine
Reward weights (λ_o, λ_b)	$(0.8, 0.2)$
Retained trajectory cap K	8
Triplet retrieval budget k_{KG}	5
Template retrieval budget k_{TPL}	1
Template edit threshold θ_{tpl}	0.75
BioHopR matching threshold τ	0.9

Table 6: Key training and retrieval hyperparameters of SSE-Bio.

Potential risks and intended use. Although SSE-Bio is evaluated only on benchmark biomedical reasoning tasks, incorrect answers or unsupported reasoning could be harmful if the system were used directly in biomedical or clinical decision-making. SSE-Bio should therefore be treated as a research system for evidence-grounded reasoning evaluation rather than as a tool for clinical diagnosis, treatment recommendation, or patient-facing deployment.

Artifact licences and intended use. We use publicly available datasets and models in accordance with their stated research-use terms and licences. BioHopR, MedHop, Humanity's Last Exam: Biomedicine, BiCA, Llama, Qwen, and the compared baseline systems are used only for research evaluation. Commercial LLM APIs, including Gemini-2.5-Pro and Claude-4.5-Sonnet, are used according to their provider terms of service. The released SSE-Bio code will be distributed for research use.

A.5 Agentic Retrieval

For KG retrieval, SSE-Bio uses the *information gap* field in the current state u_t as the query. Candidate biomedical triplets are linearised as head–relation–tail records with entity aliases, encoded with BiCA (Sinha et al., 2026), and ranked by cosine similarity. SSE-Bio then removes schema-incompatible results and duplicate entity pairs, retaining the top-5 triplets for the Manager. For template retrieval, SSE-Bio uses the *task type* field in u_t as the query and retrieves the top-1 compatible template from $\mathcal{E}$. Compatibility is determined by dense BiCA similarity plus symbolic schema and answer-cardinality checks:

$$s_{\text{tpl}}(u_t, x) = 0.60\, c_{\text{tpl}} + 0.25\, m_{\text{schema}} + 0.15\, m_{\text{ans}}, \tag{8}$$

where $c_{\text{tpl}} = \cos(\phi(\psi(u_t)), \phi(\psi(x)))$, $\psi(\cdot)$ extracts the task-type descriptor, m_{schema} indicates relation-schema compatibility, and m_{ans} indicates whether the template has the same answer cardinality. We conduct a threshold-sensitivity analysis over $\theta_{\text{tpl}} \in \{0.65, 0.70, 0.75, 0.80\}$ using training rollouts, and set $\theta_{\text{tpl}} = 0.75$ as a conservative threshold to avoid merging trajectories with only superficial task overlap. A successful trajectory is aligned with the top retrieved template only if the task type matches and $s_{\text{tpl}} \geq \theta_{\text{tpl}}$; otherwise SSE-Bio distils a new template. Fine-grained editing changes only the *reasoning flow*, *verification criteria*, *tool-use policy*, and/or *failure warning* fields identified by the Critic, while keeping the task type fixed.

We further study the sensitivity of SSE-Bio to the retrieval budgets for biomedical triplets and prior templates using training rollouts. As shown in Fig. 5, performance peaks when retrieving $k_{\text{KG}} = 5$ triplets, while larger triplet budgets introduce more redundant evidence and slightly reduce single-answer Both$_{cor}$. As shown in Fig. 6, retrieving a single template performs best, with performance remaining relatively stable across larger budgets. These results support the use of a small retrieval budget in SSE-Bio. For biomedical triplets, retrieving a moderate number of facts provides sufficient evidence without overloading the reasoning context. For templates, a single compatible template is most effective, which is consistent with the role of template memory as a planning prior rather than an evidence pool.

A.6 Proxy Policy Training

For GRPO training, each retained trajectory τ receives the reward

$$R(\tau) = 0.80 R_{\text{out}}(\tau) + 0.20 R_{\text{beh}}(\tau), \tag{9}$$

where $R_{\text{out}}(\tau) \in \{0, 1\}$ is a binary outcome reward awarded only at the conclusion of a trajectory. For BioHopR, it is 1 only when the linked 1-hop and 2-hop answers both satisfy the benchmark answer-matching criterion; otherwise it is 0. The behavioural term $R_{\text{beh}}(\tau) \in [0, 1]$ measures the fraction of required reasoning steps whose bridge or answer claim is explicitly supported by selected triplets or by a retrieved template step accepted by the Critic. This design keeps the main training signal aligned with final QA correctness while using evidence support only as a dense tie-breaker among trajectories with similar outcomes.

SFT labels are constructed from the same answer-supervised rollout pool used for GRPO group construction. For each retained structured state, we compare the four retrieval actions by the composite reward above and choose the highest-reward action as the pseudo-label. If multiple actions obtain the same outcome reward, we break ties by evidence support and then by retrieval parsimony, preferring fewer sources when the additional source does not improve support. GRPO uses all retained branches in the decision-contrastive trajectory group rather than only the best branch.

A.7 Datasets

We evaluate SSE-Bio on three biomedical reasoning benchmarks:

- **BioHopR** (Kim et al., 2025) is a biomedical benchmark for multi-hop, multi-answer reasoning over structured knowledge graphs. Built from PrimeKG, it contains linked 1-hop and 2-hop questions designed to reflect one-to-many and many-to-many biomedical relations. The dataset contains 7.63K paired instances, and we use a 7:3 train/test split, corresponding to approximately 5.34K training pairs and 2.29K test pairs.
- **MedHop** (Welbl et al., 2018) is a cross-document biomedical question answering dataset that requires models to combine evidence distributed across multiple documents in order to infer the correct answer. It was originally introduced to study multi-hop reading comprehension in the biomedical domain.

Statistic	Value	Interpretation
Initial persistent templates	0	The memory starts empty; no held-out BioHopR, MedHop, or HLE examples are used to seed templates.
Accepted post-success updates	2.47K	Training-only updates accepted after successful trajectories and critic validation.
New-template distillations	612	Updates whose task type does not align with an existing template above θ_{tpl}.
Fine-grained edit operations	1.86K	Updates applied to aligned templates without changing the task type.
Final deduplicated templates	512	Compact memory checkpoint used for all held-out evaluations.
Covered task types	24	Twelve BioHopR relation schemas under single-answer and multi-answer variants.
Templates per task type	21.3 / 19.0	Mean / median number of templates per covered task type.
Edited templates	374	73.0% of final templates receive at least one fine-grained edit.
Fields touched per edit	1.42	Average number of non-task-type fields modified per accepted edit.
Reasoning-flow edits	31.8%	Edits that revise bridge-resolution or answer-finalisation steps.
Verification-criteria edits	28.5%	Edits that strengthen hop-level grounding or answer-coverage checks.
Tool-use-policy edits	22.4%	Edits that change when triplet retrieval, template retrieval, or entity normalisation is invoked.
Failure-warning edits	17.3%	Edits that add local failure triggers and recovery actions.

Table 7: Template-memory construction and evolution statistics on the BioHopR training split. Counts are rounded after deduplication and memory validation.

- **Humanity's Last Exam: Biomedicine** (Phan et al., 2025) is a challenging subset of Humanity's Last Exam, a frontier-level benchmark designed to test closed-ended expert reasoning on difficult academic questions. HLE contains multiple-choice and short-answer questions with unambiguous, verifiable solutions, and the Biomedicine subset provides a demanding zero-shot setting for biomedical reasoning.

SSE-Bio is trained only on the BioHopR training split; the BioHopR test split, MedHop, and HLE: Biomedicine are never used for proxy tuning or template-memory construction.

A.8 Evaluation

For BioHopR, we report **Prec**$_{H1}$, **Prec**$_{H2}$, **Both**$_{cor}$, and **Both**$_{wr}$. Prec$_{H1}$ and Prec$_{H2}$ evaluate performance on the linked 1-hop and 2-hop questions, respectively, while Both$_{cor}$ and Both$_{wr}$ measure the proportions of paired instances for which both questions are correct or both are wrong.

Following BioHopR (Kim et al., 2025), answer matching first applies exact normalised entity matching. Unmatched predictions are then evaluated with embedding-based precision. Let p denote the embedding of a predicted response and $\{a_i\}_{i=1}^{n}$ denote the embeddings of the gold answers. The cosine similarity between p and a gold answer embedding a_i is defined as

$$\cos(p, a_i) = \frac{p \cdot a_i}{\|p\| \, \|a_i\|}. \tag{10}$$

A predicted response is counted as correct if

$$\max_{i \in \{1,\ldots,n\}} \cos(p, a_i) > \tau, \tag{11}$$

where τ is the cosine threshold specified in Table 6. Precision is then computed as

$$\text{Prec} = \frac{|\text{True Positives}|}{|\text{Predicted Responses}|}. \tag{12}$$

For MedHop and HLE: Biomedicine, we report accuracy.

A.9 Baselines

The main experiments compare SSE-Bio with five baseline groups: general-purpose LLMs, medical LLMs, retrieval-augmented generation systems, biomedical agent systems, and self-evolving agents. General-purpose and medical LLM baselines use the same question format and answer normalisation as SSE-Bio, but do not use SSE-Bio's structured memory or retrieval control. RAG baselines retrieve biomedical evidence before generation, but

Step type	Supported if	Not supported if
Bridge resolution	A retrieved triplet or accepted template step links the query entity to the proposed bridge with the required relation type.	The bridge is introduced only from parametric generation or by a triplet with an incompatible schema.
Answer verification	The final answer is linked to the resolved bridge by a retrieved triplet, or satisfies the retrieved template's verification criteria.	The answer is plausible but lacks explicit bridge-to-answer evidence.
Multi-answer coverage	Every returned entity has support, and the state records that candidate expansion has been exhausted within the retrieval budget.	The trajectory stops after finding only one supported entity when the task type requires a set.
No-retrieval branch	The existing retrieval state already contains the evidence needed by the current plan.	The branch avoids retrieval while the information gap remains unresolved.

Table 8: Evidence-support rules used for R_{beh}. These rules operationalise the behavioural reward in Sec. 3.4 without rewarding retrieval volume by itself.

Statistic	Value	Interpretation
BioHopR training pairs	5.34K	Training split used for trajectory construction and persistent template-memory updates.
Candidate rollouts before pruning	82.4K	Mean 15.4 rollouts per pair, close to the $4^2 = 16$ upper bound from expanding the first two retrieval decisions.
Valid parsed rollouts	76.8K	93.2% of generated rollouts have valid state, action, evidence, and answer fields.
Evidence-bearing non-duplicate rollouts	49.7K	Remaining rollouts after removing duplicate state-action paths and branches with empty selected evidence.
Retained GRPO rollouts	38.5K	Mean group size $K = 7.21$ after the $K \leq 8$ cap; median group size is 8.0.
Discarded or pruned rollouts	43.9K	53.3% of generated rollouts are removed by parsing, duplicate/evidence filters, or the group-size cap.
Gold-answer successful retained rollouts	17.3K	Retained rollouts whose final paired answer satisfies the BioHopR answer-matching criterion.
SFT-labelled state-action pairs	12.6K	Mean 2.36 labelled Proxy decisions per training pair; states without a successful answer-supervised rollout are excluded from SFT labels.
GRPO state-action pairs	38.5K	One retained state-action pair is associated with each rollout branch in the decision-contrastive trajectory groups.
SFT label: triplets + templates	43.8%	Most labelled decisions require both factual evidence and a reusable reasoning prior.
SFT label: triplets only	33.6%	Used when the current state mainly lacks biomedical bridge evidence.
SFT label: templates only	15.1%	Used when planning structure is useful but extra triplets are not needed.
SFT label: neither source	7.5%	Used when the current state already contains enough supported evidence to proceed.

Table 9: Construction statistics for SFT pseudo-labels and GRPO trajectory groups on the BioHopR training split. Counts are rounded; each training pair seeds one decision-contrastive rollout trajectory generation.

do not train a state-conditioned retrieval controller. Biomedical agent systems decompose the task or invoke biomedical tools, while self-evolving baselines update templates from previous trajectories. Biomni serves as the strongest biomedical agent baseline, and STELLA serves as the strongest self-evolving agent baseline.

- **General-purpose LLMs:** Llama-3.1-8B, Llama-3.1-70B and GPT-4o. These models serve as strong general-purpose language-model baselines without biomedical specialisation (Grattafiori et al., 2024).
- **Medical LLMs:** HuatuoGPT-o1-70B, HuatuoGPT-o1-8B, and UltraMedical-8B. These models are domain-adapted medical or biomedical language models designed for specialised biomedical reasoning (Chen et al., 2024).
- **Retrieval-augmented generation methods:** i-MedRAG and AMG-RAG. These methods combine biomedical question answering with external knowledge retrieval during infer-

ence (Xiong et al., 2024; Rezaei et al., 2025).

- **Biomedical agent systems:** KGARevion, DoctorAgent-RL, MedAgents, and Biomni. These systems use agentic decomposition, collaboration, or external biomedical tools to improve medical and biomedical reasoning (Su et al., 2024; Feng et al., 2026b; Kim et al., 2024; Huang et al., 2025a).
- **Self-evolving agents:** STELLA. This model improves reasoning behaviour over time through memory- or workflow-based evolution, and provides the most direct comparison for SSE-Bio (Jin et al., 2025).

A.10 Proxy Training and Memory Construction Statistics

Table 7 summarizes the construction and evolution of the persistent template memory. This memory starts empty and is updated from successful BioHopR training trajectories only after validation by reviewers. Most accepted updates consist of fine-tuning aligned templates, while new template distillation is employed when no compatible templates are found. After deduplication, the final memory remains compact and covers variants of all BioHopR task types, with adjustments distributed across the inference process, validation criteria, tool usage strategies, and failure warnings. These statistics support the view that SSE-Bio evolves its memory through local field-level updates rather than global prompt-level rewrites.

Table 9 reports the construction statistics for the SFT pseudo-labels and GRPO trajectory sets generated on the BioHopR training set. Each training pair provides initial conditions for decision-contrastive rolling generation by expanding alternative retrieval actions in the matched inference state. After syntactic analysis, repetition/evidence filtering, and trajectory pruning, the retained rolling results provide both state-action pairs with SFT labels and GRPO trajectory sets for agent optimization. The resulting SFT label distribution covers all four retrieval operations: simultaneous retrieval of triples and templates, retrieval of triples only, retrieval of templates only, and no retrieval from either source. This indicates that the agent model was not trained to follow a simplified, fixed retrieval strategy, but rather was supervised to make diverse retrieval decisions.

A.11 Additional Analysis of Retrieval Control and Memory Evolution

Does the Proxy outperform fixed retrieval schedules? Table 10 compares the learned Proxy with fixed retrieval policies under the same agent backbone and retrieval budget. Retrieving neither source corresponds to removing the Proxy and external retrieval, which substantially reduces performance. Fixed retrieval from a single source improves over this setting, showing that both knowledge triplets and prior templates are useful. However, always retrieving both sources remains clearly below the learned Proxy, with single-answer Both_{cor} decreasing from 16.52 to 13.18 and multi-answer Both_{cor} decreasing from 11.73 to 9.02. These results indicate that SSE-Bio's improvement is not simply due to retrieving more context; rather, training the Proxy enables state-dependent source selection that better matches the current reasoning state.

Does fine-grained editing make memory evolution more local? For the template-update quality analysis, we randomly sample 100 memory updates and compare coarse rewriting with fine-grained editing under the same template schema. *Changed fields per update* counts the average number of template fields modified by each update. *Unchanged-field preservation rate* measures the proportion of fields that should remain unchanged and are indeed preserved. *Unsupported new instruction rate* measures the proportion of updates that introduce at least one new instruction not supported by the successful trajectory, retrieved evidence, or Critic feedback. *Contradictory instruction rate* measures the proportion of updates that introduce at least one instruction conflicting with the original template, task schema, or verification criteria. All rates are computed at the update level over the 100 sampled updates.

Table 11 shows that fine-grained editing modifies fewer fields per update than coarse rewriting, reducing the average number of changed fields from 3.28 to 1.46. It also preserves unrelated fields more often, improving the unchanged-field preservation rate from 64.0% to 88.0%. Beyond locality, fine-grained editing introduces fewer unsupported new instructions and contradictory instructions, with rates decreasing from 12.0% to 4.0% and from 7.0% to 2.0%, respectively. These results support a more precise version of our claim: fine-grained editing makes template evolution more localised

Retrieval Policy	Single Answer				Multi Answer			
	Prec_{H1}	Prec_{H2}	Both_{cor}	Both_{wr}	Prec_{H1}	Prec_{H2}	Both_{cor}	Both_{wr}
Retrieve neither	39.74	15.88	10.62	55.00	30.81	11.92	7.05	64.32
Retrieve KG only	42.36	17.41	12.43	52.84	33.18	13.06	8.52	62.37
Retrieve template only	41.58	16.96	11.86	53.46	32.47	12.64	8.07	62.98
Retrieve KG + template	43.27	18.12	13.18	51.76	34.02	13.58	9.02	61.31
Learned Proxy	**47.21**	**21.38**	**16.52**	**47.93**	**38.36**	**16.24**	**11.73**	**57.13**

Table 10: Comparison of fixed retrieval strategies on BioHopR. The fixed strategies use the same Manager, Dev, Critic, retriever, and retrieval budget as SSE-Bio, but replace the trainable Proxy with a fixed retrieval decision at each reasoning round. "Retrieve neither" corresponds to the w/o proxy setting in Table 2.

Metric	Coarse rewriting	Fine-grained editing
Changed fields per update ↓	3.28	**1.46**
Unchanged-field preservation rate ↑	64.0	**88.0**
Unsupported new instruction rate ↓	12.0	**4.0**
Contradictory instruction rate ↓	7.0	**2.0**

Table 11: Template-update quality analysis on 100 randomly sampled memory updates. Rates are reported as percentages except for changed fields per update.

and easier to audit. We therefore do not claim that editing eliminates hallucinations or instruction drift; rather, the inspected cases suggest that it reduces unsupported or inconsistent template updates.

A.12 Case Study: Failure Analysis against STELLA

Table 12 illustrates a typical BioHopR case featuring a "drug → gene/protein → disease" relationship pattern. The golden intermediate entity is FOLH1, and the answer should be validated via a two-hop path from the query drug to FOLH1, and then from FOLH1 to a specific disease. The disease returned by STELLA is semantically close to the gold answer set, but its reasoning trail relies on AR as an intermediate bridge and does not verify whether the query drug is associated with that bridge. This means that even if the final disease appears reasonable, the reasoning trail is still considered invalid. In contrast, SSE-Bio retrieves relationship-specific triples, preserves the task type and reasoning process, and performs only local edits to the verification criteria fields. This requires that the final answer be supported by both the "drug-gene/protein" and "gene/protein-disease" links, enabling SSE-Bio to return a valid answer via the FOLH1 bridge.

Item	STELLA	SSE-Bio
Question pattern	Drug → Gene/Protein → Disease	Drug → Gene/Protein → Disease
Question	Name a disease that is related to a gene/protein that is associated with drug (2S)-2-{[HYDROXY(4-IODOBENZYL)PHOSPHORYL]METHYL}PENTANEDIOIC ACID.	
Gold bridge	FOLH1	
Gold answer	{neurotic disorder, colorectal neoplasm, prostate cancer, schizophrenia, dysthymic disorder, colorectal cancer, familial prostate carcinoma, colorectal carcinoma, unipolar depression, anxiety disorder, prostate carcinoma}	
Predicted bridge	AR	FOLH1
Retrieved evidence	Retrieves disease-associated context around prostate cancer and androgen-receptor signalling, but does not establish that the query drug is associated with AR.	Retrieves relation-specific triplets supporting both hops: *((2S)-2-PENTANEDIOIC ACID, associated with, FOLH1)* and *(FOLH1, related to, prostate cancer)*.
Memory/template behaviour	Coarse rewriting updates multiple fields, including reasoning flow and tool-use policy, and shifts the template towards a broad disease-association heuristic.	Local editing preserves the task type and reasoning flow, and modifies only the verification-criteria field to require both the drug–gene/protein and gene/protein–disease links to be evidence-supported.
Final answer	Prostate carcinoma	Prostate cancer
Failure/success reason	The answer is semantically close to a gold disease, but the trajectory is not valid because the selected bridge AR is not supported as the gene/protein associated with the query drug.	The answer is verified through the required two-hop path: the query drug is associated with FOLH1, and FOLH1 is related to prostate cancer.

Table 12: A BioHopR case study comparing STELLA and SSE-Bio.