

SAKE: Structured Agentic Knowledge Extrapolation for Complex LLM Reasoning via Reinforcement Learning

Jiashu He^{1*} Jinxuan Fan² Bowen Jiang¹ Ignacio Hounie¹
Dan Roth¹ Alejandro Ribeiro¹

¹University of Pennsylvania ²University of California, Berkeley
{jiashuhe, bwjiang, ihounie, danroth, aribeiro}@seas.upenn.edu
jinxuan_fan@berkeley.edu

Abstract

Knowledge extrapolation is the process of inferring novel information by combining and extending existing knowledge that is explicitly available. It is essential for solving complex questions in specialized domains where retrieving comprehensive external knowledge is impractical. We propose SAKE (Structured Agentic Knowledge Extrapolation), a RL powered agentic framework that trains LLMs to autonomously retrieve and extrapolate structured knowledge through tool-augmented reinforcement learning. SAKE defines two external KG tools: entity group construction and cross-group triplet retrieval. The model learns to interleave these 2 retrieval tools during a three-turn rollout: extracting key entities, filtering relevant concept groups, and associative reasoning by constructing new triplets through analogy. The entire pipeline is optimized end-to-end with GRPO using a reward that combines output format and answer correctness, teaching the model what to retrieve and how to reason over it. Our experiments proved that SAKE fine-tuned Qwen2.5-7B model surpasses GPT-3.5-Turbo with state-of-the-art agentic KG reasoning on both biomedical (75.4% vs. 70.1%) and commonsense (81.3% vs. 74.7%) benchmarks, while reducing token usage by over 90%. These results demonstrate that associative reasoning over incomplete structured knowledge does **not** require large models with complex, multi-step prompting, thus can be learned end-to-end by small, open-weight models through reinforcement learning with the right tools and training signal. Our code is available at <https://github.com/jxfan99/SAKE>.

1 Introduction

The ability of inferring novel facts by extending existing knowledge and accessible evidence is an essential aspect of human intelligence, especially when faced with challenging and unfamiliar questions. In this case, humans draw relation between the challenging task and the related known facts, and extrapolate through associative thinking (Beatty & Kenett, 2023; Kaufman et al., 2009). For instance, to answer whether melatonin can help treat insomnia, one may formulate the reasoning process that since melatonin is a hormone and insomnia is a mental disorder, melatonin should be able to treat insomnia as its generally known that mental disorders are treated by hormones. This kind of ability to extend beyond the available knowledge is needed by large language models (LLMs) as well. Despite recent advances in structured reasoning and planning (Xiong et al., 2025b;a), deploying AI systems in knowledge-intensive domains (Casella et al., 2023; Tian et al., 2024) remains challenging, as neither parametric knowledge from pre-training nor retrieved context at testing time could provide complete chain of reasoning. Relying on either of them leads the model to hallucination and harms the performance.

*Correspondence to jiashuhe@seas.upenn.edu.

In this paper, we propose SAKE (Structured Agentic Knowledge Extrapolation), a framework that trains LLMs to perform knowledge extrapolation over structured knowledge graphs (KGs) through tool-augmented reinforcement learning. Concretely, SAKE utilizes the structured nature of the KG as a chain of reasoning that connects the related concepts, and teach the model to *extrapolate* beyond what is explicitly retrievable. Given a query, the model autonomously retrieves relevant KG triplets, identifies structural patterns in them, and constructs new triplets by substituting semantically similar concepts. Therefore, it bridges the gap between incomplete external knowledge and the challenging corner domain question.

SAKE achieves this through an agentic, multi-turn pipeline that the model learns entirely via RL, with no pre-computed retrieval or supervised fine-tuning. The pipeline proceeds as follows. First, the model analyzes the query and extracts key concepts that it needs to analyze. It then invokes an external tool that links these concepts to a KG and returns groups of semantically similar entities. The model filters these groups for relevance, invokes a second tool that retrieves KG triplets connecting the selected groups, and finally performs associative reasoning to construct new triplets by analogy and producing a final answer. All tool invocations are triggered by special tokens that the model generates during its rollout, and the entire trajectory is optimized end-to-end with GRPO (Shao et al., 2024) using a single outcome-based reward, which propagates credits to all intermediate processes including which entities to extract and which groups to retain. This design means that the model discovers, through exploration, that accurate entity extraction, judicious group filtering, and faithful knowledge extrapolation lead to better retrieval and higher accuracy. The full pipeline is tied together through a single outcome-based reward signal.

Prior agentic frameworks have explored integrating retrieval with LLM reasoning: Search-R1 (Jin et al., 2025) and Search-O1 (Li et al., 2025) train models to issue web queries, while Graph-R1 (Luo et al., 2025) extends this to multi-turn hypergraph retrieval. AutoGraph-R1 (Tsang et al., 2025) uses RL to optimize KG construction itself, and Knowledgeable-R1 (Lin et al., 2026) addresses knowledge conflicts between parametric and retrieved knowledge. KG-based reasoning methods such as ToG (Sun et al., 2024) and GraphRAG (Edge et al., 2025) offer structured retrieval but assume complete KGs, while GIVE (He et al., 2024) addresses KG incompleteness through inference-time prompting at substantial token cost. SAKE differs fundamentally: (1) knowledge extrapolation is *learned* through RL rather than engineered through prompts, and (2) retrieval and reasoning are *unified* into a single end-to-end policy.

Our experiments demonstrate the effectiveness of SAKE across two knowledge graphs of vastly different scales: a 135-node UMLS KG for biomedical QA (PubMedQA (Jin et al., 2019), BioASQ (Krithara et al., 2023), ProcessBank (Berant et al., 2014)) and a 844K-node ConceptNet for commonsense reasoning (CommonsenseQA). SAKE consistently outperforms all baselines across prompting-based reasoning, retrieval-augmented, fine-tuning, and agentic framework categories on both model sizes including 3B and 7B. Notably, a Qwen2.5-7B model with SAKE surpasses GPT-3.5-Turbo paired with a state-of-the-art agentic KG reasoning framework on both biomedical (75.4% vs. 70.1%) and commonsense (81.3% vs. 74.7%) benchmarks, while reducing token usage by over 90% and requiring only a single inference pass. Ablation studies further confirm that each component of the pipeline: agentic retrieval, group filtering, and knowledge extrapolation contributes meaningfully, and that the gains arise from structured knowledge interaction rather than fine-tuning alone.

2 Problem Definition

Setup. Let π_θ be a policy LLM parameterized by θ , x a query with ground-truth answer a^* , and $\mathcal{G} = (\mathcal{V}_\mathcal{G}, \mathcal{R}_\mathcal{G}, \mathcal{E}_\mathcal{G})$ a knowledge graph, where $\mathcal{V}_\mathcal{G}$ is the set of entities (nodes), $\mathcal{R}_\mathcal{G}$ is the set of relations, and $\mathcal{E}_\mathcal{G} \subseteq \mathcal{V}_\mathcal{G} \times \mathcal{R}_\mathcal{G} \times \mathcal{V}_\mathcal{G}$ is the set of triplets (edges). We assume the KG may be *incomplete*: there is no guarantee that $\mathcal{E}_\mathcal{G}$ contains triplets that directly answer x . The model must therefore *extrapolate* beyond retrieved knowledge to arrive at a correct answer.

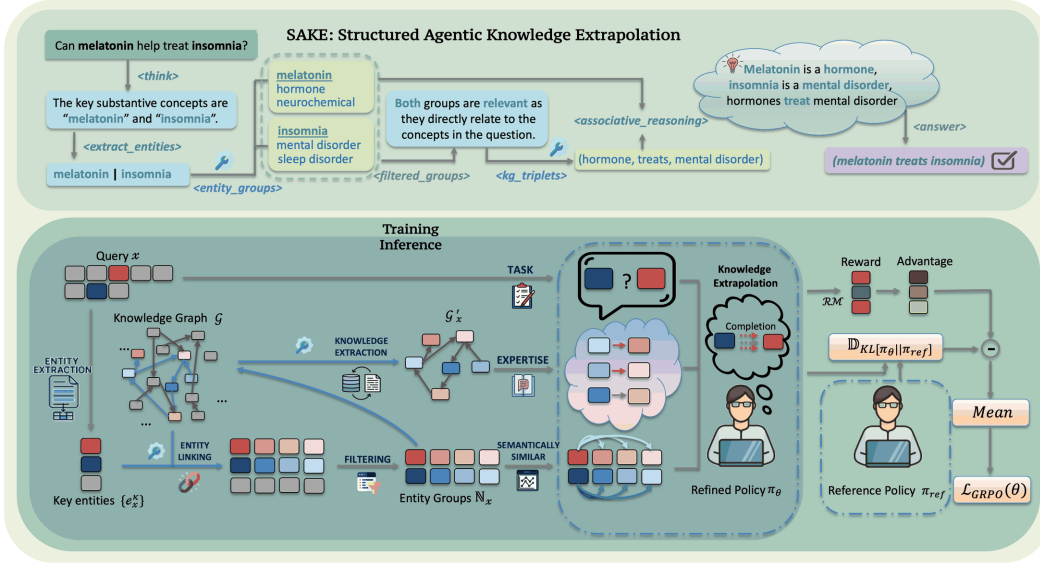


Figure 1: Overview of SAKE. **Upper:** A concrete example illustrating the three-turn agentic pipeline on a biomedical question. The model generates structured output delimited by special tokens (<extract_entities>, <filtered_groups>, <associative_reasoning>, <answer>), which trigger two external tool calls: entity group construction and cross-group triplet retrieval. The retrieved triplet (*hormone, treats, mental disorder*) is extrapolated into (*melatonin, treats, insomnia*) by substituting semantically similar concepts from the entity groups. **Lower:** The training and inference framework. For query x , the policy π_θ generates a multi-turn rollout by entity extraction, group filtering, and associative reasoning. The rollout generation process is interleaved with tool calls that interact with the knowledge graph $\mathcal{G}$. All tool invocations are triggered by the model’s own special-token outputs during the rollout; nothing is pre-computed or provided externally. The entire trajectory is optimized end-to-end with GRPO, where the reward signal flows back through all model decisions including entity extraction, group filtering and knowledge extrapolation.

Tool-Augmented Environment. We define two deterministic tool functions that the policy can invoke during generation. Together, they bridge the gap from the query’s conceptual space to the KG’s relational space:

$$\mathcal{F}_1 : E_x \rightarrow \mathbb{N}_x \quad (\text{entity linking \& group construction}) \quad (1)$$

$$\mathcal{F}_2 : \mathbb{N}_x \rightarrow \mathcal{T} \subseteq \mathcal{E}_{\mathcal{G}} \quad (\text{cross-group triplet retrieval}) \quad (2)$$

where $E_x = \{e_1, \dots, e_n\}$ denotes the set of key entities extracted from query x (which may not exist verbatim in $\mathcal{V}_{\mathcal{G}}$), $\mathbb{N}_x = \{G_1, \dots, G_n\}$ is the collection of entity groups with each $G_i \subseteq \mathcal{V}_{\mathcal{G}}$ containing e_i and its semantically similar KG concepts, and $\mathcal{T}$ is the set of retrieved cross-group triplets. $\mathcal{F}_1$ maps from the query space to the KG node space: it links each query entity to its nearest neighbors in $\mathcal{V}_{\mathcal{G}}$, handling the common case where query-specific terminology does not appear in an incomplete KG. $\mathcal{F}_2$ maps from the KG node space to the KG edge space, retrieving all triplets in $\mathcal{E}_{\mathcal{G}}$ that connect entities across different groups. Both tools are deterministic and external to the policy—they are not parameterized by θ and produce no gradient signal.

Multi-Turn Trajectory. We define a *turn* as a contiguous segment of model-generated tokens, bounded by a special token that triggers a tool call (or by the end-of-sequence token). A single rollout consists of three turns interleaved with two tool calls, producing a trajectory:

$$y = \left(\underbrace{y^{(1)}}_{\text{extract}}, \underbrace{o^{(1)}}_{\mathcal{F}_1}, \underbrace{y^{(2)}}_{\text{filter}}, \underbrace{o^{(2)}}_{\mathcal{F}_2}, \underbrace{y^{(3)}}_{\text{reason}} \right) \quad (3)$$

where each model turn $y^{(t)}$ is generated autoregressively, conditioned on the query and all preceding turns and tool outputs:

$$y^{(t)} \sim \pi_\theta \left(\cdot \mid x, y^{(1)}, o^{(1)}, \dots, y^{(t-1)}, o^{(t-1)} \right) \quad (4)$$

Tool outputs are computed deterministically from model-generated content: $o^{(1)} = \mathcal{F}_1(\text{Parse}(y^{(1)}))$ and $o^{(2)} = \mathcal{F}_2(\text{Parse}(y^{(2)}))$, where $\text{Parse}(\cdot)$ extracts structured content from the model’s tagged output. Concretely, $\text{Parse}(y^{(1)})$ yields E_x from the `<extract_entities>` tags, and $\text{Parse}(y^{(2)})$ yields a subset $S \subseteq \{1, \dots, n\}$ of selected group indices from the `<filtered_groups>` tags.

Objective. We optimize the policy end-to-end over the full trajectory:

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot | x)} [R(y, a^*)] - \beta D_{\text{KL}}(\pi_\theta(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)) \quad (5)$$

where $R(y, a^*)$ is the reward function (detailed in Section 3.5), π_{ref} is a fixed reference policy, and β controls the KL penalty. The policy is conditioned only on the query x : the retrieved entity groups $\mathbb{N}_x$ and triplets $\mathcal{T}$ are not pre-computed or provided as fixed inputs, but are determined by the model’s own outputs during the rollout. Token-level losses are computed only over model-generated tokens ($y^{(1)}, y^{(2)}, y^{(3)}$); tool outputs ($o^{(1)}, o^{(2)}$) are deterministic and masked from the loss (as described by the binary mask m in Algorithm 1). However, because the *content* of tool outputs depends on what the model generated in preceding turns, the outcome-based reward $R(y, a^*)$ implicitly supervises all upstream decisions—the model learns that better entity extraction and group filtering lead to more useful retrievals and ultimately higher accuracy.

3 SAKE

SAKE equips LLMs with associative reasoning over structured knowledge through an agentic, tool-augmented framework trained end-to-end with reinforcement learning. As illustrated in Figure 1, the model interacts with a knowledge graph (KG) through two external tools during each rollout, and learns via RL, that when and how to invoke them effectively to answer complex queries.

The framework proceeds in three generation turns interleaved with two tool calls: the model first extracts key entities from the query (Turn 1), invokes Tool 1 to obtain entity groups, filters the groups for relevance (Turn 2), invokes Tool 2 to retrieve cross-group triplets, and finally performs associative reasoning to produce an answer (Turn 3). Nothing is pre-computed or provided by a human pipeline: the model itself decides which entities to extract, which groups to retain, and how to reason over the retrieved knowledge. The entire multi-turn trajectory is optimized end-to-end with reinforcement learning. We describe each component below.

3.1 Agentic KG Interaction via Tool Use

SAKE defines two deterministic tools that serve as the model’s interface to the knowledge graph. These tools are invoked during the rollout: the model generates structured output tags, execution is paused, the tool computes a result, and the result is injected back into the rollout sequence for the model to continue generating.

Tool 1: Entity Group Construction. Given a set of extracted entities $E = \{e_1, \dots, e_n\}$ produced by the model in Turn 1, this tool links each entity to the KG and constructs an entity group containing the entity and its semantically similar siblings.

For each entity e_i , let w denote a pre-trained encoder and $\mathcal{V}_G$ the set of all KG entities. The tool retrieves the p most similar KG concepts:

$$Y_i = \underset{y \in \mathcal{V}_G}{\text{Top-}p \text{ cos}}(w(e_i), w(y)) \quad (6)$$

and returns the entity group $G_i = \{e_i\} \cup Y_i$.

The tool returns all groups $\{G_1, \dots, G_n\}$, wrapped in `<entity_groups>` tags and injected into the rollout. Each group provides the model with a neighborhood of semantically related KG concepts, enabling it to bridge the gap between query-specific terminology and the KG’s vocabulary.

Tool 2: Cross-Group Triplet Retrieval. After the model selects a subset S of relevant groups in Turn 2, this tool retrieves all KG triplets whose head and tail entities belong to different selected groups:

$$\mathcal{T} = \{(h, r, t) \in \mathcal{E}_G \mid \exists (i, j) \in S \times S, i \neq j, h \in G_i, t \in G_j\} \quad (7)$$

The retrieved triplets $\mathcal{T}$ are wrapped in `<kg_triplets>` tags and injected into the rollout. These cross-group connections provide structured evidence about how concepts from different entity groups relate to each other in the KG, serving as the foundation for associative reasoning.

3.2 Multi-Turn Rollout

The full rollout procedure is detailed in Algorithm 1 (Appendix D), including the binary token mask m that ensures GRPO loss is computed only over model-generated tokens.

Turn 1: Entity Extraction. The model receives query x and identifies key biomedical concepts, outputting them inside `<extract_entities>` tags. The model is instructed to extract specific concepts (e.g., diseases, molecules) rather than relational words (e.g., “treatment”, “cause”), as the latter do not correspond to KG entities. Tool 1 then constructs entity groups and injects them into the rollout.

Turn 2: Group Filtering. The model evaluates each returned entity group’s relevance, reasoning inside `<think>` tags and outputting selected group indices inside `<filtered_groups>` tags. This step is important: entity extraction may yield concepts whose KG neighborhoods are noisy, and the model learns through RL to discard these before triplet retrieval. Tool 2 then retrieves cross-group triplets and injects them.

Turn 3: Associative Reasoning. The model reasons over the retrieved triplets and produces a final answer. When triplets do not directly answer the query, the model constructs new triplets by substituting semantically similar concepts from entity groups into existing triplet patterns—e.g., mapping (hormone, treats, mental_disorder) to (melatonin, treats, insomnia). If no triplets are found, the model reasons using entity groups and parametric knowledge alone.

3.3 System Prompt

The system prompt (provided in full in Appendix E) defines the three-turn structure and output format for each turn. It instructs the model to extract substantive biomedical concepts (not relational words like “treatment” or “cause”), describes the associative reasoning mechanism of constructing new triplets by substituting semantically similar terms, and provides no task-specific examples. This design is to ensure that reasoning ability emerges from RL training rather than prompt engineering.

3.4 End-to-End Reinforcement Learning

The entire multi-turn trajectory is optimized jointly using GRPO (Shao et al., 2024) within the VERL (Sheng et al., 2025) framework.

Policy Optimization. Token-level losses are computed over all model-generated segments (Turns 1–3), while tool outputs are excluded from the loss computation as they are deterministic. Let $\rho_\theta(y \mid x) = \frac{\pi_\theta(y \mid x)}{\pi_{\theta_{\text{old}}}(y \mid x)}$ denote the importance ratio. The policy is optimized with:

$$\begin{aligned} \mathcal{L}_{\text{SAKE}}(\theta) = & -\mathbb{E}_{x,y} \left[\min(\rho_\theta \hat{A}, \text{clip}(\rho_\theta, 1-\epsilon, 1+\epsilon) \hat{A}) \right] \\ & - \beta \mathbb{E}_x \left[D_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}}) \right] \end{aligned} \quad (8)$$

where $y = (y^{(1)}, o^{(1)}, y^{(2)}, o^{(2)}, y^{(3)})$ denotes the full multi-turn rollout, ϵ is the clipping hyperparameter, $\hat{A}$ is the advantage estimated across rollouts within the same group, π_{ref} is a fixed reference policy, and β controls the KL penalty.

3.5 Reward Function

We use a weighted additive reward that combines structural compliance with answer correctness. Given a rollout y produced by the policy π_θ for query x with ground-truth answer a^* , the reward $R(y, a^*)$ is defined as:

$$R(y, a^*) = \lambda_{\text{fmt}} R_{\text{fmt}}(y) + \lambda_{\text{acc}} R_{\text{acc}}(y, a^*) \quad (9)$$

where $\lambda_{\text{fmt}} + \lambda_{\text{acc}} = 1$. Both R_{fmt} and R_{acc} are binary, so the reward takes one of four values, $R(y, a^*) \in \{0, \lambda_{\text{fmt}}, \lambda_{\text{acc}}, 1\}$. We set $(\lambda_{\text{fmt}}, \lambda_{\text{acc}}) = (0.3, 0.7)$ for the 3B model and $(0.2, 0.8)$ for the 7B model.

Format Reward. The format reward $R_{\text{fmt}}(y) \in \{0, 1\}$ verifies that the model has produced all four required structural tags in the rollout:

$$R_{\text{fmt}}(y) = \prod_{k=1}^4 \mathbb{1}[\tau_k \in y] \quad (10)$$

where $\{\tau_1, \tau_2, \tau_3, \tau_4\} = \{\text{</extract_entities>, </filtered_groups>, </associative_reasoning>, </answer>}\}$ are the four closing tags corresponding to entity extraction, group filtering, associative reasoning, and the final answer, respectively. The reward is 1 only when all four tags are present, ensuring the model completes the full three-turn pipeline.

Accuracy Reward. The accuracy reward $R_{\text{acc}}(y, a^*) \in \{0, 1\}$ performs an exact match between the model’s predicted answer and the ground truth:

$$R_{\text{acc}}(y, a^*) = \mathbb{1}[\text{Extract}(y) = a^*] \quad (11)$$

where $\text{Extract}(y)$ retrieves the text content between `<answer>` and `</answer>` tags in y , normalized to lowercase with whitespace stripped. If no `<answer>` tag is present, $\text{Extract}(y) = \emptyset$ and the match fails.

Reward Design. The two components are combined additively so that a rollout completing the full three-turn pipeline receives partial credit (λ_{fmt}) even when its final answer is wrong. This keeps a dense and easily satisfied signal toward completing the pipeline throughout training: under an outcome-only reward, a well-completed trajectory that reaches a wrong answer is indistinguishable from one that breaks off midway, and the policy receives no gradient toward following the multi-turn agentic structure. We place more weight on format for the 3B model ($\lambda_{\text{fmt}} = 0.3$ versus 0.2 for 7B), as the smaller base model has weaker instruction-following ability and needs a stronger anchor on structural compliance.

Training Procedure. We train the policy with GRPO (Equation 8) directly from the base instruct model, without any supervised fine-tuning stage. The additive reward removes the need for SFT initialization since the format term makes structural compliance rewarding from the very first rollouts, so that the instruction-tuned base model could acquire the multi-turn output format through RL alone, while the accuracy term drives answer quality in parallel. This purely RL-based training pipeline simplifies the overall training approach and demonstrates that the multi-turn agentic behavior can emerge entirely from reward-driven exploration.

4 Experiments

The experiments in this section are designed to test the following hypotheses: (1) Knowledge extrapolation is a learnable behavior that is robust across model sizes, enabling small

Table 1: QA accuracy (%) across two KGs of vastly different scales. Agentic KG frameworks (GraphRAG, ToG, GIVE) are reported both on GPT-3.5-Turbo, following the settings in which they were originally published, and on the same Qwen2.5 backbones used by SAKE. Best results for each model size are in **bold**. Δ : difference between SAKE and the strongest baseline *within the same model block*.

Model	Method	UMLS (135, 5.9K)				CNet (844K, 2.1M)
		PubMedQA	BioASQ	ProcBank	Wt.Avg	CSQA
GPT-3.5 Turbo	GraphRAG	23.4	10.3	71.3	27.0	—
	ToG	17.6	18.0	66.8	26.6	69.8
	GIVE	53.6	88.2	73.4	70.1	74.7
Qwen2.5 3B-Instruct	I/O Prompt	11.0	28.6	67.5	27.7	64.2
	CoT	12.0	25.0	65.0	26.3	65.0
	RAG	22.0	20.2	52.5	26.8	52.0
	GraphRAG	17.0	9.5	65.0	22.8	63.8
	ToG	11.0	11.9	62.5	20.5	64.6
	GIVE	15.0	8.3	60.0	20.5	9.8
	SFT	33.0	76.2	55.0	53.1	67.5
	GRPO	19.0	69.1	67.5	46.5	66.7
	Search-R1	16.0	39.3	42.5	29.5	66.7
	SAKE	51.0	82.1	70.0	66.1	73.2
	Δ	+18.0	+5.9	+2.5	+13.0	+5.7
Qwen2.5 7B-Instruct	I/O Prompt	23.0	78.5	70.0	52.2	76.4
	CoT	29.0	77.4	75.0	55.4	63.4
	RAG	15.0	46.4	70.0	36.6	77.6
	GraphRAG	11.0	19.0	82.5	26.8	78.9
	ToG	10.0	14.3	82.5	24.6	77.2
	GIVE	14.0	44.0	47.5	31.2	14.6
	SFT	42.0	82.1	75.0	62.9	79.3
	GRPO	42.0	66.7	72.5	56.7	79.7
	Search-R1	24.0	78.6	75.0	53.6	78.5
	SAKE	57.0	95.2	80.0	75.4	81.3
	Δ	+15.0	+13.1	-2.5	+12.5	+1.6

CSQA = CommonsenseQA. CNet = ConceptNet (844K nodes, 2.1M edges).

models to match or surpass large-model agentic frameworks (Section 4.1). (2) The improvement stems from reasoning over structured knowledge rather than from fine-tuning alone (Section 4.1). (3) SAKE generalizes across domains and KG scales, with the largest gains where knowledge is most incomplete (Section 4.1). (4) SAKE achieves competitive accuracy while reducing token usage by over 90% compared to inference-time agentic frameworks (Section 4.2). (5) Each component of the pipeline—key entity extraction, entity group construction, group filtering, agentic retrieval, and knowledge extrapolation—contributes meaningfully to the final performance (Section 4.3).

4.1 Performance on Knowledge-Intensive Benchmarks

To enable direct comparison with prior agentic KG reasoning methods, we evaluate on the same biomedical benchmarks and UMLS KG used in GIVE (He et al., 2024): PubMedQA (Jin et al., 2019), BioASQ (Krithara et al., 2023), and ProcessBank (Berant et al., 2014), with a 135-node UMLS subgraph (Li et al., 2023) containing 5,877 triplets. We additionally evaluate on CommonsenseQA (Talmor et al., 2019) with ConceptNet (Speer et al., 2018) to test generalization beyond the small-KG regime. For all datasets, we randomly select 80% of the questions for training and evaluate on the remaining 20%. Table 1 presents the main results across these knowledge-intensive QA benchmarks using two KGs of vastly different scales. We organize our analysis around three key findings:

Knowledge extrapolation is learnable and robust across model sizes. SAKE achieves the best weighted average on both Qwen2.5-3B and 7B, raising it from 27.7% to 66.1%

(+38.4%) for 3B and from 52.2% to 75.4% (+23.2%) for 7B. This consistency is notable because other fine-tuning baselines exhibit unstable gains across scales: SFT improves PubMedQA from 11.0% to 33.0% on 3B but only from 23.0% to 42.0% on 7B, while GRPO actually *decreases* BioASQ accuracy on 7B (from 78.5% to 66.7%). SAKE, by contrast, improves every benchmark on both model sizes without exception. More remarkably, these small models with SAKE surpass GPT-3.5-Turbo with GIVE, which is a state-of-the-art inference-time agentic framework that relies on multi-step prompting and iterative LLM calls. The 7B model achieves 75.4% weighted average versus GIVE’s 70.1%, and even the 3B model (66.1%) approaches GIVE’s performance despite using a model with far fewer parameters. These results demonstrate that associative reasoning over structured knowledge, with linking entity groups and constructing new triplets by analogy, can be internalized by small models through RL, without complex multi-step instructions at inference time.

The improvement stems from reasoning over structured knowledge, not from fine-tuning alone. To isolate the contribution of structured KG interaction, we compare SAKE against two fine-tuning baselines that use the same training data and optimization but lack access to KG tools. Vanilla GRPO trains with the same RL algorithm and reward signal but without KG retrieval: SAKE outperforms it by +19.6% and +18.7% weighted average accuracy on 3B and 7B, respectively. SFT, which directly fine-tunes on question-answer pairs, also falls short of SAKE by +13.0% (3B) and +12.5% (7B). These large, consistent gaps confirm that SAKE’s gains are not an artifact of additional training compute or exposure to the training data, but arise specifically from learning to retrieve and reason over structured knowledge. This contrast highlights the advantage of structured KG retrieval: compact, relational, and directly amenable to associative reasoning, over unstructured text retrieval in knowledge-intensive scientific tasks.

SAKE generalizes across domains and KG scales, with the largest gains where knowledge graph is most incomplete. To evaluate whether SAKE’s learned extrapolation strategy transfers beyond the setting it was designed for, we apply it to CommonsenseQA using ConceptNet(Speer et al., 2018), a KG that is over $6,000\times$ larger than UMLS and covers a fundamentally different domain (commonsense vs. biomedical). SAKE achieves the best accuracy on both model sizes (73.2% on 3B, 81.3% on 7B), confirming that the agentic pipeline generalizes without architectural or prompt modifications. However, the margins are notably smaller than on UMLS: +5.7% (3B) and +1.6% (7B) on CommonsenseQA versus +13.0% (3B) and +12.5% (7B) weighted average on UMLS. This pattern is consistent with SAKE’s design: its distinctive advantage of extrapolating missing knowledge by analogy is the most impactful when the KG is small and incomplete, leaving less room for improvement when retrieval alone already provides adequate information coverage.

4.2 Efficiency Analysis

A key advantage of SAKE over inference-time agentic frameworks is computational efficiency. Agentic methods such as ToG and GIVE involve multiple sequential LLM calls, where the output of each call is parsed as the input of the next. We measure *total token usage* as the sum of input tokens across all LLM calls required to answer a single question—this captures the cumulative cost of iterative refinement. As shown in Table 2, ToG and GIVE consume 7,000–19,000 tokens per question due to repeated invocations for entity pruning, relation verification, and progressive answer generation. SAKE requires only a single inference pass: the query, tool-returned entity groups, and triplets are concatenated into one context, reducing total token usage by over 90% across all datasets.

Table 2: Avg. token consumption per question. Reduction is vs. ToG / vs. GIVE.

Method	PubMedQA	BioASQ	ProcBank
ToG	12,701	7,010	11,996
GIVE	14,519	7,970	19,461
SAKE	800	639	1,156
Reduct.	93.7/94.5	90.9/92.0	90.4/94.1

4.3 Ablation Studies

We conduct ablation experiments to isolate the contribution of SAKE’s three pipeline components: group filtering, agentic retrieval, and knowledge extrapolation, by comparing the

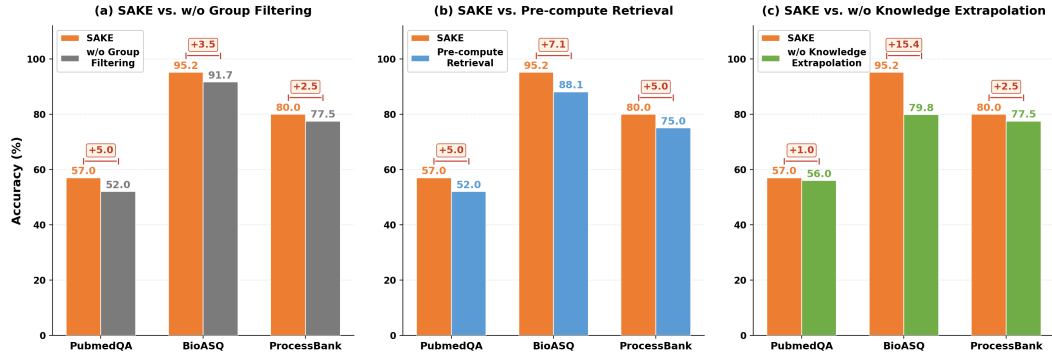


Figure 2: Component ablation on Qwen2.5-7B (Biomedical QA). (a) Full pipeline vs. removing the group filtering turn. Red annotations show the accuracy gain from each component. (b) Agentic tool use vs. pre-computed retrieval. (c) Full pipeline vs. removing the knowledge extrapolation turn.

full pipeline against variants that remove or replace each one. We also study the effect of the number of additional entities per group (p), finding that $p = 3$ yields optimal performance; this hyperparameter analysis is deferred to Appendix C. Unless otherwise stated, all ablation experiments use the three biomedical benchmarks (UMLS KG) with Qwen2.5-7B-Instruct, as the scale of ConceptNet makes repeated ablation runs prohibitively expensive.

4.3.1 Contribution of Pipeline Components

Figure 2 isolates the contribution of three key design choices in SAKE by comparing the full pipeline against variants that remove or replace each component.

Group filtering reduces noise from irrelevant entity groups. In panel (a), we remove the group filtering turn: all entity groups returned by Tool 1 are passed directly to Tool 2 without the model selecting which to retain. This variant decreases accuracy by 5.0% on PubMedQA, 3.5% on BioASQ, and 2.5% on ProcessBank. The filtering step matters because entity extraction may produce concepts whose KG neighborhoods are tangential to the question. Without filtering, Tool 2 retrieves cross-group triplets involving these irrelevant groups, introducing noise into the reasoning context. By learning to discard low-quality groups before triplet retrieval, the model ensures that the knowledge it reasons over is focused and relevant, improving the precision of the subsequent extrapolation step.

Agentic retrieval outperforms pre-computed retrieval. In panel (b), we compare SAKE’s learned agentic retrieval against a variant where entity groups and KG triplets are pre-computed offline and appended to the prompt before training, as in prior work. SAKE outperforms the pre-computed variant on all three benchmarks: +5.0% on PubMedQA, +7.1% on BioASQ, and +5.0% on ProcessBank. The advantage of agentic retrieval is that the model has the freedom to decide *what* to extract and retrieve conditioned on the specific query, rather than receiving a fixed set of knowledge determined by an external pipeline. This allows the model to adapt its retrieval strategy to each question: extracting different entities, retaining different groups, and then tailor its reasoning accordingly. The end-to-end reward signal reinforces this adaptive behavior, producing retrieval decisions that are jointly optimized with the downstream reasoning.

Knowledge extrapolation is essential for bridging incomplete knowledge. In panel (c), we remove the associative reasoning turn by eliminating the `<associative_reasoning>` tag from the prompt: after retrieving KG triplets, the model directly generates an answer without an explicit extrapolation step. This ablation degrades performance across all benchmarks, with the largest drop on BioASQ (−15.4%), followed by ProcessBank (−2.5%) and PubMedQA (−1.0%). The BioASQ result is particularly revealing: this dataset contains many questions whose answers are not directly present in the sparse UMLS KG, making the constructing new triplets by substituting semantically similar concepts critical for bridging the gap. Without it, the model must reason directly from retrieved triplets that may not connect to the query, leading to substantially lower accuracy.

5 Conclusion

We proposed SAKE, an agentic framework that trains LLMs to perform knowledge extrapolation over structured knowledge graphs through tool-augmented reinforcement learning. SAKE defines a three-turn pipeline, including entity extraction, group filtering, and associative reasoning. This rollout generation process is interleaved with two deterministic KG tool calls, all optimized end-to-end with GRPO using a single outcome-based reward. With SAKE, a Qwen2.5-7B model surpasses GPT-3.5-Turbo with state-of-the-art agentic KG traverse frameworks on both biomedical and commonsense benchmarks, while reducing token usage by over 90%. These results demonstrate that associative reasoning over limited external knowledge is a learnable behavior that small, open-weight models can acquire through reinforcement learning, and we hope SAKE encourages further exploration of tool-augmented RL for deploying knowledge-grounded reasoning in specialized domains where comprehensive knowledge bases are impractical to maintain.

References

- Roger E. Beaty and Yoed N. Kenett. Associative thinking at the core of creativity. *Trends in Cognitive Sciences*, 27(7):671–683, 2023. ISSN 1364-6613. doi: <https://doi.org/10.1016/j.tics.2023.04.004>. URL <https://www.sciencedirect.com/science/article/pii/S1364661323000943>.
- Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. Modeling biological processes for reading comprehension. In Alessandro Moschitti, Bo Pang, and Walter Daelemans (eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1499–1510, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1159. URL <https://aclanthology.org/D14-1159>.
- Marco Cascella, Jonathan Montomoli, Valentina Bellini, and Elena Bignami. Evaluating the feasibility of chatgpt in healthcare: an analysis of multiple clinical and research scenarios. *Journal of medical systems*, 47(1):33, 2023.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitanaky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025. URL <https://arxiv.org/abs/2404.16130>.
- Jiashu He, Mingyu Derek Ma, Jinxuan Fan, Dan Roth, Wei Wang, and Alejandro Ribeiro. Give: Structured reasoning with knowledge graph inspired veracity extrapolation. *arXiv preprint arXiv:2410.08475*, 2024.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Serkan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025. URL <https://arxiv.org/abs/2503.09516>.
- Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, 2019.
- Scott Barry Kaufman, Colin G. DeYoung, Jeremy R. Gray, Jamie Brown, and Nicholas Mackintosh. Associative learning predicts intelligence above and beyond working memory and processing speed. *Intelligence*, 37(4):374–382, 2009. ISSN 0160-2896. doi: <https://doi.org/10.1016/j.intell.2009.03.004>. URL <https://www.sciencedirect.com/science/article/pii/S0160289609000300>.
- Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. Bioasq-qa: A manually curated corpus for biomedical question answering. *Scientific Data*, 10:170, 2023. URL <https://doi.org/10.1038/s41597-023-02068-4>.

- Da Li, Boqing Zhu, Sen Yang, Kele Xu, Ming Yi, Yukai He, and Huaimin Wang. Multi-task pre-training language model for semantic network completion. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(11), nov 2023. ISSN 2375-4699. doi: 10.1145/3627704. URL <https://doi.org/10.1145/3627704>.
- 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, 2025. URL <https://arxiv.org/abs/2501.05366>.
- Chenyu Lin, Yilin Wen, Du Su, Hexiang Tan, Fei Sun, Muhan Chen, Chenfu Bao, and Zhonghou Lyu. Resisting contextual interference in rag via parametric-knowledge reinforcement, 2026. URL <https://arxiv.org/abs/2506.05154>.
- Haoran Luo, Haihong E, Guanting Chen, Qika Lin, Yikai Guo, Fangzhi Xu, Zemin Kuang, Meina Song, Xiaobao Wu, Yifan Zhu, and Luu Anh Tuan. Graph-r1: Towards agentic graphrag framework via end-to-end reinforcement learning, 2025. URL <https://arxiv.org/abs/2507.21892>.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys '25*, pp. 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL <http://dx.doi.org/10.1145/3689031.3696075>.
- Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge, 2018. URL <https://arxiv.org/abs/1612.03975>.
- Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph, 2024. URL <https://arxiv.org/abs/2307.07697>.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge, 2019. URL <https://arxiv.org/abs/1811.00937>.
- Shubo Tian, Qiao Jin, Lana Yeganova, Po-Ting Lai, Qingqing Zhu, Xiuying Chen, Yifan Yang, Qingyu Chen, Won Kim, Donald C Comeau, et al. Opportunities and challenges for chatgpt and large language models in biomedicine and health. *Briefings in Bioinformatics*, 25(1):bbad493, 2024.
- Hong Ting Tsang, Jiaxin Bai, Haoyu Huang, Qiao Xiao, Tianshi Zheng, Baixuan Xu, Shujie Liu, and Yangqiu Song. Autograph-r1: End-to-end reinforcement learning for knowledge graph construction, 2025. URL <https://arxiv.org/abs/2510.15339>.
- Siheng Xiong, Ali Payani, and Faramarz Fekri. Enhancing long chain-of-thought reasoning through multi-path plan aggregation. *arXiv preprint arXiv:2510.11620*, 2025a.
- Siheng Xiong, Ali Payani, Yuan Yang, and Faramarz Fekri. Deliberate reasoning in language models as structure-aware planning with an accurate world model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31900–31931, 2025b.

A Dataset Statistics

Table 3: Summary of dataset statistics. $|\mathcal{V}|$ and $|\mathcal{E}|$ are the number of nodes and edges in the associated knowledge graph. #Test is the number of held-out questions on which all methods are evaluated; the weighted average reported in Table 1 pools the three UMLS datasets over their combined 224 questions.

KG	$ \mathcal{V} $	$ \mathcal{E} $	Dataset	QA Type	#Test
UMLS (Li et al., 2023)	135	5,877	PubMedQA (Jin et al., 2019)	Yes–No	100
			BioASQ (Krithara et al., 2023)	Yes–No	84
			ProcessBank (Berant et al., 2014)	Multi-Choice	40
			<i>UMLS total</i>		224
CNet (Speer et al., 2018)	844,158	2,085,099	CSQA (Talmor et al., 2019)	Multi-Choice	246

Table 3 summarizes the statistics of all datasets used in Section 4.1.

B Computing infrastructures

We ran experiments on a workstation with 2 Nvidia A6000 GPUs, and an AMD Ryzen Threadripper 3960X CPU (24C/48T, x86_64, 2.2–3.8 GHz, single NUMA node) with 768 KiB L1d + 768 KiB L1i, 12 MiB L2, and 128 MiB L3 cache; AMD-V enabled.

C Number of Additional Entities per Group

The number of additional entities per group (p in Eq. 6) controls how broadly each entity group extends beyond the original query concept. A small p restricts each group to the most semantically similar KG concepts, yielding precise but narrow coverage. A large p includes more distant concepts, broadening coverage at the risk of introducing noise.

Figure 3 shows the effect of varying p from 1 to 4 on Qwen2.5-7B across the three UMLS benchmarks. Performance improves steadily as p increases from 1 to 3, with all three datasets reaching their peak at $p = 3$: BioASQ achieves 95.2%, ProcessBank 80.0%, and PubMedQA 57.0%. At $p = 4$, performance declines on BioASQ (−3.5) and PubMedQA (−2.0), while ProcessBank plateaus. This pattern reflects a trade-off inherent to knowledge extrapolation: too few additional entities leave the model with insufficient KG coverage to bridge the gap between query concepts and available triplets, while too many introduce semantically distant concepts that dilute the entity groups and lead to retrieval of irrelevant triplets, degrading the quality of associative reasoning.

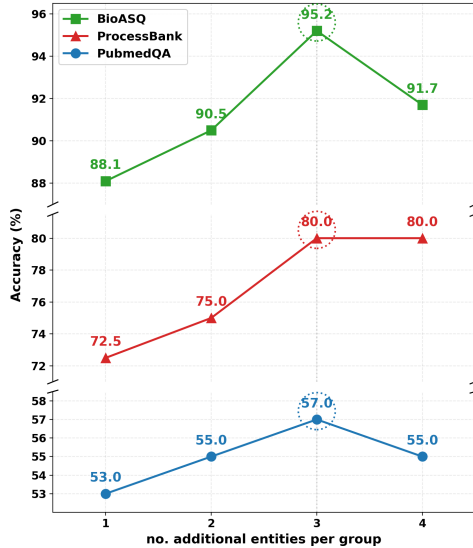


Figure 3: QA accuracy of SAKE (Qwen2.5-7B) with varying numbers of additional entities per group (p in Eq. 6). Dotted circles mark the best setting ($p = 3$).

D Rollout Algorithm

Algorithm 1 Multi-Turn KG-Guided Reasoning Rollout

Require: Input query x , policy model π_θ , KG engine $\mathcal{K}$, similarity function $\text{Sim}(\cdot)$, number of similar concepts p

- 1: Initialize empty rollout sequence y ; token mask $m \leftarrow []$
- ▷ **Turn 1: Entity Extraction**
- 2: **repeat**
- 3: Generate token $y_t \sim \pi_\theta(\cdot \mid x, y)$; $y \leftarrow y + y_t$; $m \leftarrow m + [1]$
- 4: **until** $y_t = \text{</extract_entities>}$
- 5: $E = \{e_1, \dots, e_n\} \leftarrow \text{Parse}(y, \text{<extract_entities>})$
- ▷ **Tool Call 1: Entity Group Construction** (*no gradient*)
- 6: **for** each $e_i \in E$ **do**
- 7: $G_i \leftarrow \{e_i\} \cup \text{Top-}p \text{ Sim}(e_i, \mathcal{K}.\text{concepts})$
- 8: **end for**
- 9: $o_1 \leftarrow \text{<entity_groups> } \{G_1, \dots, G_n\} \text{ </entity_groups>}$
- 10: $y \leftarrow y + o_1$; $m \leftarrow m + [0]^{|o_1|}$ ▷ Mask tool tokens
- ▷ **Turn 2: Group Filtering**
- 11: **repeat**
- 12: Generate token $y_t \sim \pi_\theta(\cdot \mid x, y)$; $y \leftarrow y + y_t$; $m \leftarrow m + [1]$
- 13: **until** $y_t = \text{</filtered_groups>}$
- 14: $S \subseteq \{1, \dots, n\} \leftarrow \text{Parse}(y, \text{<filtered_groups>})$ ▷ Selected group IDs
- ▷ **Tool Call 2: KG Triplet Retrieval** (*no gradient*)
- 15: $T \leftarrow \emptyset$
- 16: **for** each pair $(i, j) \in \binom{S}{2}$ **do**
- 17: $T \leftarrow T \cup \{(h, r, t) \in \mathcal{K}.\text{edges} \mid h \in G_i, t \in G_j\}$ ▷ Forward
- 18: $T \leftarrow T \cup \{(h, r, t) \in \mathcal{K}.\text{edges} \mid h \in G_j, t \in G_i\}$ ▷ Reverse
- 19: **end for**
- 20: $o_2 \leftarrow \text{<kg_triplets> } T \text{ </kg_triplets>}$
- 21: $y \leftarrow y + o_2$; $m \leftarrow m + [0]^{|o_2|}$ ▷ Mask tool tokens
- ▷ **Turn 3: Knowledge Extrapolation & Answer**
- 22: **repeat** ▷ Model generates reasoning + answer until EOS
- 23: Generate token $y_t \sim \pi_\theta(\cdot \mid x, y)$; $y \leftarrow y + y_t$; $m \leftarrow m + [1]$
- 24: **until** $y_t = \text{<eos>}$
- 25: **return** rollout y , mask m

Multi-turn rollout procedure for SAKE. The policy π_θ generates tokens across three turns, interleaved with two deterministic tool calls (highlighted in green). A binary mask m tracks which tokens are model-generated (1) versus tool-injected (0), ensuring that GRPO loss is computed only over the model's own outputs.

E SAKE System Prompt

SAKE System Prompt

You are a biomedical reasoning assistant. To answer questions, you follow a structured retrieval and reasoning process using a knowledge graph. Follow these steps exactly.

— STEP 1: Entity Extraction —

Think about the question. Identify the key substantive biomedical concepts – these are specific biological entities, diseases, molecules, or processes. Do NOT extract relational or intent words (for example, do not extract "treatment", "cause", "effect", "role", "relationship", "association", "mechanism", "involvement", "function").

Format your response as:

<think> [your reasoning about which concepts are substantive] </think>

<extract_entities> concept_1 | concept_2 | ... </extract_entities>

— STEP 2: Group Filtering —

You will receive entity groups: for each concept you extracted, a set of semantically related terms from the knowledge graph.

Review the groups. Keep the groups that are relevant to answering the question. Reference groups by their number.

Format your response as:

<think> [your reasoning about which groups to keep and why] </think>

<filtered_groups> 1 | 2 | ... </filtered_groups>

— STEP 3: Associative Reasoning —

You will receive knowledge graph triplets connecting terms across your selected groups.

Use these triplets to reason associatively:

- Identify which triplet relationships are relevant to the question
- If a direct answer is not present, construct new triplets by substituting semantically similar terms from the entity groups into existing triplet patterns. For example: if (hormone, treats, mental_disorder) is retrieved and melatonin is in the hormone group and insomnia is in the mental_disorder group, construct (melatonin, treats, insomnia) by analogy.
- Build a rationale connecting the retrieved knowledge to the question

Your answer must be a single word: yes, no, maybe, a, or b.

Format your response as:

<associative_reasoning> your reasoning </associative_reasoning>

<answer> your answer </answer>

If no knowledge graph triplets are found, reason using the entity groups alone combined with your general knowledge.

Question: {question}

Table 4: System prompt for SAKE, used identically during training and inference. The prompt defines the three-turn agentic pipeline and the expected output format for each turn, including the special tokens that trigger external tool calls. No task-specific examples or few-shot demonstrations are provided; the model’s reasoning ability emerges entirely from RL training. {question} is replaced with the specific query at runtime.

F Training Analysis

Figure 4 shows the training reward for SAKE on Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct, up to the checkpoint reported for each model. We log only the total reward, which combines the format and accuracy components (Equation 9), so the curves describe overall training progress rather than the behavior of either component individually.

The 3B model improves gradually, rising from approximately 0.40 to a plateau of 0.69–0.75. The 7B model starts higher and improves faster, converging to 0.73–0.79. Both models begin well above their respective λ_{fmt} levels, indicating that the instruction-tuned base models already produce a substantial fraction of well-formed rollouts before any policy updates.

Per-step batch means are noisy for both models, with individual steps ranging from below 0.2 to 1.0. The curves in Figure 4 are therefore smoothed with a 50-step centered moving average.

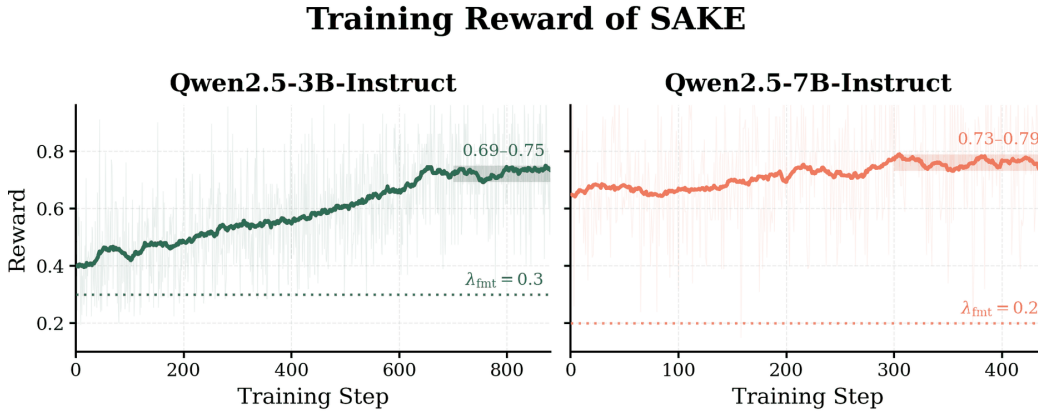


Figure 4: Training reward for SAKE on Qwen2.5-3B-Instruct (left) and Qwen2.5-7B-Instruct (right), shown up to the checkpoint reported for each model. Light lines show per-step batch means; bold lines show a 50-step centered moving average. Shaded bands mark the range over which the smoothed reward fluctuates after convergence.

G Baseline Prompts and examples

In this Section we provide detailed prompts for SAKE and each competing baselines during training and testing.

G.1 I/O Prompt

I/O Prompt User Message

Answer the given question. Provide your answer inside `<answer>` and `</answer>`.
Question: **Q.** Answer this question with yes, no or maybe.

G.2 CoT Prompt

CoT Prompt User Message

Answer the given question. Provide your answer inside `<answer>` and `</answer>`.
Question: **Q.** Answer this question with yes, no or maybe. Let's think step by step.

G.3 RAG Prompt

RAG Prompt User Message

Answer the given question based on the retrieved knowledge triplets (entity, relation, entity) and your own knowledge. Provide your answer inside <answer> and </answer>. Question: Q. Answer this question with yes, no or maybe. Knowledge triplets: **Retrieved knowledge triplets**

G.4 SFT Prompt

SFT Prompt User Message

Answer the given question. Provide your answer inside <answer> and </answer>. Question: Q. Answer this question with yes, no or maybe.

G.5 GRPO Prompt

GRPO Prompt User Message

Answer the given question. You must conduct reasoning first inside <think> and </think>. After reasoning, provide your final answer inside <answer> and </answer>. Question: Q. Answer this question with yes, no or maybe.