GENBENCH: A CROSS-MODAL MULTI-HOP GENOMIC REASONING BENCHMARK WITH CONFIDENCE-COEVOLVING AGENTIC LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models answer genomics questions fluently, but their performance degrades sharply when answering requires chaining multiple pieces of evidence rather than recalling an isolated fact. We introduce **GenBench**, a 90,379-node, 236,131-edge genomic knowledge graph benchmark built around 210 manually curated seed genes and 5,084 automatically generated, multi-hop, cross-modal question–answer (QA) items, quality-checked by an automated verification pass and audited on a sample by domain experts. On this benchmark, zero-shot Qwen3-8B achieves only 0.2304 accuracy, while GPT-4o and Claude Opus-5 remain below 0.32. Specialized biomedical baselines, including BioReason, Intern-S1-mini, and Evolla, achieve overall accuracies of 0.1676, 0.2807, and 0.2950, respectively, with BioReason dropping to 0.027 on the subset requiring sequence-level evidence. We develop an evidence-aware agent named **CoRA** (Co-evolving Retrieval Agent) that acquires cross-modal evidence from the graph during reasoning and jointly learns retrieval, trajectory-level credit assignment, and graph confidence through four coupled mechanisms: GRAD (Gated Reward Advantage Distillation), ocbook (open-/closed-book), ROC (Rescored On-policy Credit), and CoCG (Co-evolving Confidence Graph). With supervised warm-start training and graph-based tool access, Qwen3-8B reaches 0.5329 accuracy, increasing to 0.5948 with our full GRPO training procedure, a 158% relative improvement over the zero-shot baseline. The gain comes from evidence access rather than scale, which shows that **GenBench** resists parametric-only solutions and rewards explicit, confidence-aware retrieval for compositional genomic reasoning. We further identify a substantial multiple-choice/open-ended gap and provide ablations and implementation diagnostics clarifying the training framework's behavior. We release the graph, QA items with complete reasoning chains, and training code.

1 INTRODUCTION

Recent advances in LLMs and reasoning-oriented agents have substantially improved their ability to solve complex scientific and biomedical tasks by combining parametric knowledge with iterative reasoning, tool use, and external evidence. In genomics, however, many questions cannot be answered by recalling an isolated fact: they require connecting heterogeneous, distributed, and unequally reliable evidence across genes, diseases, variants, pathways, sequences, and other biological entities, and these sources are not equally reliable. Even a genomics-specialized model, BioReason, reaches only 0.027 accuracy when sequence-level evidence is required. Multi-hop reasoning over such evidence introduces two coupled challenges beyond conventional QA: *which evidence to acquire at each step* and *how much to trust it*. Both are compounded by long-horizon dependencies in which an early error can propagate through subsequent hops while a correct final answer reveals little about which intermediate actions contributed; Genomic reasoning therefore depends on acquiring and composing the right evidence over several steps, more than on recalling additional facts.

Existing benchmarks only partially capture this setting: general multi-hop QA benchmarks and their agentic-retrieval successors (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022; Mavromatis et al., 2025; Nahid & Rafiei, 2025) operate over textual evidence rather than heterogeneous biological ev-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

idence acquired during graph traversal; multimodal KGQA benchmarks such as HOME-KGQA (Egami et al., 2026) work at much smaller, non-scientific scale; and genomic benchmarks such as GeneTuring (Shang et al., 2025) and GenomeQA (Long et al., 2026a) evaluate knowledge recall or sequence understanding rather than multi-hop reasoning over a heterogeneous evidence graph. No existing resource requires composing multiple biological relations while retrieving modality-specific evidence on demand, with supervision for the reasoning path itself. Existing training methods (credit assignment for agentic RL, knowledge-graph co-evolution, RL-trained KG traversal) address this only in part (full comparison, Appendix A; benchmark and foundation-model comparison, Appendix B.10).

To address this gap, we introduce **GenBench**, a cross-modal, confidence-annotated benchmark for multi-hop genomic reasoning: 90,379 nodes and 236,131 edges built around 210 manually curated seed genes, paired with 5,084 QA items spanning eight task types and one to five reasoning hops. The graph integrates seven populated modality families (protein sequence, protein structure, sequence conservation, expression, splice scores, post-translational modification sites, and signalling roles), exposed through explicit graph relations instead of being flattened into text. Each QA item's complete reasoning chain (source/target nodes, relation, evidence type, database, and per-hop confidence) makes the reasoning path itself supervision, not only the final answer, and a gene-level train/test holdout reduces direct leakage across related questions. An expert audit of a held-out sample (§2.2.1) found 38 of 45 reviewed answers correct and 39 of 45 chains biologically valid.

GenBench provides the setting for a corresponding agentic learning problem. An effective agent must not only reason over the graph, but also decide when external evidence is needed, retrieve the appropriate modality, attribute credit to the evidence-bearing actions that contributed to an answer, and adapt its beliefs about evidence reliability as it gains experience. We therefore develop **CoRA (Co-evolving Retrieval Agent)**, an evidence-aware agentic framework in which retrieval, policy learning, evidence attribution, and graph confidence are coupled within a single GRPO optimization process. Four mechanisms address complementary aspects of this problem: **ocbook** lets **CoRA** adaptively bypass graph retrieval when parametric knowledge suffices; **GRAD** provides a teacher-free on-policy learning signal for optimizing multi-step trajectories; **ROC** assigns credit to evidence-bearing actions by measuring agreement across multiple views of the same trajectory; and **CoCG** updates graph-edge confidence alongside the policy update, letting the evidence environment evolve with **CoRA**'s experience. All four share one rollout batch: policy outcomes provide the learning signal, evidence attribution determines which actions receive credit, confidence updates alter the graph state, and the resulting graph state influences later retrieval. The interaction runs both ways: *reasoning changes which evidence the graph trusts, and graph confidence changes which evidence the policy sees next* (Figure 1).

Our contributions are as follows:

- **Problem Formulation.** We formalize confidence-aware, cross-modal multi-hop genomic reasoning: an agent must retrieve evidence across modalities mid-rollout from a per-edge-confidence-annotated knowledge graph and predict a grounded answer, rather than recovering it from parametric memory alone.
- **Benchmark.** We present **GenBench**,a 90,379-node, 236,131-edge genomic knowledge graph benchmark built around 210 curated seed genes and paired with 5,084 multi-hop QA items spanning eight task types and one to five hops, with a gene-level train/test holdout and complete, confidence-annotated reasoning chains for every item.
- **Framework.** We propose **CoRA**, an evidence-aware agent built on four coupled mechanisms sharing a single GRPO rollout batch: GRAD, a teacher-free gated policy-gradient objective; ROC, an agreement-gated multi-view credit-assignment scheme; CoCG, which co-evolves per-edge graph confidence alongside the policy; and ocbook, an adaptive, router-free open/closed-book retrieval bypass gated by **CoRA**'s own correctness signal.
- **Evaluation. CoRA** raises Qwen3-8B's accuracy from a zero-shot 0.2304 to 0.5948 — a 158% relative gain — exceeding zero-shot GPT-4o and Claude Opus-5 by 0.28 and 0.34 respectively, backed by a seven-configuration ablation isolating each mechanism's individual contribution and a forward-pass diagnostic of ROC's per-view credit signal.

We release the graph, the QA items with complete reasoning chains, and the training code, together with experimental results and analyses of the remaining limitations and ablations.

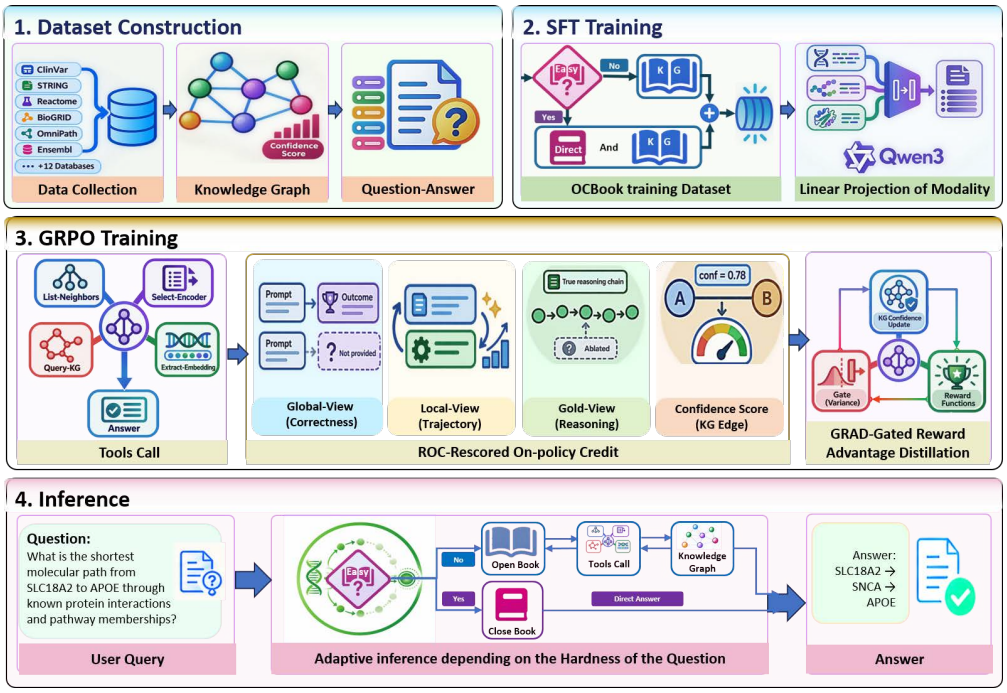


Figure 1: The **GenBench** training pipeline. (1) Dataset Construction: Genomic data from raw sources (ClinVar, STRING, Reactome, BioGRID, OmniPath, Ensembl, and others) are collected into a confidence-scored knowledge graph, from which multi-hop cross modal QA items are generated. 2.SFT Training: For sft the generated dataset is splited into open-book and closed-book trajectories and the modality embeddings from the encoders are projected for SFT warm-start training. (3) GRPO Training Pipeline: on-policy rollouts are sampled with tool calls against the graph, rescored across ROC's four views (local, global, gold, and prior-confidence), gated into a reward advantage by GRAD, and used to update both the policy and the knowledge graph's edge confidences (CoCG) within the same optimization step. (4) Inference: given a query, the trained model (with one shared adapter for both modes) decides whether to retrieve from the graph or answer directly, then produces a final answer grounded in its reasoning trajectory.

2 Dataset Pipeline

GenBench is a 90,379-node, 236,131-edge knowledge graph across 11 node types and 20 relation types, built around 210 manually curated seed genes and paired with 5,084 QA items spanning eight task types at one to five reasoning hops (Table 6). This section summarizes construction (§2.1) and the resulting benchmark's distinguishing dimensions (§2.2); full construction mechanics and a pipeline visual summary (Figure 3) are in Appendix B.

2.1 Dataset Collection

We construct **GenBench** in two layers: a confidence-scored knowledge graph built from public biomedical databases (§2.1.1), and a benchmark of QA items with human-audited reasoning chains extracted from that graph (§2.1.2).

2.1.1 Graph Construction

We choose GenBench's 210 seed genes by evidence ranking: for each disease we query four complementary sources (ClinGen/OMIM, clinically reviewed; Open Targets/PubTator, from association data and literature mining), combine their evidence additively per (disease, gene) pair into a tiered ranking, and take the deduplicated union of top-ranked candidates as seeds, entered as expert-reviewed edges before any automated expansion (per-source weights/tiers in Appendix B.1).

Eighteen live collector APIs (Table 4, Appendix B.2) then expand the graph across gene–disease association, pathway participation, protein interaction, causal signalling, functional annotation, and sequence-level structure, harmonizing results onto canonical identifiers (Ensembl, UniProt, dbSNP/ClinVar) to avoid duplicate entities.

Per-edge confidence is initialized from each source's own evidentiary semantics rather than a single global rule — ClinVar star ratings, GO evidence codes, STRING scores, among others — with curated seed edges entering at

$$c_e = \mathrm{clip}\left(\frac{\text{evidence score}}{5},\ 0,\ 1\right) \tag{1}$$

and full bi-temporal provenance retained per assertion (Appendix B), so confidence changes made during training can be traced and undone, which CoCG relies on (§3.5). Ten modality families are defined in the schema, seven populated in this release (full list, §2.2.1), each attached via an explicit relation-to-family map rather than a fixed per-node-type rule, so relevance is a property of the edge traversed, not the node (task types depending on the three unpopulated families are held out). Two filters complete construction: a non-empty-modality invariant, and a frozen, versioned snapshot of all eighteen collector responses, fixing the graph our numbers were computed against (full weights/collector table/modality map, Appendix B).

2.1.2 Benchmark Sample Construction

For each candidate query entity pair, we extract the shortest path, highest-confidence path, and all simple paths between them, excluding `DISEASE`, `TISSUE`, `GO_TERM`, and `PUBLICATION` nodes as intermediate hops (context, not a mechanistic step); every extracted path is a candidate gold reasoning chain and a ready-made gold `Query-KG` trace for warm-start training with no extra annotation. Each accepted chain becomes a QA item passed through an LLM-based rewrite/verification pass checking entailment, receives a difficulty band computed from chain features (Appendix B.5), carrying an ε-floored geometric mean of its hop confidences,

$$PC(C) = \exp\left(\frac{1}{|C|}\sum_{s\in C}\log\max(c_s,\varepsilon)\right), \qquad \varepsilon = 10^{-3} \tag{2}$$

consumed by the reward in §3.2; items per task type follow $n = \mathrm{clamp}(\mathrm{round}(17.0\sqrt{n_{\text{seed genes}}}), 30, 500)$, resolving to 246 at this release's 210 seed genes.

Every upstream source (Table 4) is a public, curated, or literature-derived database; no individual patient record or protected health information enters the graph at any stage (data-provenance details in the Ethics statement).

2.2 Dataset Characteristics

2.2.1 Dataset Dimensions

Table 6 (Appendix B.7) summarizes GenBench's overall scale, verifiable directly from the released files (full pipeline/dataset visual summary, Figure 3, Appendix B).Four further dimensions distinguish it from existing multi-hop QA resources, genomic LLM benchmarks, and genomics foundation models (Appendix B.10).

Cross-modal and genuinely multi-hop. Ten modality families are defined in the schema (Table 5), seven populated, each joined via an explicit relation-to-family map so relevance is a property of the reasoning step, not the node (§3.1); a generation-time invariant guarantees every item carries at least one real, non-empty payload. Hop counts span one to five with a real, uneven distribution: accuracy varies sharply but not monotonically with hop count, since most task types occur at a single fixed hop count, so a naive accuracy-vs-hop curve mostly re-measures task-type dominance rather than reasoning depth (Figure 7, §5.2).

Manually curated, automatically verified, and expert-audited. The 210 seed genes are manually curated (Appendix B.1), and all 5,084 items pass automated entailment, structural, and provenance checks with a repair pass (Appendix B.6). Since these checks cannot test biological validity, domain experts audited a coverage-selected sample: of 100 sampled items, 45 were reviewed, with

38 answers correct and 39 chains biologically valid, scored separately because the two can disagree (Appendix B.9). This is a quality-control sample, not exhaustive validation.

Reasoning path as supervision, confidence-annotated throughout. Every item's complete reasoning chain (§2.1.2) records source/target nodes, relation, evidence type, database, and per-hop confidence, itself a ready-made gold `Query-KG` trace, so the reasoning path is supervision, not only the final answer. Every edge also carries a source-initialized, bi-temporally tracked confidence value, and every item an aggregate path-confidence score $PC(C)$ that the training reward consumes (§3.2); sample items from each task type are given in Appendix B.11.

2.2.2 Contamination Control and Limitations

Beyond task-type stratification, the split enforces a gene-level holdout: no gene in a clean test item appears anywhere in training, which we verify empirically (Table 6). The eighteen collector APIs are live web services that drift over time, so we treat the released graph, not the collection pipeline, as the reproducible artifact; of eleven registered task types, two are dropped by a repair pass and one starved by a collector-side bug, leaving the eight released types. The closed-book eligibility rule for SFT warm-start data follows from this, with its RL-time credit-assignment effect discussed in §3.2.

3 Method: Evidence-Aware Agentic Learning

RL-trained knowledge-graph agents (Luo et al., 2025; Wang & Yu, 2026a; Sun et al., 2026) learn where to traverse, but treat all edges as equally reliable, never revise edge confidence, and learn from the final outcome alone. **CoRA**, our language-model policy, instead interacts with a confidence-aware evidence environment that evolves from its own experience, repeatedly choosing to acquire evidence or answer. Four mechanisms share a single GRPO rollout batch: ocbook (§3.2) governs whether the policy retrieves at all; GRAD (§3.3) turns rollouts into a teacher-free policy-gradient signal; ROC (§3.4) assigns that signal to evidence-bearing actions at the span level; and CoCG (§3.5) feeds ROC's credit back into the graph's edge confidence, which the next rollout then encounters.

3.1 Agent and Evidence Environment

Agent. We use Qwen3-8B as the base policy with a single LoRA adapter ($r = 16$, $\alpha = 32$, dropout $= 0.05$) for both evidence-grounded and direct-answer behavior, with no external router. Modality-projection heads, co-trained encoder MLPs, and the embedding table train in full precision as `modules_to_save` instead of through the LoRA adapter, which matters for the reference-policy computation in §3.3 (Appendix C.6). At step t, policy π_θ conditions on the question, trajectory, and evidence so far, emitting a tool action or final answer:

$$a_t \sim \pi_\theta(a \mid q, h_t), \tag{3}$$

with h_t the reasoning history so far. Interaction is constrained to five semantic actions: List-Neighbors (evidence discovery), Query-KG (path verification), Select-Encoder (modality choice), Extract-Embedding (representation retrieval), and Answer (termination).

Evidence environment. Biomedical agents either call a flat tool list (Bran et al., 2023) or return retrieved facts as text for a coordinator (Inoue et al., 2024), and multimodal assistants fold modalities into training data (Bansal et al., 2024). We instead let one policy traverse a graph and retrieve modality evidence on demand. The environment is a confidence-annotated genomic knowledge graph of genes, variants, proteins, diseases, pathways, and modality-linked evidence. List-Neighbors returns confidence-ranked neighbors; Query-KG verifies a path between two named endpoints via Dijkstra's algorithm over $1 - c_e$ edge weights, so higher-confidence evidence is preferred as lower-cost. Select-Encoder and Extract-Embedding additionally request a modality-specific payload, projected into the language-model hidden space and injected directly into the model input rather than as text (graph details in Appendix B).

3.2 Adaptive Evidence Acquisition

Search agents learn when to query and stop (Jin et al., 2025; Song et al., 2025a), but retrieval can be invoked or skipped without real dependence on its result (Ma et al., 2025), so a lucky ungrounded

guess can score like a verified answer. Unlike difficulty-aware shaping (Chen et al., 2026), ocbook needs no separate estimator. ocbook (**o**pen-/**c**losed-**book**) lets the same policy operate in open- and closed-book modes, emitting either a graph action or a direct ANSWER; the policy learns this choice without an external router. During warm-start, each item contributes an open-book trajectory from its gold reasoning chain, and single-hop items additionally contribute a closed-book trajectory; at GRPO time the retrieval decision is unconstrained, so the policy can bypass or invoke retrieval freely.

The resulting reward depends on how the trajectory actually interacts with the environment. For a rollout of T turns, with reasoning chain C_τ when verified:

$$R(\tau) = \mathbb{1}[\text{completed}] \cdot \Big[\mathbb{1}[\text{correct}] \cdot \tfrac{PC(C_\tau)}{T} \cdot (1 + w_{disc}\mathbb{1}[n_{LN} > 0]) + (1 - \mathbb{1}[\text{correct}]) \cdot \max(r_{ground}, r_{retrieved}, r_{terse})\Big], \tag{4}$$

Incorrect or incomplete trajectories score zero by default; the three partial-credit terms are strictly ordered, combine via $\max(\cdot)$ rather than summing, and each remain well below the smallest correct-trajectory reward ($PC(\emptyset) := 0$, so an ungrounded closed-book guess earns nothing through the correctness branch). An unverified-but-correct closed-book answer instead earns a smaller bonus (κ_{pe} with modality calls, κ_{uv} without), scaled by ROC's own gold-view signal so a corroborated guess earns more (full expression, Appendix C.2, Eq. 13).

3.3 TRAJECTORY-LEVEL POLICY LEARNING

Graph and search agents are trained on a single outcome reward (Luo et al., 2025; Jin et al., 2025), although reward granularity strongly affects tool learning (Qian et al., 2025); finer-grained self-distillation needs a teacher conditioned on privileged ground truth (Zhao et al., 2026a). GRAD stays teacher-free and gates reward by path confidence. Correctness is judged by exact or acceptable-answer matching. For each question, a group of trajectories $\{\tau_i\}_{i=1}^{G}$ is sampled and assigned rewards $\{R_i\}$ (Eq. 4); GRAD (**G**ated **R**eward **A**dvantage **D**istillation) converts these into group-relative advantages using the group's own trajectories as baseline:

$$A_i = \frac{R_i - \mu_g}{\sigma_g + 10^{-4}}, \tag{5}$$

with μ_g, σ_g the mean and unbiased standard deviation of rewards within the rollout group. This advantage is then modulated by ROC's action-level credit signal (§3.4, Eq. 8):

$$A_{i,t}^{\mathrm{ROC}} = A_i \left(1 + \beta_{\mathrm{ROC}} \operatorname{sign}(A_i)\, g_{i,t}\right), \tag{6}$$

so $g \equiv 0$ recovers Eq. 5 exactly. This modulated advantage enters a standard clipped importance-ratio surrogate against a reference policy (the same policy with LoRA disabled and `modules_to_save` active), together with a KL penalty for trust-region control; the full importance-ratio, KL-penalty, and aggregate-loss expressions are given in Appendix C.2 (Eqs. 14–16). On the released run, the current- and previous-policy log-probabilities are computed back-to-back with dropout zeroed, so the importance ratio is 1 and clipping has no effect; the KL penalty is the only active trust region (Appendix C.6).

3.4 EVIDENCE-AWARE CREDIT ASSIGNMENT

Dense credit for multi-turn agents usually comes from one source: a frozen reference model (Tao et al., 2026), eligibility traces (Mou et al., 2026), the outcome reward (Liu et al., 2025a; Xu et al., 2026), or a separately learned path reward model (Yan et al., 2026). A single source can be biased, e.g., replay scores partly reflect insertion cost (Yang et al., 2026). ROC uses the policy's own likelihoods across four views and trusts only their agreement. ROC (**R**escored **O**n-policy **C**redit) estimates credit at the action-span level by comparing the policy's likelihood for an action when relevant information is revealed against an otherwise matched ablated sequence:

$$q_j^{(v)} = \tanh\left(\frac{\bar{\ell}_{j,\mathrm{rev}}^{(v)} - \bar{\ell}_{j,\mathrm{abl}}^{(v)}}{2}\right), \tag{7}$$

where $\bar{\ell}_{j,\mathrm{rev}}^{(v)}$ and $\bar{\ell}_{j,\mathrm{abl}}^{(v)}$ are mean token log-probabilities under the revealed and ablated versions of the same trajectory, across four complementary views: (i) *local* (evidence just before the action that

retrieved it); (ii) *global* (only the eventual outcome); (iii) *gold* (the item's gold reasoning chain); and (iv) *prior-confidence*, a no-forward-pass signal from traversed-edge confidence (Appendix C.2, Eq. 17).

Because views can disagree, their mean is attenuated by an agreement gate:

$$\gamma_j = \mathrm{clip}\left(1 - \frac{\mathrm{Var}_v(q_j^{(v)})}{\tau_{\mathrm{ROC}}}, 0, 1\right), \qquad g_j = \gamma_j \frac{1}{V}\sum_v q_j^{(v)}, \tag{8}$$

using the biased variance across views present for a span. Each view is centered within its GRPO rollout group first, so the signal reflects relative evidence dependence rather than the graph's predominantly positive stored confidence — an earlier, uncentered version was positive by construction, fixed in Appendix C.3; a disabled view passes as neutral zero rather than being omitted. The resulting credit scatters to action tokens as $g_{i,t}$ for Eq. 6; its rollout mean g_τ feeds Eqs. 13 and 19.

3.5 Co-Evolving Graph Confidence

CoCG (**Co**-evolving **C**onfidence **G**raph) learns edge confidence from rollout experience rather than fixing it during construction, reusing the rollout batch drawn for policy optimization. Unlike methods that write trajectories into the graph (Li et al., 2026) or allow the policy to modify its structure (Lin et al., 2026), and unlike uncertainty-tracking graphs that update outside the policy-gradient step (Long et al., 2026b), CoCG updates only one scalar per existing edge, logged bi-temporally and reversible (§2.1.1). It combines two signals: the reward-contrast term Δ_e^A, which measures whether traversing an edge correlates with successful rollouts, and the ROC term Δ_e^B, which uses signed per-span credit to capture whether the edge's evidence contributed to the answer (full expressions, Appendix C.2, Eqs. 18–19). Their weighted combination updates the stored confidence:

$$\Delta_e = (1 - w_{\mathrm{ROC}})\Delta_e^A + w_{\mathrm{ROC}}\Delta_e^B, \qquad c_e^{\mathrm{new}} = \mathrm{clip}\left(c_e + \Delta_e, -2, 2\right). \tag{9}$$

The stored confidence uses the wider range $[-2, 2]$ and is mapped to $[0, 1]$ only during retrieval. This preserves headroom for upward updates, particularly because 21.9% of edges already have retrieval probability 1.0. Whether this improves correctness remains unverified (Appendices C.1, C.3).

4 Experiments

We evaluate five research questions. **RQ1:** Can existing LLMs answer GenBench from parametric knowledge alone? **RQ2:** How much do CoRA's training and graph access help? **RQ3:** What does each mechanism contribute? **RQ4:** Is ROC's credit signal informative? **RQ5:** Where does CoRA still fail? §4.2 answers RQ1 and RQ2; §5 answers RQ3 to RQ5. Each answer opens with a bold one-line summary.

4.1 Settings

Table 6 (Appendix B.7) is our test split: 574 four-way multiple-choice items and 446 open-ended generation items, so random guessing scores $(574\times0.25+446\times0)/1020 = 0.1407$ rather than 0.25. We report **Accuracy** over all test items; **F1** and **Cosine** (embedding similarity to the gold answer) over the 446 open-ended items only, since multiple-choice answers make both near-uninformative; and, in the ablation study only, **Groundedness**, the fraction of named entities in an open-ended answer that resolve to real graph entities. Baselines span zero-shot general-purpose and biomedical LLMs (Qwen3-8B, Claude Opus-5, GPT-4o, BioReason, Intern-S1-mini, Evolla) and one retrieval-augmented prompting method (SUMMER), none GenBench-trained or graph-accessed; our two trained configurations add a LoRA adapter and CoRA's agentic access (§3.1) to the same base model.

4.2 Main Results

RQ1: parametric knowledge alone is not enough. Zero-shot Qwen3-8B, GPT-4o, and Claude Opus-5 land within a narrow 0.23–0.32 band despite large differences in scale, and the biomedical models do no better (0.17–0.30).

	Tools + KG	Accuracy	F1	Cosine
Zero-shot, no tools				
Qwen3-8B (Baseline) (Team, 2025)	—	0.2304	0.0816	0.4163
Claude Opus-5 (Inc., Anthropic, 2026)	—	0.2539	0.0831	0.2842
GPT-4o (OpenAI, 2024)	—	0.3196	0.2160	0.4668
BioReason (Fallahpour et al., 2025)	—	0.1676	0.2011	0.1538
Intern-S1-mini (8B) (Zou et al., 2026)	—	0.2807	0.2271	0.4603
Evolla (10B) (Zhou et al., 2025)	—	0.2950	0.3160	0.3480
Retrieval-augmented prompting, no GenBench training				
SUMMER (Llama-3-8B) (Wu et al., 2025)	—	0.4090	**0.3480**	0.3360
Trained on GenBench (Qwen3-8B + LoRA, agentic rollout)				
ocbook SFT	✓	0.5329	0.2158	0.5025
ocbook SFT + GRPO (Full)	✓	**0.5948**	0.2755	**0.5260**

Table 1: Main results. Accuracy is over all test items (n=1020); F1 and Cosine are computed over the open-ended items only (n=446).

RQ2: training and graph access drive a large gain. Holding the base model fixed and adding only training, tool use, and graph access takes Qwen3-8B from 0.2304 to 0.5948 ($+0.3644$ absolute, 158% relative); ocbook SFT alone reaches 0.5329, and the full GRPO stage adds a further $+0.0619$ (§5.1). SUMMER (Wu et al., 2025), a retrieval-plus-reasoning prompt method with no **GenBench** training or graph access, narrows this gap (0.4090) and has the best F1, but still trails ocbook SFT alone by 12 accuracy points and the full system by 19.

Caveat on both answers. Answer length confounds the open-ended metrics, so F1 and Cosine should not be read as capability rankings across providers: Claude Opus-5 beats zero-shot Qwen3-8B on accuracy (0.2539 vs. 0.2304) but not on cosine (0.2842 vs. 0.4163), and BioReason has 0 open-ended accuracy yet non-zero F1 (0.2011).

5 Analysis and Discussion

5.1 Component Ablation

The GRPO gain in Table 1 ($+0.0619$) bundles GRAD, ROC, and CoCG, so we remove each in turn from a shared SFT+ocbook checkpoint and evaluate every configuration on the same held-out 200-item subset spanning all eight released task types.

Configuration	Accuracy	Δ vs. Full	F1 (open)	Cosine (open)	Groundedness (open)
SFT + ocbook	0.5500	-0.0800	0.2331	0.4928	0.3622
SFT (no ocbook)	0.5100	-0.1200	0.2916	0.5522	0.4699
GRPO (no ROC/GRAD/CoCG)	0.6150	-0.0150	0.2951	0.5967	0.4414
GRPO w/o ROC	0.6250	-0.0050	0.3118	**0.6169**	0.4578
GRPO w/o GRAD	0.6150	-0.0150	0.3100	0.6167	0.4267
GRPO w/o CoCG	0.6200	-0.0100	0.3051	0.6047	0.4333
Full	**0.6300**	—	**0.3207**	0.6031	**0.4980**

Table 2: Component ablation on the 200-item held-out subset (all eight released task types). Accuracy and Δ vs. Full are over all 200 items; F1, Cosine, and Groundedness are over the 83 open-ended items only. Full is the complete system (ocbook SFT followed by GRPO with GRAD, ROC, and CoCG).

RQ3: GRAD contributes most, but every effect is small. Every removal costs accuracy: removing GRAD erases the entire gain over bare GRPO (0.6150), while removing CoCG (-0.0100) or ROC (-0.0050) costs less; at n=200 each item is 0.005. Groundedness shows a similar pattern: SFT without ocbook (0.4699) outperforms all four ablated GRPO variants; only Full (0.4980) exceeds it, and removing GRAD gives the lowest value (0.4267) (single run, so suggestive only). Full (0.6031)

Configuration	$\bar{g}$ | correct	$\bar{g}$ | incorrect	Separation
All four views	-0.0187	-0.0052	-0.0135
w/o local view	-0.0226	-0.0099	-0.0127
w/o global view	-0.0240	-0.0059	-0.0181
w/o gold-path view	$+0.0095$	$+0.0051$	$+0.0044$
w/o prior-confidence view	-0.0188	-0.0050	-0.0138

Table 3: Per-view ROC ablation, 200 test items spanning all task types, GRPO step 676, forward-pass only. Method detail in Appendix C.3.

is not the highest on cosine, since three ablated variants score higher; the mechanisms trade off across metrics. Full remains best on accuracy and F1 (per-task-type breakdowns, Appendix C.4, C.5).

RQ4: ROC's credit signal is only partly informative. ROC's credit g (Eq. 8) should be higher for correct than incorrect rollouts; recomputing g on a 1,600-rollout forward-pass diagnostic with each view held out shows the full four-view g mildly mis-signed (-0.0135 separation), the gold-path view the specific source (removing it flips separation positive, $+0.0044$) and the global view well-signed (removing it worsens separation to -0.0181); Table 3, method detail in Appendix C.3.

5.2 Failure Analysis

RQ5: failures concentrate in open-ended task types. Aggregate accuracy conceals a sharp split by task type (Figure 2): multiple-choice types run 0.621–1.000, open-ended types run 0.097–0.427, with `disease_reasoning` weakest overall (Appendix C.4), a gap of at least 0.194. This is confounded with hop count, since most task types occur at one or two fixed hop counts (Figure 7); at the one hop count where both groups are populated (hop 3: 0.88 vs. 0.52, n=160/91), the gap persists. The failures concentrate in a few task types: `path_traversal` stays grounded almost regardless of configuration (0.9097–0.9877), while `counterfactual` and `disease_reasoning` produce ungrounded content under every configuration, Full included. In a traced item (`co_path_00004`, Appendix D), a malformed `Query-KG` call silently returns no path, and the rollout is correct only by luck, via `List-Neighbors`; two further items with full baseline outputs are compared qualitatively in Appendix B.12.

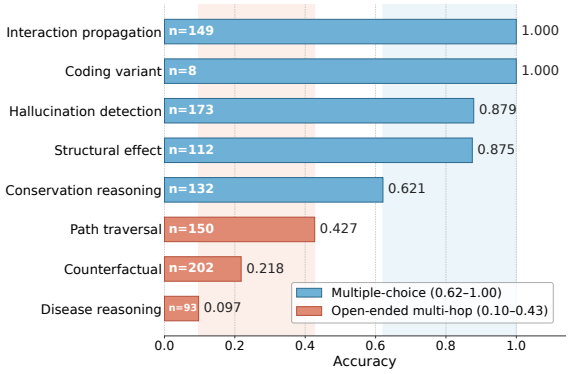


Figure 2: Accuracy by task type on the full released test set (n=1020), grouped into multiple-choice (sky blue) and open-ended multi-hop (peach red) task types. Bars are labeled with per-type n.

6 Conclusion

GenBench and **CoRA** introduce four coupled mechanisms (GRAD, ROC, CoCG, and ocbook) for jointly learning policy, evidence attribution, and graph confidence within a single GRPO rollout batch. Parametric knowledge alone is insufficient (RQ1), while training and graph access lift Qwen3-8B from 0.2304 to 0.5948 (RQ2); GRAD contributes most, though ablation effects are small (RQ3); ROC's credit is only partly informative (RQ4); and open-ended counterfactual and disease reasoning remain unsolved (RQ5). We also identified and fixed two implementation bugs (Appendices C.3, C.6) and resolved CoCG's confidence ceiling architecturally, pending empirical confirmation (§3.5). **GenBench** and **CoRA** move genomic reasoning toward systems that judge both which evidence to retrieve and how far to trust it.

AI USE STATEMENT

We used generative AI tools for disclosure-required tasks, including **implementing methods, cleaning datasets, and interpreting results**, specifically to flag discrepancies between computed figures and manuscript-reported accuracies. We did not use AI for synthetic data generation, theoretical or conceptual development, mathematical claims or proofs, hypothesis formulation, methodology or experiment design, translation, or qualitative analysis; these tasks were performed by the authors.

We also used AI for recommended-disclosure tasks, including manuscript language editing, software and analysis code, and scientific figures. All AI-assisted outputs were reviewed by the authors: code and numerical results were independently verified against source logs, interpretive claims were checked against the underlying data, text edits were individually approved, and figures were cross-checked against source data. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work uses only existing public biomedical databases and involves no participant recruitment, biospecimen collection, or new individual-level genomic data. The released **GenBench** graph contains no identifiable individual information; accordingly, no new IRB review was sought.

Population representativeness. The source databases have known ancestry and coverage biases, including European-ancestry overrepresentation. Thus, **GenBench** confidence reflects source curation and study coverage, not population-independent biological reliability, and missing or low-confidence edges may reflect under-representation.

Downstream and dual-use risk. GenBench is a reasoning benchmark, not a clinical decision-support tool. Models can produce plausible but fabricated genomic associations, so results require expert verification and should not be used for clinical decisions.

Compliance and conflicts of interest. All sources were used under their published research terms and licenses. We have no relevant conflicts of interest or sponsorships to disclose.

REPRODUCIBILITY STATEMENT

We support reproducibility by releasing the **GenBench** knowledge graph, QA benchmark, complete reasoning chains, and data-construction pipeline. Section 2 and Appendix C document data sources, collector APIs, relation types, confidence initialization, provenance, and train/test splits. Section 3 specifies the agent architecture, action grammar, GRAD, ocbook, ROC, and CoCG, with full equations in Appendix D.2; hyperparameter values are in the released configuration files. Appendix D documents the fixed implementation bugs, while Appendix E provides an end-to-end trace. Experimental details are given in Section 4. We report remaining gaps, including single-run ablations at n=200 and unverified CoCG signal quality. Code, data, and training/evaluation scripts are available at `https://anonymous.4open.science/r/genbench-agent-B3F0/`.

We also release the following supplementary material: the GenBench Sample knowledge graph and QA splits, per-item evaluation logs for CoRA and all seven baselines, and per-item logs for the mechanism-ablation and SFT open/closed-book study. All files are anonymized for double-blind review; see the accompanying README.md for a full description of each file.

REFERENCES

Žiga Avsec, Natasha Latysheva, Jun Cheng, et al. AlphaGenome: Advancing regulatory variant effect prediction with a unified DNA sequence model. *Nature*, 2026.

Hritik Bansal, Daniel Israel, Siyan Zhao, Shufan Li, Tung Nguyen, and Aditya Grover. MedMax: Mixed-modal instruction tuning for training biomedical assistants, 2024.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. ChemCrow: Augmenting large-language models with chemistry tools, 2023.

Payal Chandak, Kexin Huang, and Marinka Zitnik. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10:67, 2023.

Liuji Chen, Dianxing Tang, Xing Shi, Dingshuo Chen, Qiang Liu, Shu Wu, and Liang Wang. Learning when not to act: Mitigating tool abuse in agentic reinforcement learning (EAPO), 2026.

Zihao Cheng, Zeming Liu, Yingyu Shan, Xinyi Wang, Xiangrong Zhu, Yunpu Ma, Hongru Wang, Yuhang Guo, Wei Lin, and Yunhong Wang. Mem^{2}evolve: Towards self-evolving agents via co-evolutionary capability expansion and experience distillation, 2026.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scGPT: Toward building a foundation model for single-cell multi-omics using generative AI. *Nature Methods*, 21:1470–1480, 2024.

Phuong Tuan Dat, Qi Li, and Xinchao Wang. dOPSD: On-policy self-distillation for diffusion language models, 2026.

Vaggelis Dorovatas, Nancy Kalaj, and Rahaf Aljundi. Continual self-improvement with lightweight experiential latent memories. *arXiv preprint arXiv:2606.17803*, 2026.

Shusaku Egami, Aoi Ohta, Tomoki Tsujimura, Masaki Asada, Tatsuya Ishigaki, Ken Fukuda, Masahiro Hamasaki, and Hiroya Takamura. HOME-KGQA: A benchmark dataset for multimodal knowledge graph question answering on household daily activities, 2026. Accepted at LREC 2026.

Adibvafa Fallahpour, Andrew Magnuson, Purav Gupta, Shihao Ma, Jack Naimer, Arnav Shah, Haonan Duan, Omar Ibrahim, Hani Goodarzi, Chris J. Maddison, and Bo Wang. Bioreason: Incentivizing multimodal biological reasoning within a dna-llm model. *ArXiv*, abs/2505.23579, 2025.

Xi Fu, Shentong Mo, Alejandro Buendia, et al. A foundation model of transcription across human cell types. *Nature*, 637:965–973, 2025.

Thomas Hayes, Roshan Rao, Halil Akin, et al. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, 2020.

Inc., Anthropic. Claude opus 5. `https://platform.claude.com/docs/en/models/opus-5/overview`, 2026. URL `https://platform.claude.com/docs/en/models/opus-5/overview`. Accessed: 2026-09-26.

Yoshitaka Inoue, Tianci Song, Xinling Wang, Augustin Luna, and Tianfan Fu. DrugAgent: Reliable multi-agent integration of conflicting biomedical evidence for drug–target interaction assessment, 2024. Title evolved across preprint versions; PMC version (PMCID: PMC12036430) titled "DruGagent: Multi-Agent Large Language Model-Based Reasoning for Drug-Target Interaction Prediction"; current arXiv version titled as below. Confirm which version/title to cite.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning, 2025.

Chia Hsiang Kao, Evan Trop, McKinley Polen, Yair Schiff, Bernardo Almeida, Aaron Gokaslan, Thomas Pierrot, and Volodymyr Kuleshov. Advancing DNA language models: The genomics long-range benchmark. In *ICLR 2024 Workshop on Machine Learning for Genomics Explorations (MLGenX)*, 2024.

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

Daeyong Kwon, Soyoung Yoon, and Seung-won Hwang. SAFE: An LLM-as-verifier framework for evidence-grounded multi-hop reasoning, 2026.

Zhaoyang Li, Zenghuang Fu, Qiuyuan Ai, Ping Jiang, Haoyu Wu, Minghui Wu, Chenxu Zhao, Jie Song, and Guannan He. CoEvoKG: Co-evolving knowledge graphs with self-evolving search agents, 2026.

Xuechen Liang, Meiling Tao, Yinghui Xia, Jianhui Wang, Kun Li, Yijin Wang, Yangfan He, Jingsong Yang, Tianyu Shi, Yuantao Wang, Miao Zhang, and Xueqian Wang. SAGE: Self-evolving agents with reflective and memory-augmented abilities. *Neurocomputing*, 647:130470, 2025. doi: 10.1016/j.neucom.2025.130470.

Jiashi Lin, Changhong Jiang, Xiangru Lin, Ruifei Zhang, Xinyi Zhu, Jiyao Liu, Cheng Tang, Ye Du, Shujian Gao, Junzhi Ning, Lihao Liu, Ziyan Huang, Tianbin Li, Jin Ye, and Junjun He. EvoGraph-R1: Self-evolving multimodal knowledge hypergraphs for agentic retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 756–765, 2026.

Xinna Lin, Siqi Ma, Junjie Shan, Xiaojing Zhang, Shell Xu Hu, Tiannan Guo, Stan Z. Li, and Kaicheng Yu. BioKGBench: A knowledge graph checking benchmark of AI agent for biomedical science. *arXiv preprint arXiv:2407.00466*, 2024.

Johannes Linder, Divyanshi Srivastava, Han Yuan, Vikram Agarwal, and David R. Kelley. Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation. *Nature Genetics*, 2025.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. SimpleMem: Efficient lifelong memory for LLM agents. In *ICLR 2026 Workshop on Recursive Self-Improvement (RSI)*, 2026. Also available as arXiv preprint arXiv:2601.02553.

Sizhe Liu, Yizhou Lu, Siyu Chen, Xiyang Hu, Jieyu Zhao, Tianfan Fu, and Yue Zhao. DrugAgent: Automating AI-aided drug discovery programming through LLM multi-agent collaboration, 2024.

Xiaoqian Liu, Ke Wang, Yuchuan Wu, Fei Huang, Yongbin Li, Junge Zhang, and Jianbin Jiao. Agentic reinforcement learning with implicit step rewards (iStar), 2025a. Accepted, The Fourteenth International Conference on Learning Representations (ICLR 2026).

Yixin Liu, Guibin Zhang, Kun Wang, Shiyuan Li, and Shirui Pan. Graph-augmented LLM agents (GLA): A survey, 2025b.

Weicai Long, Yusen Hou, Junning Feng, Houcheng Su, Shuo Yang, Donglin Xie, and Yanlin Zhang. GenomeQA: Benchmarking general large language models for genome sequence understanding, 2026a.

Yunbo Long, Haolang Zhao, Ge Zheng, and Alexandra Brintrup. Helicase: Uncertainty-guided supply chain knowledge graph construction with autonomous multi-agent LLMs, 2026b.

Yuxing Lu, Wei Wu, Xukai Zhao, Rui Peng, and Jinzhuo Wang. KARMA: Leveraging multi-agent LLMs for automated knowledge graph enrichment, 2025. Spotlight paper, NeurIPS 2025. Author list expanded between v1 (Feb 2025, 2 authors) and current version (v2, Jan 2026, 5 authors); listed here per current version.

Haoran Luo, Haihong E, Guanting Chen, Qika Lin, Yikai Guo, Fangzhi Xu, Zemin Kuang, Meina Song, Xiaobao Wu, Yifan Zhu, and Luu Anh Tuan. Graph-R1: Towards agentic GraphRAG framework via end-to-end reinforcement learning, 2025. Accepted, ICML 2026 (confirmed via official GitHub repository).

Shengjie Ma, Chenlong Deng, Jiaxin Mao, Jiadeng Huang, Teng Wang, Junjie Wu, Changwang Zhang, and Jun Wang. Proof-of-use: Mitigating tool-call hacking in deep research agents, 2025.

Costas Mavromatis, Soji Adeshina, Vassilis N. Ioannidis, Zhen Han, Qi Zhu, Ian Robinson, Bryan Thompson, Huzefa Rangwala, and George Karypis. BYOKG-RAG: Multi-strategy graph retrieval for knowledge graph question answering, 2025.

Chaoli Mou, Zhan Zhuang, Xinning Chen, and Yu Zhang. Beyond uniform credit assignment: Selective eligibility traces for RLVR, 2026.

Md Mahadi Hasan Nahid and Davood Rafiei. PRISM: Agentic retrieval with LLMs for multi-hop question answering, 2025.

Duy Nguyen, Hanqi Xiao, Archiki Prasad, Zaid Khan, Anirban Das, Austin Zhang, Sambit Sahu, Hyunji Lee, Elias Stengel-Eskin, and Mohit Bansal. AVSD: Adaptive-view self-distillation by balancing consensus and teacher-specific privileged signals, 2026. Consensus/gated-residual design proposed as a candidate replacement for ROC's Eq. 18 aggregation, to be run as an ablation; see Appendix C.

OpenAI. Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`, 2024. Accessed: 2026-09-26.

Anish Pahilajani, Devasha Trivedi, Jincen Shuai, Khin S. Yone, Samyak Rajesh Jain, Namyong Park, Ryan A. Rossi, Nesreen K. Ahmed, Franck Dernoncourt, and Yu Wang. GRS-QA: Graph reasoning-structured question answering dataset, 2024.

Jinyoung Park, Sanghyeok Lee, Omar Zia Khan, Hyunwoo J. Kim, and Joo-Kyung Kim. ProGraph-R1: Progress-aware reinforcement learning for graph retrieval augmented generation. *arXiv preprint arXiv:2601.17755*, 2026.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. ToolRL: Reward is all tool learning needs, 2025.

Yuanhao Qu, Yingzhou Lu, Xinming Tu, Serena Zhang, Tianwei She, Alexander Glenn Shaw, Jou-Ho Shih, Bingqing Zhao, Minjie Shen, Haochen Yang, Jielin Yan, Rongchuan Zhang, Xinze Wu, Tingting Li, Le Cong, Xiaobo Hu, Yuan Jiang, Jiayun Dong, Tao Peng, Jure Leskovec, and Kexin Huang. BiomniBench: Process-level evaluation of LLM agents for real-world biomedical research. *bioRxiv*, 2026. doi: 10.64898/2026.05.12.724604.

Owen Queen et al. CGBench: Benchmarking language model scientific reasoning for clinical genetics research. In *Advances in Neural Information Processing Systems*, 2025.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025. Industry whitepaper, Zep AI.

Xinyi Shang, Xu Liao, Zhicheng Ji, and Wenpin Hou. Benchmarking large language models for genomic knowledge with GeneTuring. *Briefings in Bioinformatics*, 26, 2025.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-Searcher: Incentivizing the search capability in LLMs via reinforcement learning, 2025a.

Yanlin Song, Ben Liu, Víctor Gutiérrez-Basulto, Zhiwei Hu, Qianqian Xie, Min Peng, Sophia Ananiadou, and Jeff Z. Pan. Plan then retrieve: Reinforcement learning-guided complex reasoning over knowledge graphs. *arXiv preprint arXiv:2510.20691*, 2025b.

Jia Ao Sun, Hao Yu, Fengran Mo, Zhan Su, Yuchen Hui, Bang Liu, and Jian-Yun Nie. Search-on-graph-R1: Training large language models to search knowledge graphs with reinforcement learning, 2026.

Zhiquan Tan and Yinrong Hong. PAINT: Partial-solution adaptive interpolated training for self-distilled reasoners, 2026.

Leitian Tao, Baolin Peng, Wenlin Yao, Tao Ge, Hao Cheng, Mike Hang Wang, Jianfeng Gao, and Sharon Li. TRACE: Turn-level reward assignment via credit estimation for long-horizon agents, 2026.

Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Christina V. Theodoris, Ling Xiao, Anant Chopra, et al. Transfer learning enables predictions in network biology. *Nature*, 618:616–624, 2023.

Ruilin Tong and Dong Gong. MILES: Modular instruction memory with learnable selection for self-improving llm reasoning. *arXiv preprint arXiv:2607.06974*, 2026.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10, 2022.

Changhao Wang, Yanfang Liu, Xinxin Fan, Anzhi Zhou, Lao Tian, and Yunfeng Lu. DTKG: Dual-track knowledge graph-verified reasoning framework for multi-hop qa. *arXiv preprint arXiv:2510.16302*, 2025.

Shuai Wang and Yinan Yu. KG-Hopper: Empowering compact open LLMs with knowledge graph reasoning via reinforcement learning, 2026a.

Shuai Wang and Yinan Yu. KG-Reasoner: A reinforced model for end-to-end multi-hop knowledge graph reasoning, 2026b.

Xiao Wang et al. HiCFoundation: A generalizable foundation model for chromatin architecture. *Nature Methods*, 2026.

Wuzhenghong Wen, Chao Xue, Su Pan, Yuwei Sun, and Minlong Peng. Reinforcement learning enhanced multi-hop reasoning for temporal knowledge question answering. *arXiv preprint arXiv:2601.01195*, 2026.

Jinyang Wu, Shuo Yang, Zhengxi Lu, Fan Zhang, Yuhao Shen, Lang Feng, Haoran Luo, Zheng Lian, Shuai Zhang, Zhengqi Wen, and Jianhua Tao. SEED: Self-evolving on-policy distillation for agentic reinforcement learning, 2026. Motivates a proposed additive fifth ROC view for closed-book (ocbook-routed) rollouts; see Appendix C.

Menghua Wu, Russell Littman, Jacob Levine, Lin Qiu, Tommaso Biancalani, David Richmond, and Jan-Christian Huetter. Contextualizing biological perturbation experiments through language. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yuanda Xu, Zhengze Zhou, Hejian Sang, Xiaomin Li, Jiaxin Zhang, Xinchen Du, Sen Na, Zhipeng Wang, and Alborz Geramifard. Triage: Role-typed credit assignment for agentic reinforcement learning. *arXiv preprint arXiv:2606.32017*, 2026.

Shiqi Yan, Yubo Chen, Ruiqi Zhou, Zhengxi Yao, Shuai Chen, Tianyi Zhang, Shijie Zhang, WeiQiang Zhang, Yongfeng Huang, Haixin Duan, and Yunqi Zhang. Explore-on-graph: Incentivizing autonomous exploration of large language models on knowledge graphs with path-refined reward modeling. *arXiv preprint arXiv:2602.21728*, 2026.

Xue Yan, Zijing Ou, Mengyue Yang, Yan Song, Haifeng Zhang, Yingzhen Li, and Jun Wang. Memory-driven self-improvement for decision making with large language models. *arXiv preprint arXiv:2509.26340*, 2025.

Yi Yang, Cong Qin, Xiaodan Liu, Chishui Chen, Qing Dong, Yan Zhang, Cao Liu, Zhao Yang, Lu Pan, Jiaye Lin, and Yi Feng. Agentic reinforcement learning with observation-calibrated self-distillation, 2026. The URL originally given for this title (2605.20643) resolves to a different paper (AVSD, above); this is the correct arXiv ID, found independently. Identifies a confound directly relevant to ROC's gold-view ablation, treated as a fix rather than a citation; see Appendix C.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Ming Yin, Yuanhao Qu, Dyllan Liu, Ling Yang, Le Cong, and Mengdi Wang. Genome-bench: A scientific reasoning benchmark from real-world expert discussions. *bioRxiv*, 2025.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. MemSkill: Learning and evolving memory skills for self-evolving agents. *arXiv preprint arXiv:2602.02474*, 2026.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026a.

Ziqi Zhao, Xinyu Ma, Liu Yang, Yujie Feng, Daiting Shi, Jingzhou He, Xin Xin, Zhaochun Ren, and Xiao-Ming Wu. ROSD: Reflective on-policy self-distillation for language model reasoning across domains, 2026b.

Xibin Zhou, Chenchen Han, Yingqi Zhang, Huan Du, Jiayuan Tian, Jin Su, Renju Liu, Kai Zhuang, Shiyu Jiang, Anthony Gitter, et al. Decoding the molecular language of proteins with evolla. *bioRxiv*, pp. 2025–01, 2025.

Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana V. Davuluri, and Han Liu. DNABERT-2: Efficient foundation model and benchmark for multi-species genomes. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Yicheng Zou, Dongsheng Zhu, Lin Zhu, Tong Zhu, Yunhua Zhou, Peiheng Zhou, Xinyu Zhou, Dongzhan Zhou, Zhiwang Zhou, Yuhao Zhou, Bowen Zhou, Zhanping Zhong, Zhijie Zhong, Haiteng Zhao, Penghao Zhao, Xiaomeng Zhao, Zhiyuan Zhao, Yechen Zhang, Jin Zhang, Wenwei Zhang, Hongjie Zhang, Zhuo Zhang, Wenlong Zhang, Bo Zhang, Chao Zhang, Chen Zhang, Yuhang Zang, Fei Yuan, Jiakang Yuan, Jiashuo Yu, Jinhui Yin, Haochen Ye, Qian Yao, Bowen Yang, Danni Yang, Kaichen Yang, Ziang Yan, Jun Xu, Yicheng Xu, Wanghan Xu, Xuenan Xu, Chao Xu, Ruiliang Xu, Shuhao Xing, Long Xing, Xinchen Xie, Ling-I Wu, Zijian Wu, Zhenyu Wu, Lijun Wu, Yue Wu, Jianyu Wu, Wen Wu, Fan Wu, Xilin Wei, Qi Wei, Bingli Wang, Rui Wang, Ziyi Wang, Zun Wang, Yi Wang, Haomin Wang, Yizhou Wang, Lintao Wang, Yiheng Wang, Longjiang Wang, Bin Wang, Jian Tong, Zhongbo Tian, Huanze Tang, Chen Tang, Shixiang Tang, Yu Sun, Qiushi Sun, Xuerui Su, Qisheng Su, Chenlin Su, Demin Song, Jin Shi, Fukai Shang, Yuchen Ren, Pengli Ren, Xiaoye Qu, Yuan Qu, Jiantao Qiu, Yu Qiao, Runyu Peng, Tianshuo Peng, Jiahui Peng, Qizhi Pei, Zhuoshi Pan, Linke Ouyang, Wenchang Ning, Yichuan Ma, Zerun Ma, Ningsheng Ma, Runyuan Ma, Chengqi Lyu, Haijun Lv, Han Lv, Lindong Lu, Kuikun Liu, Jiangning Liu, Yuhong Liu, Kai Liu, Hongwei Liu, Zhoumianze Liu, Mengjie Liu, Ziyu Liu, Wenran Liu, Yang Liu, Liwei Liu, Kaiwen Liu, Junyao Lin, Junming Lin, Tianyang Lin, Dahua Lin, Jianze Liang, Linyang Li, Peiji Li, Zonglin Li, Zehao Li, Pengze Li, Guoyan Li, Lingkai Kong, Linglin Jing, Zhenjiang Jin, Feifei Jiang, Qian Jiang, Junhao Huang, Zixian Huang, Haian Huang, Zhouqi Hua, Han Hu, Linfeng Hou, Yinan He, Conghui He, Tianyao He, Xu Guo, Qipeng Guo, Aijia Guo, Yuzhe Gu, Lixin Gu, Jingyang Gong, Qiming Ge, Jiaye Ge, Songyang Gao, Jianfei Gao, Xinyu Fang, Caihua fan, Yue Fan, Yanhui Duan, Zichen Ding, Shengyuan Ding, Xuanlang Dai, Erfei Cui, Ganqu Cui, Pei Chu, Tao Chu, Guangran Cheng, Yu Cheng, Kai Chen, Yongkang Chen, Chiyu Chen, Guanzhou Chen, Qiaosheng Chen, Sitao Chen, Xin Chen, Haojiong Chen, Yicheng Chen, Weihan Cao, Yuhang Cao, Qinglong Cao, and Lei Bai. Intern-s1-pro: Scientific multimodal foundation model at trillion scale, 2026. URL https://arxiv.org/abs/2603.25040.

APPENDIX CONTENTS

A Related Work **18**

A.1 Cross-modal, multi-hop knowledge graph question answering 18

A.2 Genomic and biomedical LLM benchmarks 18

A.3 Credit assignment in agentic reinforcement learning 18

A.4 Co-evolving knowledge and policy 19

A.5 Reinforcement-learning agents for knowledge-graph traversal and verification . . . 19

A.6 Automated knowledge graph construction 20

A.7 On-policy self-training without an external teacher 20

A.8 Adaptive tool use and retrieval bypass 20

A.9 Broader agentic tool use for scientific domains 21

A.10 Self-evolving agent memory and skill accumulation 21

B Additional Details on GenBench **21**

B.1 Seed Gene Curation 21

B.2 Graph Expansion: Eighteen Collector APIs 23

B.3 Modality Families and the Relation-to-Family Map 24

B.4 Reasoning-chain and confidence provenance fields 25

B.5 Difficulty scoring 26

B.6 Task-type accounting 27

B.7 Split construction 27

B.8 Dataset Reproducibility 28

B.9 Expert Validation of a Held-Out Item Sample 28

B.9.1 Validation Objective 28

B.9.2 Validation Dimensions 28

B.9.3 Validation Procedure 28

B.9.4 Expert Validation Results 28

B.9.5 Representative Validation Cases 29

B.9.6 Implications for GenBench 29

B.10 Comparison to related genomic and multi-hop reasoning benchmarks 30

B.11 Sample GenBench items 33

B.12 Qualitative Case Studies: CoRA versus Zero-Shot Baselines 33

C Implementation Details, Diagnosed Failure Modes, and Concurrent Methods **33**

C.1 Architecture and Environment Implementation Details 34

C.2 Full Training-Objective and Credit-Assignment Equations 35

C.3 A diagnosed and fixed failure mode: ROC's views were positive by construction . . 36

C.4 Per-task-type breakdown of the component ablation 38

C.5 Per-task-type groundedness for the open-ended split 39
C.6 A diagnosed and fixed bug: the KL reference reverted trained modality projections to random initialization 40
C.7 OCSD and ROC's gold view 41
C.8 AVSD and ROC's aggregation rule 41
C.9 SEED and closed-book credit 42
C.10 Summary 42

D A Worked Trace of the Full System on One Item **43**

A Related Work

This appendix expands the related-work summary in Section 1.

A.1 Cross-modal, multi-hop knowledge graph question answering

Multi-hop QA has largely been studied over encyclopedic text: HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022) all require chaining evidence across disjoint Wikipedia passages, and recent agentic retrieval methods (BYOKG-RAG (Mavromatis et al., 2025); PRISM (Nahid & Rafiei, 2025) improve accuracy on this family of benchmarks by having an LLM generate graph artifacts (entities, paths, queries) that anchor iterative retrieval. Multimodal KGQA is comparatively rare: HOME-KGQA (Egami et al., 2026) is, to our knowledge, the closest existing benchmark in spirit, pairing a multimodal knowledge graph of household activities with multi-hop natural-language questions and graph-query-language targets, and reporting that LLM-based KGQA methods degrade sharply relative to their performance on text-only benchmarks. **GenBench** differs from this lineage in domain, in graph scale (90k nodes and 236k edges versus household-scale graphs), and in a design choice these benchmarks do not share: modality access in **GenBench** is conditioned on the *reasoning step*, through an explicit relation-to-family map, rather than being a fixed property of an entity that the model always has available once it locates that entity. A separate line trains multimodal biomedical assistants via instruction tuning: MedMax (Bansal et al., 2024) mixes modalities into a single fine-tuned model's training data instead of retrieving a modality payload on demand during a graph traversal. This is a fine-tuning-time versus inference-time distinction that matters for updating evidence, since GenBench's agent-retrieved payloads stay tied to the confidence-annotated edge they came from, while a fine-tuned assistant's multimodal knowledge is baked into weights with no per-fact confidence to revise.

A.2 Genomic and biomedical LLM benchmarks

A separate line of work benchmarks LLMs on genomic and biomedical knowledge directly: GeneTuring (Shang et al., 2025) probes genomic knowledge with short-form questions, GenomeQA (Long et al., 2026a) evaluates general-purpose LLMs on genome sequence understanding, and Genome-Bench (Yin et al., 2025) draws a scientific reasoning benchmark from real-world expert discussion threads. Sequence-only foundation-model benchmarks such as the Genomics Long-Range Benchmark (Kao et al., 2024) and DNABERT-2 (Zhou et al., 2024) evaluate a different question, namely representation quality on raw sequence, not reasoning over curated biomedical relationships. None of these resources pairs a large, heterogeneous relational graph with agent-retrievable non-text modalities or with per-item, per-hop confidence; they evaluate either parametric recall (can the model state the fact) or single-modality sequence understanding (can the model represent the sequence), not multi-hop reasoning that requires retrieving and combining evidence of different kinds along a graph path. In the general (non-biomedical) domain, GRS-QA (Pahilajani et al., 2024) provides a graph-reasoning-structured QA dataset with explicit reasoning-graph annotations, which is the closest general-domain analogue to GenBench's chain-level supervision; it differs in domain, in the absence of per-edge confidence, and in having no retrievable non-text modality attached to its graph. This is the gap **GenBench** is built to occupy.

A.3 Credit assignment in agentic reinforcement learning

Assigning credit within a multi-turn tool-use trajectory from a single terminal reward is an active problem independent of domain. TRIAGE (Xu et al., 2026) assigns role-typed credit to the environment-facing actions inside a trajectory when supervision arrives only as a final verifier outcome; TRACE (Tao et al., 2026) derives dense, turn-level reward from a frozen reference model via temporal-difference increments applied at tool-call boundaries; S-trace (Mou et al., 2026) replaces GRPO's implicit assumption of uniform, trajectory-wide credit with a critic-free eligibility-trace mechanism that concentrates credit on high-entropy tokens; iStar (Liu et al., 2025a) sources implicit step rewards without additional rollouts by exploiting structure already present in (un)verifiable outcome rewards. Each of these methods derives its dense signal from a single source: a reference model, an eligibility trace over tokens, or the outcome reward itself. ROC differs by drawing on *four independent views* (local narration, global outcome, gold chain, and prior edge confidence)

and discounting a span's credit by how much those views disagree with one another (Eq. 8); to our knowledge, agreement-gating across independently-constructed credit signals, rather than a single signal source refined in isolation, has not been used elsewhere in this literature.

A.4 Co-evolving knowledge and policy

Closest in spirit to CoCG is a small but growing set of methods that let an agent's knowledge source evolve alongside its policy during RL training. CoEvoKG (Li et al., 2026) turns a knowledge graph into both a source of verifiable training tasks and a persistent evidence memory, writing verified, path-supported solver trajectories back into the graph as new evidence after each update; EvoGraph-R1 (Lin et al., 2026) implements a similar closed loop through an explicit `Graph-Edit` action, so the policy can add or revise graph content directly. Helicase (Long et al., 2026b) pursues a related goal from the opposite direction, tracking calibrated uncertainty at the action, trajectory, and memory layers of an autonomously-constructed supply-chain knowledge graph, though without folding that uncertainty update into a shared policy-gradient step. On the memory side rather than the training side, Zep's (Rasmussen et al., 2025) Graphiti engine maintains a temporal knowledge graph for agent memory with explicit event-time and ingestion-time tracking, which is architecturally close to GenBench's bi-temporal provenance record (Section 2.1.1); the difference is purpose rather than mechanism. Zep's temporal graph records what an agent has learned across a deployment for later recall, while GenBench's bi-temporal record exists so that a training-induced confidence trajectory can be audited and, if needed, rolled back. CoCG is narrower than the graph-editing approaches: it never adds, removes, or rewrites a fact, only reweights the *confidence* already attached to an existing edge, using the same rollout batch and the same GRPO step that updates the policy, with no separate write-back phase. We restrict CoCG this way because reweighting confidence is reversible and auditable through the bi-temporal provenance record (Section 2.1.1), whereas structural graph edits made by a policy mid-training are harder to walk back if a later check finds them to be reward-hacking artifacts.

A.5 Reinforcement-learning agents for knowledge-graph traversal and verification

A cluster of concurrent work trains LLM policies with RL directly against knowledge graphs, and overlaps most closely with our training mechanisms collectively. Graph-R1 (Luo et al., 2025) frames agentic GraphRAG as an end-to-end RL problem, training a policy's retrieval actions under outcome reward. Explore-on-Graph (EoG; Yan et al., 2026) incentivizes autonomous KG exploration with a path-refined reward model that scores partial traversals rather than only the final answer, making it the closest analogue to ROC's per-span credit. However, EoG's reward model is a separately learned component, whereas ROC derives credit directly from the policy's own log-probabilities under revealed-versus-ablated views, with no auxiliary model to train or maintain. KG-Hopper (Wang & Yu, 2026a) and Search-on-Graph-R1 (Sun et al., 2026) both train compact open LLMs to search knowledge graphs via RL, the latter with an explicit query-then-traverse action space resembling our `Query-KG`/`List-Neighbors` split (Section 3.1); neither attaches confidence to graph edges or updates it during training, which is CoCG's contribution. MRE (T-GRPO; Wen et al., 2026) applies GRPO to multi-hop temporal knowledge graph QA and is the closest existing work to our bi-temporal design (Section 2.1.1), but T-GRPO treats temporal validity as a property of the query to reason about, not as a confidence value the policy itself revises during training. SAFE (Kwon et al., 2026) uses an LLM-as-verifier to give stepwise, atomic feedback for error correction in multi-hop reasoning, which is complementary to but distinct from ROC: SAFE's feedback comes from a separate judge model called after generation, while ROC's four-view agreement gate is computed from the policy's own tokens with no external verifier call. DTKG's (Wang et al., 2025) dual-track verification and KG-Reasoner's (Wang & Yu, 2026b) end-to-end reinforced multi-hop reasoning pursue related goals through a single verification or reward track, rather than ROC's agreement gate across four independent views. Plan Then Retrieve (Song et al., 2025b) and ProGraph-R1 (Park et al., 2026) both study when-to-retrieve and progress-aware reward shaping for graph RAG, addressing a version of the question ocbook answers, but without ocbook's gold-corroboration-gated discount tying the bypass decision back to ROC's credit signal. We take this cluster as a whole to be the closest prior art to our four training mechanisms; to our knowledge, no existing system combines per-edge confidence co-evolution, multi-view agreement-gated credit, and a policy-controlled retrieval bypass gated by that same credit signal inside a single shared GRPO step.

A.6 AUTOMATED KNOWLEDGE GRAPH CONSTRUCTION

A separate line of work grows a knowledge graph automatically from unstructured text using multi-agent LLM pipelines. KARMA (Lu et al., 2025) uses nine collaborative agents (entity discovery, relation extraction, schema alignment, conflict resolution) to enrich a KG directly from scientific literature, reporting tens of thousands of new LLM-verified entities extracted from PubMed abstracts. This is a construction-time approach: agents propose graph content before any downstream QA or RL training begins. GenBench's construction pipeline is comparable in spirit (18 collectors expanding a curated seed, Appendix B.2) but keeps a stronger separation between graph construction and policy training: collectors run once to produce a static, versioned snapshot, and no training-time agent adds or edits graph structure; only edge confidence changes during training (Section 3.5), and only within edges the graph already contains. We keep this boundary because entity- and relation-level enrichment during RL training raises the same reward-hacking and rollback concerns noted for structure-editing co-evolution methods in Appendix A.4.

A.7 ON-POLICY SELF-TRAINING WITHOUT AN EXTERNAL TEACHER

GRAD (Section 3.3) belongs to a growing set of on-policy methods that need no separate teacher. On-Policy Self-Distillation (OPSD; Zhao et al., 2026a) lets one model act as both teacher and student, conditioning the teacher on privileged, instance-specific ground truth and training the student to match the teacher's per-token distribution over its own rollouts; follow-up work extends this idea to diffusion language models (dOPSD; Dat et al., 2026), partial-solution interpolation (PAINT; Tan & Hong, 2026), and reflective cross-domain transfer (ROSD; Zhao et al., 2026b). GRAD shares the teacher-free setting but is a policy-gradient method with path-confidence reward gating and group-relative normalization (Eqs. 4, 5), and its gold-chain rescoring enters ROC (Section 3.4) rather than the loss itself.

Two further pieces of concurrent work sharpen this comparison. SEED (Self-Evolving On-Policy Distillation; Wu et al., 2026) converts completed on-policy trajectories into hindsight skills and distills their behavioral effect back into the policy, jointly optimized with outcome-based RL. It is closer to GRAD in spirit than OPSD is, since both let the policy's own completed trajectories generate the training signal rather than a privileged-context teacher view, though SEED distills natural-language skill descriptions rather than rescoring reward under a GRPO objective. More directly relevant as a threat to validity for our gold-path view is a concurrent critique of OPSD itself: Observation-Calibrated Self-Distillation (OCSD; Yang et al., 2026) identifies a confound in privileged-replay self-distillation, where the support a teacher view provides may reflect not the privileged information itself but score shifts induced by the replay scaffold that presents it, making it difficult to attribute credit specifically to the privileged content. Since the gold-path view feeds into ROC (Section 3.4), this confound may explain gold-view credit that ROC's other three views do not corroborate; Table 3 is consistent with this, and a matched ablation is left to future work. AVSD (Adaptive-View Self-Distillation; Nguyen et al., 2026) addresses a related problem from the multi-view side: it separates a consensus target (fires only when views agree in direction) from an arithmetic-marginal residual (fires when even one view strongly favors a token), admitting the residual only under a two-component alignment-and-magnitude gate. This is the closest precedent to ROC's four-view agreement gate (Eq. 8), though AVSD operates on distillation targets over token distributions, while ROC operates on a credit/advantage signal derived from log-probability shifts under revealed-versus-ablated conditioning. Appendix C compares all three with ROC at the equation level.

A.8 ADAPTIVE TOOL USE AND RETRIEVAL BYPASS

Deciding *whether* to retrieve at all, not just how, is well studied in text-only RAG and search-agent RL: Search-R1 (Jin et al., 2025) and R1-Searcher (Song et al., 2025a) train policies to decide what to query and when to stop within a search loop, and ToolRL (Qian et al., 2025) studies reward granularity for tool-use learning directly. Closer to our concern with reward hacking on the bypass decision, EAPO (Chen et al., 2026) applies difficulty-aware reward shaping specifically to discourage tool use when tool-free reasoning already suffices, and Proof-of-Use (Ma et al., 2025) is motivated by the observation that retrieval tools are comparatively easy for a policy to invoke or skip without genuine informational dependence, unlike execution-style tools whose contribution is directly observable.

ocbook (Section 3.2) addresses the same failure mode, in which a lucky ungrounded guess scores the same as a verified answer. It does so with a discount on the two ungrounded reward tiers that is itself gated by the rollout's own gold-corroboration signal from ROC (Eq. 13), rather than by a separately-trained difficulty estimator or an auxiliary usefulness verdict.

A.9 Broader agentic tool use for scientific domains

GenBench's agent sits within a wider trend of giving LLMs domain tools rather than domain text. ChemCrow (Bran et al., 2023) equips an LLM with expert chemistry tools in a flat ReAct-style loop and demonstrated real, physically executed syntheses planned without a human chemist in the loop; DrugAgent (Liu et al., 2024) uses a two-stage planner/instructor multi-agent architecture for drug-discovery programming tasks; BiomniBench (Qu et al., 2026) evaluates LLM agents at each stage of a biomedical research process rather than on a single end task. Closer to GenBench's own domain, the drug–target DrugAgent of Inoue et al. (2024) applies a coordinator-based multi-agent architecture to drug-target interaction assessment, with a dedicated knowledge-graph agent retrieving from curated biomedical databases (DrugBank, DGIdb, STITCH, CTD) alongside machine-learning-prediction and literature-search agents; this is the closest work we found to combining a knowledge graph with an agentic biomedical reasoning system, but the KG agent there returns retrieved facts for a coordinator to synthesize across agents, rather than being traversed step-by-step with per-edge confidence by a single policy under RL, as GenBench's agent does. None of these targets multi-hop reasoning over a confidence-annotated graph: ChemCrow's tools are a flat, ungraphed list, and DrugAgent's and BiomniBench's evaluation units are pipeline stages rather than graph traversal steps. We note the broader survey of Graph-Augmented LLM Agents (Liu et al., 2025b), which categorizes graph use in LLM agents into planning graphs, memory graphs, tool graphs, and multi-agent-system graphs; GenBench's knowledge graph functions as what that taxonomy calls a knowledge-memory graph, but one that is additionally cross-modal, per-edge confidence-annotated, and updated (via CoCG) by the same optimization step that trains the policy consuming it, none of which the taxonomy's existing knowledge-memory exemplars provide.

A.10 Self-evolving agent memory and skill accumulation

A distinct cluster of work lets an agent's memory or skill repertoire evolve over time, rather than evolving a knowledge graph's edge confidence or a policy's weights directly. Mem^{2}Evolve (Cheng et al., 2026) co-evolves an experience memory with an asset memory, using accumulated experience to guide the creation of new tools or expert agents and vice versa. It also uses the term "co-evolution", but for a different pairing than our Figure 1 (per-edge confidence and policy weights within one GRPO step, not memory and capability across a longer horizon). A broader family, including Memory-Driven Self-Improvement (Yan et al., 2025), Continual Self-Improvement (Dorovatas et al., 2026), MILES (Tong & Gong, 2026), MemSkill (Zhang et al., 2026), and SAGE (Liang et al., 2025), accumulates and selectively retrieves experiential or procedural memory across episodes to improve future decisions, and SimpleMem (Liu et al., 2026) targets efficient lifelong memory specifically. None of these methods attaches confidence to individual facts in a structured knowledge source or updates such confidence inside an RL policy-gradient step; their unit of evolution is a memory record, skill description, or tool, not an edge-level belief.

B Additional Details on GenBench

This appendix expands on the design details summarized in Section 2, for readers who wish to reproduce or extend individual components of the benchmark.

B.1 Seed Gene Curation

GenBench's 210 seed genes are selected, not sampled: every candidate gene is scored against a fixed set of curated and large-scale evidence sources, and only the highest-scoring candidates are retained as seeds around which the graph is subsequently expanded (§B.2).

For each disease under consideration, four sources are queried: ClinGen and OMIM, whose gene–disease relationships are manually reviewed by clinical experts, and the Open Targets Platform and

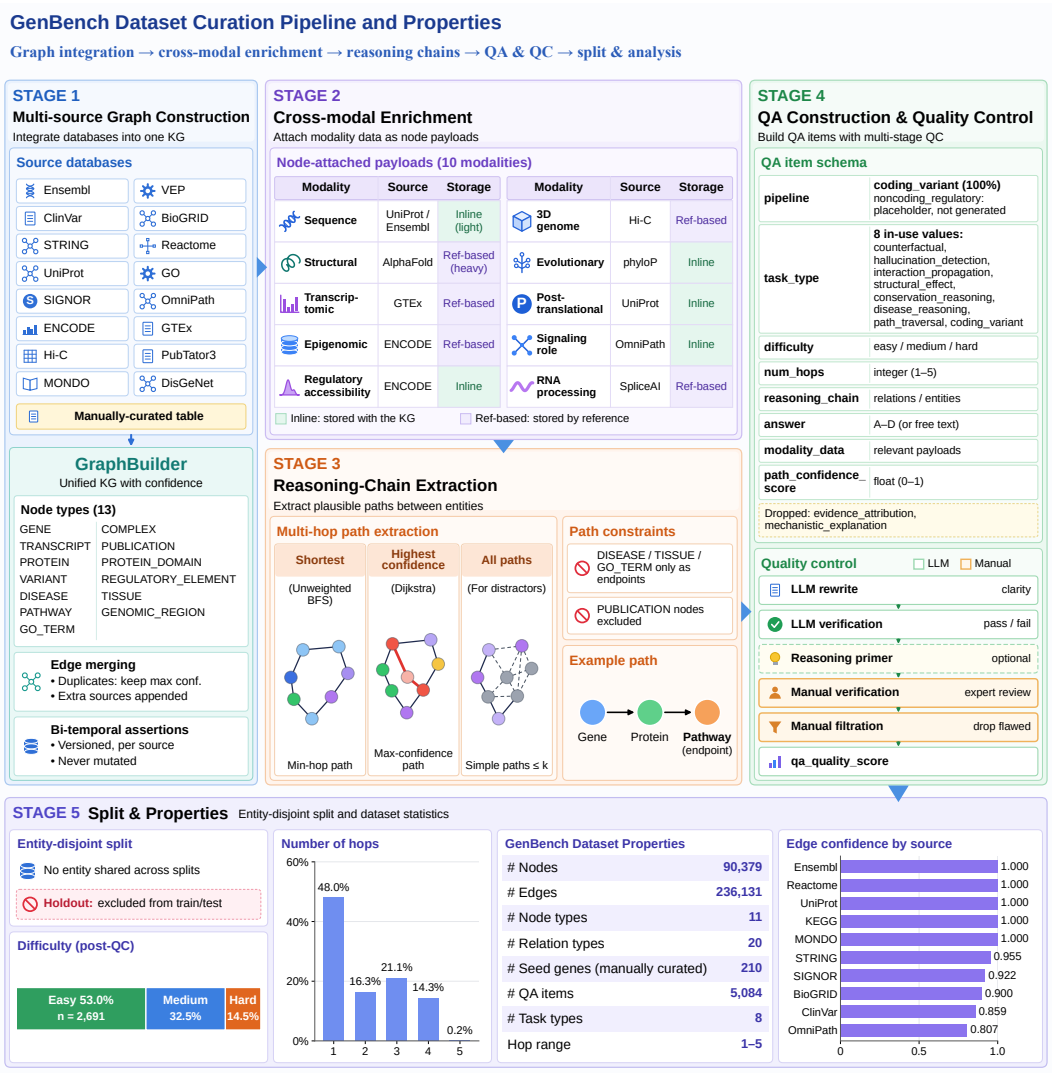


Figure 3: The **GenBench** dataset curation pipeline and resulting dataset properties. Stage 1 integrates eighteen collector sources into a unified, confidence-scored knowledge graph via GRAPHBUILDER. Stage 2 attaches the seven populated modality families through relation-level links. Stage 3 extracts shortest, highest-confidence, and all-simple-path reasoning chains under biological-plausibility constraints. Stage 4 constructs and quality-controls QA items via LLM rewrite/verification and manual expert review. Stage 5 performs the gene-level holdout split and reports the released set's scale and distributions.

PubTator, whose relationships are derived from large-scale genetic association data and literature mining, respectively. Each source contributes independently and additively to a per-(disease, gene) evidence score: a ClinGen record contributes a score set by its clinical validity classification (Definitive: $+5$, Strong: $+4$, Moderate: $+3$, Limited: $+1$, Disputed: -2, Refuted: -5); an OMIM disease-gene record contributes a flat $+4$; an Open Targets record contributes $+3$ only if its association score exceeds 0.8 (itself applied on top of an initial minimum-score prefilter used when querying Open Targets); and each PubTator literature record contributes $+1$, so a gene supported by multiple independent PMIDs accumulates a correspondingly larger literature score rather than being capped at a single point. Candidate genes are ranked within each disease by this summed score (ties broken alphabetically) and additionally assigned a source-composition tier: genes corroborated by ClinGen, OMIM, and Open Targets jointly are Tier 1; genes corroborated by OMIM and Open Targets without

ClinGen are Tier 2; genes supported by PubTator alone are Tier 3; all other combinations are labeled Evidence-supported.

The 210 released seed genes are the union, across all diseases under consideration, of each disease's top-ranked candidates, deduplicated by gene identifier and retained by their best (lowest) per-disease rank. This ranked, tiered seed table is itself inserted into the graph as a small set of `Curated GENE_ASSOCIATED_WITH_DISEASE` edges (Eq. 1) before any collector expansion begins, so that the highest-confidence, most expert-reviewed associations in the graph are attributable to this stage specifically rather than to a downstream collector.

B.2 Graph Expansion: Eighteen Collector APIs

Starting from the 210 seed genes, the graph is expanded by eighteen live collector APIs, each responsible for one or more relation types (Table 4). Collectors are queried per-gene (or per-disease, for the four seed-curation sources reused here) and their results are harmonized onto canonical identifiers before insertion: gene symbols are resolved against Ensembl (including previous symbols and aliases), proteins against UniProt accessions, and variants against dbSNP/ClinVar identifiers where available.

Collector	Relation(s) produced	Node types touched	Evidence class
ClinGen	`GENE_ASSOCIATED_WITH_DISEASE`	GENE, DISEASE	curated
OMIM	`GENE_ASSOCIATED_WITH_DISEASE`	GENE, DISEASE	curated
Open Targets	`GENE_ASSOCIATED_WITH_DISEASE`	GENE, DISEASE	multi-evidence
PubTator	`GENE_ASSOCIATED_WITH_DISEASE`	GENE, DISEASE	literature-inferred
KEGG	`GENE_PARTICIPATES_IN_PATHWAY`, `GENE_ASSOCIATED_WITH_DISEASE`	GENE, PATHWAY, DISEASE	curated
Reactome	`GENE_PARTICIPATES_IN_PATHWAY`	GENE, PATHWAY	curated
PROGENy	`GENE_PARTICIPATES_IN_PATHWAY`	GENE, PATHWAY	computational
BioGRID	`PROTEIN_PHYSICALLY_INTERACTS_WITH`	PROTEIN	physical
STRING	`PROTEIN_FUNCTIONALLY_ASSOCIATED_WITH`	PROTEIN	multi-channel

continued on next page

Table 4 – continued from previous page

Collector	Relation(s) produced	Node types touched	Evidence class
OmniPath	PROTEIN_MEMBER_OF_COMPLEX	PROTEIN, COMPLEX	curated
DisGeNet	GENE_ASSOCIATED_WITH_DISEASE	GENE, DISEASE	literature-inferred
SIGNOR	PROTEIN_ACTIVATES, PROTEIN_INHIBITS, PROTEIN_REGULATES	PROTEIN	causal
GO	GENE_ANNOTATED_WITH_GO_TERM	GENE, GO_TERM	annotation (evidence-coded)
UniProt	GENE_ENCODES_PROTEIN	GENE, PROTEIN	curated
Ensembl	GENE_HAS_TRANSCRIPT, TRANSCRIPT_ENCODES_PROTEIN	GENE, TRANSCRIPT, PROTEIN	structural
Ensembl VEP	VARIANT_HAS_VEP_CONSEQUENCE	VARIANT, GENE	computational
ClinVar	VARIANT_IN_GENE, VARIANT_HAS_CLINICAL_SIGNIFICANCE	VARIANT, GENE, DISEASE	curated
GTEx	GENE_EXPRESSED_IN_TISSUE	GENE, TISSUE	expression

Table 4: The eighteen live collector APIs used to expand the graph beyond its 210 curated seed genes, the relation types each is responsible for, and the node types and evidence class each contributes. GENE_ENCODES_PROTEIN edges without an explicit UniProt record are additionally inserted as structural bridge edges during harmonization, connecting the gene and protein layers deterministically rather than through an external evidentiary claim.

Because these are live web services, results are cached by (collector, query-key) at retrieval time; the released graph is a single frozen snapshot of these cached responses rather than a live-refreshed artifact (Section 2.2.2).

B.3 Modality Families and the Relation-to-Family Map

GenBench's schema reserves ten modality families, all listed in Table 5. Seven are populated in the released collection: protein sequence (ESM-2 embeddings over the UniProt sequence, anchored to a GENE or PROTEIN node), protein structure (AlphaFold, anchored to PROTEIN), sequence conservation (phyloP, anchored to VARIANT), tissue expression (GTEx, anchored to GENE, reusing the per-tissue TPM profiles that populate GENE_EXPRESSED_IN_TISSUE edges), splice-site effect (SpliceAI, anchored to VARIANT), post-translational modification sites (anchored to PROTEIN),

and signalling role (anchored to PROTEIN, reusing SIGNOR/OmniPath edge evidence). The other three (chromatin accessibility, single-cell expression, and protein–protein interaction interface structure) have reserved relation mappings but no collector yet, and task types that would depend on them are held out of the current release (§B.6).

Modality family (data type)	Representative encoder / source	Populated
Protein sequence	Frozen ESM-2	Yes
Protein structure	AlphaFold	Yes
Sequence conservation	phyloP	Yes
Tissue expression	GTEx	Yes
Splice-site effect	SpliceAI	Yes
Post-translational sites	PTM annotation databases	Yes
Signalling role	Pathway / signalling DBs	Yes
Chromatin accessibility		No
Single-cell expression		No
Protein–protein interaction interface structure		No

Table 5: Modality families defined in the **GenBench** schema.

Each modality family is joined to the graph through an explicit relation-to-family map (e.g., `PROTEIN_PHYSICALLY_INTERACTS_WITH` and `PROTEIN_MEMBER_OF_COMPLEX` both map to the signalling-role and structure families; `VARIANT_IN_GENE` and `VARIANT_HAS_CLINICAL_SIGNIFICANCE` map to the conservation and splice families), rather than a fixed modality being attached to a node type irrespective of which edge is being traversed.

At generation time, every released item is checked against a non-empty-modality invariant: an item is only emitted if at least one hop in its reasoning chain resolves, through the relation-to-family map, to a modality family with a real (non-placeholder) payload for the node in question. Items that would otherwise resolve only to one of the three unpopulated families are discarded rather than emitted with an empty payload.

B.4 Reasoning-chain and confidence provenance fields

Each **GenBench** item is accompanied by an explicit reasoning chain that records the provenance of every hop in the underlying knowledge graph. The resulting structure forms a gold Query-KG trace, providing an explicit correspondence between the natural-language question, the graph traversal required to answer it, and the evidence supporting each intermediate assertion. This trace can therefore be used directly to construct supervised warm-start trajectories without requiring additional annotation of the reasoning path.

For each hop h in a reasoning chain, we record the following fields:

- Source node: the entity from which the hop originates.
- Target node: the entity reached by the hop.
- Relation type: the typed relationship connecting the source and target.
- Evidence type: the evidentiary category supporting the relation.
- Originating database: the source database from which the assertion was obtained.
- Edge confidence: a normalized measure of the evidentiary strength associated with the assertion.

The provenance representation is designed to preserve not only what relationship was traversed, but also where it originated and what type of evidence supports it. This is particularly important in a heterogeneous knowledge graph, where relationships originating from curated clinical resources, experimental interaction databases, expression resources, and computational or literature-derived sources carry different native evidence.

Source-native confidence initialization. Edge confidence is initialized from the evidentiary semantics provided by the originating database rather than by applying a single global confidence rule to all relationships. Representative mappings include:

- **ClinVar**: confidence derived from the reported review/star rating.
- **Gene Ontology (GO)**: confidence derived from the associated evidence code.
- **STRING**: confidence derived from the normalized interaction score.
- **BioGRID**: confidence derived from the experimental throughput/evidence class.
- **GTEx**: confidence derived from the corresponding log-scaled expression value.

Thus, the initial confidence value represents the strength of the evidence available for a particular assertion within its source database. The resulting value is subsequently normalized into the common confidence representation used by the graph and downstream reasoning framework.

For a reasoning chain

$$C = (e_1, e_2, \ldots, e_H),$$

where H is the number of hops and e_h denotes the h-th graph assertion, the chain-level confidence is computed from the constituent edge confidences as defined in Eq. 2. This provides a confidence-monotonic quantity that can be propagated through multi-hop reasoning and subsequently consumed by the confidence-aware reward in Section 3.2.

Importantly, source-derived confidence should be interpreted as evidence strength within the integrated knowledge graph, rather than as a calibrated probability that a biological assertion is true. Different databases encode different notions of evidentiary support, and the normalization provides a common computational representation while retaining the originating provenance.

B.5 Difficulty scoring

Rather than being manually assigned during question generation, **GenBench** difficulty emerges from the evidential and structural characteristics of each reasoning chain. The goal is to use characteristics of the underlying KG trace to describe the intrinsic complexity of a QA item.

A 13-point difficulty score is formed by combining five signals, then banded into three difficulty levels: easy, medium, and hard.

1. Hop count: the number of graph transitions needed to complete the reasoning chain.
2. Node-type diversity: the number and variety of node types encountered along the chain.
3. Relation diversity: the variety of relation types encountered along the chain.
4. Provenance diversity: the number and variety of originating databases contributing evidence to the chain.
5. Evidence-weakness penalty: an added contribution for chains containing relatively poorly supported assertions.

The final score is meant to capture complementary aspects of reasoning complexity. For instance, two chains with the same hop count can differ substantially in difficulty: one may stay within a single relation type inside one modality, while the other requires transitions across multiple biological entity types, relation classes, and evidence sources.

Difficulty does not use the path-confidence score. The evidence-weakness term counts weakly supported assertions as extra work for the solver, not as a judgment of reliability, so the difficulty score should not be read as an estimate of whether a chain is biologically correct; it describes the structural and evidentiary complexity a model must navigate.

This distinction matters because difficulty and confidence capture different properties. A long chain may consist entirely of strongly supported assertions, giving high structural difficulty but high evidence confidence. Conversely, a short chain may contain a single weakly supported assertion, giving low structural difficulty but reduced evidence confidence.

Accordingly, the path-confidence quantity $PC(C)$ (Eq. 2) has no bearing on the difficulty score: the reward in Section 3.2 consumes the confidence-monotonic quantity, while the difficulty score is instead used for stratified analysis and benchmark characterization.

B.6 TASK-TYPE ACCOUNTING

The **GenBench** generation pipeline registers eleven task types covering complementary forms of genomic and graph-based reasoning. Task templates are instantiated automatically from the available graph structure, provenance information, evidence annotations, and reasoning chains.

During benchmark construction, the registered task types undergo quality control and repair. Of the eleven:

- Two are dropped by the repair pass, because the generated items exhibit documented data-quality issues that prevent reliable benchmark release.
- One is starved by a collector-side data-collection bug, yielding an insufficient number of valid instances for release.
- Eight satisfy the release criteria and constitute the task families reported in Table 6.

This accounting is kept explicit so that the distinction between registered generation templates and released benchmark tasks remains clear: registration of a task type does not by itself imply that it contributes valid items to the final benchmark, since an item must first pass the corresponding structural, provenance, and data-quality checks. The pipeline records whether each task type was removed during quality-control repair, failed from insufficient data, or was retained for release, which keeps benchmark construction reproducible.

B.7 SPLIT CONSTRUCTION

Task type	Train	Test	Total	Test %
Structural effect	677	112	789	14.2
Interaction propagation	651	149	800	18.6
Hallucination detection	627	173	800	21.6
Conservation reasoning	624	132	756	17.5
Counterfactual	598	202	800	25.3
Disease reasoning	523	93	616	15.1
Path traversal	263	151	414	36.5
Coding variant	101	8	109	7.3
Total	**4,064**	**1,020**	**5,084**	**20.1**

Table 6: Released split composition. The non-uniform per-type test fraction is a designed consequence of the gene-level holdout described below, not an artifact of the sampling process.

The gene-level holdout is selected via a deterministic stable hash over gene identifiers, ensuring the same holdout set is reproduced across regenerations of the benchmark. Each QA item is then partitioned into one of three categories:

- **Clean**: every gene touched by the item is in the holdout set.
- **Mixed**: some, but not all, genes touched by the item are in the holdout set.
- **Eligible**: no gene touched by the item is in the holdout set.

Clean and mixed items are both forced into the test set, since placing a mixed item in training would reintroduce a held-out gene into the training gene set and violate the holdout guarantee for every other item sharing that gene. Task-type stratification is applied only within the remaining eligible pool, and because the forced-test items are layered on top of an allocation computed at a reduced effective fraction, the resulting per-type test proportions in Table 6 are non-uniform by design rather than by sampling artifact.

B.8 Dataset Reproducibility

The eighteen collector APIs used to expand the graph beyond its curated seed are live external web services and are expected to drift over time (schema changes, deprecations, updated records). Accordingly, the released snapshot, not the collection pipeline, is the artifact intended for reproducible use; re-running the collectors against live endpoints is not guaranteed to reproduce the released graph exactly. The gene-level holdout guarantee (Section B.7) is verified empirically by the split script against the released snapshot, rather than assumed from the sampling procedure.

B.9 Expert Validation of a Held-Out Item Sample

We manually audited a sample of held-out test items to check whether the programmatically constructed QA items reflect biologically meaningful reasoning rather than syntactically valid but biologically incorrect graph traversal. This is a quality-control audit of a subset, not a full validation of the benchmark.

B.9.1 Validation Objective

The objective is to determine, for each sampled item, (i) whether the generated final answer is correct, (ii) whether the reasoning chain connecting question to answer is itself biologically valid, and (iii) whether these two judgments agree, since a benchmark in which they diverge (a correct answer reached through an invalid chain, or an incorrect answer reached through a chain the expert otherwise accepts) is informative in a way that answer-only checking cannot surface.

B.9.2 Validation Dimensions

Each reviewed item was assessed on the following dimensions:

- **Answer correctness** (`annotator_1_answer_correct`): a Yes/No judgment on whether the item's proposed answer is correct.
- **Reasoning-chain validity** (`annotator_1_chain_valid`): whether the sequence of relationships connecting the question to the answer is itself biologically sound; recorded values include *Yes Chain Valid*, *Context Chain is valid*, *partially*, *Wrong/Weak evidences picked up*, and *NO THE CHAIN IS NOT VALID*.
- **Overall quality** (`annotator_1_overall_quality`): a High/Medium/Low holistic rating.
- **Adjudication commentary** (`adjudicated_label`): free-text expert notes explaining a judgment, populated for items where the reviewer noted a specific issue (e.g., a weak evidence base, a mismatched entity, or a database artifact).

Evidence/source adequacy and entity-mapping correctness were assessed as part of the chain-validity and adjudication-commentary fields above rather than as separate columns; several of the representative cases in Table 8 illustrate this distinction (a chain can be numerically/structurally consistent while resting on an incorrect entity mapping).

B.9.3 Validation Procedure

Review was conducted by a single primary annotator per item, with adjudication commentary recorded for a subset rather than as a fully independent second-annotator pass (§B.9.4).

B.9.4 Expert Validation Results

The audit sample contains 100 held-out items spanning ten task types (including `mechanistic_explanation` and `evidence_attribution`, which were subsequently excluded from the eight released task types of Table 6 for the data-quality and collector-coverage reasons given in Appendix B.6). Of these 100, **45** carry a completed primary-annotator judgment on both answer correctness and chain validity; we report statistics over these 45. Second-annotator labels were not collected in usable form, so this is a single-annotator audit with adjudication commentary on a subset, and we do not report inter-annotator agreement.

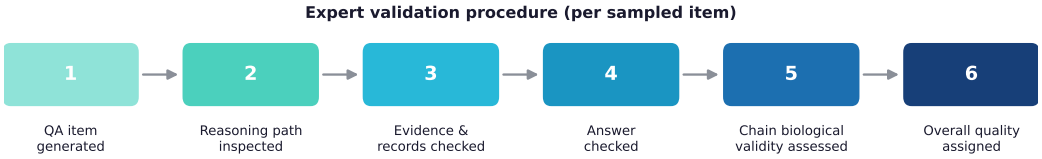


Figure 4: The six-stage expert validation procedure applied to each sampled held-out item, in order: (1) a QA item is generated; (2) the reviewer inspects the knowledge-graph reasoning path underlying it; (3) the supporting evidence and database records behind that path are inspected; (4) the final answer is checked for correctness; (5) the biological validity of each relationship in the chain is assessed; and (6) an overall quality rating is assigned. Steps 2, 3, and 5 together determine the `annotator_1_chain_valid` judgment, step 4 the `annotator_1_answer_correct` judgment, and step 6 the `annotator_1_overall_quality` rating reported in Table 7.

Quantity	Count (of 45 reviewed)
Answer judged correct	38
Answer judged incorrect	7
Chain judged valid	39
Chain judged invalid	4
Chain judged partially valid / weak evidence	2
Overall quality: High	26
Overall quality: Medium	12
Overall quality: Low	6
Items with adjudication commentary	26

Table 7: Expert-review outcomes over the 45 items with a completed judgment, out of a 100-item candidate audit sample. Quality is unrated for one of the 45 items.

Task-type distribution among the 45 reviewed items: `path_traversal` (8), `mechanistic_explanation` (5), and four each for `coding_variant`, `disease_reasoning`, `conservation_reasoning`, `counterfactual`, `evidence_attribution`, `hallucination_detection`, `interaction_propagation`, and `structural_effect`. Hop-depth distribution: 1-hop (13), 2-hop (4), 3-hop (11), 4-hop (8), 5-hop (9). This sample is not claimed to be representative of the released test set's task-type or hop-depth composition (Table 6); the sample was selected for task-type coverage rather than drawn at random, so it should be read as an audit sample, not a statistically representative one.

Answer correctness and chain validity disagree on three of the 45 items: one item (`co_stru_00653`, Table 8) has a correct answer reached via a chain the expert flagged as biologically flawed, and two items have an expert-validated chain paired with an incorrect answer. Scoring the two dimensions separately is what exposes these cases.

B.9.5 Representative Validation Cases

Table 8 gives five reviewed items illustrating distinct outcomes, quoted from the expert's `adjudicated_label` comments where present.

B.9.6 Implications for GenBench

This audit provides evidence that GenBench's reasoning chains are, for the sampled items, largely biologically meaningful (38/45 correct, 39/45 chain-valid), and that the benchmark can expose the specific failure modes it is meant to guard against: a false-positive relation traceable to shared terminology rather than genuine interaction (`co_mech_00098`, from a task type later excluded from the release), and an answer that is arithmetically correct but rests on a mismatched entity

QA ID	Task	Outcome	Key expert observation
co_coun_00061	Counterfactual	Correct / valid / high	"the relational logic precisely outlines a verified human disease-gene architecture"
co_mech_00098	Mechanistic explanation	Incorrect / invalid / low	"The connection between APC and SNCA is artificial because it relies on a shared descriptive keyword ('microtubule binding') rather than an actual biological dependency"; a false-positive relation rejected by the expert (task type excluded from the release)
co_stru_00653	Structural effect	Correct / flawed context / medium	"the context is flawed because it mixes up two distinct genes"; the structural-loss arithmetic was internally consistent, but computed on AOPEP's residue count while the variant nomenclature named FANCC; correct answer, wrong entity mapping
co_cons_00032	Conservation reasoning	Incorrect / valid / medium	"The answer is wrong, It should be B"; the expert identifies the specific phyloP-score misinterpretation behind the wrong choice
co_codi_00092	Coding variant	Correct / valid / high	"The variant node has been just reported once in ClinVar"; answer and chain accepted, but flagged as resting on a single, low-multiplicity evidence record

Table 8: Five representative reviewed items spanning acceptance, rejection, correct-answer/flawed-context, incorrect-answer-with-diagnosed-cause, and accepted-but-weak-evidence outcomes. QA IDs and quotations are taken from the expert review.

(co_stru_00653). It does *not* demonstrate that all 5,084 QA items are biologically correct, nor that the 45 reviewed items are statistically representative of the full benchmark or even of the full test set: they are a non-random audit sample, reviewed by a single primary annotator with adjudication commentary on a subset, covering ten task types of which two are absent from the final release. We report this as an independent quality-control audit of a subset, consistent with the reading in Section 2, and not as exhaustive validation of GenBench.

B.10 Comparison to related genomic and multi-hop reasoning benchmarks

Tables B.10 and B.10 compare **GenBench** with recent genomics foundation models and with recent knowledge-graph and multi-hop genomic reasoning benchmarks, respectively. They make concrete the gap described qualitatively in Appendix A.2.

Model	Venue (Year)	Protein seq.	Protein struct.	Conservation	Expression	Splice site	PTM	Signalling
AlphaGenome (Avsec et al., 2026)	Nature (2026)	×	×	×	✓	✓	×	×
Borzoi (Linder et al., 2025)	Nat. Genet. (2025)	×	×	×	✓	✓	×	×
ESM3 (Hayes et al., 2025)	Science (2025)	✓	✓	×	×	×	×	×
GET (Fu et al., 2025)	Nature (2025)	×	×	×	✓	×	×	×
scGPT (Cui et al., 2024)	Nat. Meth. (2024)	×	×	×	✓	×	×	×
Geneformer (Theodoris et al., 2023)	Nature (2023)	×	×	×	✓	×	×	×
HiCFoundation (Wang et al., 2026)	Nat. Meth. (2026)	×	×	×	×	×	×	×
GenBench (ours)	—	✓	✓	✓	✓	✓	✓	✓

Table 9: Coverage of the seven modality families GenBench populates (§2.2.1, Table 5) by recent genomics foundation models. No model covers more than two of the seven, and none covers post-translational sites or signalling roles, which GenBench attaches from curated PTM annotations and SIGNOR/OmniPath evidence rather than from a trained predictor (Appendix B.3).

Benchmark	Venue (Year)	KG	Multi-hop	Genomics	Modality	Reasoning task
BioHopR (Kim et al., 2025)	ACL Findings (2025)	PrimeKG	✓ Explicit	Low	Text	Open-ended KGQA
BioReason (Fallahpour et al., 2025)	NeurIPS (2025)	KEGG-based	✓ Implicit	High	DNA + text	Pathway reasoning
CGBench (Queen et al., 2025)	NeurIPS (2025)	×	×	High	Text	Clinical-genetics reasoning scored against expert rationales
PrimeKG (Chandak et al., 2023)	Sci. Data (2023)	✓ Core KG	Indirect	Moderate	Text + relations	KG resource, not a reasoning benchmark
GeneTuring (Shang et al., 2025)	Brief. Bioinform. (2025)	×	×	High	Text, sequence	Short-form genomic QA
BioKGBench (Lin et al., 2024)	arXiv (2024)	✓ Composite KG	×	Moderate	Text	KGQA and claim verification
GenBench (ours)	—	✓ 90k nodes, 236k edges, per-edge confidence	✓ Explicit, 1–5 hops	High	Cross-modal, 7 populated families	Multi-hop QA with released reasoning chains for training

Table 10: Genomic knowledge-graph and multi-hop reasoning benchmarks compared with GenBench.

No resource in Table B.10 combines everything **GenBench** provides. **BioReason** is the closest analogue to a reasoning-trace dataset, since it provides step-by-step explanations rather than only a final label. Its graph, however, is limited to KEGG pathways, a single and comparatively narrow biological namespace, with no cross-modal evidence to retrieve during reasoning and no per-edge confidence to condition on or update. **CGBench** scores a model's explanations against expert-written rationales, a check **GenBench** does not perform. Those rationales are used to *score* outputs and are not *released* as per-item training supervision. **GenBench** takes the opposite position: every item's complete, confidence-annotated reasoning chain (§2.1.2) is released so that it can be trained on, not only evaluated against. **BioHopR** and **BioKGBench** test knowledge-graph QA, but over general or composite biomedical graphs rather than a genomics-specific one, with no retrievable non-text modality and no confidence attached to edges. **PrimeKG** is a graph resource that other benchmarks (including BioHopR) build on, not a reasoning benchmark. **GeneTuring** tests genomic knowledge through short-form factual questions rather than multi-hop composition over a graph.

Table B.10 makes a complementary point about representation learning. No genomics foundation model we found covers more than two of GenBench's seven populated families: ESM3 covers protein sequence and structure; AlphaGenome and Borzoi cover expression and splicing; GET, scGPT, and Geneformer cover expression only; and HiCFoundation, which models chromatin contacts, covers none of the seven. None reports post-translational or signalling-role information. **GenBench** obtains these two families by attaching curated annotations directly to the graph (Appendix B.3), a route not open to a model that must predict every modality from sequence or accessibility data. No single released model therefore provides cross-modal genomic evidence on its own. **GenBench** instead makes modality access a property of a graph relation that an agent retrieves *during* reasoning (§3.1), rather than a representation that one model must already have fused into its weights.

To our knowledge, **GenBench** is the only resource that jointly offers (i) a large knowledge graph with per-edge confidence, (ii) empirically verified multi-hop structure (1–5 hops, §2.2.1), (iii) genomics coverage beyond a single pathway database, (iv) retrievable cross-modal evidence across seven populated modality families, and (v) complete released reasoning chains usable as training supervision rather than only as an evaluation rubric.

Item	Question, answer, and gold chain
Coding variant	(`coding_variant`, multiple choice)
`co_codi_00000` 1-hop, easy	**Q:** According to Ensembl VEP, what is the predicted functional consequence of NM_000038.6(APC):c.1673del(p.Asn558fs) in APC? **Options:** A. missense_variant B. synonymous_variant C. splice_region_variant **D. frameshift_variant** ✓ *Chain:* NM_000038.6(APC):c.1673del(p.Asn558fs) VARIANT_HAS_VEP_CONSEQUENCE APC
Conservation reasoning	(`conservation_reasoning`, multiple choice)
`co_cons_00000` 1-hop, easy	**Q:** Using the phyloP conservation track provided for GRCh38/hg38 9q34.3(chr9:135204722-138394717) (in gene LOC126860794), classify the evolutionary constraint across the variant's genomic window. **Options:** A. weakly conserved (mean phyloP 0.0-1.5) **B. not conserved (mean phyloP < 0.0)** ✓ C. highly conserved (mean phyloP > 1.5) *Chain:* GRCh38/hg38 9q34.3(chr9:135204722-138394717) VARIANT_IN_GENE LOC126860794
Counterfactual	(`counterfactual`, open-ended)
`co_coun_00000` 3-hop, medium	**Q:** If BARD1 were knocked out, the BRCA1-B complex complex would lose one of its members, indirectly affecting BRIP1, another member of the same complex. Which disease association of BRIP1 could therefore be indirectly affected? **A:** Fanconi anemia complementation group J *Chain:* BARD1 PROTEIN_MEMBER_OF_COMPLEX BRCA1-B complex → BRIP1 PROTEIN_MEMBER_OF_COMPLEX BRCA1-B complex → BRIP1 GENE_ASSOCIATED_WITH_DISEASE Fanconi anemia complementation group J
Disease reasoning	(`disease_reasoning`, open-ended)
`co_dise_00000` 2-hop, medium	**Q:** According to ClinVar, name a variant in BRAF classified as Pathogenic or Likely Pathogenic for cardiofaciocutaneous syndrome 1. **A:** NM_004333.6(BRAF):c.1497A>T (p.Lys499Asn) *Chain:* NM_004333.6(BRAF):c.1497A>T (p.Lys499Asn) VARIANT_IN_GENE BRAF → NM_004333.6(BRAF):c.1497A>T (p.Lys499Asn) VARIANT_HAS_CLINICAL_SIGNIFICANCE cardiofaciocutaneous syndrome 1
Hallucination detection	(`hallucination_detection`, multiple choice)
`co_hall_00000` 3-hop, medium	**Q:** One of the following protein interactions is NOT supported by any entry in BioGRID or STRING. Which one is fabricated? **Options:** A. AXL interacts with HSPA1A **B. UNG interacts with ADNP** ✓ C. EGF interacts with ERBB2 D. EZR interacts with BRCA1 *Chain:* AXL GENE_ENCODES_PROTEIN AXL → AXL PROTEIN_PHYSICALLY_INTERACTS_WITH HSPA1A → HSPA1A GENE_ENCODES_PROTEIN HSPA1A
Interaction propagation	(`interaction_propagation`, multiple choice)
`co_inte_00000` 1-hop, easy	**Q:** How many direct STRING interaction partners does SPTLC2 have with a combined association score ≥ 400? **Options:** A. 11 B. 8 C. 9 **D. 10** ✓ *Chain:* SPTLC2 PROTEIN_FUNCTIONALLY_ASSOCIATED_WITH SPTLC1; also >= 400: KDSR, ORM1, ORMDL1, ORMDL2, ORMDL3, SPTLC3, SPTSSA, SPTSSB, ZDHHC9
Path traversal	(`path_traversal`, open-ended)
`co_path_00004` 2-hop, easy	**Q:** What is the shortest molecular path from SLC18A2 to APOE through known protein interactions and pathway memberships? **A:** SLC18A2 → SNCA → APOE *Chain:* SLC18A2 PROTEIN_FUNCTIONALLY_ASSOCIATED_WITH SNCA → SNCA PROTEIN_FUNCTIONALLY_ASSOCIATED_WITH APOE
Structural effect	(`structural_effect`, multiple choice)
`co_stru_00000` 1-hop, easy	**Q:** NM_013254.4(TBK1):c.1770_1771del (p.Tyr591fs) introduces a premature truncation at residue 591 of TBK1's 729-residue protein product (AlphaFold model, mean pLDDT 89.7). Approximately how much of the protein is lost downstream of the truncation? **Options:** A. a substantial portion (25-50%) **B. a small C-terminal portion ($<$25%)** ✓ C. more than half the protein ($>$50%) *Chain:* NM_013254.4(TBK1):c.1770_1771del (p.Tyr591fs) VARIANT_IN_GENE TBK1

Table 11: Sample **GenBench** items, one per task type. Gold options are in bold and marked ✓; *Chain* is the gold reasoning chain (→ separates hops).

B.11 SAMPLE GENBENCH ITEMS

Table 11 lists one item from each of the eight released task types, taken from the released dataset. For each item we show its ID, hop count, and difficulty; the question; the answer options for multiple-choice task types, with the gold option in bold and marked ✓, or the gold answer for open-ended task types; and the gold reasoning chain in the templated form stored in the item's `context` field. The structured `reasoning_chain`, `modality_data`, and `evidence` fields are omitted for space. Item `co_path_00004` is the item traced end to end in Appendix D.

B.12 QUALITATIVE CASE STUDIES: CORA VERSUS ZERO-SHOT BASELINES

Figures 5 and 6 show two individual test items, one `disease_reasoning` item and one `path_traversal` item, with verbatim outputs from three zero-shot baselines (GPT-4o, Claude Opus-5, Intern-S1-mini) alongside **CoRA**'s full agentic trajectory. These are single-item case studies, not aggregate results; Tables 1 and 2 report the corresponding averages. **Cos** is the embedding cosine similarity defined in §4; **Path F1** is the same open-ended F1 metric, reported here for a `path_traversal` item specifically. Item `co_path_00784` is a different, correctly answered item from the `co_path_00004` trace given in Appendix D, which instead illustrates a failure mode.

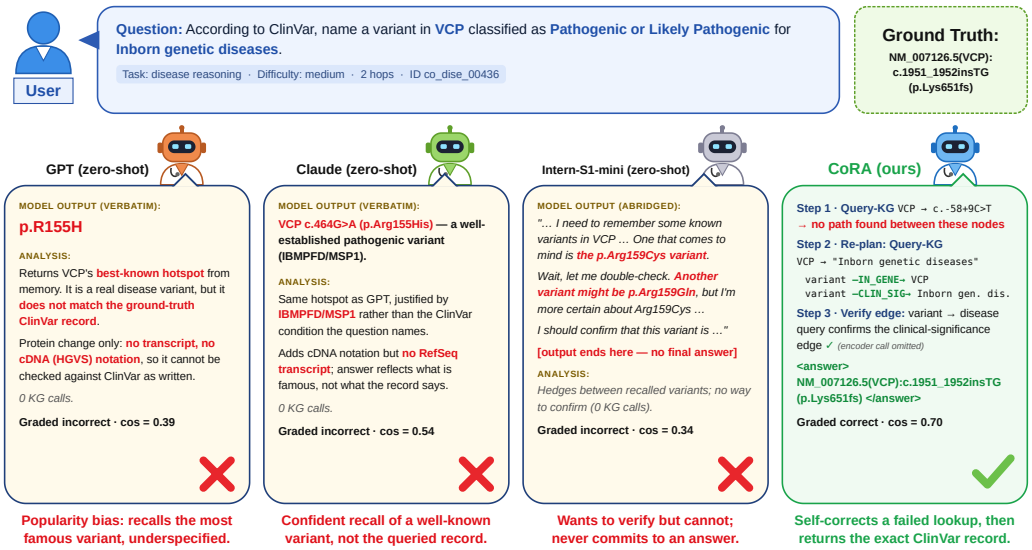


Figure 5: `disease_reasoning` item `co_dise_00436` (medium difficulty, 2 hops). GPT-4o and Claude Opus-5 recall a well-known VCP variant associated with a different disease than the one asked about; Intern-S1-mini hedges between recalled variants without committing to an answer. **CoRA**'s initial `Query-KG` call finds no path, re-plans with a query over the named disease label, verifies the resulting edge, and returns the exact ClinVar record.

C IMPLEMENTATION DETAILS, DIAGNOSED FAILURE MODES, AND CONCURRENT METHODS

This appendix gives implementation details and full equations (§C.1, §C.2), two diagnosed and fixed bugs (§C.3, §C.6), per-task and groundedness breakdowns of the ablation (§C.4, §C.5), and an equation-level comparison of ROC with three concurrent methods, OCSD, AVSD, and SEED (§C.7–§C.9).

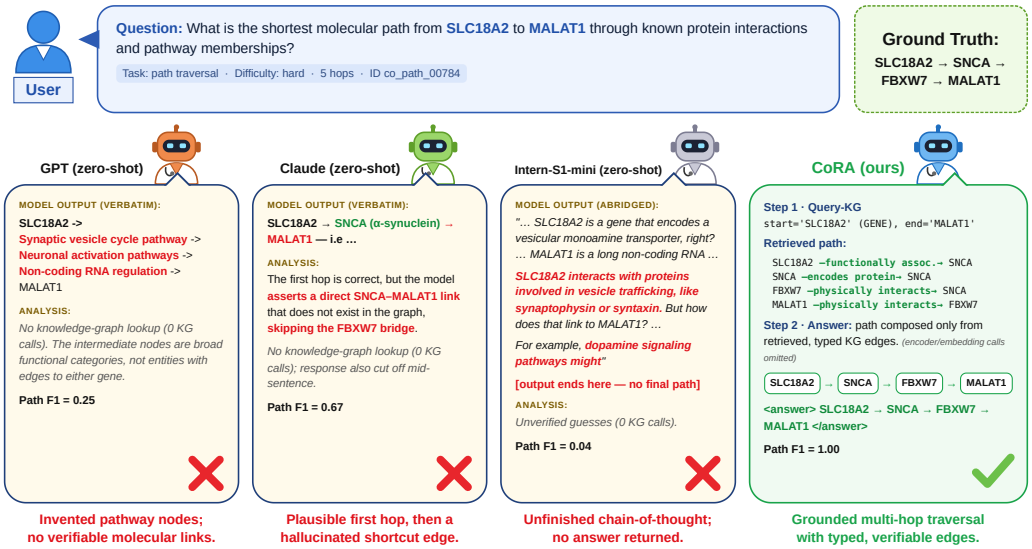


Figure 6: `path_traversal` item `co_path_00784` (hard difficulty, 5 hops). GPT-4o invents pathway-level intermediate nodes with no corresponding graph entities; Claude Opus-5 gets the first hop right but asserts an unverified direct edge that skips an intermediate gene; Intern-S1-mini does not commit to a final path. **CoRA**'s `Query-KG` call retrieves a path composed entirely of typed, verifiable edges, matching the gold chain exactly.

C.1 Architecture and Environment Implementation Details

This subsection gives implementation details for Section 3.1: the action interface, encoders, and turn budgets.

Action interface. Five actions are parsed from the first matching line of a generation; free-form reasoning before the first valid action is kept in context, and the first grammar-matching action determines the environment transition:

Action	Role
`List-Neighbors(node, node_type, relation)`	Discovery: what is connected here, top-k by confidence
`Query-KG(start, start_type, end, end_type)`	Verification: shortest path between two named endpoints
`Select-Encoder(node, node_type, family)`	Choose a modality family
`Extract-Embedding(node, node_type)`	Emit `<modality_embed>`
`Answer: ...`	Terminate

`List-Neighbors` returns confidence-sorted neighbors, capped at 60 results when filtered to a specific relation and 20 when unfiltered. Node resolution matches on `(label, type)` and tolerates unambiguous near-misses. `VARIANT`-type results additionally carry ClinVar significance and condition fields, without which the two constraints in a disease-reasoning question are not visible to the policy at all.

Multimodal encoders. Ten encoder families are defined: sequence, structural, transcriptomic, evolutionary, epigenomic, regulatory-accessibility, 3D-genome, post-translational, signalling-role, and RNA-processing. `sequence` is a frozen ESM-2; the remaining nine are small MLPs co-trained end-to-end with the policy. Encoders for families with no data in the released graph exist in code but are never invoked. Each co-trained encoder is a two-layer MLP, and one linear projection per

family maps its output into the language model's hidden size d_{LM}:

$$E_f(x) = W_2\,\mathrm{ReLU}\big(W_1\,\mathrm{LayerNorm}(x) + b_1\big) + b_2, \qquad P_f : \mathbb{R}^{d_f} \to \mathbb{R}^{d_{LM}}, \tag{10}$$

and a payload becomes a token embedding as

$$h_{b,p} = \begin{cases} P_f(E_f(v)) & \text{raw features, MLP runs in-graph} \\ P_f(v) & \text{frozen embedding, cached,} \end{cases} \tag{11}$$

scattered directly into `inputs_embeds` at the `<modality_embed>` position, so a real embedding enters the forward pass rather than a textual placeholder describing the modality.

Turn and modality budgets. Two independent budgets apply: T (the turn count used in Eq. 4) bounds reasoning turns, incremented only by the two graph actions, while M bounds modality calls and resets on every new graph action, giving a termination backstop

$$\text{hard_cap} = T(M+1) + M + 1. \tag{12}$$

CoCG operational details. The undirected-view cache used for retrieval is invalidated on every confidence write, so later rollouts within the same training run see the updated confidence immediately. A periodic stability check on a held-out slice can roll back a batch of CoCG updates: if accuracy regresses beyond tolerance, ledgered edge deltas since the last checkpoint are reverted and a confidence-bank snapshot is restored, while policy weights are left untouched. Downstream retrieval converts the internally stored confidence c_e (Eq. 9, range $[-2, 2]$) into an edge probability, $\mathrm{edge_probability}(c_e) = \mathrm{clip}(c_e, 0, 1)$, before ranking or path search; the internal range is kept wider than this consumer range because 21.9% of edges (effectively all `VARIANT_IN_GENE` edges) already sit at $\mathrm{edge_probability}(c_e) = 1.00$, where a $[0, 1]$-clamped store would silently absorb further upward updates. Whether this widening enables further correctness-correlated movement remains unverified: the one measurement available ($+0.0055$ separation between correct- and incorrect-rollout confidence trajectories) is indistinguishable from noise at this scale and has flipped sign across runs.

C.2 Full Training-Objective and Credit-Assignment Equations

This subsection gives the remaining equations for Section 3: standard machinery, secondary refinements, and terms folded into the main-text equations.

ocbook's gold-corroborated closed-book bonus (§3.2). An unverified-but-correct closed-book answer ($C_\tau = \emptyset$, so the correctness branch of Eq. 4 is zero) instead earns κ_{pe} (with modality calls) or κ_{uv} (without), adjusted by ROC's rollout-mean credit g_τ:

$$\kappa^{eff} = \mathrm{clip}(\kappa + \beta_{cg}\, g_\tau,\ 0,\ 1), \tag{13}$$

so a closed-book answer corroborated by the gold view earns more than an uncorroborated one.

GRAD's clipped surrogate and KL trust region (§3.3). The ROC-modulated advantage $A^{\mathrm{ROC}}_{i,t}$ (Eq. 6) enters a clipped importance-ratio surrogate. The log-ratio between current and previous policy is masked before exponentiation and clamped to $[-10, 10]$ for numerical safety:

$$\rho_{i,t} = \exp\Big(\mathrm{clip}\Big(\ell^{\theta}_{i,t} - \ell^{\theta_{old}}_{i,t},\ -10,\ 10\Big)\Big), \tag{14}$$

together with a KL term (the K3 estimator) against a reference (the same policy with LoRA disabled and `modules_to_save` active):

$$\delta_{i,t} = \ell^{ref}_{i,t} - \ell^{\theta}_{i,t}, \qquad \mathrm{KL}_{i,t} = e^{\delta_{i,t}} - \delta_{i,t} - 1, \tag{15}$$

combined into the aggregate loss

$$\mathcal{L}_{\mathrm{GRAD}} = \frac{1}{N}\sum_{i,t} m_{i,t}\Big[-\min\big(\rho_{i,t}A^{\mathrm{ROC}}_{i,t},\ \mathrm{clip}(\rho_{i,t}, 1{-}\epsilon, 1{+}\epsilon)A^{\mathrm{ROC}}_{i,t}\big) + \beta_{\mathrm{KL}}\mathrm{KL}_{i,t}\Big], \tag{16}$$

where $m_{i,t}$ masks non-action tokens and $N = \sum_{i,t} m_{i,t}$. As noted in the main text, $\rho \equiv 1$ on the released run (dropout zeroed, single inner epoch), so ratio clipping has no effect and the KL penalty is the only active trust region (Appendix C.6).

ROC's fourth view (§3.4). Alongside the local, global, and gold views, a no-forward-pass *prior-confidence* view reads directly off the traversed edges' stored confidence:

$$q_\tau^{(\text{conf})} = \frac{1}{|E_\tau|} \sum_{e \in E_\tau} \big(2\,\text{edge_probability}(c_e) - 1\big), \tag{17}$$

zeroed at an edge probability of 0.5 to match $\tanh$'s zero point, so it combines with the other three views in Eq. 8 on the same scale.

CoCG's two input signals (§3.5). Eq. 9's update combines Δ_e^A, a reward-contrast signal from trajectories that traverse edge e successfully versus those that fail while touching it:

$$\Delta_e^A = \eta \left(\frac{1}{|\mathcal{S}_e|} \sum_{\tau \in \mathcal{S}_e} R_\tau - \frac{1}{|\mathcal{F}_e|} \sum_{\tau \in \mathcal{F}_e} R_\tau \right), \tag{18}$$

which is near one-sided since a failed rollout earns at most small partial credit (Eq. 4), with Δ_e^B, a complementary, non-one-sided ROC-based signal:

$$\Delta_e^B = \eta \frac{1}{|\mathcal{G}_e|} \sum_{\tau \in \mathcal{G}_e} g_\tau. \tag{19}$$

C.3 A DIAGNOSED AND FIXED FAILURE MODE: ROC'S VIEWS WERE POSITIVE BY CONSTRUCTION

This subsection traces a failure in an earlier, uncentered version of ROC on one rollout and confirms it across the test set. The group-centering fix (Section 3.4) is now part of the released method.

The rollout. On item `co_path_00004` (`SLC18A2` → `SNCA` → `APOE`), the policy produced:

```
[gen 1] Query-KG(start=SLC18A2, ..., end=SNCA, ...)
[inj 1] -> SLC18A2 PROTEIN_FUNCTIONALLY_ASSOCIATED_WITH
SNCA
[gen 2] Query-KG(start=SNCA, ..., end=APOE, ...)
[inj 2] -> SNCA PROTEIN_FUNCTIONALLY_ASSOCIATED_WITH
APOE
[gen 3] Answer:  SLC18A2 -> SNCA -> APOE
```

Each view builds two rewritten copies of this sequence and re-scores both under the policy; the residual per action span is $e = \bar{\ell}^{\,\text{revealed}} - \bar{\ell}^{\,\text{ablated}}$, then $q = \tanh(e/2)$, matching Eq. 7.

Local view. Moves each hop's narration to just before its own action, so the model sees the answer while producing the query:

Revealed		Ablated
`-> SLC18A2 ..._WITH SNCA` `Query-KG(...)`	then	`not provided` then `Query-KG(...)`

Global view. A single marker placed after the prompt:

Revealed	Ablated
`[Outcome: this rollout will end up CORRECT]`	`[Outcome: not provided]`

Gold view. The item's own gold chain:

Revealed	Ablated
`[Gold path: SLC18A2 ..._WITH SNCA; SNCA ..._WITH APOE]`	`[Gold path: not provided]`

Prior-confidence view. No forward pass is needed; the view is the mean of $2c_e - 1$ over the resolved edges. Here SLC18A2→SNCA has $c = 0.98$ ($+0.96$) and SNCA→APOE has $c = 0.995$ ($+0.99$), so $q^{(\text{conf})} = +0.975$.

The gate, with illustrative numbers. Suppose the three forward-pass views return $q^{(\text{local})} = +0.45$, $q^{(\text{global})} = +0.30$, $q^{(\text{gold})} = +0.60$, alongside $q^{(\text{conf})} = +0.975$. Applying Eq. 8 with $\tau_{\text{ROC}} = 0.25$:

$$\bar{q}_j = \tfrac{1}{4}(0.45 + 0.30 + 0.60 + 0.975) = +0.581, \qquad \text{Var}_j = 0.0629,$$
$$\gamma_j = 1 - 0.0629/0.25 = 0.748, \qquad g_j = 0.581 \times 0.748 = +0.435.$$

This positive g_j both scales the advantage (Eq. 6) and drives $\Delta_e^B = \eta \cdot \text{mean}(g_\tau)$ (Eq. 19) upward for every edge this rollout touched, regardless of whether SLC18A2 → SNCA → APOE was actually the correct answer to the question asked.

The structural problem, as originally measured across the full test set. Uncentered, $q^{(\text{conf})} > 0$ for 1,019 of 1,019 test items (median $+0.933$, minimum $+0.382$, 52.6% above $+0.9$); it was never negative, because the curated knowledge graph stores high confidence almost everywhere. The other three views were positive by construction almost as reliably, for a different reason: revealing the true narration, the true outcome, or the true gold path all raise the log-probability of tokens the model already wrote, whether or not writing them was the right thing to do. With all four views agreeing positively in the overwhelming majority of cases, the variance gate γ_j stayed open and g_j came out positive almost unconditionally. A training run before the fix confirmed this: 1,468 edges changed, all 1,468 upward. No edge lost confidence during the whole run, which indicates a signal that cannot separate correct from incorrect rollouts, not one that separates them and simply finds correctness more common.

Diagnosis. ROC as specified by Eqs. 7–8, run without centering, is a confidence signal rather than a correctness signal: it measures whether revealing evidence makes the associated text more likely under the policy, which is nearly always true independent of whether the rollout's answer was right. The prior-confidence view was the least discriminating of the four by design, since it is a fixed property of the knowledge graph, identical across every rollout that touches the same edge regardless of that rollout's outcome.

The fix, and confirmation that it changes the diagnosis. $\beta_{\text{roc}} = 0.5$ is a default, non-zero setting, so an uncentered gate would be active in the released configuration; we therefore treat this as a required correction. The released mechanism (Section 3.4) centers each view, including $q^{(\text{conf})}$, within its GRPO group before aggregating, so only relative differences across the sampled group survive. On the traced item, the post-centering prior-confidence view reads $q^{(\text{conf})} = +0.005$ instead of $+0.975$. Across rollouts, centering removes the uncentered $+0.783$ mean, so this view no longer forces the gate open on 98.5% of spans. Centering removes the positive bias but does not by itself make g discriminative: Table 3 shows the centered four-view signal is still mildly mis-signed.

What centering does not fix. Removing the bias from g_τ does not create confidence headroom. Before storage was widened to $[-2, 2]$ (Eq. 9), a $[0, 1]$ clamp silently absorbed any upward update on an edge at the ceiling, where 21.9% of edges (all VARIANT_IN_GENE edges) sit. On the traced item, the centered update writes a small positive Δ_e onto both gold-chain edges; the SLC18A2–SNCA edge moves from 0.98 to 0.980479, while the SNCA–APOE edge, at 0.995, has almost no room left below the ceiling. This is a separate problem from positivity: a bounded store cannot record upward updates for edges already at or near its ceiling (Section 3.5).

Method note: the per-view ablation (Section 5.1) is forward-pass only. No training or gradient updates are performed: all four views' scores are computed once against the frozen checkpoint and recombined offline per configuration in Table 3. This measures correlation with an already-observed outcome, not the causal effect of training with a view disabled, because removing a view during training would also change the policy's later trajectories. That causal question is addressed by the training-time "GRPO w/o ROC" configuration in Table 2 (Section 5.1), which removes ROC from training itself. Read the two together: this diagnostic is correlational, the training-time ablation causal.

C.4 PER-TASK-TYPE BREAKDOWN OF THE COMPONENT ABLATION

Table 2 reports accuracy aggregated over all 200 items. Splitting by task type shows the aggregate is not uniform: most of the five multiple-choice task types are near ceiling for most configurations, while two of the three open-ended task types stay weak across every configuration, including Full. We report the multiple-choice (n=117) and open-ended (n=83) groups separately, since they occupy different accuracy bands throughout this section.

Configuration	coding variant (n=8)	halluc. detection (n=27)	interaction propagation (n=27)	structural effect (n=27)	conservation reasoning (n=28)
SFT + ocbook	1.0000	1.0000	1.0000	0.8148	0.4643
SFT (no ocbook)	0.8750	0.9259	0.6667	0.6296	0.5357
GRPO (no ROC/GRAD/CoCG)	1.0000	0.9630	1.0000	0.8519	0.6071
GRPO w/o ROC	1.0000	0.9630	1.0000	0.8889	0.5714
GRPO w/o GRAD	1.0000	0.9630	1.0000	0.8519	0.6429
GRPO w/o CoCG	1.0000	0.9630	1.0000	0.8889	0.6429
Full	1.0000	0.9630	1.0000	0.9259	0.5714

Table 12: Per-task-type accuracy, multiple-choice task types, matching the configurations in Table 2. `coding_variant` has only 8 items in this 200-item subset, so its column is indicative only. Each item is worth $1/n$ of its column (0.036–0.125).

Configuration	counter-factual (n=28)	disease reasoning (n=28)	path traversal (n=27)
SFT + ocbook	0.0714	0.0714	0.3333
SFT (no ocbook)	0.1071	0.0714	0.5556
GRPO (no ROC/GRAD/CoCG)	0.1429	0.1071	0.5556
GRPO w/o ROC	0.2143	0.1071	0.5556
GRPO w/o GRAD	0.1071	0.0714	0.5926
GRPO w/o CoCG	0.1429	0.1071	0.5185
Full	0.1786	0.1071	0.5926

Table 13: Per-task-type accuracy, open-ended task types, matching the configurations in Table 2.

The split shows three things.

First, for every GRPO configuration (bare and all three ablations), three of the five multiple-choice types (`coding_variant`, `hallucination_detection`, `interaction_propagation`) sit at or within a few points of ceiling; removing ROC, GRAD, or CoCG individually from an already-RL-trained model has essentially nothing left to explain on these three types. `structural_effect` (0.8519–0.9259 across GRPO variants) and `conservation_reasoning` (0.4643–0.6429) are the multiple-choice types that move with configuration; on the latter, Full (0.5714) is below three ablated variants.

Second, the plain SFT baseline without ocbook is a clear outlier on two of those same three "near-ceiling" types: `interaction_propagation` drops to 0.6667 (a 33-point gap to the GRPO configurations, the single largest swing of any task type in this study) and `coding_variant` drops to 0.8750. Every GRPO variant, including the one with none of ROC/GRAD/CoCG active, closes both gaps completely. The ocbook SFT stage (or RL in general, since bare GRPO also closes the gap) is what these two task types need, not any of the three RL mechanisms specifically, so their near-ceiling numbers in Table 12 should not be read as evidence for ROC, GRAD, or CoCG individually.

Third, and most consequentially, `counterfactual` and `disease_reasoning` never exceed 0.2143 and 0.1071 respectively in *any* of the seven configurations, Full included; no mechanism, alone, combined, or absent, closes this gap on this 200-item subset. This is consistent with the low groundedness of these two task types (§5.2, Appendix C.5): the model is not failing to apply a mechanism correctly on these items so much as fabricating ungrounded content on a task type the ablated mechanisms were not designed to fix. `path_traversal` (0.3333–0.5926, a 26-point range) sits in between and shows the most configuration-to-configuration movement among the open-ended types.

Hop-count breakdown. The task-type split of §5.2 is confounded with hop count, since most task types occur predominantly at one or two fixed hop counts in this test set. The combined curve is

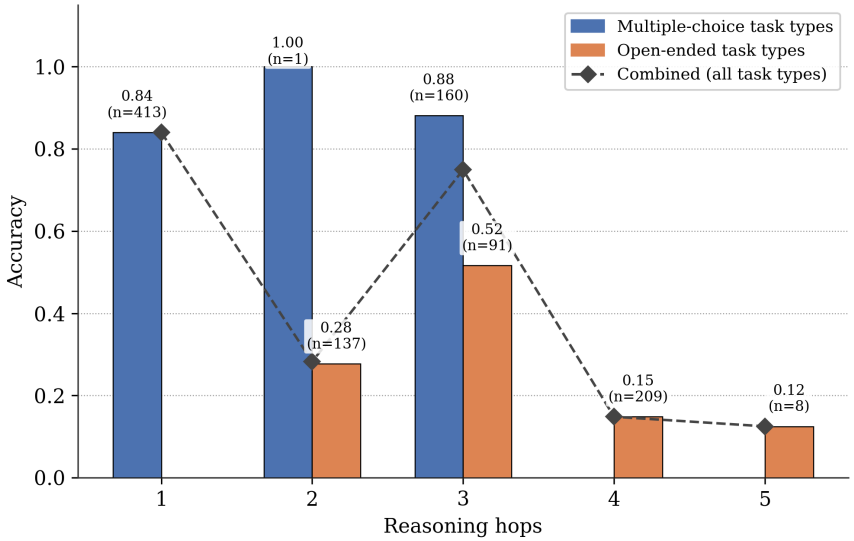


Figure 7: Accuracy by reasoning-hop count on the full released test set, split by task-type category, with per-hop n labeled. The combined curve (black, dashed) is non-monotonic because hop count is confounded with task-type composition: hops 1 and 3 are multiple-choice-heavy, hops 2, 4, and 5 are open-ended-heavy, so this plot mostly reflects which group dominates each hop level rather than isolating reasoning depth.

non-monotonic (0.84 at hop 1, n=413; 0.28 at hop 2, n=137; 0.75 at hop 3, n=251; 0.15 at hop 4, n=209; 0.12 at hop 5, n=8) because hop 1 and hop 3 are dominated by multiple-choice items while hops 2, 4, and 5 are dominated by open-ended items, not because accuracy itself rises and falls with depth. A fully hop-controlled analysis across all five hop levels would require more open-ended items at hops 1 and 5 than this release currently provides.

C.5 Per-task-type groundedness for the open-ended split

The aggregate Groundedness column in Table 2 is computed over the 83 open-ended items: `counterfactual` (n=28), `disease_reasoning` (n=28), and `path_traversal` (n=27). The multiple-choice `conservation_reasoning` items are scored for accuracy only.

Configuration	counter-factual	disease_ reasoning	path_ traversal
SFT + ocbook	0.1250	0.0714	0.9097
SFT (no ocbook)	0.2738	0.2143	0.9383
GRPO (no ROC/GRAD/CoCG)	0.2560	0.1071	0.9802
GRPO w/o ROC	0.2976	0.1071	0.9877
GRPO w/o GRAD	0.2554	0.0714	0.9728
GRPO w/o CoCG	0.2321	0.1429	0.9432
Full	**0.3095**	**0.2143**	**0.9877**

Table 14: Per-task-type groundedness (fraction of named entities that resolve to real graph entities) for the three open-ended task types that carry a groundedness score, matching the configurations in Table 2. The weighted average of these three columns (28:28:27) reproduces each configuration's aggregate Groundedness figure in Table 2 exactly.

The three columns behave differently enough that the aggregate figure in Table 2 masks large differences. **Path traversal** is grounded almost regardless of configuration (0.9097–0.9877, an 8-point range): even the weakest configuration fabricates on only 1–2 of 27 items, so this column's contribution to the aggregate is close to a constant offset rather than a signal the ablation moves. **Counterfactual** moves the most (0.1250–0.3095, an 18-point range) and drives most of the aggregate's configuration-to-configuration change: Full (0.3095) and the RL-trained configurations generally sit well above SFT+ocbook (0.1250), and within the GRPO variants, removing CoCG costs the most (0.2321, -0.0774 vs. Full) while removing ROC costs almost nothing (0.2976, -0.0119 vs. Full). On this small single-run sample, CoCG appears to matter more than ROC for keeping counterfactual answers grounded. **Disease reasoning** is the flattest and lowest of the three among GRPO variants (0.0714–0.1429) and is the one column where the plain SFT-without-ocbook baseline (0.2143) is not beaten by any GRPO ablation and only matched by Full (0.2143): no combination of ROC, GRAD, or CoCG improves on it over that simple baseline at this sample size, consistent with the persistent `disease_reasoning` accuracy weakness already noted above.

C.6 A DIAGNOSED AND FIXED BUG: THE KL REFERENCE REVERTED TRAINED MODALITY PROJECTIONS TO RANDOM INITIALIZATION

A second bug affected the KL reference computation (Eq. 15). It is fixed in the released method (Section 3.3).

The bug. Computing the Schulman K3 reference for Eq. 15 requires a version of the policy with LoRA's contribution removed. An earlier implementation did this with a call that disables *every* trainable component at once (LoRA adapters and `modules_to_save`), not LoRA alone. Because the modality projections and co-trained encoders are trained exclusively through `modules_to_save` (Section 3.1), disabling them for the reference pass did not recover a meaningful "pre-LoRA" checkpoint; it reverted them to random initialization, since they have no pretrained state independent of this training run to fall back to. Every `<modality_embed>` position in the reference sequence was therefore populated with a projection of encoder output through effectively random weights, rather than through the current trained projection.

The consequence, measured. $\ell^{\text{ref}}_{i,t}$ at modality-embed positions bore no relationship to a real trained model's log-probabilities, so $\delta_{i,t}$ at those positions could be arbitrarily large. The measured effect on released runs was loss spikes as large as 7.37×10^5, several orders of magnitude above any value a correctly computed KL term should produce. Mean reward over the same period showed no indication of a problem: reward is computed from the policy's own generations under its own weights, which were never corrupted by this bug, so a reward-only monitoring dashboard could not have surfaced it. Only inspecting the loss or KL term directly reveals the pathology.

The fix. The reference computation now disables only the LoRA adapter layers, leaving `modules_to_save` at its current trained values; the corrected reference is a real "policy with the LoRA update removed but current projections intact" rather than an internally inconsistent hybrid.

We now also log the KL term as its own metric rather than folding it into the aggregate loss, so a recurrence would be visible in monitoring. (The $[-10, 10]$ clamp in Eq. 14 applies to the importance ratio, not to $\delta_{i,t}$, and does not guard against this bug.)

Implication. This bug is an instance of a general pattern for any architecture combining LoRA with additional full-precision trainable components: "disable the adapter" is ambiguous whenever more than one class of trainable parameter exists, and the ambiguous version will silently pick a default that may not correspond to any meaningful reference model. Aggregate reward or accuracy curves are not a reliable signal for detecting this class of bug, since they depend only on the policy's forward pass, not the reference computation used solely for the KL term; a bug isolated to the reference pass can coexist indefinitely with reward curves that look entirely healthy.

C.7 OCSD AND ROC'S GOLD VIEW

What OCSD does. Observation-Calibrated Self-Distillation (Yang et al., 2026) contrasts two *structurally matched* replay views: Full (real future observation present) and Observation-Ablated (same scaffold, observation removed). It takes the difference as an "observation residual" and uses it to modulate token-level GRPO updates. The paper's central point is that a naive present-versus-absent contrast mixes the information in the observation with the cost of *presenting* it: scores shift partly because new text was inserted, not only because of what the text says.

The local view. Our local view (Eq. 7) moves each hop's environment narration to just before its own action, and its ablation substitutes a fixed placeholder for every hop, so the residual reflects narration content rather than shape. This is the matched design OCSD recommends, so the local view needs no change.

The gold view. The gold view is currently ablated as "[Gold path: not provided]" (Appendix C.3), a shorter string than the revealed chain. It therefore has the confound OCSD describes: part of $q_j^{(gold)}$ reflects the chain's content, and part reflects the cost of inserting a chain-shaped block of text. Because gold-view credit enters ROC's aggregate g_j (Eq. 8), the confound reaches the policy gradient and, through Δ_e^B (Eq. 19), CoCG's confidence updates. This may explain why the gold view is the mis-signed component in Table 3.

Proposed matched ablation. A matched gold contrast, analogous to the local view, would use:

- $\ell_j^{(gold),rev}$ = log-prob under the true gold chain inserted at the standard position (current behavior)
- $\ell_j^{(gold),abl}$ = log-prob under a **structurally identical but content-scrambled gold chain** (same token count and hop-tag formatting, with source, target, and relation fields replaced by plausible but wrong values) inserted at the same position, *not* simply omitted

$q_j^{(gold)}$ would then be a calibrated residual rather than a revealed-versus-absent contrast. This requires changing only the gold-view ablation string; we leave the experiment to future work.

C.8 AVSD AND ROC'S AGGREGATION RULE

What AVSD does. Given M privileged views, AVSD (Nguyen et al., 2026) computes two pooled targets: a *consensus* target

$$A_t^G(v) = \frac{1}{M} \sum_m \Delta_t^{(m)}(v)$$

(the average signed per-view update, which is conservative and fires only when views agree in direction) and an *arithmetic marginal* target

$$q_t^A(v) = \frac{1}{M} \sum_m q_t^{(m)}(v)$$

(union support, which fires if even one view strongly favors a token and is noisy on its own). It then gates a *residual* (arithmetic minus geometric) back onto the consensus signal, admitting the residual only when a two-component gate confirms views are (a) aligned in direction and (b) proportionate in magnitude, so a single outlier view cannot flip the direction the consensus has set.

Relation to Eq. 8. Our current aggregation is

$$\gamma_j = \mathrm{clip}(1 - \mathrm{Var}_j/\tau_{\mathrm{ROC}},\ 0,\ 1)\,, \qquad g_j = \bar{q}_j \cdot \gamma_j,$$

which averages the four views and scales the result down by their disagreement. This is a coarser version of what AVSD does: we have one blended signal (mean) and one blended discount (variance-derived scalar), where AVSD keeps two distinct signals (conservative consensus vs. permissive union) and a *vector* gate rather than a scalar one. The difference matters when one view is strongly informative and the other three are weak or neutral (a disabled view is passed as zero). Our variance term then *penalizes* the strong view, even though the disagreement is expected. AVSD's marginal target is designed for this case: strong support from one view is not washed out by weak support from the others, as long as it agrees with the sign of the consensus.

Possible variant. An AVSD-style replacement for Eq. 8 would be

$$\begin{aligned}
\bar{q}_j^{G} &= \tfrac{1}{4}\textstyle\sum_v q_j^{(v)} \qquad \text{(current mean, kept as the conservative signal)},\\
\bar{q}_j^{A} &= \max_v q_j^{(v)} \cdot \mathbb{1}[\text{signs agree}],\\
g_j &= \bar{q}_j^{G} + \mathrm{gate}(\mathrm{align}_j, \mathrm{magnitude}_j) \cdot \big(\bar{q}_j^{A} - \bar{q}_j^{G}\big).
\end{aligned}$$

This changes what "agreement" means, from a variance penalty to directional consensus plus a gated residual, so we treat it as an ablation for future work (does it change which task types benefit from ROC?) rather than a substitution.

C.9 SEED AND CLOSED-BOOK CREDIT

What SEED does. SEED (Wu et al., 2026) has the *same policy checkpoint* both act (collect trajectories) and analyze (extract a natural-language "hindsight skill", such as a reusable workflow, decisive observation, or failure-avoidance rule, from its own completed trajectory). It then re-scores the same sampled actions under ordinary vs. skill-augmented context, and the resulting probability shift is a dense, token-level distillation signal, jointly optimized with GRPO. SEED uses no external teacher and no gold reference; the privileged information is generated by the same model looking back at what it just did.

The closed-book gap. A closed-book rollout has no injected segments, so its revealed and ablated local sequences are identical and $q^{(\mathrm{local})} \equiv 0$ without special-casing. It also touches no edges, so $q^{(\mathrm{conf})} = 0$. For an **ocbook-routed closed-book rollout that answers correctly** (Section 3.2), ROC therefore has only two usable views: the global view, which carries the trajectory-level correctness bit, and the gold view, which scores the answer against the item's gold chain even though the rollout never retrieved it. Two of the four views are unavailable for the rollouts ocbook is designed to reward.

Possible extension. A SEED-style fifth view, $q^{(\mathrm{skill})}$, would use the same policy, post hoc, to extract a short natural-language justification of *why* a closed-book answer was correct (what prior knowledge or pattern licensed it), then re-score the original tokens under that self-generated justification vs. an ablated/generic version of it. This would give ocbook's correct-but-unverified tier (Eq. 13, κ_{uv}) a per-token credit signal instead of relying only on the trajectory-level correctness bit. Unlike the OCSD and AVSD changes, it is additive and leaves ROC's four existing views untouched. We propose it as an extension for closed-book credit, not a replacement for any existing view.

The gap matters most for multi-hop items. Closed-book warm-start trajectories exist only for single-hop items (Section 3.2), so the model has no warm-start precedent for a correct closed-book answer to a multi-hop question. On those rollouts, the global view adds nothing beyond the trajectory-level reward, leaving the gold view as ROC's only per-token signal, and Table 3 suggests that view is the least reliable. A $q^{(\mathrm{skill})}$ view is the most direct way we see to add credit for this case.

C.10 SUMMARY

The AVSD variant changes the same equation (Eq. 8) as the centering fix, but for a different reason. Centering removed a positive bias, while AVSD's split would change how much a single strongly informative view can outweigh the rest. AVSD should therefore be evaluated on top of the centered

aggregation, not instead of it. None of the three literature-motivated items changes CoCG's update rule (Eqs. 18–9), but the OCSD and AVSD changes both alter the g_τ that Δ_e^B (Eq. 19) consumes, so either would require re-running CoCG's confidence-trajectory checks. The storage-range row differs from the other five: the widening makes correctness-correlated movement *possible*, but the one measurement so far ($+0.0055$) cannot yet show that it helps.

Source	Maps onto	Action
Measured (this appendix)	ROC's aggregation (Eq. 8) was positive by construction	**Fixed.** Group-centering applied (Section 3.4); bias removed on the traced item ($q^{(\text{conf})}$: $+0.975 \rightarrow +0.005$) and in aggregate ($+0.783$ mean removed); separation remains weak (Table 3)
Measured (this appendix)	The KL reference reverted trained modality projections to random init (Eq. 15)	**Fixed.** The reference now disables only LoRA, not `modules_to_save`; resolved loss spikes up to 7.37×10^5 that mean-reward monitoring alone never surfaced (Section 3.3)
Measured (this appendix)	CoCG's confidence storage could not move ceiling-pinned edges (Eq. 9)	**Fix applied; effect on correctness unverified.** Storage widened to $[-2, +2]$ (Section 3.5); the one available signal-quality measurement ($+0.0055$ separation) is not distinguishable from noise and has flipped sign across runs
OCSD	ROC's gold-view ablation	**Future work.** Matched-ablation (scrambled-content) gold contrast, same pattern as the existing local view
AVSD	ROC's aggregation (Eq. 8)	**Future ablation.** AVSD-style consensus+gated-residual vs. current mean+variance-discount
SEED	Closed-book credit gap in ROC/ocbook	**Proposed extension.** Additive fifth view for the rollout class where ROC currently has no informative signal

Table 15: Summary of implementation implications from the three measured findings and from SEED, AVSD, and OCSD.

D A Worked Trace of the Full System on One Item

Appendix C.3 traces ROC's computation alone, on this same item, to diagnose one specific failure mode. This appendix traces the full system (SFT demonstration, inference, reward, advantage, ROC, and CoCG) end to end on item `co_path_00004`, for a different rollout than the one in Appendix C.3, using the post-centering (fixed) ROC configuration throughout. The per-view revealed/ablated construction is not repeated here; see Appendix C.3 for that detail.

The item. *What is the shortest molecular path from SLC18A2 to APOE?* Gold chain: `SLC18A2` $\xrightarrow{\text{conf } 0.98}$ `SNCA` $\xrightarrow{\text{conf } 0.995}$ `APOE` , giving $PC(C) = \sqrt{0.98 \times 0.995} = 0.9875$ (Eq. 2).

Stage 1: what SFT demonstrates. The gold trajectory is `Think` $\rightarrow$ one `List-Neighbors` $\rightarrow$ one `Query-KG` on the question's own endpoints (`SLC18A2`$\rightarrow$`APOE`, not the intermediate `SNCA`, following the warm-start convention of querying only the question's endpoints for multi-hop `path_traversal` items) $\rightarrow$ nine `Select-Encoder`/`Extract-Embedding` pairs across the `sequence`, `structural`, and `transcriptomic` families $\rightarrow$ `Answer`. Most of the roughly 30 demonstrated lines are modality calls; the closed-book variant of the same item, when eligible, is two lines. This asymmetry matters for training: absent other correction, the much shorter closed-book-style trajectory has a structural advantage in likelihood terms simply by being shorter, independent of correctness.

Inference, including an illustrative bug. On the traced rollout, `List-Neighbors` correctly surfaces `SNCA` as `SLC18A2`'s top neighbor by confidence. The subsequent `Query-KG` call, however, is issued with a malformed type field (`end_type='GENe'` rather than `GENE`), and node resolution at the time of this trace returns no path found for the malformed query, despite a real path

existing. The released configuration resolves nodes with tolerance for unambiguous near-misses (Appendix C.1); this trace illustrates the failure mode that motivated that tolerance. The rollout still answers correctly only because `List-Neighbors` had already surfaced the correct intermediate; on other `path_traversal` items where discovery does not independently succeed, the same class of failure produces a wrong or incomplete answer.

Stage 2: reward and advantage. Reward for this rollout is $R = 0.9875 \times 1.2 = 1.185$, where $1.2 = 1 + w_{\text{disc}}$ reflects the discovery bonus for a non-empty `List-Neighbors` call (Eq. 4). Across a sampled group of 8 rollouts on this item with rewards $[1.185, 1.185, 0.05, 0, 1.185, 0.05, 0, 0]$, group-relative normalization (Eq. 5) gives advantages $[+1.21, +1.21, -0.67, -0.76, +1.21, -0.67, -0.76, -0.76]$. The two intermediate values (0.05) come from sibling rollouts that earned only small partial credit or a closed-book bonus (Section 3.2, Eq. 13), and the zeros from rollouts that earned neither. This spread is what produces a gradient: if every rollout in the group scored zero, every advantage would be zero and the item would contribute no gradient.

Stage 3: ROC, post-centering. For the span corresponding to the `Query-KG` action, the four views (Eq. 7–17) return $q^{(\text{local})} = +0.45$, $q^{(\text{global})} = +0.15$, $q^{(\text{gold})} = -0.31$, and $q^{(\text{conf})} = +0.005$ after group-centering (versus $+0.975$ uncentered, Appendix C.3). The gate (Eq. 8, $\tau_{\text{ROC}} = 0.25$) combines these into a small positive $g \approx +0.052$ (mean 0.074, $\gamma = 0.70$), which nudges the ROC-modulated advantage (Eq. 6) from its base value up slightly. The gold view's negative value shows the kind of disagreement an uncentered signal could not express: one view dissents even though the rollout is correct overall.

Stage 4: CoCG update. The small positive g drives the ROC-based term Δ_e^B (Eq. 19), which Eq. 9 combines with Δ_e^A and writes back onto both edges of the gold chain. The `SLC18A2–SNCA` edge moves from 0.98 to 0.980479, and the change is visible through edge_probability. The `SNCA–APOE` edge, at 0.995, has little room left below the probability ceiling, so repeated updates will soon stop changing its ranking in `List-Neighbors`. This edge illustrates the limitation in Section 3.5: the edges a model relies on to answer correctly (high-confidence, frequently corroborated) are the ones most likely to sit at or near the ceiling. The ceiling rate (21.9% of all edges, including all 28,705 `VARIANT_IN_GENE` edges) is therefore not a tail effect on rarely used edges. Whether the wider storage range changes such edges' trajectories over many updates remains open (Section 3.5).