

# EvoGraph-R1: Self-Evolving Multimodal Knowledge Hypergraphs for Agentic Retrieval

## Supplementary Material

### Prompt for Textual Entity and High-Order Relation (Hypergraph) Extraction

**Purpose.** Extract structured high-order relational knowledge from raw text documents by identifying semantically coherent segments and their associated entities.

**Task.** Given a paragraph or document and a predefined list of entity types, identify knowledge segments and extract all textual entities associated with each segment.

**Step 1: Extract textual knowledge segments.** Divide the text into the minimum number of self-contained, fact-rich segments. Each segment must be independently interpretable without reading surrounding text.

For each segment output:

(Knowledge\_segment): the self-contained factual text.  
(Completeness\_score): A score from 0 to 10 indicating how complete and self-contained the segment is.

Format: ("hyper-relation"{tuple\_delimiter}  
<knowledge\_segment>{tuple\_delimiter}  
<completeness\_score>)

**Step 2: Extract textual entities.**

For each entity appearing in a segment, extract:

(entity\_name): Use the full explicit name as stated in the text.  
(entity\_type): Assign the correct entity type (e.g., person, organization, product, event, concept).  
(entity\_summary): Provide a context-rich description of the entity's role in that specific segment, including temporal, spatial, numerical, and relationship details.  
(confidence\_score): Assign a score from 0 to 100 reflecting the certainty in the extraction.

Format:

("entity"{tuple\_delimiter}<entity\_name>  
{tuple\_delimiter}<entity\_type>  
{tuple\_delimiter}<entity\_description>  
{tuple\_delimiter}<key\_score>)

**Step 3: Output format.** Return all extracted hyper-relations and entities as a single list.

Use {record\_delimiter} to separate records.

**Step 4: Completion marker.** End with:

{completion\_delimiter}

Figure 1. Prompt template for textual entity and high-order relation extraction.

## A. Prompts Used in EvoGraph-R1

### A.1. Prompts for Multimodal Hypergraph Construction

As shown in Figure. 1 and Figure. 2, we use a structured prompt to support multimodal hypergraph construction, guiding the extraction of high-order relations and associated entities from raw input.

### Prompt for Visual Entity and High-Order Relation Extraction

**Purpose.** Extract structured high-order visual knowledge from raw images by identifying semantically coherent visual subscenes and their associated entities.

**Task.** Given a raw image, its associated texts (caption, OCR, meta-data), and a predefined list of entity types, identify visual knowledge segments and extract all visual entities associated with each segment.

**Step 1: Extract visual knowledge segments.** Divide the image into the minimum number of self-contained, semantically coherent subscenes. Each segment must capture one coherent visual event, interaction, or functional cluster involving one or more entities simultaneously.

For each segment output:

(visual\_segment): a concise description of the subscene grounded in both image and associated text.  
(completeness\_score): A score from 0 to 10 indicating how complete and self-contained the segment is.

Format: ("hyper-relation"{tuple\_delimiter}  
<visual\_segment>{tuple\_delimiter}  
<completeness\_score>)

**Step 2: Extract visual entities.**

For each entity appearing in a segment, extract:

(entity\_name): Use the full explicit name as visible in the image or stated in associated text.  
(entity\_type): Assign the correct entity type from the predefined list.  
(entity\_summary): Provide a context-rich description of the entity's role in that specific segment, including spatial, relational, and visual attribute details.  
(confidence\_score): Assign a score from 0 to 100 reflecting the certainty in the extraction.

Format:

("entity"{tuple\_delimiter}<entity\_name>  
{tuple\_delimiter}<entity\_type>  
{tuple\_delimiter}<entity\_description>  
{tuple\_delimiter}<key\_score>)

**Step 3: Output format.** Return all extracted hyper-relations and entities as a single list.

Use {record\_delimiter} to separate records.

**Step 4: Completion marker.** End with:

{completion\_delimiter}

Figure 2. Prompt template for visual entity and high-order relation extraction.

### A.2. Prompt-Driven Knowledge Evolution

As shown in Figure. 3 and Figure. 4, we design an Agentic Knowledge Evolution Prompt to drive retrieval and editing decisions over the evolving knowledge hypergraph.

## Agentic Knowledge Evolution Prompt

**System.** You are a helpful assistant.  
**Instruction.** Answer the user’s question using the provided knowledge base. You may issue multiple queries if needed.  
**Reasoning Rule.** Always reason first inside: `<think> ... </think>` If additional information is required, call a tool inside: `<tool_call> ... </tool_call>` following the tool specifications in `<tools>...</tools>`. **Never** call a tool without preceding reasoning.  
**Available Tools.** Available functions are provided within `<tools></tools>` as JSON definitions:  
`<tools>`  

```
{ "name": "graph_search", "description":  
"...", ... }  
{ "name": "web_search", "description":  
"...", ... }  
{ "name": "insert", "description": "...",  
... }  
{ "name": "update", "description": "...",  
... }  
{ "name": "delete", "description": "...",  
... }  
</tools>
```

  
**Retrieval Priority.** Use `graph_search` first for any query. Use `web_search` only when `graph_search` is insufficient or irrelevant.  
**Knowledge Maintenance.** After using `web_search`, if new information, conflicts, or corrections are found, immediately maintain the knowledge state via: `insert / update / delete`.  
**Final Answer.** Wrap the final answer inside: `<answer> ... </answer>`

Figure 3. Prompt template for agentic knowledge evolution system.

## B. Datasets

### B.1. E-VQA

Encyclopedic-VQA [3] is a large-scale visual question answering dataset targeting fine-grained entity and attribute recognition. It contains 221k unique question–answer pairs and roughly 1 million associated image samples. Each question is accompanied by structured knowledge extracted from a subset of Wikipedia as supporting evidence, enabling evaluation of a model’s retrieval and answering capabilities with respect to a controlled external knowledge base. The dataset mainly consists of single-hop attribute questions, with a subset of two-hop questions constructed via bridging entities. It does not involve deeper multi-hop reasoning and is commonly used to assess a model’s generalization ability in fine-grained knowledge understanding and external knowledge retrieval settings.

### B.2. InfoSeek

InfoSeek [1] is an open-domain, knowledge-intensive visual question answering dataset. It frames each task as a set of Hierarchically Constrained Sub-Problems (HCSP) and

organizes the reasoning process through a “research tree.” The dataset uses real-time web search and dynamically retrieved Wikipedia evidence, rather than a fixed knowledge base. Most questions require 4–6 reasoning nodes to integrate evidence from multiple sources and produce a single verifiable answer. InfoSeek emphasizes open-domain, multi-hop, and time-sensitive knowledge acquisition, and is widely used to assess a model’s retrieval and reasoning generalization on unseen entities and unseen questions.

### B.3. OK-VQA

OK-VQA [5] is a dataset designed for open-domain, knowledge-based visual question answering. It contains 14,055 open-ended question–answer pairs paired with 14,031 real-world images. Each question is annotated with five independently provided correct answers, enabling evaluation of model robustness under diverse natural language expressions. The dataset spans 10 categories of external knowledge—including commonsense, natural science, culture, geography, materials, and attributes. Its questions go beyond the explicit visual content of an image and require reasoning that combines visual information with external knowledge. As a result, OK-VQA serves as an important benchmark for assessing knowledge-enhanced multimodal models.

### B.4. 2WikiMultiHopQA

2WikiMultiHopQA [2] is a multi-hop question answering dataset in which every question requires cross-document reasoning between two Wikipedia pages. The dataset contains approximately 192k examples, with about 167k for training and around 12.5k each for validation and testing. Each sample provides an explicit evidence chain and a clearly defined reasoning path. These annotations enable evaluation of a model’s ability to integrate information across documents, construct multi-hop reasoning chains, and produce interpretable explanations.

### B.5. HotpotQA

HotpotQA [6] is an English Wikipedia dataset designed for multi-hop reasoning and explainable question answering. It contains roughly 113k human-annotated question–answer pairs. The dataset requires models to retrieve and reason across multiple documents to answer each question. For every sample, HotpotQA provides sentence-level supporting fact annotations, enabling evaluation of whether a model can produce an interpretable reasoning process. It also includes several types of reasoning tasks, such as comparison questions. Under both the distractor setting and the full-wiki retrieval setting, HotpotQA is widely used as a benchmark for multi-hop reading comprehension and retrieval-augmented QA methods.

## Tool Interface Specification

```
[
  {
    "name": "graph_search",
    "description": "Search the internal knowledge base and return matched results. Supports both text and image queries. For image search, use '<img>' as the query.",
    "parameters": {
      "type": "object",
      "properties": {
        "query": {
          "type": "string",
          "description": "The search string. For image-based search, use '<img>' as the query."
        }
      }
    },
    "required": ["query"]
  },
  {
    "name": "web_search",
    "description": "Search the internet for external information.",
    "parameters": {
      "type": "object",
      "properties": {
        "query": {
          "type": "string",
          "description": "The internet search query string."
        }
      }
    },
    "required": ["query"]
  },
  {
    "name": "insert",
    "description": "Insert new factual knowledge that does not currently exist in the KB.",
    "parameters": {
      "type": "object",
      "properties": {
        "content": {
          "type": "string",
          "description": "The full text content to be added as new knowledge."
        }
      }
    },
    "required": ["content"]
  },
  {
    "name": "update",
    "description": "Update an existing knowledge item in the KB.",
    "parameters": {
      "type": "object",
      "properties": {
        "content": {
          "type": "string",
          "description": "Existing knowledge item content to match."
        },
        "new_content": {
          "type": "string",
          "description": "Replacement content for this knowledge item."
        }
      }
    },
    "required": ["content", "new_content"]
  },
  {
    "name": "delete",
    "description": "Soft-delete a knowledge item from the KB.",
    "parameters": {
      "type": "object",
      "properties": {
        "content": {
          "type": "string",
          "description": "The content of the knowledge item to be deleted."
        }
      }
    },
    "required": ["content"]
  }
]
```

Figure 4. Tool interface specification for knowledge base operations.

Table 1. Hyperparameter settings for baselines and our method.

Method	Backbone	Batch Size	Max Length	Top-K	Algo	Epochs
NaiveRAG	Qwen2.5 / GPT-4o-mini	–	$\infty$	N/A	–	–
GraphRAG	GPT-4o-mini	–	$\infty$	60	–	–
LightRAG	GPT-4o-mini	–	$\infty$	60	–	–
HippoRAG2	GPT-4o-mini	–	$\infty$	60	–	–
HyperGraphRAG	GPT-4o-mini	–	$\infty$	60	–	–
Echosight	LLaMA-3-8B	–	4096	20	–	–
MMKB-RAG	Qwen2.5	512	4096	5 / Turn	LoRA	–
Search-R1	Qwen2.5	128	4096	5 Chunks / Turn	GRPO	3
Graph-R1	Qwen2.5/Qwen2.5-VL	128	4096	5 / Turn	GRPO	3
MMSearch-R1	Qwen2.5/Qwen2.5-VL	128	4096	5 / Turn	GRPO	3
EvoGraph-R1	Qwen2.5/Qwen2.5-VL	128	4096	5 / Turn	GRPO	3

## B.6. Natural Questions

Natural Questions (NQ) [4, 6] is a large-scale single-hop QA dataset built from real user queries submitted to Google Search. The public version contains approximately 307,373 training examples, 7,830 validation examples, and 7,842 test examples annotated by five human labelers. The dataset aligns natural user queries with their corresponding Wikipedia pages. It provides human-annotated long answers at the paragraph level and short answers at the entity or span level. These annotations enable evaluation of a model’s real-world performance in open-domain retrieval, complex evidence localization, and end-to-end reading comprehension. NQ places particular emphasis on reflecting authentic search scenarios and supporting diverse answer formats.

To ensure consistency across datasets and maintain manageable training and evaluation workloads, we uniformly sample 5,120 instances per dataset for training and 128 instances for testing.

## C. Implement Details

As summarized in Table 1, we train Qwen2.5 and Qwen2.5-VL policies with GRPO in veRL using a batch size of 512, 4096-token limits, a learning rate of  $5 \times 10^{-7}$ , and 5 rollouts per prompt. A low-variance KL penalty (0.001) is applied to a frozen reference model, and input tokens are masked during training.

Tool usage follows our unified search setup with  $\leq 5$  retrieved chunks per turn and  $\leq 10$  tool calls per rollout. For web search tool, we use the Google Search API for URL discovery and Jina Reader for content extraction, retrieving SERP results and converting HTML into LLM-friendly Markdown. Training is conducted for 3 epochs. The visual entity extraction prompt is used for hypergraph construction. As in prior work, we mask input tokens so that

optimization focuses solely on the generated outputs.

## D. Evaluation Setting and Metric Details

**Evaluation Setting:** We conduct model inference using the veRL framework by setting the `val_only` parameter of the trainer to `True`. The inference engine is based on vLLM, with `top_p` set to 1.0 and temperature set to 0. A single response is generated for each sample in the test set, as described in Section B.

**Evaluation Metrics:** We adopt the following metrics:

**(i) F1 Score.** The F1 score measures the token-level overlap between the predicted answer  $y_i$  and the ground-truth answer  $y_i^*$  using the harmonic mean of precision and recall:

$$F_1 = \frac{1}{N} \sum_{i=1}^N \frac{2 \cdot |\text{tokens}(y_i) \cap \text{tokens}(y_i^*)|}{|\text{tokens}(y_i)| + |\text{tokens}(y_i^*)|} \quad (1)$$

**(ii) Accuracy.** We adopt an LLM-as-Judge approach. Specifically, we use the GPT-4o-20241120 model with `top_p` = 0.1 and temperature = 0, and apply a predefined prompt template (see Figure. 5) to elicit binary judgments ("Yes"/"No") based on image-question-ground truth alignment. Final accuracy is computed from the model’s binary outputs.

**(iii) Generation Evaluation (G-E).** G-E reflects generation quality. Let  $s_{i,d}$  be GPT-4o-mini scores across 7 evaluation criteria, as illustrated in Figure 6 and Figure 7. These scores are obtained by prompting GPT-4o-mini with structured templates and aggregated into overall generation quality metrics:

$$\text{G-E} = \frac{1}{N} \sum_{i=1}^N \left( \frac{1}{7} \sum_{d=1}^7 s_{i,d} \right). \quad (2)$$

## LLM-as-judge Prompt

### Image, Question, and Model Response Evaluation

**Question:** {question}

**Ground Truth Answer:** {ground truth answer}

**Candidate Answers:** {candidate answers}

**Model Response:** {model response}

### Evaluation Instructions.

Evaluate whether the Model Response is correct based on the Image, Question, Ground Truth Answer, and Candidate Answers. Follow the predefined judgment rules and provide a clear **Yes/No** answer along with a justification.

### Output Format.

```
<judge>Yes/No</judge>
<reason>Detailed reasoning following the
evaluation principles.</reason>
```

Figure 5. Prompt template for LLM-as-judge evaluation system.

Table 2. Ablation of foundation model on different sizes.

Backbone	E-VQA	InfoSeek	OK-VQA	Avg.
Qwen2.5-VL-3B	13.6	20.8	27.9	20.8
EvoGraph-R1-3B	<b>35.9</b>	<b>36.7</b>	<b>57.8</b>	<b>41.5</b>
Qwen2.5-VL-7B	18.8	26.4	35.0	26.7
EvoGraph-R1-7B	<b>43.6</b>	<b>42.3</b>	<b>68.6</b>	<b>51.5</b>

## E. Backbone-Agnostic Evaluation

To assess the generality of our framework, we conduct ablations across different foundation model backbones. Table 2 reports the corresponding results. EvoGraph-R1 yields substantial and consistent improvements across all tested backbones, achieving +20.7 Avg. on the 3B model and +24.8 on the 7B model. These backbone-invariant gains demonstrate that the performance improvements primarily arise from the EvoGraph-R1 framework itself rather than from any backbone-specific optimization, underscoring the robustness and architectural generality of our approach.

## F. Case Study

To demonstrate the practical effectiveness of EvoGraph-R1’s self-evolving mechanism, we present three representative cases that illustrate how the agent dynamically refines the knowledge hypergraph through iterative retrieval, web search, and graph editing operations.

Case 1 (Figure. 8) demonstrates how web search enables precise knowledge enrichment. The query concerns which dense plants goldfish use for egg attachment. Ini-

tial retrieval returns a vague statement: “eggs attach to dense aquatic vegetation.” Recognizing this lacks specificity, the agent invokes `WEBSEARCH`, which identifies concrete species: Cabomba, Elodea, and spawning mops. The agent then executes `UPDATE` to replace the original vague statement with specific plant names. This illustrates how the framework bridges the gap between incomplete internal knowledge and precise external evidence, transforming generic statements into specific facts through dynamic refinement.

Case 2 (Figure. 9) demonstrates cross-modal multi-hop reasoning. The query requires identifying a pigment type starting from an image. The agent first visually identifies *Honckenya peploides*, then retrieves its taxonomic classification (Caryophyllaceae), and finally searches for pigment information. After successfully completing the first two hops within the hypergraph, the agent finds only generic family descriptions when searching for pigment properties. The agent explicitly notes: “The search results from the knowledge base do not directly contain pigment-specific information.” Recognizing this gap, the agent conducts web search, which returns that Caryophyllaceae contains anthocyanins rather than betalains. An `INSERT` operation permanently adds this missing biochemical knowledge, completing the reasoning chain from visual appearance to chemical properties.

Case 3 (Figure. 10) illustrates persistent evolution. The query asks for the death location of Henry of Berg’s father. Initial graph retrieval identifies Adolf VII of Berg as the father, but returns no death location. Recognizing this gap, the agent conducts `WEBSEARCH` and discovers Adolf VII died in Neuss during a 1259 tournament. Rather than using this information once and discarding it, the agent executes an `INSERT` operation to permanently add this fact to the hypergraph. This enables reuse in subsequent queries about Adolf VII, Neuss, or tournament deaths without redundant searches. The ablation study in Table 3 of the main paper shows that removing `INSERT` increases retrieval rounds from 2.57 to 3.48 on 2Wiki, confirming the efficiency gains from persistent evolution.

These three cases follow a consistent pattern: the agent attempts internal graph retrieval, identifies specific knowledge gaps, retrieves targeted external evidence through web search, and permanently integrates findings through graph modifications. This closed-loop process directly addresses the frozen-structure and rigid-retrieval limitations of static GraphRAG identified in Section 1 of the main paper. The combination of gap identification and web search prevents both knowledge staleness and unnecessary queries, explaining the substantial improvements over static graph methods and transient web-search systems.

```

{
  (
    "comprehensiveness": (
      "comprehensiveness",
      "whether the thinking considers all important aspects and is thorough",
      "Scoring Guide (0--10):"
      -- 10: Extremely thorough, covering all relevant angles and considerations with depth.
      -- 8--9: Covers most key aspects clearly and thoughtfully; only minor omissions.
      -- 6--7: Covers some important aspects, but lacks depth or overlooks notable areas.
      -- 4--5: Touches on a few relevant points, but overall lacks substance or completeness.
      -- 1--3: Sparse or shallow treatment of the topic; misses most key aspects.
      -- 0: No comprehensiveness at all; completely superficial or irrelevant."
    ),

    "knowledgeability": (
      "knowledgeability",
      "whether the thinking is rich in insightful, domain-relevant knowledge",
      "Scoring Guide (0--10):"
      -- 10: Demonstrates exceptional depth and insight with strong domain-specific knowledge.
      -- 8--9: Shows clear domain knowledge with good insight; mostly accurate and relevant.
      -- 6--7: Displays some understanding, but lacks depth or notable gaps.
      -- 4--5: Limited knowledge shown; understanding is basic or somewhat flawed.
      -- 1--3: Poor grasp of relevant knowledge; superficial or mostly incorrect.
      -- 0: No evidence of meaningful knowledge."
    ),

    "correctness": (
      "correctness",
      "whether the reasoning and answer are logically and factually correct",
      "Scoring Guide (0--10):"
      -- 10: Fully accurate and logically sound; no flaws in reasoning or facts.
      -- 8--9: Mostly correct with minor inaccuracies or small logical gaps.
      -- 6--7: Partially correct; some key flaws or inconsistencies present.
      -- 4--5: Noticeable incorrect reasoning or factual errors throughout.
      -- 1--3: Largely incorrect, misleading, or illogical.
      -- 0: Entirely wrong or nonsensical."
    ),

    "relevance": (
      "relevance",
      "whether the reasoning and answer are highly relevant and helpful to the question",
      "Scoring Guide (0--10):"
      -- 10: Fully focused on the question; highly relevant and helpful.
      -- 8--9: Mostly on point; minor digressions but overall useful.
      -- 6--7: Generally relevant, but includes distractions or less helpful parts.
      -- 4--5: Limited relevance; much of the response is off-topic or unhelpful.
      -- 1--3: Barely related to the question or largely unhelpful.
      -- 0: Entirely irrelevant."
    )
  )
}

```

Figure 6. Seven evaluation dimensions for generation quality (part a).

```

{
  (
    "diversity": (
      "diversity",
      "whether the reasoning is thought-provoking, offering varied or novel perspectives",
      "Scoring Guide (0--10):"
      -- 10: Exceptionally rich and original; demonstrates multiple fresh and thought-provoking
      ideas.
      -- 8--9: Contains a few novel angles or interesting perspectives.
      -- 6--7: Some variety, but generally safe or conventional.
      -- 4--5: Mostly standard thinking; minimal diversity.
      -- 1--3: Very predictable or monotonous.
      -- 0: No diversity or originality at all."
    ),

    "logical_coherence": (
      "logical coherence",
      "whether the reasoning is internally consistent, clear, and well-structured",
      "Scoring Guide (0--10):"
      -- 10: Highly logical, clear, and easy to follow throughout.
      -- 8--9: Well-structured with minor lapses in flow or clarity.
      -- 6--7: Some structure and logic, but a few confusing or weakly connected parts.
      -- 4--5: Often disorganized or unclear; logic is hard to follow.
      -- 1--3: Poorly structured and incoherent.
      -- 0: Entirely illogical or unreadable."
    ),

    "factuality": (
      "factuality",
      "whether the reasoning and answer are based on accurate and verifiable facts",
      "Scoring Guide (0--10):"
      -- 10: All facts are accurate and verifiable.
      -- 8--9: Mostly accurate; only minor factual issues.
      -- 6--7: Contains some factual inaccuracies or unverified claims.
      -- 4--5: Several significant factual errors.
      -- 1--3: Mostly false or misleading.
      -- 0: Completely fabricated or factually wrong throughout."
    )
  )
}

```

Figure 7. Seven evaluation dimensions for generation quality (part b).



What is a dense plant to which this fish attach their eggs?



**<think>** The fish in the image appears to be a goldfish. To answer which dense plant goldfish attach their eggs to, I first query the internal knowledge base about goldfish reproduction and egg-laying. **</think>**



**<graph\_search>** Goldfish reproduction **</graph\_search>**  
**<information>** (1) "Goldfish, like all cyprinids, are egg-layers." coherence="1.833"  
(2) "Their eggs are adhesive and attach to aquatic vegetation, typically dense plants." coherence="1.0"  
(3) "Goldfish eggs hatch within 48 to 72 hours after being laid." coherence="0.88"  
**</information>**



**<think>** The current knowledge base only states that goldfish eggs attach to "dense aquatic vegetation," without specifying which plants these are. To obtain more precise information, I now conduct a broader web-style search focused on identifying the specific dense plants commonly used by goldfish for egg attachment. **</think>**



**<graph\_search>** Which dense plants do goldfish typically attach their eggs to?  
**</graph\_search>**  
**<information>** (1) "Goldfish eggs are adhesive and attach to aquatic vegetation, typically dense plants such as Cabomba or Elodea, or roots."  
(2) "Artificial spawning mops are widely used in aquaria as dense substrates mimicking vegetation for goldfish to attach their eggs." **</information>**



**<update>** old\_knowledge="Their eggs are adhesive and attach to aquatic vegetation, typically dense plants." new="Their eggs are adhesive and attach to dense aquatic substrates such as Cabomba, Elodea, or an artificial spawning mop." **</update>**



**<think>** Goldfish lay adhesive eggs that must attach to dense submerged structures. The complete knowledge now identifies Cabomba, Elodea, and spawning mops as the typical dense substrates used during egg-laying. **</think>**  
**<answer>** Cabomba; Elodea; also spawning mops as dense artificial substrates  
**</answer>**

Figure 8. Case study 1: Graph update through web evidence integration.



What pigment do plants in the family that this plant belongs to have that distinguishes them from Caryophyllineae?



**<think>** To answer the pigment question, I must first identify which plant appears in the image. I will perform multimodal kb\_search to retrieve candidate species linked to this visual appearance. **</think>**



**<graph\_search>** **<img>**.png**</img>** **</graph\_search>**  
**<information>** (1) entity\_id=... entity\_label="Honckenya peploides"  
entity\_type="Species" source="visual\_retrieval" coherence="0.92" ..... **</information>**



**<think>** The strongest match is Honckenya peploides. Next, I query the knowledge base for the taxonomic family to which this species belongs. **</think>**



**<graph\_search>** Honckenya peploides family classification **</graph\_search>**  
**<information>** (1) "Honckenya peploides is the only species in the genus Honckenya of the plant family Caryophyllaceae." **<coherence= 1.5>** ..... **</information>**



**<think>** To answer the pigment question, I must determine which pigment characterizes Caryophyllaceae relative to Caryophyllineae. I query the knowledge base for pigment-related information about Caryophyllaceae. **</think>**



**<graph\_search>** Caryophyllaceae pigment anthocyanin betalain **</graph\_search>**  
**<information>** (1) "Caryophyllaceae is described as a cosmopolitan family of mostly herbaceous plants, particularly diverse in temperate regions." **<coherence= 1.0>** ..... **</information>**



**<think>** The search results from the knowledge base do not directly contain pigment-specific information for Caryophyllaceae. I will perform a web search to find this information. **</think>**



**<web\_search>** Caryophyllaceae anthocyanins betalains pigment difference Caryophyllineae **</web\_search>**  
**<information>** (1) "Caryophyllaceae were historically distinguished from other Caryophyllineae because they contain anthocyanins rather than betalain pigments." ..... **</information>**



**<think>** The websearch provided several sources indicating that Caryophyllaceae contain anthocyanins rather than betalain pigments, which distinguishes them from other Caryophyllineae. The knowledge base does not contain this pigment information, so we can maintain it using the insert function. **</think>**



**<insert>** Caryophyllaceae contain anthocyanin pigments rather than betalains, distinguishing them from most Caryophyllineae **</insert>**



**<think>** The plant in the image is Honckenya peploides, which belongs to Caryophyllaceae. External evidence shows this family produces anthocyanins rather than betalains. Thus, anthocyanins are the pigment distinguishing Caryophyllaceae from Caryophyllineae. **</think>**  
**<answer>** Anthocyanins **</answer>**

Figure 9. Case study 2: Multi-hop reasoning with knowledge base insert.

Where was the place of death of Henry Of Berg, Lord Of Windeck's father?



### S1:Graph Retrieval

**<think>** Let's start by searching the knowledge base to find information about Henry of Berg, Lord of Windeck and his father's place of death. **</think>**  
**<graph\_search>** Henry of Berg, Lord of Windeck father place of death **</graph\_search>**  
**<information>** (1) "Adolf VII of Berg (1220–1259) was the father of Henry of Berg, Lord of Windeck." coherence="1.33"  
(2) "Henry of Berg, Lord of Windeck, was the son of Adolf VII of Berg and Margaret of Hochstaden." coherence="1.25" **</information>**



### S2:WebSearch

**<think>** The retrieved sentences confirm that Adolf VII of Berg was the father. The knowledge base does not yet contain his place of death, so I will retrieve it using external web-style search. **</think>**  
**<web\_search>** Where did Adolf VII of Berg die? **</web\_search>**  
**<information>** (1) "He died on 22 April 1259 as a result of wounds received during a tournament at Neuss." coherence="0.98"  
(2) "Adolf VII of Berg died in Neuss, Holy Roman Empire, at the age of 39." coherence="0.96"  
(3) "Wikidata lists the place of death of Adolf VII of Berg as Neuss." coherence="0.90"  
**</information>**



### S3:GraphEdit

**<think>** The internal knowledge previously lacked Adolf VII's place of death. I now insert the most complete version of the sentence into the knowledge base for future reasoning. **</think>**  
**<insert>** "Adolf VII died on 22 April 1259 in Neuss, Holy Roman Empire, after injuries received during a tournament." **</insert>**



### Final Reasoning & Answer

**<think>** Henry of Berg, Lord of Windeck's father was Adolf VII of Berg. Web search results consistently show that Adolf VII of Berg died in Neuss. Therefore, the place of death is Neuss. **</think>**  
**<answer>** Neuss **</answer>**

Figure 10. Case study 3: Sequential graph retrieval and editing.

## References

- [1] Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14948–14968, 2023. [2](#)
- [2] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, 2020. [2](#)
- [3] Chengyue Huang, Brisa Maneechotesuwan, Shivang Chopra, and Zsolt Kira. Frames-vqa: Benchmarking fine-tuning robustness across multi-modal shifts in visual question answering. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3909–3918, 2025. [2](#)
- [4] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020. [4](#)
- [5] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3190–3199. IEEE, 2019. [2](#)
- [6] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380, 2018. [2](#), [4](#)