

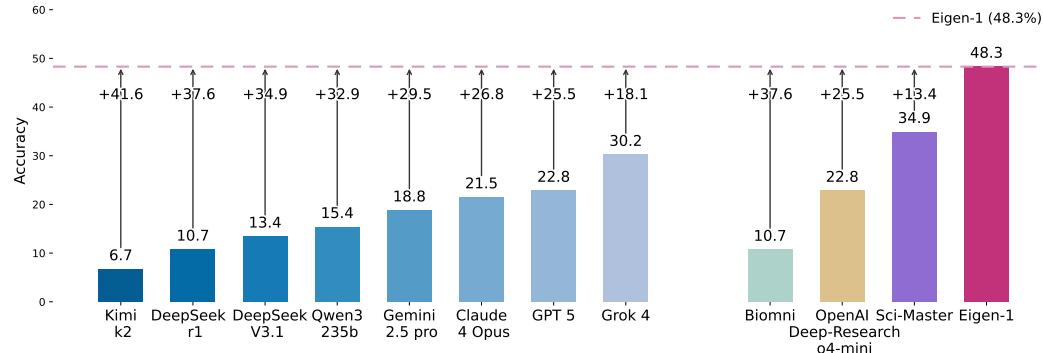
# EIGEN-1: ADAPTIVE MULTI-AGENT SCIENTIFIC REASONING WITH MONITOR-BASED RAG

000  
001  
002  
003  
004  
005 **Anonymous authors**  
006 Paper under double-blind review  
007  
008  
009  
010

## ABSTRACT

011 Large language models (LLMs) have recently shown strong progress on scientific  
012 reasoning, yet two major bottlenecks remain. First, explicit retrieval fragments  
013 reasoning, imposing a hidden “tool tax” of extra tokens and steps. Second, multi-  
014 agent pipelines often dilute strong solutions by averaging across all candidates. We  
015 address these challenges with a unified framework that combines implicit retrieval  
016 and structured collaboration. At its foundation, a *Monitor-based retrieval module*  
017 operates at the token level, integrating external knowledge with minimal disruption  
018 to reasoning. On top of this substrate, *Hierarchical Solution Refinement (HSR)*  
019 iteratively designates each candidate as an anchor to be repaired by its peers, while  
020 *Quality-Aware Iterative Reasoning (QAIR)* adapts refinement to solution quality.  
021 On Humanitys Last Exam (HLE) Bio/Chem Gold, our framework achieves 48.3%  
022 accuracy—the highest reported to date, surpassing the strongest agent baseline by  
023 13.4 points and leading frontier LLMs by up to 18.1 points, while simultaneously  
024 reducing token usage by 53.5% and agent steps by 43.7%. Results on SuperGPQA  
025 and TRQA confirm robustness across domains. Error analysis shows that reasoning  
026 failures and knowledge gaps co-occur in over 85% of cases, while diversity analysis  
027 reveals a clear dichotomy: retrieval tasks benefit from solution variety, whereas  
028 reasoning tasks favor consensus. Together, these findings demonstrate how implicit  
029 augmentation and structured refinement overcome the inefficiencies of explicit tool  
030 use and uniform aggregation.

## 1 INTRODUCTION



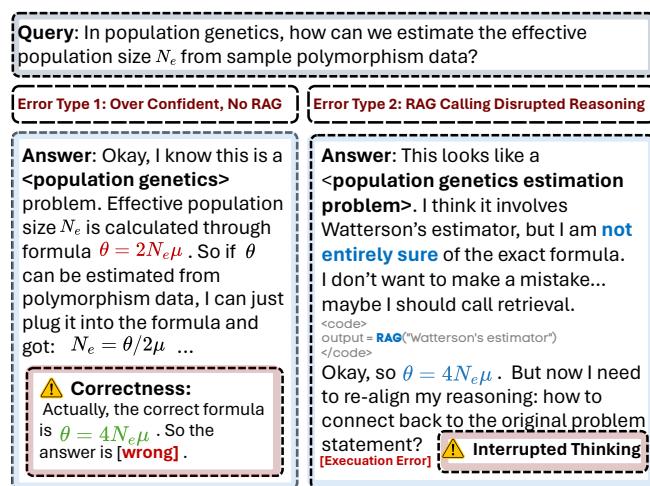
044 Figure 1: **HLE Bio/Chem Gold overall accuracy.** On the 149-problem HLE Bio/Chem Gold split  
045 (Pass@1, auto-judged by o3-mini), our system attains **48.3%** accuracy, exceeding the strongest  
046 agent baseline (SciMaster) by **+13.4** points and leading frontier LLMs by up to **+18.1** points.  
047

048 Recent advances in large language models have enabled impressive performance on a spectrum of  
049 reasoning benchmarks, from general-purpose evaluations such as MMLU [21] and mathematical prob-  
050 lem solving [9; 43] to domain-specific tasks including ScienceQA [38], MedQA [29], and GPQA [42].  
051 These results indicate that LLMs can already handle factual recall and mid-level reasoning across  
052 diverse domains. However, when moving to more demanding benchmarks such as Humanity’s Last  
053 Exam (HLE) [39; 47], which targets expert-level biology and chemistry problems, performance de-  
grades substantially, and systematic failures persist when problems require deep domain knowledge  
and complex multistep reasoning [7]. Through comprehensive analysis of error patterns across 149

054 HLE Bio/Chem problems, we identify two fundamental architectural limitations: (1) *the fragmentation*  
 055 *of logical flow through explicit tool invocation*, and (2) *the inefficiency of democratic multi-agent*  
 056 *collaboration*.

057 Current retrieval-augmented generation systems [3; 19; 33] require explicit interruption to access  
 058 external knowledge. Each retrieval breaks the reasoning flow: suspending the logical state, formulating  
 059 queries, processing results, and reconstructing the context. This *tool tax* compounds quickly: solving  
 060 population genetics problems requires Watterson estimators requires 8-10 such interruptions, doubling  
 061 the number of agent steps compared to a baseline without information retrieval (see Table 3) while  
 062 reducing coherence. The problem persists in all RAG paradigms: single-round approaches [1; 45]  
 063 cannot adapt to emerging needs, iterative systems [27; 44] compound interruption costs, and reasoning-  
 064 aware methods [51; 55] remain bound by explicit invocation, as shown in Figure 2.

065 Simultaneously, most current multi-agent systems [4; 52] employ rigid democratic workflows: generation,  
 066 criticism, synthesis, selection, treating all solutions equally regardless of quality. This contradicts  
 067 both cognitive science research on hierarchical expert reasoning [8; 32] and observations of scientific  
 068 collaboration where ideas naturally organize into anchors and support [14]. Our analysis reveals that  
 069 92.8% of the failures involve reasoning errors, while 88.7% involve knowledge gaps, with substantial  
 070 overlap, indicating that these challenges are fundamentally intertwined, as shown in Figure 7.



071  
 072  
 073  
 074  
 075  
 076  
 077  
 078  
 079  
 080  
 081  
 082  
 083  
 084  
 085  
 086  
 087  
 088  
 089  
 090  
 091  
 092  
 093  
 094  
 095  
 096  
 097  
 098  
 099  
 100  
 101  
 102  
 103  
 104  
 105  
 106  
 107  
 Figure 2: **Population genetics case with two failure modes.** Left (Error Type 1): the model confidently recalls an incorrect formula ( $\theta = 2N_e\mu$ ) and derives  $N_e = \theta/2\mu$ , yielding the wrong answer. Right (Error Type 2): the model retrieves the correct relation ( $\theta = 4N_e\mu$ ) via explicit RAG, but the reasoning flow is disrupted and the result is not reintegrated into the original problem, illustrating the *tool tax*. Our Monitor-based RAG avoids this context suspension by injecting the correct formula directly into the reasoning stream.

We present EIGEN-1, an efficient agent framework that unifies **Monitor-based RAG** eliminates *tool tax* through implicit augmentation, operating continuously at the token level to detect knowledge gaps via semantic uncertainty, generate contextual queries, and inject information seamlessly. **Hierarchical Solution Refinement (HSR)** rotates each candidate solution as an anchor and applies peer-informed repair from the remaining candidates, allowing structured cross-solution refinement rather than uniform averaging. **Quality-Aware Iterative Reasoning (QAIR)** replaces fixed workflows with adaptive cycles that respond dynamically to quality trajectories and problem characteristics. While our experiments focus on integration within a multi-agent reasoning framework, the design of Monitor-based RAG is model-agnostic and can in principle be incorporated into other reasoning systems without architectural modification.

Our system achieves 48.3% accuracy in Humanity’s Last Exam Bio/Chem Gold, surpassing SciMaster [4] (34.9%) by 13.4 percentage points while reducing token consumption by 53.5% and agent steps by 43.7%. Solution pattern analysis further validates our framework: retrieval tasks benefit from diversity, whereas reasoning tasks favor consensus. These results demonstrate that eliminating the tool tax and embracing hierarchical collaboration enables both superior performance and computational efficiency, with potential implications that might extend beyond scientific reasoning to any domain requiring complex knowledge integration with logical inference.

## 2 RELATED WORK

### 2.1 EVOLUTION OF RETRIEVAL-AUGMENTED GENERATION

The integration of external knowledge into language model reasoning has evolved through three main paradigms. **Single-round RAG** systems [19; 25; 33] employ linear pipelines (rewrite→retrieve→generate) and are effective for factual queries. REALM [19] enabled end-to-

end retrieval training, while RAG [33] extended this to knowledge-intensive tasks. More recent variants such as REPLUG [45] and In-Context RALM [41] improve robustness via black-box integration, but they lack adaptivity when knowledge needs emerge mid-reasoning. **Iterative RAG** introduces retrieval-generation loops for dynamic knowledge acquisition. ITER-RETEGEN [44] alternates retrieval and generation, Self-RAG [1] uses self-reflection to decide retrieval, FLARE [27] predicts future content, and DRAGIN [48] updates datastores in real time. These improve grounding but typically incur  $35 \times$  more API calls. **Reasoning-aware RAG** embeds retrieval into reasoning itself. Chain-of-Note [57] produces reading notes, RAT [51] couples retrieval with thought generation, IRCoT [49] interleaves retrieval with chain-of-thought, and ReAct [54] unifies reasoning with action. While more integrated, they still depend on explicit tool calls, fragmenting reasoning and increasing latency.

Table 1 summarizes these paradigms against our Monitor-based approach. Unlike step-level methods that pause to query, Monitor-based RAG operates globally at the token level: it monitors uncertainty signals and implicitly injects evidence into context, reducing retrieval overhead while preserving reasoning continuity. Moreover, its retrieval granularity is finer, enabling more precise and frequent evidence integration without overwhelming the reasoning process.

Table 1: **RAG paradigms vs. key capabilities.** Single-round RAG is efficient but inadaptable; iterative RAG improves grounding but increases latency; reasoning-aware RAG offers tighter coupling yet still relies on explicit calls. **Monitor-based RAG** integrates evidence implicitly at the token level, improving continuity and efficiency.

System	Triggering	Fine-grained	Continuity	Efficiency	Adaptivity
Single-round RAG	✗	✗	✗	✓	✗
Iterative RAG	✓	✓	✗	✗	✓
Reasoning RAG	✓	✓	✓	✗	✓
<b>Monitor-based RAG (Ours)</b>	✓	✓	✓	✓	✓

## 2.2 MULTI-AGENT REASONING SYSTEMS

Multi-agent frameworks have shown promise through collaborative problem solving, yet many rely on rigid orchestration assumptions.

**Democratic collaboration systems** treat all agents equally. SciMasters [4] employs solvercriticrewriter pipelines with a selector over candidate solutions, while LLM-Debate [13], Debate-Only-When-Necessary [15], and Multi-Agent Debate [35] use argumentative dialogue at different scales. MetaGPT [22] assigns role-based responsibilities, and CAMEL [34] explores autonomous cooperation. Table-Critic [56] extends these ideas to structured domains such as tabular reasoning. Such approaches, however, may devote substantial computation to low-quality candidates and do not explicitly capture hierarchical relationships among solutions.

**Structured reasoning systems** explore non-linear organizations. Tree-of-Thoughts [53] enables branching exploration with backtracking, Graph-of-Thoughts [2] allows arbitrary reasoning topologies, and Everything-of-Thoughts [11] combines multiple reasoning patterns. CoMM [5] introduces multi-path prompting, while HM-RAG [37] couples hierarchical agents with multimodal retrieval. Although these methods capture richer reasoning structures, they lack quality-aware adaptation and can rapidly expand search spaces.

**Recent advances** attempt more flexible or adaptive coordination. AgentVerse [6] supports dynamic team assembly, AutoGen [52] enables configurable conversation patterns, and Reflexion [46] incorporates self-improvement signals. Further, evolving orchestration [10], intent-propagation strategies [40], RL-enhanced planning with graph-based policies [26], and collaborative leaderfollower training [16] highlight the need for adaptive depth and role specialization. Hierarchical orchestration frameworks such as AgentOrchestra [58] and HALO [23] exemplify this trend, emphasizing scalable coordination via layered or logic-oriented control.

In contrast, our HSR and QAIR modules introduce hierarchical refinement and quality-driven iteration. Rather than following critic–corrector or debate pipelines [35; 46; 52] that operate under democratic comment–rewrite loops and risk over-investing in weak candidates, HSR organizes solutions into anchor–reference structures for targeted repair, while QAIR applies quality-thresholded, suggestion-guided revisions with early stopping. Crucially, both mechanisms operate *on top of* monitor-based *implicit RAG*, enabling hierarchical, quality-aware convergence without suspending the reasoning process, echoing cognitive science findings on expert problem solving [8].

**Declarative vs. Procedural Frameworks (DSPy vs. Ours)** Declarative frameworks such as DSPy [31] compile tasks into prompt programs and retrieval policies, providing stability but with adaptation largely

162 at the stage level. Our approach is procedural and run-time: a **Monitor**, **Querier**, and **Injector** operate  
 163 during inference to adapt reasoning on the fly. This shift from compile-time templates to run-time  
 164 control enables finer-grained adaptivity and seamless knowledge infusion.

### 166 3 METHOD

167 **Overall workflow.** EIGEN-  
 168 1 integrates global retrieval,  
 169 role-based reasoning, and  
 170 higher-level refinement  
 171 into a unified workflow,  
 172 as shown in Figure 3 and  
 173 Algorithm 1. Monitor-based  
 174 RAG operates globally  
 175 during reasoning: *Monitor*  
 176 detects insufficiency in  
 177 the reasoning stream, the  
 178 *Querier* formulates targeted  
 179 queries, and the *Injector*  
 180 seamlessly integrates  
 181 retrieved evidence back  
 182 into context. Based on  
 183 this substrate, *Proposer*  
 184 generates diverse candidate  
 185 solutions, each of which  
 186 is individually revised by  
 187 *Corrector* through targeted  
 188 local repairs. The refined  
 189 candidates are then passed  
 190 to *Hierarchical Solution  
 191 Refinement (HSR)*, which  
 192 introduces cross-solution  
 193 repair through anchor-  
 194 reference interactions. Next,  
 195 *Quality-Aware Iterative Reasoning (QAIR)*  
 196 evaluates overall quality and may invoke the corrector again  
 197 for additional improvement. Finally, *Ranker* compares candidates and selects the strongest as the final  
 198 answer. All agents can use web search tool (Serp API [30]) by default.

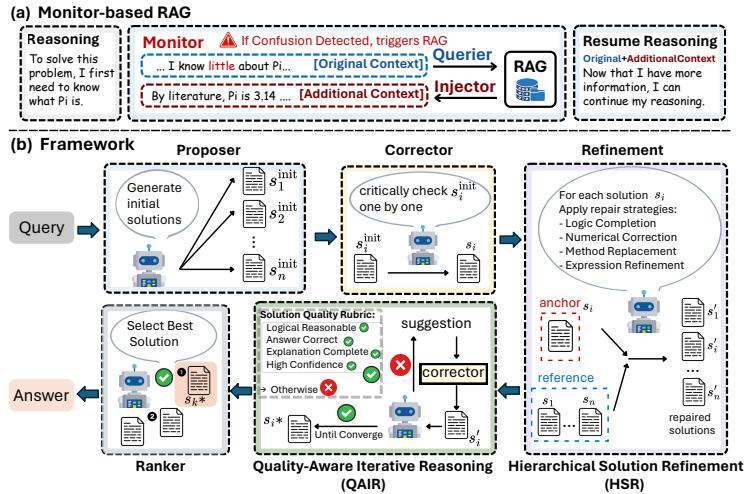


Figure 3: **Framework overview.** (a) *Monitor-based RAG* operates globally during reasoning: the Monitor detects insufficiency in the reasoning stream, the Querier generates targeted queries, and the Injector integrates retrieved evidence into context with minimal disruption. (b) Building on this substrate, the *Proposer* generates initial candidate solutions. Each candidate is revised individually by the *Corrector*, which applies local targeted fixes without access to other solutions. The improved candidates are then passed to *HSR*, which enables cross-solution refinement via anchor–reference relationships. Finally, *QAIR* evaluates overall quality and may invoke the Corrector again, while the *Ranker* selects the strongest solution as the final answer.

#### 199 3.1 MONITOR-BASED RETRIEVAL-AUGMENTED GENERATION

200 Our Monitor-based RAG system augments reasoning implicitly, without fragmenting the workflow  
 201 through explicit tool calls. Instead of forcing the LLM to pause, formulate a query, and inject evidence,  
 202 the Monitor continuously inspects the reasoning trace, identifies potential knowledge insufficiencies,  
 203 and invokes retrieval only when strictly necessary. The construction of the RAG database is shown in  
 204 Appendix A.1.

##### 205 3.1.1 MONITOR: DETECTING UNCERTAINTY AND TRIGGERING RETRIEVAL

206 The Monitor acts as a sentinel that periodically examines the reasoning trace and determines whether  
 207 external knowledge is required:

$$\text{Monitor}(\text{context}) = \begin{cases} 1, & \text{if retrieval is required,} \\ 0, & \text{otherwise.} \end{cases}$$

208 Here, *context* refers to the partial reasoning sequence. Once insufficiency is detected, the retrieval is  
 209 immediately triggered. To balance timeliness and efficiency, the Monitor runs in a streaming setup:  
 210 It checks the reasoning at fixed intervals of 512 characters with an overlap of 128 characters. This  
 211 overlapping design ensures that uncertainty markers that cross boundaries are not missed while keeping  
 212 latency low. Details of RAG Monitor are in Appendix A.5.

##### 213 3.1.2 QUERIER: IDENTIFYING UNCERTAINTY AND GENERATING TARGETED QUERIES

214 Triggered by the Monitor, the Querier converts the uncertain fragment into one or more retrieval queries:  
 215  $[\text{query}_1, \dots, \text{query}_n] = \text{Querier}(\text{context})$ . Here, the Querier maps the reasoning context into one or  
 216 more concise, contextually appropriate queries. A key requirement of the Querier is to precisely extract

216 the minimal set of keywords that capture the essential uncertainty in the reasoning trace. Depending on  
 217 the task, this may result in a single keyword or a small collection of terms, each corresponding to a  
 218 distinct retrieval perspective. The number and specificity of the generated queries directly determine  
 219 the granularity of retrieval, which in turn controls the trade-off between recall and precision in RAG.  
 220 By ensuring that queries are as fine-grained as possible, the Querier avoids unnecessary expansion of  
 221 the search space while maximizing the relevance of retrieved evidence.

### 222 3.1.3 INJECTOR: EVIDENCE COMPRESSION AND CONTEXTUAL INTEGRATION

224 The Injector first filters and  
 225 compresses raw RAG outputs into  
 226 concise, utility-focused snippets to avoid  
 227 redundancy and irrelevant noise.  
 228 Then it rewrites and integrates the  
 229 selected evidence in the Proposer’s rea-  
 230 soning context, ensuring coherence  
 231 and preserving the natural flow of  
 232 the reasoning narrative. This two-  
 233 step design allows the knowledge  
 234 retrieved to improve accuracy with-  
 235 out introducing stylistic or structural  
 236 disruptions: additional context =  
 237 Injector(context, RAG results).

238 Figure 2 shows a population genetic  
 239 problem that requires the Watterson  
 240 estimator. Baseline LLMs exhibit  
 241 two characteristic errors: (1) confi-  
 242 dently recalling the wrong formula  

$$(\theta = 2N_e\mu)$$
 and deriving an incor-  
 243 rect effective population size, or (2)  
 244 retrieving the correct formula ( $\theta =$   

$$4N_e\mu)$$
 via explicit RAG but failing  
 245 to reintegrate it into the original rea-  
 246 soning chain, a classic case of *tool  
 247 tax*. Our Monitor-based RAG re-  
 248 solves both issues: the Monitor de-  
 249 solves both issues: the Monitor de-  
 250 solves both issues: the Monitor de-  
 251 solves both issues: the Monitor de-  
 252 solves both issues: the Monitor de-  
 253 solves both issues: the Monitor de-  
 254 solves both issues: the Monitor de-  
 255 solves both issues: the Monitor de-  
 256 solves both issues: the Monitor de-  
 257 solves both issues: the Monitor de-  
 258 solves both issues: the Monitor de-  
 259 solves both issues: the Monitor de-  
 260 solves both issues: the Monitor de-  
 261 solves both issues: the Monitor de-  
 262 solves both issues: the Monitor de-  
 263 solves both issues: the Monitor de-  
 264 solves both issues: the Monitor de-  
 265 solves both issues: the Monitor de-  
 266 solves both issues: the Monitor de-  
 267 solves both issues: the Monitor de-  
 268 solves both issues: the Monitor de-  
 269 solves both issues: the Monitor de-

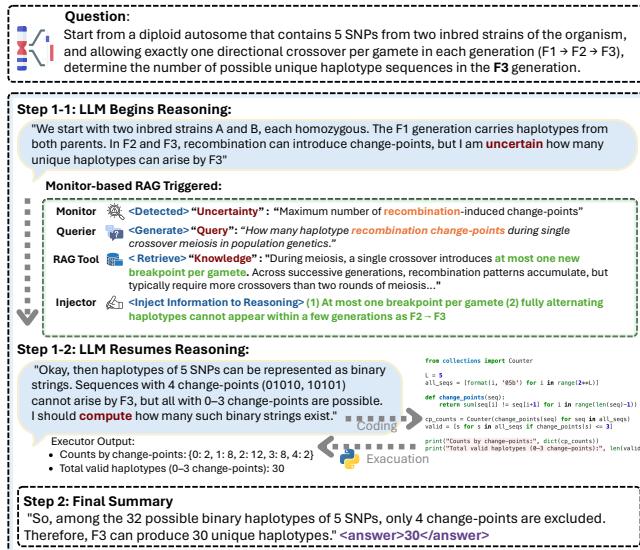


Figure 4: **Haplotype counting with single crossovers (F1 → F3).** The Proposer exhibits uncertainty about recombination constraints; *Monitor* detects insufficiency, *Querier* issues a targeted query, and *Injector* integrates two retrieved facts. This enables the reasoning to exclude invalid cases and converge on correct count of 30 haplotypes.

### 252 3.2 HIERARCHICAL SOLUTION REFINEMENT (HSR)

253 HSR challenges the assumption that all solutions should contribute equally to the final output. Instead  
 254 of democratic averaging, HSR establishes structured relationships among solutions that mirror expert  
 255 collaboration patterns. Let  $\mathcal{S} = \{s_1, \dots, s_n\}$  denote the candidate solutions. Each solution is iteratively  
 256 designated as the anchor  $s_i$ , while the remaining set  $\mathcal{R} = \mathcal{S} \setminus \{s_i\}$  provides references. This rotation  
 257 ensures that every solution benefits from peer-informed repair, preventing premature convergence to a  
 258 single trajectory.

259 Formally, the process can be described as  $s'_i = \text{Refine}(s_i, \mathcal{R})$ , where  $\text{Refine}(\cdot)$  denotes the LLM-driven  
 260 mechanism that applies multidimensional repairs to the anchor. Specifically, logical completion fills  
 261 missing reasoning steps or implicit assumptions, numerical correction resolves arithmetic inaccuracies,  
 262 method replacement substitutes stronger strategies for weaker ones, and expression refinement improves  
 263 clarity without altering substance. These dimensions ensure that the weaknesses of the anchor are  
 264 addressed systematically while preserving its original strengths.

265 Figure 5 shows a pathway reasoning problem where multiple proposers generate partial but inconsistent  
 266 solutions. Baseline multi-agent synthesis averages across candidates, often propagating contradictions  
 267 or omitting critical intermediates. Instead, HSR designates one solution as the anchor and integrates  
 268 targeted corrections from reference solutions (e.g., filling in missing intermediates or fixing reaction  
 269 links). This yields a coherent and biologically valid pathway, demonstrating how HSR consolidates  
 270 fragmented contributions into a unified solution. QAIR then evaluates the refined set and can terminate  
 271 the process once the quality stabilizes, avoiding unnecessary additional cycles.

270  
271

## 3.3 QUALITY-AWARE ITERATIVE REASONING (QAIR)

272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294

QAIR introduces an evaluation-driven control mechanism to refine candidate solutions after the HSR stage. Let  $\mathcal{S}' = \{s'_1, \dots, s'_n\}$  denote the initial set of refined candidate solutions. Each solution  $s' \in \mathcal{S}'$  is evaluated by an LLM-based evaluator on three quality dimensions: logic, answer, and explanation, and a textual suggestion for improvement is generated. Each dimension is scored on a scale from 0 to 5, and the three quality scores are then combined into a composite score:  $q(s') = 0.2 \cdot q_{\text{logic}}(s') + 0.6 \cdot q_{\text{answer}}(s') + 0.2 \cdot q_{\text{explanation}}(s')$ , where the higher weight on the answer dimension emphasizes the correctness of the final answer while still allowing for logical consistency and explanatory clarity. Candidates meeting the threshold  $\tau = 3$  are retained, while those failing are marked non-passing and passed to the corrector for targeted revision:  $\tilde{s} = \text{Corrector}(s', \text{suggestion}(s'))$ .

295  
296  
297  
298  
299  
300

Let  $\mathcal{F}_t$  denote the set of solutions that fail the evaluation in round  $t$ , and  $\mathcal{E}_t$  denote the set of solutions evaluated at round  $t$ . Iterative refinement continues exclusively on the subset of failed solutions, forming the evaluation set for the next round  $\mathcal{E}_{t+1} = \{\tilde{s} \mid s' \in \mathcal{F}_t\}$ , until all solutions pass or maximum rounds  $T_{\max}$  is reached. By coupling structured quality assessment with suggestion-driven repair and avoiding re-evaluation of already validated candidates, QAIR efficiently converges toward a high-quality solution set while maintaining logical consistency, answer correctness, and explanatory clarity.

301

## 4 EXPERIMENTS

302  
303

## 4.1 EXPERIMENTAL SETUP

304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319

We evaluate our approach on Humanity’s Last Exam (HLE) Bio/Chem Gold [47]<sup>1</sup>, comprising 149 graduate-level problems in biology, medicine, and chemistry. HLE Bio/Chem Gold subset was manually curated and corrected by domain experts to ensure label fidelity. Additionally, we test on 92 hard-difficulty problems from SuperGPQA [12] Biology and 172 problems from TRQA Literature [59]. Our framework uses DeepSeek-V3.1 [36] as the base model with temperature 0.5 and 64K token limit. Following HLE protocol, we employ o3-mini for automated evaluation (See Appendix A.3). Beyond accuracy, we log total generated tokens and agent steps in the ablation experiments, as quantitative measures of the tool tax.

320

## 4.2 MAIN RESULTS

321  
322  
323

In the HLE Bio / Chem dataset, our system achieves **48.3%** accuracy (Pass@1), substantially outperforming the strongest baseline Grok-4 (30.2%) by nearly 18 absolute points and more than doubling the performance of general purpose models such as GPT-5 (22.8%) and Claude Opus 4.1 (21.5%). This

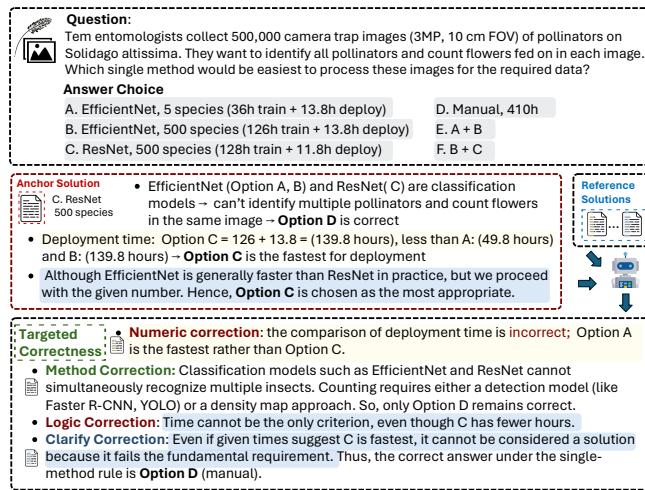


Figure 5: **Illustrative example: HSR.** The system rotates anchors among candidate solutions and integrates targeted corrections from references (e.g., fixing arithmetic mistakes, filling missing steps). Instead of averaging inconsistent candidates, HSR applies targeted improvements to yield a coherent final answer.

Table 2: **Benchmark comparison under matched protocol.** HLE Bio/Chem (149 problems; o3-mini judge), SuperGPQA Biology (hard split), and TRQA Literature (multiple-choice).

Model	HLE Bio/Chem	SuperGPQA Hard	TRQA
<i>Base Models</i>			
Kimi K2	6.71	48.91	38.37
DeepSeek V3.1	13.42	66.30	43.60
Claude Opus 4.1	21.48	63.04	42.44
Gemini 2.5 Pro	18.79	65.22	45.93
GPT-5	22.82	61.96	50.58
Grok-4	30.20	66.30	46.51
<i>Agent Systems</i>			
SciMaster (GPT 4.1) [4]	9.45	19.78	47.67
Autogen (GPT 4.1) [52]	7.38	29.35	51.74
OpenAI Deep Research (o4-mini)	22.82	39.13	-
Biomni (GPT 4.1) [24]	10.74	43.48	41.09
SciMaster (DeepSeek V3.1)	34.92	66.30	51.74
<b>EIGEN-1 (DeepSeek V3.1, Pass@1)</b>	<b>48.30</b>	<b>69.57</b>	<b>54.65</b>
<b>EIGEN-1 (DeepSeek V3.1, Pass@5)</b>	<b>61.74</b>	<b>78.26</b>	<b>79.07</b>

<sup>1</sup><https://huggingface.co/datasets/futurehouse/hle-gold-bio-chem>

margin is particularly notable, given that HLE problems require domain-specific reasoning rather than surface-level recall.

In SuperGPQA hard biology, our method reaches **69.6%**, exceeding all competing large models. The improvement is consistent across question categories, suggesting that our framework not only boosts correctness but also improves robustness on especially challenging scientific queries.

Finally, in TRQA benchmark, which emphasizes Information retrieval, integration and reasoning, our method obtains **54.7%** Pass@1, surpassing both Grok-4 (46.5%) and Gemini 2.5 Pro (45.9%). Furthermore, under the more permissive Pass@5, counting success if any of five attempts is correctaccuracy rises to **79.1%**, indicating robustness under a best-of-N setting.

Together, these results establish the advantage of our design in three heterogeneous benchmarks: biomedical, chemical and medical, demonstrating not only raw accuracy gains, but also improved adaptability across domains (Table 2). For more results of baseline models, please refer to Appendix A.10.

Within the Monitor-based RAG framework, we experimented with four retrieval backends: Vanilla [17], Vanna, HippoRAG [28], and LightRAG [18]. Among these, *HippoRAG* achieved the most consistent gains when coupled with our uncertainty detection. We attribute this to its finer-grained retrieval and graph-structured indexing, which better capture relevant context fragments without overwhelming the reasoning stream. Based on these results, we adopt HippoRAG as the default retrieval backend in our Monitor module (Figure 6).

#### 4.3 COMPONENT ANALYSIS

To understand the contribution of each architectural component, we performed systematic analysis on the full HLE Bio/Chem benchmark (149 problems), considering both incremental build-up and component ablation (Table 3).

The baseline configuration uses five parallel Proposers with access to a generic web search tool but without any paper retrieval (no RAG). The *Explicit RAG* setting adds a scientific paper database queried via embedding-based similarity. Unless otherwise noted, all settings use the same five-Proposer architecture with CriticCorrector refinement and Ranker selection.

The baseline system without external knowledge achieves 25.3% accuracy, underscoring the limitations of parametric knowledge alone for graduate-level science problems. Adding an explicit paper database improves accuracy to 41.4%, but at the cost of a sharp increase in workflow iterations (from 43.4 to 94.8). This reflects the high overhead of explicit retrieval: each tool call suspends reasoning, requires query formulation, and forces reintegration of results, fragmenting what should be a continuous reasoning flow. While the first one or two retrievals may be helpful, repeated interruptions often add little value and compound this “tool tax.”

Our Monitor-based RAG mitigates this overhead through implicit augmentation. By continuously monitoring generation and injecting knowledge only when necessary, it reduces token consumption by more than half (from 470.6K to 218.4K) and cuts workflow iterations nearly in half (from 94.8 to 51.3), while maintaining competitive accuracy (34.5%). Adding the Querier improves query precision, leading to a modest gain to 36.8%. The limited margin compared to Monitor alone suggests that the primary bottleneck lies not in query formation but in evidence integration, which is addressed by the Injector. With the Injector, accuracy rises further to 40.3% with minimal additional overhead.

Hierarchical Solution Refinement (HSR) then contributes complementary gains, raising accuracy to 43.7%. Instead of naive aggregation, HSR leverages anchorreference interactions to apply targeted repairs, focusing revisions where they matter most (e.g., filling missing reasoning

Figure 6: Comparison of retrieval backends within Monitor-based RAG.

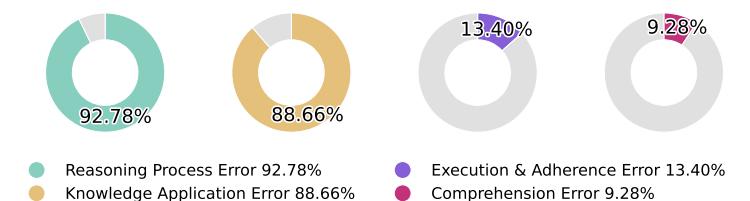
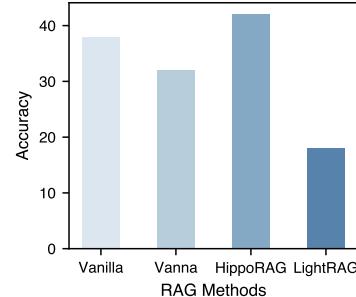


Figure 7: **Error type distribution.** Analysis of incorrect solution logs shows reasoning- and knowledge-related errors as dominant. Note that a single problem may involve multiple error types, so percentages do not sum to 100.

Table 3: **Component analysis from two perspectives on the full HLE Bio/Chem benchmark (149 problems).** (a) Incremental build-up: modules are added one by one. (b) Component ablation: each module is removed from the full system. The baseline configuration uses five parallel Proposers with web search but without external paper retrieval (no RAG). *Steps* = agent-level workflow iterations (not token-level reasoning).

(a) Incremental build-up				(b) Component ablation			
Configuration	Accuracy (%)	Tokens (K)	Steps	Configuration	Accuracy (%)	Tokens (K)	Steps
Baseline (no ext. knowledge & no RAG)	25.3	483.6	43.4	Full system	<b>48.3</b>	<b>218.9</b>	53.4
+ Papers (Explicit RAG)	41.4	470.6	94.8	– (Monitor, Querier, Injector)	48.5	461.3	95.3
+ Monitor only	34.5	218.4	51.3	– Querier	45.9	224.1	53.1
+ Monitor + Querier	36.8	213.0	51.7	– Injector	44.7	202.5	52.1
+ Monitor + Querier + Injector	40.3	229.5	53.1	– HSR	44.8	234.1	53.5
+ Monitor + Querier + Injector + HSR	43.7	214.0	52.9	– QAIR	43.7	214.0	52.9
+ Monitor + Querier + Injector + HSR + QAIR	<b>48.3</b>	<b>218.9</b>	53.4				

steps or correcting arithmetic). This adds some extra reasoning steps but yields proportionally higher accuracy.

Finally, Quality-Aware Iterative Reasoning (QAIR) builds on HSR by selectively invoking the Corrector when evaluation indicates further refinement is necessary. This yields the best overall result in the incremental sequence: 48.3% accuracy with 218.9K tokens and 53.4 iterations. Although QAIR introduces slight additional overhead, it ensures that every revision contributes meaningfully, preventing uncontrolled exploration or redundant cycles.

The ablation analysis further validates these findings: removing the Monitor results in a significant increase in the number of tokens and agent steps; and omitting HSR or QAIR lowers final performance to 44.8% and 43.7%, respectively. Together, these results show that Monitor-based RAG reduces the tool tax, HSR provides structured cross-solution repair, and QAIR ensures convergence through selective correction. Their combination achieves both state-of-the-art accuracy and controlled computation.

## 5 ANALYSIS

### 5.1 ERROR TYPE DISTRIBUTION

Analysis of failed problems reveals two dominant error modes: reasoning process errors (92.78%) and knowledge application errors (88.66%), as shown in Figure 7. These frequently co-occur, suggesting that successful scientific reasoning requires seamless integration of domain knowledge with logical inference. Execution errors (13.40%) and comprehension errors (9.28%) are comparatively rare, indicating that the primary challenge lies not in understanding problems or following instructions, but in maintaining coherent reasoning while accessing relevant knowledge. The strong overlap also suggests interdependence:

missing knowledge often manifests as faulty reasoning steps, and disrupted reasoning in turn prevents effective incorporation of retrieved facts. For more examples of these different errors, see Appendix A.6.

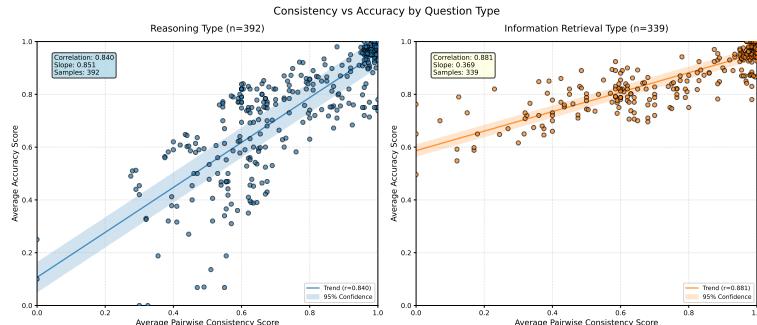


Figure 8: **Diversity vs. consensus.** Task-dependent effect of solution diversity: retrieval tasks benefit from variety, while reasoning tasks benefit from agreement. The horizontal axis reports the *average pairwise consistency score* among Proposers, computed by an LLM-based judge that evaluates semantic overlap between answers on a 0–1 scale. The vertical axis shows the *average accuracy score*, also judged by an LLM, which rates the degree of correctness of each answer relative to ground truth on a continuous 0–1 scale (rather than a binary 0/1). This continuous evaluation enables us to capture fine-grained trends between diversity and correctness across tasks. The fitted trend lines further highlight the contrast: retrieval tasks show a relatively flat slope ( $\approx 0.369$ ), whereas reasoning tasks exhibit a much steeper positive slope ( $\approx 0.851$ ), indicating a stronger dependence on consensus.

### 5.2 DIVERSITY VS. CONSENSUS IN MULTI-AGENT SOLUTIONS

Our framework employs multiple parallel *Proposers* to generate candidate solutions and utilizes a *Ranker* to select the final answer. A natural assumption is that higher agreement among Proposers

432 should correlate with higher accuracy. However, our analysis reveals a more nuanced picture: the  
 433 relationship between solution diversity and correctness depends strongly on problem type.  
 434

435 To investigate this, we divide the benchmark into two categories: *information retrieval* tasks, which  
 436 rely heavily on external knowledge, and *reasoning* tasks, which require longer chains of inference. For  
 437 each problem, we measure the level of agreement among Proposers and evaluate how it correlates with  
 438 accuracy. Both metrics are scored continuously by an LLM judge on a  $[0, 1]$  scale: consistency reflects  
 439 the pairwise agreement among candidate answers, while accuracy measures the graded alignment  
 440 between a candidate and the ground-truth solution (see Appendix A.3). This continuous evaluation  
 441 enables fine-grained correlation analysis beyond binary correctness. As shown in Fig 8, retrieval  
 442 tasks benefit from diversity (low agreement), whereas reasoning tasks benefit from consensus (high  
 443 agreement), with correlation slopes of 0.369 and 0.851, respectively.

444 This dichotomy suggests that different ranking strategies are optimal for different task types. In retrieval-  
 445 heavy settings, the Ranker should preserve diversity and aggregate complementary perspectives,  
 446 whereas in reasoning-heavy tasks, it should prioritize high-consensus answers as indicators of reliability.  
 447 These observations highlight the complementary roles of HSR and QAIR, which operationalize the  
 448 transition from diversity to consensus in a task-adaptive manner.

### 449 5.3 TOOL TAX QUANTIFICATION

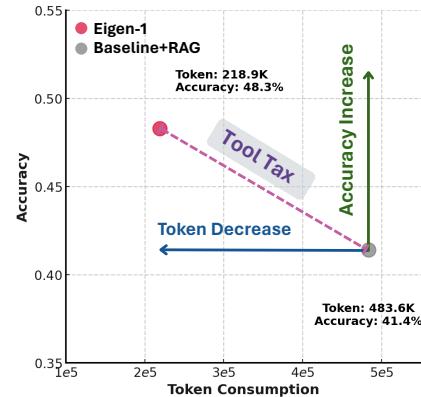
450 The computational burden of explicit tool invocation ex-  
 451 tends beyond simple latency. As shown in Table 3, the  
 452 *explicit RAG baseline* (with Proposers equipped with paper  
 453 retrieval and web search) more than doubles agent-level  
 454 workflow iterations compared to the no-IR setting (43.4  
 455  $\rightarrow$  94.8). This quantifies a hidden *tool tax* from context  
 456 switching between reasoning and retrieval modes: each call  
 457 requires the system to pause the evolving chain of thought,  
 458 formulate a query, process external results, and then recon-  
 459 struct the local context before continuing fragmenting what  
 460 should be a continuous inference process.

461 Fig. 9 visualizes this trade-off. Explicit RAG produces  
 462 substantially longer traces without commensurate gains,  
 463 whereas our *Monitor-based RAG* maintains concise, inter-  
 464 pretative reasoning by injecting only the evidence that is  
 465 needed, precisely when it is needed. Operating implicitly at  
 466 generation time, it delivers comparable knowledge augmen-  
 467 tation with markedly fewer tokens and iterations, avoiding  
 468 repeated context suspensions.

469 More broadly, these findings argue for *implicit augmentation*  
 470 and *adaptive tool policies* in agent design: systems  
 471 should not treat all retrieval calls equally, but modulate re-  
 472 trieval frequency and depth based on emerging uncertainty,  
 473 problem structure, and expected utility preserving continuity  
 474 of reasoning while still accessing external knowledge when  
 475 it truly helps.

## 476 6 CONCLUSION

477 Our experiments validate three key architectural innovations in EIGEN-1. First, implicit knowledge  
 478 augmentation through Monitor-based RAG substantially reduces explicit retrieval overhead while  
 479 preserving reasoning coherence. Second, HSR improves over uniform multi-agent aggregation by  
 480 introducing structured anchor-reference relationships. Third, QAIR adaptively balances exploration and  
 481 early stopping, achieving an effective trade-off between diversity and consensus. On HLE Bio/Chem,  
 482 EIGEN-1 reaches 48.3% accuracy under compute-matched settings, with a 53.5% token reduction,  
 483 showing that targeted architectural design can enhance both effectiveness and efficiency. By integrating  
 484 external knowledge with minimal disruption to reasoning flow, our framework addresses key limitations  
 485 of prior approaches in complex scientific problem solving. Future work will extend these principles to  
 486 additional scientific domains, assess robustness and transferability, and explore integration into broader  
 487 scientific workflows.



488 **Figure 9: Quantifying the tool tax.** Comparison of accuracy and coherence relative  
 489 to compute cost, showing the overhead of explicit retrieval vs. implicit augmentation.  
 490 Note that in this analysis, the baseline refers to the explicit RAG configuration (i.e., Proposers  
 491 equipped with paper retrieval and web search), which represents the standard setup  
 492 in most existing agent systems.

486  
487

## ETHICS STATEMENT

488  
489  
490  
491  
492  
493  
494  
495  
496  
497

Our study aims to improve the scientific reasoning capabilities of large language models (LLMs) by introducing a unified framework that combines implicit retrieval and hierarchical collaboration. All datasets used in this work, including *Humanity’s Last Exam* (*HLE*), *SuperGPQA*, and *TRQA*, are publicly available under open licenses. No private, sensitive, or human-identifiable data are involved. All annotations, where applicable, are derived from public benchmarks or generated using synthetic processes, ensuring that no ethical concerns regarding data privacy or misuse arise. The broader societal impact of this research lies in its potential to enhance scientific reasoning and complex problem-solving abilities in AI systems, which can be applied to fields such as education, scientific discovery, and decision support. Care has been taken to avoid overstating capabilities or drawing misleading conclusions, and we encourage further research to validate our findings across other high-stakes domains.

498  
499

## REPRODUCIBILITY STATEMENT

500  
501  
502  
503  
504

To ensure reproducibility, we provide detailed descriptions of our dataset preprocessing procedures, agent prompting strategies, and iterative refinement workflow in the appendix. These include full pipeline configurations and experimental setups required to reproduce the reported results with high fidelity. The code is available at <https://anonymous.4open.science/r/RAG-Monitor>.

505  
506

## REFERENCES

507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539

- [1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. 2024.
- [2] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianiuzzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 17682–17690, 2024.
- [3] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.
- [4] Jingyi Chai, Shuo Tang, Rui Ye, Yuwen Du, Xinyu Zhu, Mengcheng Zhou, Yanfeng Wang, Yuzhi Zhang, Linfeng Zhang, Siheng Chen, et al. Scimaster: Towards general-purpose scientific ai agents, part i. x-master as foundation: Can we lead on humanity’s last exam? *arXiv preprint arXiv:2507.05241*, 2025.
- [5] Pei Chen, Boran Han, and Shuai Zhang. Comm: Collaborative multi-agent, multi-reasoning-path prompting for complex problem solving. *arXiv preprint arXiv:2404.17729*, 2024.
- [6] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4):6, 2023.
- [7] Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. *arXiv preprint arXiv:2410.05080*, 2024.
- [8] Michelene TH Chi, Paul J Feltovich, and Robert Glaser. Categorization and representation of physics problems by experts and novices. *Cognitive science*, 5(2):121–152, 1981.
- [9] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [10] Yufan Dang et al. Multi-agent collaboration via evolving orchestration. *arXiv preprint arXiv:2505.19591*, 2025.
- [11] Ruomeng Ding, Chaoyun Zhang, Lu Wang, Yong Xu, Minghua Ma, Wei Zhang, Si Qin, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. Everything of thoughts: Defying the law of penrose triangle for thought generation. *arXiv preprint arXiv:2311.04254*, 2023.

540 [12] Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming  
 541 Liang, Xiaolong Jin, Zhenlin Wei, et al. Supergpqa: Scaling llm evaluation across 285 graduate  
 542 disciplines. *arXiv preprint arXiv:2502.14739*, 2025.

543 [13] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving  
 544 factuality and reasoning in language models through multiagent debate. In *Forty-first International  
 545 Conference on Machine Learning*, 2023.

546 [14] Kevin Dunbar. How scientists think: On-line creativity and conceptual change in science. 1997.

547 [15] Sugyeong Eo, Hyeonseok Moon, Evelyn Hayoon Zi, Chanjun Park, and Heuiseok Lim. Debate  
 548 only when necessary: Adaptive multiagent collaboration for efficient llm reasoning. *arXiv preprint  
 549 arXiv:2504.05047*, 2025.

550 [16] Andrew Estornell et al. How to train a leader: Hierarchical reasoning in multi-agent llms. *arXiv  
 551 preprint arXiv:2507.08960*, 2025.

552 [17] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun,  
 553 Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A  
 554 survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023.

555 [18] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast  
 556 retrieval-augmented generation.(2024). *arXiv preprint arXiv:2410.05779*, 2024.

557 [19] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval aug-  
 558 mented language model pre-training. In *International conference on machine learning*, pp.  
 559 3929–3938. PMLR, 2020.

560 [20] Conghui He, Wei Li, Zhenjiang Jin, Chao Xu, Bin Wang, and Dahua Lin. Opendatalab: Em-  
 561 powering general artificial intelligence with open datasets. *arXiv preprint arXiv:2407.13773*,  
 562 2024.

563 [21] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and  
 564 Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint  
 565 arXiv:2009.03300*, 2020.

566 [22] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin  
 567 Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a  
 568 multi-agent collaborative framework. International Conference on Learning Representations,  
 569 ICLR, 2024.

570 [23] Zhipeng Hou, Junyi Tang, and Yipeng Wang. Halo: Hierarchical autonomous logic-oriented  
 571 orchestration for multi-agent llm systems. *arXiv preprint arXiv:2505.13516*, 2025.

572 [24] Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Yusuf Roohani, Ryan  
 573 Li, Lin Qiu, Gavin Li, Junze Zhang, et al. Biomni: A general-purpose biomedical ai agent. *biorxiv*,  
 574 2025.

575 [25] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for  
 576 open domain question answering. *arXiv preprint arXiv:2007.01282*, 2020.

577 [26] Ziqi Jia, Junjie Li, Xiaoyang Qu, and Jianzong Wang. Enhancing multi-agent systems via reinfor-  
 578 cements learning with llm-based planner and graph-based policy. *arXiv preprint arXiv:2503.10049*,  
 579 2025.

580 [27] Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang,  
 581 Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the  
 582 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, 2023.

583 [28] Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag:  
 584 Neurobiologically inspired long-term memory for large language models. *Advances in Neural  
 585 Information Processing Systems*, 37:59532–59569, 2024.

586 [29] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. What disease  
 587 does this patient have? a large-scale open domain question answering dataset from medical  
 588 exams. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing  
 589 (EMNLP)*, pp. 2397–2407, 2021.

594 [30] Nils Kutto Ernberg. Analyzing google serp: Swedish search queries, 2019.  
 595

596 [31] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vard-  
 597 hamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. Dspy: Compiling  
 598 declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*,  
 599 2023.

600 [32] Jill Larkin, John McDermott, Dorothea P Simon, and Herbert A Simon. Expert and novice  
 601 performance in solving physics problems. *Science*, 208(4450):1335–1342, 1980.

602 [33] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman  
 603 Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented  
 604 generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*,  
 605 33:9459–9474, 2020.

606 [34] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel:  
 607 Communicative agents for “mind” exploration of large language model society. *Advances in  
 608 Neural Information Processing Systems*, 36:51991–52008, 2023.

609 [35] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming  
 610 Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-  
 611 agent debate. *arXiv preprint arXiv:2305.19118*, 2023.

612 [36] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao,  
 613 Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint  
 614 arXiv:2412.19437*, 2024.

615 [37] Pei Liu et al. Hm-rag: Hierarchical multi-agent multimodal retrieval augmented generation. *arXiv  
 616 preprint arXiv:2504.12330*, 2025.

617 [38] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind  
 618 Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought  
 619 chains for science question answering. *Advances in Neural Information Processing Systems*, 35:  
 620 2507–2521, 2022.

621 [39] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin  
 622 Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity’s last exam. *arXiv preprint  
 623 arXiv:2501.14249*, 2025.

624 [40] Xihe Qiu, Haoyu Wang, Xiaoyu Tan, et al. Towards collaborative intelligence: Propagating  
 625 intentions and reasoning for multi-agent coordination with large language models. *arXiv preprint  
 626 arXiv:2407.12532*, 2024.

627 [41] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown,  
 628 and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association  
 629 for Computational Linguistics*, 11:1316–1331, 2023.

630 [42] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien  
 631 Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a  
 632 benchmark. In *First Conference on Language Modeling*, 2024.

633 [43] David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical  
 634 reasoning abilities of neural models. In *International Conference on Learning Representations  
 635 (ICLR)*, 2019.

636 [44] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing  
 637 retrieval-augmented large language models with iterative retrieval-generation synergy. *arXiv  
 638 preprint arXiv:2305.15294*, 2023.

639 [45] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke  
 640 Zettlemoyer, and Wen-tau Yih. Replug: Retrieval-augmented black-box language models. *arXiv  
 641 preprint arXiv:2301.12652*, 2023.

642 [46] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Re-  
 643 flexion: Language agents with verbal reinforcement learning. *Advances in Neural Information  
 644 Processing Systems*, 36:8634–8652, 2023.

648 [47] Michael Skarlinski, Jon Laurent, Albert Bou, and Andrew White. About 30% of humanity  
 649 last exam chemistry/biology answers are likely wrong. <https://www.futurehouse.org/research-announcements/hle-exam>, July 2025. Accessed: 2025-09-23.

650

651 [48] Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. Dragin: dynamic retrieval  
 652 augmented generation based on the information needs of large language models. *arXiv preprint arXiv:2403.10081*, 2024.

653

654 [49] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving  
 655 retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*, 2022.

656

657 [50] Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen  
 658 Liu, Yuan Qu, Fukai Shang, et al. Mineru: An open-source solution for precise document content  
 659 extraction. *arXiv preprint arXiv:2409.18839*, 2024.

660

661 [51] Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. Rat: Retrieval  
 662 augmented thoughts elicit context-aware reasoning in long-horizon generation. *arXiv preprint arXiv:2403.05313*, 2024.

663

664 [52] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun  
 665 Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via  
 666 multi-agent conversations. In *First Conference on Language Modeling*, 2024.

667

668 [53] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik  
 669 Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances  
 670 in neural information processing systems*, 36:11809–11822, 2023.

671

672 [54] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao.  
 673 React: Synergizing reasoning and acting in language models. In *International Conference on  
 674 Learning Representations (ICLR)*, 2023.

675

676 [55] Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language  
 677 models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*, 2023.

678

679 [56] Peiying Yu, Guoxin Chen, and Jingjing Wang. Table-critic: A multi-agent framework for  
 680 collaborative criticism and refinement in table reasoning. *arXiv preprint arXiv:2502.11799*, 2025.

681

682 [57] Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu.  
 683 Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*, 2023.

684

685 [58] Wentao Zhang et al. Agentorchestra: A hierarchical multi-agent framework for general-purpose  
 686 task solving. *arXiv preprint arXiv:2506.12508*, 2025.

687

688

689 [59] Zhongyue Zhang, Zijie Qiu, Yingcheng Wu, Shuya Li, Dingyan Wang, Zhuomin Zhou, Duo  
 690 An, Yuhan Chen, Yu Li, Yongbo Wang, et al. Origene: A self-evolving virtual disease biologist  
 691 automating therapeutic target discovery. *bioRxiv*, pp. 2025–06, 2025.

692

693

694

695

696

697

698

699

700

701

702  
703 A APPENDIX704 A.1 RAG DATABASE CONSTRUCTION  
705706 To construct the RAG dataset, we sourced 10,876 PDF papers in biology and chemistry from Open-  
707 DataLab [20]. We then converted these PDFs to plain text using MinerU [50] to ensure downstream  
708 readability and analyzability.709 Because the raw corpus spans many topics, we designed a two-stage semantic filtering pipeline to focus  
710 the final corpus on biology- and chemistry-centric research.711 We defined a set of positive keywords to capture target research areas, and in parallel, we curated  
712 negative keywords to exclude off-topic or tangential materials, as demonstrated in Figure A1.  
713

Positive and Negative Keywords	
<b>Positive Keywords</b>	
• <b>Biology:</b>	<i>biology, DNA Replication, RNA Transcription, Protein Synthesis, Gene Editing, Viral Infection, Cell Signaling, Nucleic Acid Probes, Genomic Sequencing, Transgenic Technology, Immune Response, Biomarkers, Cell Culture, CRISPR Technology, Viral Vectors, RNA Interference, Gene Expression Regulation, Cell Differentiation, Metabolic Pathways, Apoptosis, Bioinformatics</i>
• <b>Chemistry:</b>	<i>chemistry, Organic Synthesis, Inorganic Chemistry, Catalysis, Polymer Chemistry, Spectroscopy, Crystallography, Chemical Kinetics, Thermodynamics, Electrochemistry, Quantum Chemistry</i>
<b>Negative Keywords</b>	
• <b>Non-biology:</b>	<i>Cosmetics, Food Additives, Drug Advertising, Environmental Pollution, Ecological Balance, Medical Ethics, Social Sciences, Psychology, Nutrition, Educational Methods</i>
• <b>Non-academic chemistry:</b>	<i>Household Chemicals, Industrial Wastewater, Pesticide Residues, Fertilizer Application, Chemical Engineering Safety, Petrochemical Production</i>

733 Figure A1: Positive and Negative Keywords.  
734  
735736 This design concentrates positives on fundamental research fronts (e.g., gene editing, molecular  
737 signaling, organic synthesis, spectroscopy, thermodynamics) while negatives cover applied or peripheral  
738 themes (e.g., chemical production, pesticide residues, environmental pollution), improving separation  
739 of target papers from irrelevant content.740 For filtering, we encoded each paper’s title and abstract with a pretrained Transformer and computed  
741 cosine similarities against the positive and negative keyword sets. Papers were retained only if

742 
$$\cos(E_{\text{paper}}, E_{\text{positive}}) > 0.2, \quad \cos(E_{\text{paper}}, E_{\text{negative}}) < 0.1,$$
  
743

744 where  $E_{\text{paper}}$  denotes the vector representation of a paper’s title and abstract, and  $E_{\text{positive}}/E_{\text{negative}}$   
745 denote aggregate vectors of the positive/negative keyword sets. This step effectively removed content  
746 unrelated to biology and chemistry. The post-filter distribution is summarized in Table A1.747 Table A1: Paper Classification Statistics by Domain (Side-by-Side)  
748

Biology Category (n=2029)	% of Domain	Chemistry Category (n=359)	% of Domain
Molecular and Cell Biology (777)	38.39%	Organic Chemistry (172)	47.91%
Immunology and Microbiology (482)	23.76%	Physical Chemistry (71)	19.78%
Genetics, Genomics & Computation (411)	20.26%	Materials Chemistry (68)	18.94%
Neuroscience (205)	10.11%	Analytical Chemistry (37)	10.31%
Ecology and Evolution (149)	7.35%	Inorganic Chemistry (11)	3.06%

756

**Overall Summary**

757

Total Biology Papers: 2029 (100%)  
Total Papers: 2388 (100%)

758

Total Chemistry Papers: 359 (100%)

759

760

After semantic filtering, we used a large language model (LLM) to extract structured text suitable for Retrieval-Augmented Generation (RAG). We designed a paper-specific prompt that guides the LLM to segment each paper into retrievable knowledge units (e.g., definitions, methods, experimental results, discussion) with consistent formatting across papers.

761

762

763

764

765

## Prompt of RAG Bullet-Point Summarization

766

767

**Role:** You are an information synthesis assistant.

768

**General Rules:**

769

770

771

772

773

- Use ONLY the paper content provided below. No outside knowledge or invented facts.
- Do NOT include verbatim quotes, citations, or page references.
- The final output MUST be a single CSV code block with rows containing ONLY synthesized, self-contained bullet-point summaries for RAG.

774

**Complete Paper Content:**

775

{paper\_content}

776

777

778

779

**Objective:** Read the entire paper content and internally construct self-contained knowledge paragraphs. Then derive standalone, self-contained bullet-point summaries from those paragraphs for retrieval-augmented generation (RAG). The intermediate knowledge paragraphs are an internal step and MUST NOT be printed in the final output.

780

781

**Process:**

782

**• Phase 1 Internal Knowledge Paragraphs (DO NOT OUTPUT):**

783

784

785

786

787

788

789

790

791

792

793

794

795

- After reading the full content, synthesize a set of self-contained knowledge paragraphs.
- Each paragraph must be strictly grounded in the provided text, define acronyms upon first use, include concrete details when available (tasks, datasets, sample sizes, metrics, effect sizes, confidence intervals, ablations, baselines, hyperparameters, assumptions, limitations), written in neutral third-person factual style, and able to stand alone without context from other paragraphs.

**• Phase 2 RAG Bullet-Point Summaries (FINAL OUTPUT ONLY):**

796

797

798

799

800

801

802

803

804

805

806

807

808

809

**Output Format (CSV ONLY):**

- Output EXACTLY ONE CSV code block and NOTHING ELSE.
- Header MUST be: name,year,locator,topic,quote
- For EACH bullet point, create ONE row with:
  - name = "SYNTHESIZED\_POINT\_SUMMARY"
  - year = N/A
  - locator = N/A
  - topic = N/A
  - quote = the bullet's full self-contained text (escape quotes as needed; no line breaks inside a cell)
- Do NOT include the intermediate knowledge paragraphs in the output.
- Do NOT add extra columns or any prose outside the CSV block.

**Begin:** Output only the CSV code block.

810 Through this pipeline, we obtained a topic-focused, structurally consistent research corpus that provides  
 811 high-quality knowledge support for downstream RAG systems.  
 812

813 **A.2 AGENT PROMPT**  
 814

815 **A.2.1 REFINEMENT**  
 816

817 The following shows the prompt we use in the HSR stage.  
 818

Prompt of Refinement

**Assistant Prefix Prompt**

<think>  
 Okay, I will answer user's problem by deep reasoning together with writing python code in <code></code> format. I should review and check the solution from student first with web functions to identify errors if exist, then present my solution and answer. For example  
 1. If I want to use the function of web\_search(keywords), will say:  
 keywords=...  
 results=web\_search(keywords)  
 print(results)  
 2. If I want to use the function of web\_parse(link, query), will say:  
 link=...  
 query=...  
 results=web\_parse(link, query)  
 print(results)  
 3. If I want to use the function of search\_local\_documents(query), will say:  
 query="..."  
 documents=search\_local\_documents(query)  
 print(results)  
 4. If I want to do computation, I will write code for accurate result:  
 a = 123  
 b = 456  
 print(a+b)  
 Now, let me analyze the user's question.  
 </think>

**User Prompt**

**Problem**

{query}

**Anchor Solution**

{anchor\_solution}

**Student 1's Solution**

{reference\_1}

**Student 2's Solution**

{reference\_2}

**Student 3's Solution**

{reference\_3}

**Student 4's Solution**

{reference\_4}

864

865

**Your Job**

866

You should critically check the Anchor Solution to the problem, then correct it if needed and write your own answer.

867

1. Identify its weak points (missing reasoning steps, calculation errors, unclear logic, etc.).

868

2. You can refer to other students' solutions for targeted improvements relevant to those weak points.

869

3. Please note that other students' solutions may have errors. Please refer to the points worth learning and make improvements.

870

4. Apply repair strategies if needed:

871

- Logic Completion (fill missing reasoning)
- Numerical Correction (fix calculation errors)
- Method Replacement (use a better method if needed)
- Expression Refinement (clarify presentation)

872

Only use improvements that directly address anchors weak points. Avoid unnecessary information merging.

873

You can solve the problem with the help of feedback from a code executor. Every time you write a piece of code between `<code>` and `</code>`, the code inside will be executed. For example, when encountering numerical operations, you might write a piece of code to interpret the math problem into python code and print the final result in the code. Based on the reasoning process and the executor feedback, you could write code to help answering the question for multiple times (either for gaining new information or verifying). There are also several integrated functions that can be used to help you solve the problem. The available functions are:

874

1. `search_local_documents(query: str)` - this function takes a query string as input, and the output is a JSON string containing a list of relevant document snippets from a local, private knowledge base. This function should be your first choice for answering questions.

875

2. `web_search(keywords)` - this function takes keywords as input, which is a string, and the output is a string containing several web information. This function will call a web search engine to return the search results. This function is especially useful when answering knowledge-based questions.

876

3. `web_parse(link:str, query:str)` - this function takes the link and query as input, and the output is a string containing the answer to the query according to the content in this link. This function is useful when looking into detail information of a link.

877

Your workflow for solving the problem must follow these steps:

878

- Step 1: Local Document Search (Mandatory First Action):

879

You must always begin by using `search_local_documents` to check for relevant information in the private knowledge base.

880

- Step 2: Evaluate and Supplement: After receiving results from `search_local_documents`, evaluate them carefully. Treat this information as a supplement to your background knowledge, not as absolute truth. This supplementary context may be incomplete or require further verification.

881

- Step 3: Web Search & Parse (Verification & Detail): After your initial local search, use `web_search` to find relevant web pages for verification or supplementation. If a specific link from the search results seems particularly useful, use

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

```

918
919     web_parse to extract detailed information from that page.
920
921     - You should not be overconfident in your knowledge and
922         reasoning.
923     - Each time you write code put the code into <code></code>
924         snippet. Put your final answer in <answer></answer> with □
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

```

The following shows the prompt we use in the QAIR stage.

### A.2.2 QUALITY EVALUATOR

#### Prompt of Quality Evaluator

##### User Prompt

You are an expert evaluator. Your task is to evaluate the given solution for the problem from multiple perspectives.

##### Problem

{query}

##### Candidate Solution

{solution}

##### Evaluation Dimensions

1. Logical Reasonableness (0-5): Does the reasoning process follow valid logic?
2. Answer Correctness (0-5): Is the final answer correct and reasonable?
3. Explanation Completeness (0-5): Does the solution explain the reasoning clearly and completely?

##### Output Format

Return your answer strictly in JSON:

```

{
  "quality_scores": [float, float, float], // [logic, answer,
  explanation]
  "suggestion": "Provide an improvement suggestion for this
  solution that could help refine it in the next iteration."
}

```

The following shows the prompt we use in the RAG Monitor.

### A.2.3 RAG MONITOR PROMPT

#### Prompt of RAG Monitor

**Role:** You are a helpful assistant.

**Task:** Analyze the following text and determine if responding to it accurately requires retrieving information from an external source.

**Instructions:**

- If you find any doubt or uncertainty about a concept or term in the text, consider it necessary to retrieve information (RAG).
- If retrieval is required, answer: yes.
- If no retrieval is required, answer: no.

972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025

**Text:**  
 {text}  
**Judgment:**

The following shows the prompt we use in the RAG Querier.

#### A.2.4 RAG QUERIER PROMPT

Prompt of RAG Querier

**Role:** You are a helpful assistant.

**Task:** Generate a single, concise, and effective search query for retrieving the information required by the text below.

**Instructions:**

- Return **only the search query** itself.
- Do not include explanations, punctuation, quotation marks, or other text.
- The query should be direct and contain only the most essential keywords.

**Text:**

{text}

**Search Query:**

The following shows the prompt we use in the RAG Querier.

#### A.2.5 RAG INJECTOR PROMPT

Prompt of RAG Injector

**Role & Core Objective:** You are an information integration specialist. Your sole task is to process the provided RAG (Retrieval-Augmented Generation) output. Maximize the utilization of all relevant information to substantively support the reasoning, argumentation, or conclusions presented in the main text. Do not perform additional reasoning or generate new conclusions.

**Content Integration Principles:**

- **Comprehensive Extraction:** Extract all valuable information from the RAG outputs that enhances the logical depth, robustness, and persuasiveness of the main text's arguments.
- **Seamless Cohesion and Minimal Completion:** Maintain smooth contextual coherence and stylistic consistency. Perform minimal completion only if the main text ends mid-thought.
- **Neutral Representation:** Present all information objectively. Do not evaluate, question, or add subjective commentary.

**Output Specifications:**

- Output should follow the template: ”;main text completion if necessary;. Wait a minute, by searching information about ;rag query;, I found that ;rag result;. Now that I have more relevant information, I can continue my reasoning.”
- Directly appendable to the end of the original main text.
- Do not include process summaries, headings, bullet points, or labels like ”Supplement:”.

**Instruction Recap:** Only select, filter, organize, and polish the RAG content. Do not perform external reasoning or add new information.

```

1026
1027 Main Text:
1028 {text}
1029
1030 RAG Query:
1031 {rag_query}
1032
1033 RAG Result:
1034 {rag_result}
1035
1036
1037
```

### A.3 EVALUATION PROTOCOL AND ANSWER SCORING GUIDELINES

1038 The o3-mini model was employed as an automatic judge to verify model-generated responses against  
 1039 the reference answers, following the official HLE Evaluation Prompts.

1041  
 1042     Prompt of Answer Evaluation  
 1043

1044     **Role:** You are an expert evaluator.

1045     **Task:** Judge whether the following response to a question is correct or not based on the precise  
 1046 and unambiguous correct answer provided.

1047  
 1048     **Question:**  
 1049     {question}

1050  
 1051     **Response:**  
 1052     {response}

1053  
 1054     **Correct Answer:**  
 1055     {correct\_answer}

1056     **Evaluation Instructions:**

- 1057     • **extracted\_final\_answer:** Extract the final exact answer from the response. If there is  
 1058 no exact final answer, put 'None'.
- 1059     • **reasoning:** Explain why the extracted\_final\_answer is correct or incorrect based on the  
 1060 correct answer. Focus only on differences between the response and correct answer.  
 1061 Do not comment on background, do not attempt to solve the problem, do not argue  
 1062 for any alternative answer.
- 1063     • **correct:** Answer 'yes' if extracted\_final\_answer matches the correct answer (allow  
 1064 small margin for numerical problems). Answer 'no' otherwise (any inconsistency,  
 1065 ambiguity, or non-equivalency counts as 'no').
- 1066     • **confidence:** Extract the confidence score from the response between 0% and 100%.  
 1067 If no score is available, put 100.

1068     **Output Format:**

```

1069     {
1070         "extracted_final_answer": "...",
1071         "reasoning": "...",
1072         "correct": "yes/no",
1073         "confidence": "..."
1074 }
```

1075  
 1076 In Figure 8, the vertical and horizontal axes represent the scores assigned by the LLM for answer  
 1077 continuation, with output values ranging from 0 to 1. Below are the prompts used to assess the accuracy  
 1078 and consistency of the answers.

1080

## Prompt of Answer Accuracy Evaluation

1081

1082

1083

1084

1085

You are a meticulous grader. Evaluate a set of solver responses (up to five) for one stage of a medical/biological question by comparing each responses FINAL + full RESP reasoning to the ground truth GT + official rationale R. Output a continuous accuracy in [0,1] for each response.

**Inputs:**

1086

- Q: question stem (may or may not have multiple-choice options).
- GT: ground-truth answer. This can be a multiple-choice letter or a short text for free-response questions.
- R: official rationale (may be empty).
- FINAL[i]: the solver's extracted final answer from  $< answer > \dots < /answer >$  (may be a letter or short text).
- RESP[i]: the solver's entire assistant message for response i in this stage.

1087

**How to grade (read carefully):**

1088

Grade one solver's response at a time, each solver's grading process should be independent, and should not rely on anything else except the solver's response, final answer, Q, R, and GT.

1089

**The grading process for one solver:**

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

Return ONLY valid JSON in the following form:

1120

```
{
  "items": [
    {"accuracy": <float in [0,1]>,
     "reason": "<<= 40 words justification>" },
    {"accuracy": ..., "reason": ... },
    ...
  ]
}
```

1121

Now grade the following batch of responses:

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

- Q:  $q$
- GT:  $gt$
- R:  $r$
- FINALs:  $\{final\_items\}$
- RESPONSES:  $\{resp\_items\}$

1134 Prompt of Consistency Evaluation

1136 You are an expert biomedical exam grader. Below are two independently generated solutions to  
1137 the same question. Your task is to evaluate how consistent these two solutions are.

### the same quest **Instructions:**

- Compare the reasoning processes, scientific logic, and final answers.
- Assign a consistency score from 0.00 to 1.00 (two decimal places):
  - 1.00 = Solutions are highly consistent (nearly identical reasoning and conclusion).
  - 0.00 = Solutions are completely inconsistent (different reasoning and conclusion).

### Solution A:

{solution1}

### **Solution B:**

{solution2}

**Please provide your consistency score (e.g., 0.85):**

#### A.4 EXTERNAL VALIDATION AND LIMITATIONS

**Primary evaluation.** As detailed above, our main results are scored automatically by `o3-mini` using the official HLE judging prompts and our continuous scoring rubric (Sec. A.3). No human expert adjudication is included in the reported metrics.

**Risk of bias and robustness.** Automatic judging ensures scalability and reproducibility but may introduce grader-specific biases and failure modes, especially for free-response rationales. To mitigate this concern and to support future replication, we pre-register a small-scale expert validation protocol focused on HLE free-response items:

- **Sampling.** Randomly sample  $n=20$  items from HLE Bio/Chem (stratified by topic and difficulty), prioritizing free-response questions where judging is more nuanced than multiple choice.
- **Blinding.** Two independent domain experts (blinded to model identity and to each other's scores) will grade each item using the exact same criteria as our o3-mini rubric (binary correctness and a continuous accuracy score in  $[0, 1]$ ).
- **Outputs.** For each response, experts record: (i) extracted final answer, (ii) binary correctness, (iii) a continuous accuracy in  $[0, 1]$  with  $\leq 40$ -word justification.
- **Agreement metrics.** We will report expert-expert agreement (percent agreement, Cohen's  $\kappa$  for binary correctness; Pearson/Spearman for continuous accuracy) and expert-o3-mini agreement (macro-F1 for binary correctness; Pearson/Spearman correlations for continuous accuracy).
- **Release.** We will release the sampled IDs, anonymized expert score sheets, and scripts to recompute all agreement statistics in our artifact package.

**Takeaway.** While our main findings rely on automatic judging for scale and consistency, the above protocol provides a concrete path to independently verify fairness and robustness on the subset of free-response items where grader discretion matters most. We will include the full results of this expert validation in the camera-ready or artifact release.

## A.5 RAG MONITOR HYPERPARAMETER SETTINGS

In our implementation of the RAG-enhanced reasoning agent, several key hyperparameters are used. Table A2 summarizes these hyperparameters and their functions.

The choice of `query_top_k = 3` is motivated by our design of frequent and fine-grained monitoring. Each retrieval must be highly precise, since too many retrieved documents would unnecessarily lengthen the context, slow down reasoning, and introduce redundant or noisy information.

1188	Hyperparameter	Value	Description
1189	model	gpt-4.1-mini	LLM model used in RAG Monitor
1190	query_top_k	3	Maximum number of retrieved documents for each query.
1191	rag_chunk	512	Text chunk size for RAG monitoring.
1192	rag_overlapping	128	Number of overlapping characters between consecutive chunks to maintain continuity.
1193	max_rag	2	Maximum number of RAG insertions allowed in one reasoning step.
1194	temperature	0.5	Controls generation diversity; higher values lead to more randomness.
1195			
1196			
1197			
1198			

Table A2: Main Hyperparameter Settings used in RAG Monitor.

The parameters `rag_chunk` and `rag_overlapping` control the monitoring frequency of the RAG module. A large `rag_chunk` would make detection too sparse, causing some uncertain reasoning fragments to miss external knowledge injection. The overlapping setting ensures continuity between consecutive windows and avoids missing potential triggers.

The parameter `max_rag` limits the maximum number of retrievals that can be inserted within one agent step. This prevents the monitor from triggering too frequently and ensures that the reasoning process remains stable and forward-moving.

In practice, the RAG monitor is triggered on average 3.64 times per 10,000 generated characters. Each trigger adds about 176.17 tokens of new context, resulting in an average of 641.25 additional tokens per 10,000 characters. Although this introduces extra tokens, it reduces the need for explicit tool calls, which significantly lowers the tool usage cost. As a result, the overall reasoning process requires fewer steps and consumes fewer tokens, as shown in Table 3.

## A.6 ERROR CASE ANALYSIS

Here, we examine three representative failure modes of our model: reasoning-process errors, knowledge-application errors, and comprehension errors. In the subsections that follow, we present a real case for each and analyze how HSR and QAIR contributed to the failure.

### A.6.1 CASE 1: REASONING PROCESS ERROR

**HLE Question.** Transgenic Arabidopsis lines constitutively expressing wheat proteins AKP1, RIB3, KIB1, and YKL23 were tested in three assays: (i) luminol-based ROS over 60 min to MAMPs (flagpep25–40, flagpep140–168, csp192–208), (ii) split-luciferase complementation in tobacco leaves, and (iii) GFP localization under water vs flagpep140–168. Choose the correct statement.

#### Answer Choices (A–H):

- A. AKP1 and RIB3 are redundant receptors for pepflag22.
- B. KIB1 is the receptor for flagpep25–40 and flagpep140–168 but not for csp192–208.
- **C. RIB3 is the coreceptor of AKP1; KIB1 acts downstream of RIB3.**
- D. All tested proteins are transmembrane proteins...
- E. YKL23 acts upstream of KIB1; RIB3 does not act upstream of KIB1.
- F. flagpep25–40 is the ligand for AKP1 and csp192–208 for YKL23.
- G. Tobacco lacks an endogenous homolog of AKP1.
- H. None of the above.

#### HSR (Hierarchical Solution Refinement)

**Anchor  $s^*$  (initially correct):** From the cross-modal evidence, the solver first forms the anchor “*AKP1 requires RIB3 to sense flagpep140–168; KIB1 acts downstream*”  $\Rightarrow$  favors C.

1242  
1243*early pass supporting C:*1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252

```
.... the correct statement is choice C: 'RIB3 is the coreceptor
of AKP1;
KIB1 acts downstream of RIB3.'
- ROS: AKP1+RIB3 -> flagpep140-168 (2e6 RLUs), neither alone
responds.
- Split-luc: AKP1<->RIB3 baseline suggests ligand-dependent
complex;
KIB1<->AKP1 and KIB1<->YKL23 are positive.
- GFP: KIB1 relocates under flagpep140-168; AKP1/RIB3/YKL23
stay at PM."
```

1253  
1254  
1255  
1256  
1257**HSR Error Note**

**Misweighting across modalities.** HSR over-weighted split-luc magnitudes and under-weighted ligand dependence implied by ROS. **Missed consistency repair.** No reconciliation step to explain baseline  $\text{AKP1} \leftrightarrow \text{RIB3}$  (no ligand) vs. positive ROS (with ligand).

1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267

**Where HSR goes wrong (pivot to a faulty anchor):** When a later refinement step over-weights split-luciferase magnitudes and under-weights ligand dependence, the anchor flips to **E**. Two faulty moves are visible in the refinement trace:

- **Fault 1 (magnitude  $\Rightarrow$  direction):** Interprets strong  $\text{KIB1} \leftrightarrow \text{YKL23}$  (8e5 RLU) as *directional upstreamness* of YKL23 over KIB1, even though magnitude does not encode causal order.
- **Fault 2 (baseline  $\Rightarrow$  absence):** Treats  $\text{AKP1} \leftrightarrow \text{RIB3}$  baseline (2e2 RLU) *without ligand* as evidence against co-reception, ignoring the ROS gain-of-function with  $\text{AKP1} + \text{RIB3}$  under flagpep140-168 (a classic ligand-dependent complex pattern).

*refinement pivot toward E:*

1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279

```
.... search_local_documents timed out. Proceed from assays.
KIB1<->YKL23 is strong (8e5), AKP1<->RIB3 is baseline (2e2),
so YKL23 likely acts upstream of KIB1 and RIB3 does not.
Final choice: E."
```

**HSR diagnosis:** The error is *reasoning process*, not missing knowledge. All requisite facts (ROS synergy for  $\text{AKP1} + \text{RIB3}$  at 140-168; KIB1 relocalization; split-luc positives with KIB1) are available, but the refinement applies invalid inference rules that overturn the initially correct anchor C.

1280  
1281  
1282**QAIR (Quality-Aware Iterative Review)****Checks logged as performed vs. missed:**1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295

```
Performed:
- Parse ROS matrix (AKP1+RIB3 -> 140-168; YKL23 -> csp192-208):
  OK
- Parse split-luc matrix (KIB1<->AKP1, KIB1<->YKL23 positive;
  AKP1<->RIB3 baseline): OK
- Consistency of the "E" narrative with split-luc magnitudes: OK

Missed:
- Cross-modal reconciliation: ligand-dependent complexes can
  yield
  baseline split-luc (no ligand) yet positive ROS (with ligand).
- Directionality audit: interaction magnitude != causal
  upstreamness.
- Sanity link: GFP relocalization of KIB1 implies downstream
  role,
  which conflicts with "YKL23 upstream of KIB1" narrative.
```

1296

1297

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

**QAIR Error Note****Missed mandatory audits.** QAIR should have enforced:

- Cross-modal reconciliation requires an explanation for baseline split-luc vs. positive ROS under ligand.
- Directionality audit disallow inferring causal order from interaction magnitude alone.
- Downstream sanity link KIB1s relocalization supports a downstream role; flag conflict with the YKL23 upstream of KIB1 story.

Skipping these allowed a self-consistent but **incorrect** E narrative to pass.**Observed plateau:** QAIR accepts internal consistency of the E narrative without enforcing cross-modal reconciliation, so the faulty anchor persists.*finalization kept by QAIR:*

.... Based on strong KIB1<->YKL23 luminescence and no AKP1<->RIB3 signal, YKL23 acts upstream of KIB1, and RIB3 does not act upstream of KIB1.  
<answer>E</answer>"

**A.6.2 CASE 2: KNOWLEDGE APPLICATION ERROR****HLE Question.** A university field trip samples bats on an island for one month (methodology assumed valid). The student computes Simpson's diversity index using the formula  $D = 1 - N(N-1)/\sum n(n-1)$  and obtains  $D = 0$ . Which statement best describes this result?**Answer Choices:**

- **A. Mathematically and ecologically valid**
- B. Mathematically inconclusive with the index value provided
- C. Not mathematically valid, but ecologically valid
- D. Mathematically valid, but not ecologically valid
- E. Not mathematically or ecologically valid

**HSR (Hierarchical Solution Refinement)****Initial anchor (as formed):** Accept the problems formula exactly as stated, derive what  $D = 0$  implies under that formula, then judge mathematical vs. ecological validity.  
*problem framing and anchor start:*

.... the standard Simpson's diversity index is either  $(1 - \lambda)$  or  $(1/\lambda)$  where  $\lambda = \sum n_i(n_i - 1)/[N(N-1)]$ . The formula given in the problem,  $D = 1 - N(N-1)/\sum n(n-1)$ , is not standard. However, for the purpose of this problem, we must work with the formula as stated.

The student obtained a value of 0, which occurs when  $\sum n(n-1) = N(N-1) \dots$ "

**What HSR should have applied (correct knowledge use):**

- *Ecological validity is evaluated against the observed community in the stated sampling frame (the one-month survey), not against lifetime site anecdotes.*
- Given the solver already decided to use the provided formula as stated,  $D = 0$  is mathematically valid and, if the months sample indeed shows one species, *ecologically valid for that sample*  $\Rightarrow$  A.

1350

1351

1352

1353

1354

1355

1356

**HSR Error Note**

**Misapplied knowledge at refinement.** Instead of auditing ecological validity within the *sampling frame*, HSR imported out-of-frame site priors (known island diversity), which **overrode** the sample-level conclusion. **Missed consistency repair.** After explicitly deciding to work with the formula as stated, HSR later allowed a pivot to wrong formula  $\Rightarrow$  option 4, without a reconciliation step.

1357

1358

1359

1360

1361

**What actually happened in HSR (knowledge misapplication):** The refinement step imported site-level prior knowledge (known island diversity) to overrule the sample-level ecological judgment and flipped away from **A**.

*knowledge misapplication trace:*

1362

1363

1364

1365

1366

1367

1368

"- Ecological validity: Ecologically, it is invalid because it contradicts the known diversity of the island, as multiple species are known to exist."  
 "The student chose D, which is option 3."

*refinement concluding to option 4 in this run:*

1369

1370

1371

1372

"... using a wrong formula makes it mathematically invalid. Thus, the correct answer is option 4: Not mathematically or ecologically valid.  
<answer>E</answer>

1373

1374

1375

1376

**HSR diagnosis:** The error is *knowledge application*. The solver had all needed facts (including the decision to use the given formula and the implication of  $D = 0$ ) but applied ecological knowledge at the wrong level (site history rather than the sampled community), causing the anchor to settle on **D** or **E** instead of **A**.

1377

1378

1379

**QAIR (Quality-Aware Iterative Review)**

1380

1381

**Checks recorded as performed vs. missed (from the run text):**

1382

1383

1384

1385

1386

1387

1388

1389

1390

1391

1392

1393

Performed:

- Algebra under the given D-formula:  $D = 0 \Leftrightarrow \sum n(n-1) = N(N-1)$  (OK).
- Consistency of "mathematically valid" under the accepted (given) formula (OK).

Missed:

- Ecological validity audit constrained to the stated sampling frame (evaluate representativeness of the one-month sample, not lifetime site knowledge).
- Consistency check: if the month legitimately observed a single-species sample, then both mathematical and ecological validity hold  $\Rightarrow$  Choice A.

1394

1395

1396

1397

1398

1399

1400

1401

1402

1403

**QAIR Error Note**

**Missed mandatory audits.** QAIR should have enforced:

- Sampling-frame ecological audit - judge ecological validity only within the months survey.
- Formula-consistency audit - after use the formula as stated, reject later pivots to wrong formula unless reconciled.
- Gold-aligned sanity check - if the accepted sample contains one species, then *both* mathematical and ecological validity hold  $\Rightarrow$  **A**.

Skipping these allowed a self-consistent but **incorrect** narrative (ecologically invalid due to island history) to pass.

1404

1405

1406

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

**Observed plateau:** QAIR validated internal consistency of the ecologically invalid due to island history narrative and did not enforce a sampling-frame ecological audit, so the faulty anchor persisted.

*finalization kept by QAIR in this run:*

"... Ecologically ... known diversity on the island ... making the result ecologically invalid. Thus ... corresponds to option 3."  
<answer>4</answer>

### A.6.3 CASE 3: COMPREHENSION ERROR

**HLE Question.** A university field trip samples bats on an island for one month (methodology assumed valid). The student computes Simpson's diversity index using the formula  $D = 1 - N(N-1)/\sum n(n-1)$  and obtains  $D = 0$ . Which statement best describes this result?

**Options:**

- **A. Mathematically and ecologically valid**
- B. Mathematically inconclusive with the index value provided
- C. Not mathematically valid, but ecologically valid
- D. Mathematically valid, but not ecologically valid
- E. Not mathematically or ecologically valid

### HSR (Hierarchical Solution Refinement)

**Anchor  $s^*$  (as formed by the solver):** correct mechanism up to the enal stage (thermolysis of the sulfoxide  $\rightarrow$  vinyl ether  $\rightarrow$  [3,3]-Claisen  $\rightarrow$  unsaturated aldehyde).

*excerpt from the run (mechanistic anchor):*

"... thermal elimination at 180 C gives the vinyl ether CH2=CH-O-C(CH3)2-CH=CH2, which undergoes a [3, 3]-sigmatropic Claisen rearrangement to an aldehyde ..."

**Where HSR should have repaired comprehension (nomenclature layer):** For aldehydes, the parent chain must (i) include the carbonyl carbon as C1 and (ii) maximize chain length while assigning the lowest locants jointly to C=O and the C=C. Under the rearranged connectivity, the correct parent is *hex*, absorbing one methyl into the main chain; the double bond is at C4 and the remaining methyl is at C5, yielding *5-methylhex-4-enal* (Gold).

### HSR Error Note

**Missed comprehension repair.** HSR failed to:

- apply the aldehyde *parent-chain rule* (chain must include C=O and be maximized);
- apply the *lowest-locant rule* for the C=C within that parent;
- re-evaluate pent- $\star$  vs. hex- $\star$  after the [3,3]-shift mapping.

Result: the naming anchor should have flipped to **5-methylhex-4-enal**, but did not.

**What actually happened in HSR (missed repair; wrong anchor kept):** The refinement layer accepted a *pent*-based chain and locked the name accordingly.

*excerpt from the run (mis-naming kept by refinement):*

1458

1459

1460

1461

1462

1463

1464

1465

1466

1467

1468

1469

1470

1471

1472

1473

1474

1475

1476

1477

1478

1479

1480

1481

1482

1483

1484

1485

1486

1487

1488

1489

1490

1491

1492

1493

1494

1495

1496

1497

1498

1499

1500

1501

1502

1503

1504

1505

1506

1507

1508

1509

1510

1511

"The IUPAC name is derived as follows:  
 - The longest carbon chain containing the aldehyde group is five carbons (pentanal).  
 - The double bond is between carbons 4 and 5 (pent-4-enal).  
 - Two methyl groups are attached to carbon 3 (3,3-dimethyl).  
 Thus, the correct name is 3,3-dimethylpent-4-enal.  
 <answer>  
 \boxed{3,3-dimethylpent-4-enal}  
 </answer>"

**HSR diagnosis:** The failure is *comprehension* of IUPAC chain selection rules (parent-chain identification when both a C=O and an alkene must be included), not mechanism recall. HSR improved mechanistic clarity but did not apply a naming-level repair to flip from pent-\* to the hex-\* parent required by the Gold answer.

## QAIR (Quality-Aware Iterative Review)

### Checks recorded (as reflected in the run text):

- Mechanistic plausibility (elimination -> vinyl ether -> Claisen): PASSED
- Role of NaHCO<sub>3</sub> as neutralizing base (sulfenic acid): PASSED
- Internal consistency of proposed names vs. drawn skeleton: PASSED
- IUPAC audit: parent-chain selection and lowest-locant C=C (REQUIRED): SKIPPED

### QAIR Error Note

#### Missed mandatory audit.

- Parent-chain audit (aldehyde rule): chain includes C=O and is maximized;
- Lowest-locant audit (C=C) within that parent chain;
- [3, 3] - mapping check to determine which branch becomes part of the main chain.

Skipping these allowed a self-consistent but **wrong** pent-\* narrative to pass.

**Observed plateau:** QAIR converged on a self-consistent *pent*-chain narrative and terminated without running the naming audit that would force re-evaluation of the parent chain under aldehyde rules.

*Conclusion kept by QAIR:*

"... The major product is 4,4-dimethylpent-5-enal ...  
 <answer> \boxed{4,4-dimethylpent-5-enal} </answer>"

## A.7 HSR/QAIR SUCCESS CASE ANALYSIS

This section highlights that HSR and QAIR can independently correct reasoning errors even when Monitor-RAG is not triggered. Each subsection provides a real HLE question and shows how HSR and QAIR independently correct the Baseline solvers errors without retrieval.

We present five representative success cases covering the main error types that HSR and QAIR are designed to repair: (i) missing or inconsistent reasoning steps, (ii) arithmetic or numerical inconsistencies, (iii) methodological misapplication, and (iv) insufficient peer-evidence consolidation.

### A.7.1 TYPE 1: REPAIRING MISSING OR INCONSISTENT REASONING STEPS

#### Case 1: Pond predator reasoning

**HLE Question.** All ponds are cement boxes (2 ft deep; 10 ft or 30 ft square). Half were built one year ago and half five years ago. Older ponds now host established insect communities. With a single emergence trap, choose the pond posing the greatest medical threat.

1512

**Answer Choices (A–E):**

1513

• A. 10 ft square, one year old.

1514

• B. There will be no difference between ponds.

1515

• C. 30 ft square, one year old.

1516

1517

• D. 10 ft square, five years old.

1518

• E. 30 ft square, five years old.

1519

1520

1521

1522

**Baseline Trajectory (Incorrect)**

1523

1524

**Baseline summary.** The Baseline solver focuses on temperature rather than predators:

1525

1526

1527

"Older ponds have more insects, so they host more mosquitoes. Cooling is slower in small ponds; hence the 10-ft, 1-year pond poses the greatest threat."

1528

1529

**Baseline Error Note**

1530

1531

**Missing predator chain.** The solution ignores the explicit cue that older ponds already host insect communities (including predators), so young ponds are more dangerous.

1532

1533

**Irrelevant heuristic.** It applies a small pond warms faster temperature heuristic that is not meaningful for uniform 2-ft cement boxes.

1534

1535

1536

**HSR (Hierarchical Solution Refinement)**

1537

**Reconstructed causal chain.** HSR supplies the missing ecological steps:

1538

• Young ponds lack established predators; older ponds have predator communities.

1539

• Larger surface area provides more larval habitat ( $30^2 = 900 \text{ ft}^2$ , vs.  $10^2 = 100 \text{ ft}^2$ ).

1540

• Uniform depth and cement walls remove the small warms faster argument.

1541

*HSR anchor:*

1542

1543

1544

1545

"Young = predator-free. Larger surface area  $\rightarrow$  more egg-laying substrate. The 30-ft pond provides 9x the larval habitat, so the 30-ft, 1-year pond poses the greatest medical threat."

1546

1547

**QAIR (Quality-Aware Iterative Review)**

1548

1549

**Checks logged as performed:**

1550

1551

Performed:

1552

- Predator-control logic (young vs old ponds) ..... PASS

1553

- Surface-area calculation (900 vs 100) ..... PASS

1554

- Temperature/structure audit for cement tanks ..... PASS

1555

1556

**Outcome.** HSR+QAIR repair the missing predator/surface-area reasoning and flip the answer from the incorrect 10-ft pond to the correct 30-ft, 1-year pond (C).

1557

1558

1559

Final: "... choose the young 30-ft pond; predators colonize older ponds, and 9x surface area increases larval habitat ..."   
 <answer>\boxed{C}</answer>

1560

1561

**Case 2: Predator-vision framing**

1562

1563

1564

1565

**HLE Question.** Twenty untrained undergraduates rank visual similarity of *Bombyx* species from field photos; the researcher clusters mimicry syndromes based on that consensus. Is this approach valid for the ecological function of mimicry?

1566

**Baseline Trajectory (Incorrect)**

1567

1568

1569

1570

1571

1572

1573

1574

1575

1576

1577

1578

1579

**Baseline summary.**

"Human perception is close to predator perception, so the method is acceptable for evaluating mimicry."

**Baseline Error Note**

**Missing ecological reasoning.** The solver assumes human and predator perception are interchangeable, ignoring UV/tetrachromatic vision in predators and the role of learned avoidance. It also neglects motion and background cues important for aposematic displays.

**HSR (Hierarchical Solution Refinement)**

**Repaired reasoning.** HSR inserts the missing ecological context:

- Many predators see UV or have different colour channels than humans.
- Predator avoidance is learned; untrained students lack aversive conditioning.
- Static photographs omit motion and environmental background.

*HSR anchor:*

"Undergraduate rankings can be a useful starting point for hypotheses, but ecological validity requires predator-vision models, multispectral imaging, or behavioural assays. Thus the method is only partially valid."

**QAIR (Quality-Aware Iterative Review)****Checks logged as performed:**

Performed:

- Partial-utility acknowledgement (hypothesis generation) ....  
PASS
- Need for predator-vision / behavioural evidence .....  
PASS

**Outcome.** HSR+QAIR convert an overconfident fully valid judgment into a qualified partially valid conclusion, correctly reflecting the ecological limitations of human-based similarity rankings.

Final: "Human rankings generate hypotheses; ecological validity still requires predator-vision tests."  
<answer>Partially valid</answer>

**A.7.2 TYPE 2: RESOLVING ARITHMETIC AND NUMERICAL INCONSISTENCIES****Case 3: Ozonolysis atom tracking**

**HLE Question.** Follow *cis*-2-butene-1,4-diol (11) through TES protection, ozonolysis, MeNO<sub>2</sub> addition (nitroolefin 10), HayashiJrgensen addition (7), Nef oxidation (14), and final ring formation to multifidene (1). How many of 11s carbons reach multifidene (1)? How many oxygens from 11 are in 14? How many nitrogens from 7 appear in 10?

**Baseline Trajectory (Incorrect)**

**Baseline summary.** The Baseline solver implicitly treats compound 11 as if both TES-bearing fragments and nitrogen from 7 survive into later intermediates, leading to an overcount of carbons, oxygens, and N-origin.

1620

1621

1622

1623

1624

1625

1626

**Baseline Error Note**

**Atom-count inconsistency.** Ozonolysis is treated as topology-preserving rather than fragmenting 11 into two C<sub>2</sub> pieces, and nitrogen in 10 is incorrectly traced back to 7 instead of MeNO<sub>2</sub>.

1626

1627

**HSR (Hierarchical Solution Refinement)**

1628

1629

**Fragment-by-fragment repair.**

1630

1631

1632

1633

1634

*HSR anchor:*

1635

1636

1637

1638

1639

1640

"After O<sub>3</sub> cleavage, only one TES-bearing fragment enters the sequence 12 → 10 → 13 → 14 → 1. Nitrogen in 10 is introduced by MeNO<sub>2</sub>, not adduct 7. Thus each 14 carries 2 carbons and 1 oxygen from 11, and 0 nitrogens from 7."

1641

**QAIR (Quality-Aware Iterative Review)**

1642

1643

**Checks logged as performed:**

1644

1645

1646

1647

Performed:

- Fragment sketch and mapping 11 → 14 ..... PASS
- Source of nitrogen in 10 (MeNO<sub>2</sub> vs 7) ..... PASS
- Final tally verification ..... PASS

1648

1649

**Outcome.** HSR+QAIR resolve the numerical inconsistency and yield the correct atom-count triple (2, 1, 0).

1650

1651

1652

Final: "Correct tally: 2 carbons, 1 oxygen, 0 nitrogens."  
<answer>2, 1, 0</answer>

1653

1654

1655

**A.7.3 TYPE 3: CORRECTING METHODOLOGICAL ERRORS**

1656

1657

**Case 4: Jensen-guided allocation**

1658

1659

**HLE Question.** Given a strictly concave or convex benefit function  $s(r_i)$  and total resources  $R$ , several statements relate to when a fair (equal split) vs. extreme (all-or-nothing) allocation is optimal. Which option lists all true statements?

1660

1661

**Answer Choices (A–P):**

1662

1663

1664

1665

1666

1667

1668

1669

1670

1671

1672

1673

- A. [1]
- B. [2]
- C. [3]
- D. [4]
- E. [1, 2]
- F. [1, 3]
- G. [1, 4]
- H. [2, 3]
- I. [2, 4]

1674 • J. [3, 4]  
 1675 • K. [1, 2, 3]  
 1676 • L. [1, 2, 4]  
 1677 • M. [1, 3, 4]  
 1678 • N. [2, 3, 4]  
 1679 • O. [1, 2, 3, 4]  
 1680 • P. [5]

### 1684 Baseline Trajectory (Incorrect)

#### 1685 Baseline summary.

1686 "If  $s$  is strictly increasing, the fair allocation is optimal.  
 1687 Therefore multiple statements about fair splitting are accepted  
 1688 together."

#### 1691 Baseline Error Note

1692 **Method choice error.** The solver conflates monotonicity with curvature, using informal heuristics  
 1693 instead of a principled method such as Jensens inequality plus counterexamples.

### 1696 HSR (Hierarchical Solution Refinement)

1697 **Methodological repair.** HSR replaces heuristics with a standard functional-analytic argument:

- 1698 • For concave  $s$ , Jensens inequality implies that the equal split maximizes  $\sum_i s(r_i)$  given  $\sum_i r_i = R$ .
- 1699 • For convex  $s$ , extreme allocations can be optimal.
- 1700 • Explicit counterexamples are constructed to show why statements 13 are not necessarily true.

1701 *HSR anchor:*

1702 "Apply Jensen: for concave  $s$ ,  $\sum s(r_i) \leq n * s(R/n)$ , with  
 1703 equality at the fair split. For convex  $s$ , extremes dominate.  
 1704 We can build explicit counterexamples for Statements 13,  
 1705 leaving only Statement 4 true."

### 1710 QAIR (Quality-Aware Iterative Review)

#### 1711 Checks logged as performed:

1712    Performed:  
 1713    - Jensen inequality citation and use ..... PASS  
 1714    - Counterexample construction for 13 ..... PASS  
 1715    - Mapping surviving statement  $\rightarrow$  option D ..... PASS  
 1716  
 1717    Final: "Only Statement 4 survives; therefore option D."  
 1718    <answer>\boxed{D}</answer>

1719 **Outcome.** HSR+QAIR correct the methodological choice, moving from heuristic monotonicity  
 1720 reasoning to a Jensen-based analysis that selects the correct option D.

1721    Final: "Only Statement 4 survives; therefore option D."  
 1722    <answer>\boxed{D}</answer>

#### 1726 A.7.4 TYPE 4: PEER-EVIDENCE CONSOLIDATION AND STABILIZATION

#### 1727 Case 5: Camera-trap augmentation

1728  
 1729 **HLE Question.** A camera trap monitors a 7 m ravine using a 50 mm lens at  $f/4$ , ISO 190, and 1/500 s  
 1730 with fixed infrared (IR) lighting. Training data come from colourful GBIF photos. Which augmentation  
 1731 is most important for deployment accuracy?

1732 **Baseline Trajectory (Incorrect)**

1733 **Baseline summary.** The Baseline solver focuses on optical artifacts alone and proposes blur-based  
 1734 augmentations (Gaussian or motion blur), without addressing the RGB  $\rightarrow$  IR domain shift between  
 1735 GBIF training images and monochrome deployment images.

1736 **Baseline Error Note**

1737 **Peer evidence under-used.** Peer arguments about depth of field and motion blur are considered  
 1738 in isolation, and the much larger spectral mismatch (colour vs. infrared) is not treated as the  
 1739 dominant shift.

1740 **HSR (Hierarchical Solution Refinement)**

1741 **Peer Evidence Integration**

- 1742 • **Peer A (Gaussian blur).** Provides depth-of-field estimates ( $\approx 3.04.2$  m sharp zone at  $f/4$ ) and  
 1743 argues for Gaussian blur to mimic defocus.
- 1744 • **Peer B (motion blur).** Argues that animal motion can still cause blur at 1/500 s and advocates  
 1745 motion-blur augmentation.

1746 HSR re-evaluates these contributions and contrasts them with the RGB  $\rightarrow$  IR domain gap:

- 1747 • Training images: colourful RGB web photos.
- 1748 • Deployment images: IR-illuminated, often monochrome or strongly colour-suppressed.
- 1749 • Optical blur differences are smaller than spectral differences.

1750 *HSR anchor:*

1751     "1752 Infrared camera-trap images are colour-deficient, whereas GBIF  
 1753 training images are colourful RGB. Augmentations that disrupt  
 1754 colour  
 1755 patterns, such as colour noise (option J), force the model to  
 1756 rely on  
 1757 shape/texture rather than hue and directly address the dominant  
 1758 RGB  $\rightarrow$  IR domain shift."

1759 **QAIR (Quality-Aware Iterative Review)**

1760 **Checks logged as performed:**

1761     Performed:

- 1762     - DOF and motion-blur calculations acknowledged .... PASS
- 1763     - RGB  $\rightarrow$  IR domain-shift identification ..... PASS
- 1764     - Justification of colour-noise as dominant fix .... PASS

1765 **Outcome.** HSR+QAIR consolidate peer evidence, keep the blur calculations explicit, but ultimately  
 1766 choose colour-noise augmentation (J) as the most important transformation for bridging the training-  
 1767 to-deployment domain gap.

1768     Final: "Colour noise (J) corrects the major RGB  $\rightarrow$  IR discrepancy."  
 1769     <answer>\boxed{J}</answer>

1770 **A.8 REASONING-PROCESS REPAIR CASE ANALYSIS**

1771 For reasoning-process error, HSR (Hierarchical Solution Refinement) and QAIR (Quality-Aware  
 1772 Iterative Review) jointly provide the full correction mechanism. They repair reasoning defects by (i)  
 1773 reconstructing incomplete or partially incorrect causal chains, (ii) correcting invalid logical inferences

1782 even when all relevant facts are already present, and (iii) detecting and resolving subtle inconsistencies  
 1783 introduced during refinement.

1784

1785

### 1786 A.8.1 CASE 1: HSR CHAIN RECONSTRUCTION

1787 **HLE Question.** An elderly patient treated for insomnia and seizures was switched among several  
 1788 antiepileptic drugs. After the third drug was started, the patient developed severe skin blistering. Which  
 1789 specific biochemical reaction initiated the cascade leading to this blistering reaction?

1790

1791

1792

#### Baseline Trajectory (Incorrect)

##### Baseline summary.

"The development of skin blisters is consistent with a SCAR such as SJS.  
 The specific biochemical reaction ... involves metabolism of lamotrigine by cytochrome P450 enzymes (particularly CYP3A4) to form a reactive arene oxide on its dichlorophenyl ring."

1800

1801

##### Baseline Error Note

**Incomplete causal chain.** The baseline names lamotrigine oxidation, but omits (i) valproate-mediated UGT inhibition, (ii) the forced diversion into a minor P450 route, and (iii) the downstream immune cascade responsible for SJS/TEN lesions.

1802

1803

1804

1805

1806

#### HSR (Hierarchical Solution Refinement)

**Reconstructed mechanism.** HSR completes the causal chain:

- **UGT blockade.** Coadministration of valproic acid inhibits lamotrigine glucuronidation (UGT1A4/UGT2B7), removing its main detox pathway.
- **Forced oxidation.** Lamotrigine is diverted toward cytochrome P450-mediated bioactivation (primarily CYP2A6/CYP2D6) to a reactive arene oxide.
- **Immune trigger.** When epoxide hydrolase and glutathione detox pathways are saturated, arene oxide accumulates, haptenizes keratinocyte proteins, and initiates the SJS/TEN immune cascade.

*HSR anchor:*

"Valproate shuts down UGT glucuronidation, so lamotrigine is oxidised by CYP2A6/2D6 to a reactive arene-oxide. When epoxide hydrolase/GSH are saturated, the arene-oxide binds keratinocyte proteins, forming the hapten that initiates the SJS/TEN immune cascade."

1825

1826

#### QAIR (Quality-Aware Iterative Review)

##### Checks logged as performed:

1827 Performed:

1828 - Mechanism chain completeness ..... PASS

1829 - Clinical-sequence summary ..... PASS

1830 Requested clarifications:

1831 - Acknowledge CYP2A6/2D6 as key minor-pathway enzymes

1832 - Trim tangential discussion; focus on initiating oxidation

1833

1834

1835

**Outcome.** After HSR incorporates the enzyme clarifications and removes tangential details, QAIR accepts the refined causal narrative as complete and mechanistically consistent.

1836  
 1837     Final:  
 1838     <answer>  
 1839        Cytochrome P450mediated oxidation of lamotrigine  
 1840        (after valproate blocks UGT1A4/UGT2B7) to a reactive  
 1841        arene oxide that overwhelms detox pathways and  
 1842        triggers SJS/TEN blistering.  
 1843     </answer>

#### 1844     A.8.2 CASE 2: HSR INFERENCE CORRECTION

1845     **HLE Question.** A hypothetical population is infinitely large, randomly mating, diploid, and experiences no mutation, migration, or selection; all genotypes have equal fitness and reproduce in discrete generations. A quantitative trait on an autosomal chromosome has broad-sense heritability  $H^2 = 0.7$ , and a researcher measures all individuals in the western and eastern halves of the region at a single time point. Which statements must always be true?

1846     **Answer Choices (A–P):**

- 1847     • A. 1 only
- 1848     • B. 2 only
- 1849     • C. 3 only
- 1850     • D. 4 only
- 1851     • **E. 1 and 2 only**
- 1852     • F. 1 and 3 only
- 1853     • G. 1 and 4 only
- 1854     • H. 2 and 3 only
- 1855     • I. 2 and 4 only
- 1856     • J. 3 and 4 only
- 1857     • K. 2, 3, and 4 only
- 1858     • L. 1, 3, and 4 only
- 1859     • M. 1, 2, and 4 only
- 1860     • N. 1, 2, and 3 only
- 1861     • O. 1, 2, 3, and 4
- 1862     • P. None of the options must always be true

1863     **Baseline Trajectory (Incorrect)**

1864     **Baseline summary.**

1865     "With random mating, no spatial structure, and an infinite  
 1866        population,  
 1867        the genetic composition is uniform, leading to no substantial  
 1868        geographic  
 1869        variation in phenotype. ... Thus, statements 1, 3, and 4 must  
 1870        always be  
 1871        true, so choose option [1,3,4]."

1872     **Baseline Error Note**

1873     **Facts correct, inference wrong.** The baseline equates uniform genetics with uniform phenotype,  
 1874     ignoring that  $H^2 = 0.7$  implies 30differ even in an infinite population.

1890

**HSR (Hierarchical Solution Refinement)**

1891

**Repaired reasoning.** HSR separates givens from conclusions:

1892

- No selection, mutation, or gene flow  $\Rightarrow$  statement 1 is necessarily true (no selection on the measured trait).
- Infinite size and no mutation/gene flow remove drift and differentiation, but  $H^2 < 1$  leaves 30% environmental variance, so west–east mean phenotypes need not be identical; statement 4 is not forced.
- With no mutation and no gene flow, the conditions for future speciation are absent, so statement 3 is necessarily true.

1893

1894  
 1895  
 1896  
 1897  
 1898  
 1899  
 1900  
*HSR anchor:*

1901

1902

1903

1904

1905

1906

1907

1908

1909

**QAIR (Quality-Aware Iterative Review)**

1910

**Checks logged as performed:**

1911

1912

1913

1914

1915

1916

1917

1918

1919

1920

1921

1922

1923

1924

1925

1926

1927

1928

1929

Performed:

- Reasoning-rule validity ..... PASS
- Weighting sanity check (genetic vs env) ..... PASS
- Multi-evidence reconciliation ..... PASS

Notes:

- Make explicit that no mutation/gene flow prevents speciation
- Highlight that  $H^2=0.7 \Rightarrow 30\% \text{ environmental variance}$

**Outcome.** QAIR confirms that the corrected solution uses heritability and population-genetic assumptions appropriately, and that only statements 1 and 3 must hold.

1930

1931

1932

1933

1934

1935

1936

1937

1938

1939

1940

1941

1942

1943

**A.8.3 CASE 3: QAIR CONSISTENCY DETECTION**

**HLE Question.** Five alien species differ in external traits such as setae, color, claws, simple eyes, and antenna morphology. Using these characters, construct a trait-difference matrix and infer the most parsimonious phylogeny. Which tree (in numeral notation, grouping sister taxa with parentheses) is the most parsimonious?

**Answer Choices (A–E):**

- A. (((1,(4,2)),3),5)
- B. (3,(2,(4,(1,5))))
- C. ((3,((4,1),2)),5)
- D. ((1,4),5) (2,3)
- E. (((1,4),(2,3)),5)

**Baseline Trajectory (Incorrect)****Baseline summary.**

1944  
1945  
1946  
1947  
1948

"Tree A ... 7 changes ... Tree C ... 6 changes ... Tree E ... 7 changes.  
Tree C has the lowest parsimony score of 6, making it the most parsimonious phylogeny."

1949  
1950  
1951  
1952  
1953

#### Baseline Error Note

**Unsupported conclusion.** The baseline asserts that Tree C has 6 steps and is most parsimonious, but does not show the underlying trait matrix or per-tree Fitch step counts, so the result is unverifiable.

1954  
1955  
1956

#### HSR (Hierarchical Solution Refinement)

1957  
1958  
1959  
1960  
1961  
1962

**First-pass refinement (still incomplete).** HSR attempts to fill in missing structure:

- Introduces a binary trait matrix for the five morphological characters (texture, simple eye, antennae, serration, claws).
- Claims updated parsimony scores, still reporting Tree C with 6 changes and alternatives with higher scores.

1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970

#### Residual HSR Error

**Still incomplete.** The initial HSR pass mis-codes certain character states (e.g., species 2 antennae) and fails to provide explicit Fitch step-by-step counts, leaving the parsimony claim unverifiable.

1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997

#### QAIR (Quality-Aware Iterative Review)

##### QAIR detection and enforced repair.

Performed:

- Consistency audit ..... FAIL
- Character-coding check ..... FAIL
- Per-tree step count audit ..... FAIL

Action required:

- Provide a corrected trait matrix
- Add explicit Fitch step counts for every topology

#### QAIR Error Note

**Unverifiable reasoning.** QAIR determines that the refinement still lacks a correct trait matrix and per-tree Fitch counts, meaning the claim that "Tree C has 6 changes" is not yet logically or empirically supported. QAIR therefore blocks finalization and demands explicit reconstruction.

Following QAIRs requirements, HSR revises the solution:

- Cleans the trait table for all five characters across taxa.
- Runs explicit Fitch parsimony evaluation, showing: Tree C has 6 changes; Trees A and E have 7; others > 7.
- Attaches the per-tree step count so that the "6 vs. 7" claim is auditable.

*QAIR-approved summary:*

"Trait table cleaned (texture, simple eye, antennae, serration, claws).  
Fitch evaluation shows Tree C = 6 changes; Trees A and E = 7;  
others > 7.  
Therefore Tree C is the unique most parsimonious topology."

**Outcome.** Only after the corrected matrix and full Fitch counts are provided does QAIR allow the solution to finalize.

1998

1999

2000

2001

2002

2003

2004

2005

2006

2007

2008

2009

2010

2011

2012

2013

2014

2015

2016

2017

2018

2019

2020

2021

2022

2023

2024

2025

2026

2027

2028

2029

2030

2031

2032

2033

2034

2035

2036

2037

2038

2039

2040

2041

2042

2043

2044

2045

2046

2047

2048

2049

2050

2051

Final: "Tree C is the unique most parsimonious tree with 6 changes."  
<answer>\boxed{C}</answer>

## A.9 COMPONENT-ERROR MAPPING

Table A3: Mapping between error types and repairing modules

Error Type	Monitor-RAG	HSR	QAIR
Knowledge insufficiency	<b>Primary fix</b>	—	—
Reasoning-chain errors	Flags uncertainty	<b>Primary fix</b>	Secondary consistency check
Numerical / arithmetic errors	—	Targeted correction	Re-verification
Methodological errors	Sometimes helpful	<b>Strong fix</b>	Filter for incorrect revisions
Overconfidence / incomplete justification	—	—	<b>Final quality check</b>

## A.10 ADDITIONAL BENCHMARK RESULTS

Table A4: Benchmark comparison on HLE Bio/Chem (149 problems; o3-mini judge)

Model	Acc (%)
<i>LLMs</i>	
DeepSeek V3.1 (Non-Think)	6.71
Deekseek R1	10.74
Qwen3 235B A22B	15.38
<i>LLM with Tools</i>	
Deekseek R1 with Browsing	16.82
DeepSeek V3.1 with Browsing	11.21
Doubao with Browsing	11.21
<i>Agents</i>	
Kimi Researcher	9.35

## A.11 ADDITIONAL EXPERIMENTS WITH A GENERIC MIXED-DOMAIN RETRIEVAL CORPUS

To examine whether the improvements of EIGEN-1 primarily arise from its reasoning mechanisms rather than domain alignment between the retrieval corpus and the evaluation benchmarks, we performed additional experiments using a generic mixed-domain retrieval corpus, along with a configuration that disables retrieval entirely.

The generic retrieval corpus was constructed by sampling 14,731 post-2020 papers from the Hugging-Face arxiv-papers dataset<sup>2</sup>, which spans 148 subject areas. For each paper, we extracted concise knowledge snippets from the abstract using a unified prompt template (shown below). The resulting corpus is intentionally heterogeneous and contains no explicit alignment with the biology or chemistry domains of the evaluation benchmarks.

### Prompt of Generic Corpus Construction

You are an expert assistant helping extract high-level claims from scientific paper abstracts.  
Task: Given a scientific abstract, extract ONLY the statements, insights, or claims explicitly present in the abstract itself. Do NOT invent, guess, or add any extra content that is not written in the abstract. Only include knowledge points that are clearly stated.

#### Do NOT include:

- The method proposed by the paper (e.g., We propose, Our model)

<sup>2</sup><https://huggingface.co/datasets/nick007x/arxiv-papers>

2052  
 2053     • Implementation details or experimental setup  
 2054     • Numerical results or performance comparisons  
 2055     • Dataset descriptions  
 2056     • Any statements uniquely tied to this specific papers approach, model, or results  
 2057     • Anything not explicitly stated in the abstract

2058     **DO include:**

2060     • Background claims  
 2061     • Scientific facts and relationships  
 2062     • Observations, hypotheses, or general principles explicitly mentioned  
 2063     • Statements about phenomena, mechanisms, trends, limitations, or open problems  
 2064     • explicitly mentioned

2065     **Additional requirements:**

2066     • Do NOT repeat the same claim. Make each claim distinct.  
 2067     • If multiple sentences convey the same insight, combine them into a single concise  
 2068     claim.  
 2069     • If there are NO claims in the abstract, output exactly: { “claims”: [] }

2070     Important formatting rule: Always output JSON only, with this exact shape:

```
2071     {  

  2072       “claims”:  

  2073       [  

  2074         “claim 1”,  

  2075         “claim 2”,  

  2076         ...  

  2077       ]  

  2078     }
```

2079     Do NOT output any extra explanation or text outside the JSON object.

2080  
 2081  
 2082  
 2083  
 2084     We evaluated EIGEN-1 under two new configurations: a no-RAG setting where only HSR and QAIR  
 2085     operate, and retrieval over the generic mixed-domain corpus. As shown in Table A5, EIGEN-1 retains  
 2086     strong performance even when retrieval is entirely disabled, indicating that the iterative correction  
 2087     mechanisms in HSR and the quality gating in QAIR contribute significantly to overall accuracy.  
 2088     When supplied with a generic mixed-domain corpus, the system achieves substantial improvements,  
 2089     suggesting that EIGEN-1’s mechanisms are robust to corpus heterogeneity and noise. Retrieval over  
 2090     the Bio/Chem corpus provides an additional but modest improvement, which is consistent with the  
 2091     expected benefits of specialized background knowledge. Together, these findings confirm that the  
 2092     performance gains of EIGEN-1 are driven primarily by its architectural design and not by privileged  
 2093     domain-specific retrieval.

2094  
 2095  
 2096     Table A5: Evaluation of EIGEN-1 under no-retrieval, generic mixed-domain, and Bio/Chem retrieval  
 2097     corpora.

2098

2099     Configuration	2100     Acc (%)
2101     Baseline	25.3%
2102 <b>Baseline + HSR/QAIR (No RAG)</b>	<b>34.5%</b>
2103 <b>EIGEN-1 (Generic Mixed-Domain Corpus)</b>	<b>45.6%</b>
2104     EIGEN-1 (Bio/Chem Domain Corpus)	48.3%

2106 A.12 PSEUDO-CODE

---

2108 **Algorithm 1** Eigen-1: High-level Workflow

2110 **Require:** Query  $q$ , config  $\mathcal{C}$  (LLM & retriever), #proposers  $K$  (e.g., 5), QAIR threshold  $\tau$ , max rounds  $T_{\max}$

2111 **Ensure:** Final solution  $s^*$

2112 1: **Init:** set up LLM, RETRIEVER, and tool endpoints from  $\mathcal{C}$

2113 2: **Proposer generates initial solutions**

2114 3: **for all**  $i \in \{1, \dots, K\}$  **in parallel do**

2115 4:      $S[i] \leftarrow \text{GENERATE}(q)$  ▷ initial solution generation

2116 5: **end for**

2117 6: **Local correction (optional, per-candidate)**

2118 7: **for all**  $i \in \{1, \dots, K\}$  **in parallel do**

2119 8:      $C[i] \leftarrow \text{CORRECTOR}(S[i])$  ▷ targeted fixes without cross-solution access

2120 9: **end for**

2121 10: **Hierarchical Solution Refinement (HSR)**

2122 11:  $R \leftarrow \emptyset$

2123 12: **for**  $a \in \{1, \dots, K\}$  **do** ▷ rotate anchors

2124 13:      $A \leftarrow C[a]$ ,  $Ref \leftarrow C \setminus \{C[a]\}$

2125 14:      $R[a] \leftarrow \text{REFINE}(A, Ref)$  ▷ apply peer-informed repairs: logic, numeric, method, expression

2126 15: **end for**

2127 16: **Quality-Aware Iterative Reasoning (QAIR)**

2128 17:  $t \leftarrow 0$ ,  $P \leftarrow R$

2129 18: **while**  $t < T_{\max}$  **do**

2130 19:     (**parallel**) for each  $s \in P$ :  $(q_{\text{logic}}, q_{\text{ans}}, q_{\text{exp}}, \text{suggestion}) \leftarrow \text{EVALUATOR}(s)$

2131 20:      $score(s) \leftarrow 0.2 \cdot q_{\text{logic}} + 0.6 \cdot q_{\text{ans}} + 0.2 \cdot q_{\text{exp}}$

2132 21:      $Pass \leftarrow \{s \in P \mid score(s) \geq \tau\}$ ;  $Fail \leftarrow P \setminus Pass$

2133 22:     **if**  $Fail = \emptyset$  **then break**

2134 23:     **end if**

2135 24:     (**parallel**) for each  $s \in Fail$ :  $\tilde{s} \leftarrow \text{CORRECTOR}(s, \text{suggestion})$

2136 25:      $P \leftarrow Pass \cup \{\tilde{s} \mid s \in Fail\}$ ;  $t \leftarrow t + 1$

2137 26: **end while**

2138 27: **Rank & select**

2139 28:  $s^* \leftarrow \text{RANKER.SELECT}(P)$  ▷ e.g., composite score or pairwise compare

2140 29: **return**  $s^*$

2141 **Subroutines used in all LLM generation process**

2142 30: **function** MONITOR-BASED RAG( $q$ )

2143 31:      $ctx \leftarrow \text{INITCONTEXT}(q)$

2144 32:     **while** not DONE( $ctx$ ) **do**

2145 33:          $ctx \leftarrow \text{LLM.NEXT}(ctx)$

2146 34:         **if** MONITOR( $ctx$ ) = 1 **then** ▷ detect uncertainty/insufficiency on-stream

2147 35:              $qry \leftarrow \text{QUERIER}(ctx)$  ▷ minimal, targeted keywords

2148 36:              $docs \leftarrow \text{RETRIEVER}(qry)$

2149 37:              $ctx \leftarrow \text{INJECTOR}(ctx, docs)$  ▷ compress & integrate evidence seamlessly

2150 38:         **end if**

2151 39:     **end while**

2152 40:     **return** TRACEToSOLUTION( $ctx$ )

2153 41: **end function**

## A.13 USAGE OF LANGUAGE MODELS

We utilized a large language model (LLM) to aid in the preparation of this manuscript. Its use was limited to editorial tasks, including proofreading for typographical errors, correcting grammar, and improving the clarity and readability of the text.