RareAgent: Self-Evolving Reasoning for Drug Repurposing in Rare Diseases

Anonymous ACL submission

Abstract

Computational drug repurposing for rare diseases is especially challenging when no prior associations exist between drugs and target diseases. Therefore, knowledge graph completion and message-passing GNNs have little reliable signal to learn and propagate, resulting in poor performance. We present RareAgent, a self-evolving multi-agent system that reframes this task from passive pattern recognition to active evidence-seeking reasoning. RareAgent organizes task-specific adversarial debates in which agents dynamically construct evidence graphs from diverse perspectives to support, refute, or entail hypotheses. The reasoning strategies are analyzed *post hoc* in a self-evolutionary loop, producing textual feedback that refines agent policies, while successful reasoning paths are distilled into transferable heuristics to accelerate future investigations. Comprehensive evaluations reveal that RareAgent improves the indication AUPRC by 18.1% over reasoning baselines and provides a transparent reasoning chain consistent with clinical evidence.

1 Introduction

Drug repurposing, the discovery of new therapeutic uses for approved drugs, is pivotal for accelerating drug development while markedly reducing costs and clinical risks (Liu et al., 2021; Huang et al., 2024; de la Fuente et al., 2025). Its value is particularly pronounced in rare diseases, where fewer than 10% of the thousands of identified rare diseases currently have approved therapies, leaving millions of patients with unmet medical needs (Greene et al., 2023; Cummings et al., 2025; Cipriani et al., 2025).

Recent computational work on drug repurposing typically organizes biomedical evidence as knowledge graphs (KGs) (Himmelstein et al., 2017; Ioannidis et al., 2020; Zhang et al., 2021; Chandak et al., 2023) and applies graph neural networks (GNNs) to infer new drug-disease associations (Zitnik et al., 2018; Li et al., 2024). Yet their efficacy can be limited by the scarcity of curated **drug-disease** associations, especially when both the candidate drugs and the target diseases (*e.g.*, rare diseases) lack reliable knowledge. In practice, the *drug side* is inherently *sparse* as a drug typically has only a few officially recognized indications, while the *disease side* becomes a *degree-zero* node in the case where our target is a rare disease for which no drug-disease associations have been established within the current body of human knowledge. We refer to this as the **sparse-zero bipartite setting**, which defines the key challenge central to drug repurposing for rare diseases.

Under this condition, the reliance of KGs and GNNs on message passing between neighboring nodes becomes a critical liability: associations from the target disease to a candidate drug are too weak to carry signals, limiting the model's generalization capacity. One common response is to densify knowledge graphs with similarity or co-mention edges to sustain message passing (Liu et al., 2020; Li et al., 2022; Chandak et al., 2023), but spurious links inevitably dilute genuine signals and obscure truly informative evidence. As a result, it *masks* the underlying scarcity and *recasts* rare-disease repurposing as a conventional GNN cold-start problem, and models trained on these inflated graphs learn unreliable representations and produce many implausible candidate associations (Jiang et al., 2022; Yu et al., 2024; Ma et al., 2024), leaving the core challenge unresolved.

In this paper, we advocate a paradigm shift from passive pattern recognition on static graphs to active, evidence-seeking reasoning that simulates the scientific discovery process. We introduce **RareAgent**, a self-evolving research framework designed for this challenge. **RareAgent** casts drug repurposing for rare diseases as a dynamic investigation conducted by multiple AI agents with complementary roles: a **Principal Investigator (PI)** provides strategic oversight, an **Explorer** formu-

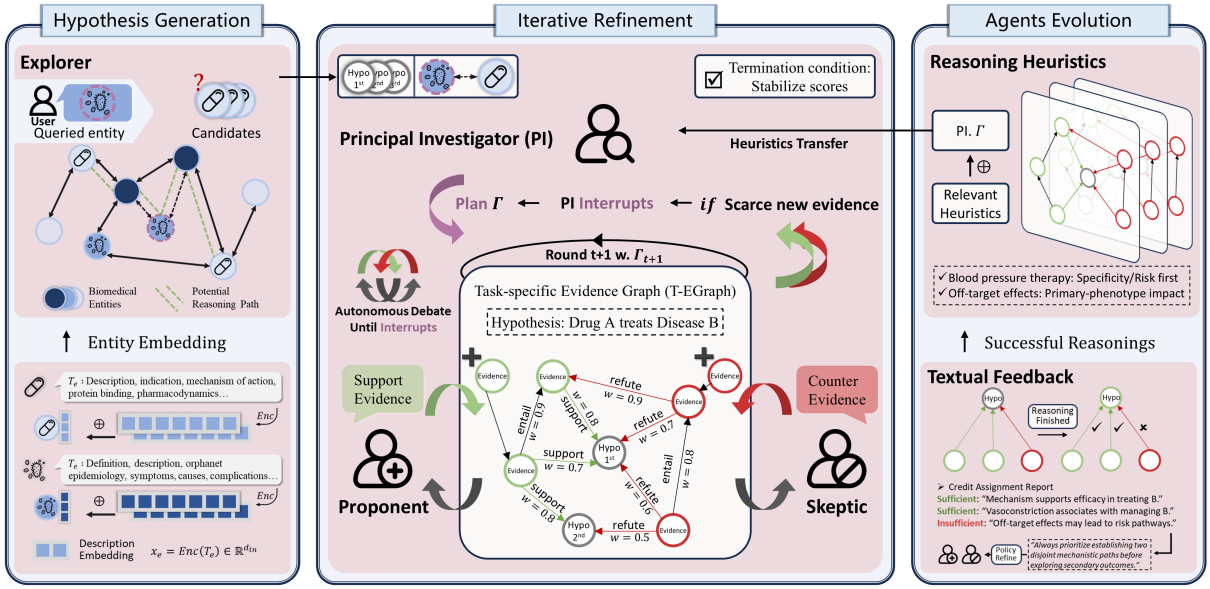

Figure 1: **RareAgent** overview. Explorer seeds a task-specific evidence graph (T-EGraph) with initial candidates based on the user query.[1] The Proponent and Skeptic freely add and challenge evidence on a shared whiteboard, while the PI monitors, interrupts, and revises. Iteration yields ranked candidates with auditable trails. Finally, PI analyzes reasoning paths to generate feedback and distill heuristics, continuously improving the system's policies.

lates initial hypotheses, and **Proponent** and **Skeptic** agents engage in an adversarial debate to gather and evaluate evidence. As shown in Figure 1, the overall process builds on three mechanisms:

- **Evidence Reasoning as Dynamic Graph Learning.** To address the lack of existing relational paths, agents maintain a **Dynamic Shared Whiteboard** instantiated as a task-specific evidence graph (**T-EGraph**). Nodes represent the hypotheses proposed by the **Explorer**, as well as the support and counter evidence collected by the **Proponent** and **Skeptic**. Edges encode support, refutation, and entailment. Each action (hypothesize, gather, argue) updates the graph. We formulate reasoning by selecting the next action, conditioned on the current graph state, to assemble a complete evidence chain efficiently.
- **Reasoning Policy Self-Evolution via Textual Feedback.** In the absence of ground truth reasoning paths to learn from, **RareAgent** introduces a self-evolutionary loop driven by natural language feedback. After each T-EGraph construction, the **PI** issues a textual **Credit Assignment Report** that marks pivotal versus unproductive evidences/steps, automatically revises agent prompts, and steers future actions toward high-yield behaviors.
- **Distilling Reasoning Trajectories into Transferable Heuristics.** To avoid relearning common reasoning patterns for each new zero-shot case, **RareAgent** accumulates scientific intuition. **RareAgent** meta-analyzes trajectories to extract recurring successful reasoning patterns, distills them into transferable heuristics, and stores them in a shared library of policy priors. For novel but related tasks, the **PI** initializes prompts with relevant heuristics, which narrows the search space and accelerates convergence.

RareAgent actively constructs plausible drug-disease hypotheses supported by a verifiable, auditable evidence trail and full reasoning path. The final output is not just a prediction but a comprehensive research report detailing the recommended pairs, the assembled body of evidence, and the complete reasoning steps leading to each conclusion, thus improving reliability and interpretability.

Our contributions can be summarized as follows:

- We identify ***sparse-zero bipartite setting*** as a critical yet underexplored challenge in drug repurposing for rare diseases, characterized by the absence of prior knowledge on both the drug and the target disease.
- We propose **RareAgent**, a multi-agent research framework where agents collaboratively debate to derive and score hypotheses, grounding their reasoning in a dynamical evidence graph.
- We introduce a self-evolutionary learning paradigm that improves reasoning policies without direct supervision. We use an automated feedback loop of textual critiques to refine agent

[1] "Disease→Drug" in Explorer is just an example; either disease or drug can be the query condition.

behaviors and distill transferable heuristics from successful reasoning trajectories.

- Experimental results demonstrate **RareAgent** outperforms domain-specific GNN baselines and large language models on drug repurposing tasks for rare diseases, with verifiable evidence for its predictions. Specifically, RareAgent achieves state-of-the-art results on the rare disease drug repurposing task (0.438 AUPRC / 0.662 AUROC) and on the BioHopR benchmark (17.68% 2-hop precision).

2 Related Work

Graph-based Methods for Drug Repurposing. Graph neural networks (GNNs) over biomedical knowledge graphs (KGs) have become central to computational drug repurposing (Perdomo-Quinteiro and Belmonte-Hernández, 2024; Wei et al., 2025). Early systems such as Decagon (Zitnik et al., 2018) and DTD-GNN (Li et al., 2024) established modeling paradigms for biomedical graphs, while more recent approaches like TxGNN (Huang et al., 2024) improve generalization to rare diseases via metric learning and neighborhood aggregation. Despite these advances, performance remains coupled to graph structure and edge quality, and apparent density in KGs can be misleading. For example, in PrimeKG, 94.7% of edges related to diseases and drugs capture underspecified drug-drug associations (Chandak et al., 2023), diluting therapeutically meaningful signals and revealing limitations of GNNs when dense, high-quality relational paths are scarce (Jiang et al., 2022; Huang and Li, 2023; Ma et al., 2024; Jamadandi et al., 2024).

LLM-powered Agents for Scientific Discovery. LLM-based agents integrate planning, tool use, and automated experimentation for scientific workflows. Coscientist demonstrates autonomous robotic experiments (Boiko et al., 2023), while ChemCrow tailors domain tools to chemistry and drug discovery. Multi-agent pipelines such as the AI Scientist series report fully AI-generated manuscripts (Lu et al., 2024; Yamada et al., 2025), and systems like PaperQA enable retrieval-augmented question answering over scientific literature (M. Bran et al., 2024). However, these agents typically assume a well-scoped problem or some initial evidence to ground reasoning. Extending them to the **sparse-zero bipartite setting** where such grounding is absent remains a key open challenge.

Algorithm 1 RareAgent: Iterative Refinement

Input: Initial Hypotheses $\{v_{h_i}\}$, Initial Graph $G_E^{(0)}$, Max rounds $T_{\max}$, Stop threshold δ_{stop}
$\Gamma_0 \leftarrow$ InitialPlan()
for $t = 0, 1, \ldots, T_{\max} - 1$ **do**
  $G_E^{(t+1)} = \mathcal{M}(G_E^{(t)}, \Delta G_{E,\text{pro}}^{(t)}, \Delta G_{E,\text{ske}}^{(t)})$ ▷ Inner runs until PI interrupts (graph change $\Delta G_E < \epsilon$)
  for all v_{h_i} **do**
    $s_{t+1}(v_{h_i}) \leftarrow$ PI.score$(G_E^{(t+1)}, v_{h_i})$
  end for
  if $\max_i |s_{t+1}(v_{h_i}) - s_t(v_{h_i})| \leq \delta_{\text{stop}}$ and $t > 0$ **then**
    break
  end if
  $\Gamma_{t+1} \leftarrow$ PI.revise$(G_E^{(t+1)}, \{s_{t+1}(v_{h_i})\})$
end for
Output: Ranked list $\langle (c_i, G_E^{(T)}) \rangle$ sorted by final scores $s_T(v_{h_i})$.

Self-improving and Evolutionary AI Systems. Work on self-improvement provides mechanisms for refining reasoning policies over time. Reflexion leverages verbalized reflection for reinforcement (Shinn et al., 2023), Self-Refine operationalizes a generate-critique-refine loop (Madaan et al., 2023), and evolutionary strategies such as PromptBreeder optimize prompts via self-referential mutation (Fernando et al., 2023). Complementary views frame reasoning as structured search, externalized in Tree-of-Thoughts (Yao et al., 2023) and Graph-of-Thoughts (Besta et al., 2024). TextGrad formalizes language feedback as *textual gradients* for systematic refinement (Yuksekgonul et al., 2025). A persistent frontier is to develop systems that autonomously improve from sparse, noisy, and unstructured feedback, especially in domains lacking explicit reward signals, where domain-specific intuition must be accrued gradually.

3 Methodology

3.1 Problem Formulation

A biomedical knowledge graph can be formalized as $G_K = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ is the set of entities (e.g., drugs, diseases), $\mathcal{R}$ is the set of relation types, and $\mathcal{T}$ is the set of known triplets (h, r, t). A drug repurposing query is a tuple $Q = (q, r_{\text{target}})$, where q is the query entity, either a drug for which we seek potential therapeutic diseases or a disease for which we seek candidate drugs, and r_{target} is the tar-

get relation, such as *indication* or *contraindication*. The goal is to identify a set of candidate entities $\mathcal{C} \subset \mathcal{E}$ such that the triplet $(q, r_{\text{target}}, c)$ for each $c \in \mathcal{C}$ is a valid but unobserved link.

The challenge arises when attempting to infer a relationship (q, r_{type}, c) under the **sparse-zero bipartite setting**, where both endpoints lack meaningful prior evidence in the graph. For any entity $e \in \mathcal{E}$, its neighborhood is $\mathcal{N}(e) = \{v \in \mathcal{E} \mid (e, r, v) \in \mathcal{T} \vee (v, r, e) \in \mathcal{T}\}$, and the sparse-zero setting for a candidate link (q, r_{type}, c) is satisfied when both neighborhoods are sparse or empty:

$$|\mathcal{N}(q)| \leq \epsilon_1 \quad \wedge \quad |\mathcal{N}(c)| \leq \epsilon_2, \tag{1}$$

where ϵ_1, ϵ_2 are typically small non-negative integers (e.g., 0 or 1). This setting is fundamentally more challenging than conventional cold-start problems, which typically involve only one entity with sparse connections, thereby limiting the generalization ability of GNN-based models that rely on message-passing mechanisms.

Our objective under this setting is not merely to predict a score, but to generate a ranked list of candidate pairs, each accompanied by a verifiable body of evidence. The final output for a query Q is a ranked list $\mathcal{C}^*$:

$$\mathcal{C}^* = \langle (c_1, G_{E,1}), \ldots, (c_k, G_{E,k}) \rangle \tag{2}$$

where c_i is a candidate entity and $G_{E,i}$ is a task-specific evidence graph (T-EGraph) that provides an auditable reasoning path justifying the proposed link $(q, r_{\text{target}}, c_i)$.

The overall objective is to learn a reasoning policy π that maps an input query to a ranked list of candidates and their evidence graphs: $\pi : (q, r_{\text{type}}) \mapsto \mathcal{C}^*$. The policy π is optimized to maximize both the relevance and accuracy of the candidates c_i, and the logical coherence and completeness of the supporting evidence graphs $G_{E,i}$.

3.2 RareAgent Framework

RareAgent operationalizes the scientific discovery process using a team of four AI agents with specialized roles, who collaborate on a shared whiteboard by constructing T-EGraph.

- **Principal Investigator (PI)**: The strategic orchestrator. The PI does not directly manipulate evidence but sets research goals, evaluates the state of the T-EGraph, identifies knowledge gaps, issues directives to other agents, and is responsible for self-evolutionary learning loops.
- **Explorer**: The hypothesis generator. The Explorer initiates the investigation by querying the external KG using GNN to propose an initial set of plausible drug-disease hypotheses.
- **Proponent**: The constructive advocate. The Proponent's goal is to build a coherent, mechanistic case for a given hypothesis. It searches for and adds supporting evidence to the T-EGraph, aiming to form complete causal chains (e.g., Drug $\rightarrow$ Target $\rightarrow$ Pathway $\rightarrow$ Phenotype).
- **Skeptic**: The adversarial challenger. The Skeptic's role is to challenge the current hypothesis by finding counter-evidence, identifying risks, and exposing logical fallacies. It adds refuting evidence, such as off-target effects or safety concerns, to the T-EGraph.

The overall workflow proceeds in three phases: (1) **Hypothesis Generation**, where the Explorer seeds the investigation; (2) **Iterative Refinement**, an autonomous adversarial debate where the Proponent and Skeptic build upon the T-EGraph under the PI's guidance; and (3) **Self-Evolution**, where the PI analyzes completed investigations to refine agent policies and distill reusable knowledge. All four agents are realized as LLM modules with role-specific policies, and the backbone is swappable and orthogonal to our contributions.

3.3 Hypothesis Generation

The **Explorer** surveys potential drug-disease connections and proposes a tractable set of initial hypotheses in the *sparse-zero* setting, where usable paths in G_K are absent. It derives semantic representations from biomedical ontologies and descriptions, and applies a relation-aware scoring module to rank candidates for a query $Q = (q, r_{\text{target}})$. We select the top-$k$ candidates to form the initial hypotheses and construct the initial T-EGraph $G_E^{(0)}$ for subsequent refinement. Implementation and modeling details are provided in Appendix A.1.

3.4 Iterative Refinement

Once initial hypotheses are proposed, the framework enters a refinement loop. This phase forms the core of **RareAgent**'s reasoning process, where the **Proponent** and **Skeptic** agents engage in an adversarial debate, dynamically constructing a T-EGraph under the strategic guidance of the **PI**.

Construction of the Task-specific Evidence Graph (T-EGraph). For a given query $Q = (q, r_{\text{type}})$, a T-EGraph is a directed typed graph

$G_E = (V_E, E_E, \tau, \rho)$, where $\tau : V_E \to \{\text{query}, \text{hypothesis}, \text{evidence}\}$ assigns node roles and $\rho : E_E \to \{\text{entails}, \text{supports}, \text{refutes}\}$ assigns edge semantics. Each edge carries an **evidence weight** $w \in [0, 1]$ assigned by the creating agent, representing the confidence in or strength of that relational claim.

Adversarial Debate of Graph Refinement Loop. At each round t, the system state is the current T-EGraph, $G_E^{(t)}$. The loop, detailed in Algorithm 1, consists of three steps:

(1) **Adversarial Debate (`Proponent/Skeptic: debate`)**: A round begins with the Proponent and Skeptic engaging in an autonomous debate. They *freely* add evidence to the T-EGraph, reacting to the current state of evidence and each other's arguments. The agents execute their assigned `graph_actions`. Their policies, π_{pro} and π_{ske}, map the current state and the directive to a graph update ΔG_E. This produces the next graph state:

$$G_E^{(t+1)} = \mathcal{M}(G_E^{(t)}, \Delta G_{E,\text{pro}}^{(t)}, \Delta G_{E,\text{ske}}^{(t)}) \quad (3)$$

where $\mathcal{M}$ is a graph manager command conducted by PI that merges updates and ensures consistency. Notably, this process is competitive and interactive rather than a rigid turn-taking exchange.

(2) **Supervised Interruption and Scoring (`PI: monitor & score`)**: The PI monitors the debate and interrupts the process if lack of substantive new evidence. Then PI evaluates the T-EGraph for each hypothesis. It computes a multi-faceted score $s_t(v_h)$ using an aggregation command $\mathcal{A}$ that combines structural graph properties. The command is guided by a set of **scoring weights** that are dynamically instantiated by the PI from the available evidence, which prioritize different aspects of the evidence (e.g., placing a higher weight on mechanistic coherence than on safety signals in early rounds).

$$s_t(v_h) = \mathcal{A}\left(\sum w_{\text{sup}}, \sum w_{\text{ref}}, C_{\text{mech}}, D_{\text{path}}, \ldots\right) \quad (4)$$

where $\sum w_{\text{sup}}$ and $\sum w_{\text{ref}}$ are the aggregated weights of supporting and refuting evidence, C_{mech} is a metric for mechanistic connectivity, and D_{path} measures the disjointness of evidence paths.

(3) **Strategic Revision (`PI: revise`)**: Based on the scores, the PI generates a strategic plan Γ_t, a set of structured directives (`graph_actions`) for the Proponent and Skeptic. An action might instruct the Proponent to "find a second, disjoint mechanistic path supporting H1" or task the Skeptic to "stress-test the risk of supine hypertension for H1".

Termination Condition. The iterative loop terminates when one of two conditions is met: (1) the hypothesis scores and rankings stabilize, i.e., the maximum change in any score falls below a predefined threshold δ_{stop}, or (2) a maximum number of refinement rounds, $T_{\max}$, is reached. This ensures that the process concludes when a consensus is approached or resources are exhausted.

3.5 Agents Evolution

RareAgent achieves long-horizon learning by distilling and reusing *reasoning patterns* extracted from completed trajectories.

Learning from Textual Feedback. After each complete investigation, the PI performs a final audit and generates a **Credit Assignment Report**, R_{final}. This is a structured natural language critique of the entire reasoning trajectory, identifying pivotal motifs (successful reasoning patterns), unproductive paths, and residual gaps. This report is then used to automatically refine the agents' policies, which are implicitly encoded in their instructions, via a refinement operation:

$$P_{\text{agent}}^{(k+1)} \leftarrow \mathcal{U}(P_{\text{agent}}^{(k)}, R_{\text{final}}^{(k)}) \quad (5)$$

where $P_{\text{agent}}^{(k)}$ is the instruction for an agent on task k, and $\mathcal{U}$ is an update command executed by PI that synthesizes the feedback into concrete improvements for the agent's instructions (e.g., adding *"Always prioritize establishing two disjoint mechanistic paths before exploring secondary outcomes."*).

Distilling and Transferring Reasoning Heuristics. To accumulate "scientific intuition" and avoid relearning common reasoning patterns, **RareAgent** distills successful strategies into a library of transferable heuristics.

(1) **Distillation**: The PI meta-analyzes a collection of high-quality Credit Assignment Reports to identify recurring successful patterns. It then abstracts them into concise, conditional rules, or **heuristics** (e.g., *"WHEN disease is neurogenic, THEN prioritize agents acting on α_1-receptors and check for supine hypertension risk."*). A new heuristic H_{new} is generated and added to a shared library, $\mathcal{L} \leftarrow \mathcal{L} \cup \{H_{\text{new}}\}$.

(2) **Transfer**: When a new query Q_{new} arrives, the PI retrieves the top-J most relevant heuristics from the library $\mathcal{L}$ and injects them into the initial prompts for the PI. This bootstrapping mechanism

Method	Indication				Contraindication			
	AUPRC↑	AUROC↑	P@10↑	R@10↑	AUPRC↑	AUROC↑	P@50↑	R@50↑
TxGNN	0.373	0.552	0.02	0.12	0.241	0.448	0.00	0.00
gpt-4o-mini	0.361	0.544	0.01	0.06	0.244	0.458	0.00	0.00
gpt-4o	0.421	0.594	0.03	0.19	0.283	0.495	0.0012	0.0588
o3-mini	0.376	0.554	0.02	0.14	0.269	0.490	0.00	0.00
gpt-4o-mini (w.Con)	0.361	0.524	0.01	0.11	0.245	0.436	0.00	0.00
gpt-4o (w.Con)	0.421	0.599	0.03	0.20	0.245	0.443	0.0012	0.0716
o3-mini (w.Con)	0.432	0.607	0.03	0.26	0.260	0.487	0.00	0.00
ToT	0.371	0.528	0.01	0.09	0.251	0.442	0.00	0.00
GoT	0.206	0.479	0.01	0.09	0.260	0.444	0.0012	0.0196
TextGrad	0.138	0.490	0.00	0.00	0.241	0.433	0.00	0.00
RareAgent	**0.438**	**0.662**	**0.04**	**0.27**	**0.289**	**0.497**	**0.0024**	**0.0735**

Table 1: **Rare-disease drug repurposing on the 100,000 Genomes Project.** We compare model predictions with real-world experimental outcomes on rare-disease cohorts. Metrics: *AUPRC/AUROC*; $P@k$, precision among the top-k predicted drugs per disease; $R@k$, recall among the top-k.

primes the agents with relevant prior knowledge, effectively narrowing the search space and accelerating convergence on new but related problems.

4 Experiments

In this section, we evaluate **RareAgent** through four research questions (RQs). First, we examine drug repurposing for rare diseases under the challenging sparse-zero bipartite setting (RQ1). Second, we assess its performance on general biomedical reasoning tasks (RQ2). Then, we perform ablation studies to quantify the contributions of key components (RQ3). Finally, we present a case study illustrating **RareAgent**'s ability to produce auditable and interpretable evidence trails (RQ4).

4.1 Datasets and implementation details

We evaluate RareAgent on three datasets: (1) 100,000 Genomes Project. We use the disease list from the Genomes Project to enhance clinical relevance. Diseases are from the project's *Current Rare Disease List v1.9.0*[2], and ground-truth drug-disease pairs are matched from clinical trials[3], 230 diseases and 74 drugs survive quality filters, yielding 1188 candidate pairs. (2) PrimeKG (Chandak et al., 2023). A large, multimodal biomedical KG integrating 20 curated sources (129,375 nodes across 10 types, 4M edges) with strong rare disease coverage and natural language node descriptions. (3) BioHopR (Kim et al., 2025). A PrimeKG-derived benchmark for multi-hop biomedical reasoning, we use it to assess generalization beyond repurposing and to compare against state-of-the-art baselines.

4.2 RQ1: Rare disease drug repurposing

We evaluate rare-disease cohorts from the Genomics England 100,000 Genomes Project, using registered interventional trials as the ground truth. We analyze RareAgent and compare against (i) GNN baseline, (ii) general-purpose LLMs, and (iii) reasoning-augmented frameworks. Baselines marked "w.Con" enable self-consistency (Wang et al., 2022), and all reasoning frameworks use the same base model (gpt-4o-mini) with **RareAgent** and are budget-matched in LLM call counts. Comprehensive results are in Table 1.
(1) **Indication.** RareAgent outperforms reasoning frameworks under identical computational constraints, yielding an 18.1% increase in AUPRC compared to ToT, highlighting the value of evolving, dynamic evidence graphs over scaffolded deliberation alone. (2) **Contraindication.** All methods exhibit near-zero top-K precision/recall, but RareAgent still outperforms general and reasoning baselines in ranking, underscoring the robustness of adversarial evidence aggregation when positive signals are rare. (3) Notably, despite using gpt-4o-mini as the base, RareAgent outperforms o3-mini (w.Con): +1.4% AUPRC, +9.1% AUROC on Indication. This indicates that structured evidence aggregation and self-evolution can outweigh raw model capacity.

To further investigate the cumulative impact of our self-evolution mechanism, we visualize RareAgent's performance across three distinct self-evolution rounds in Figure 2. The results reveal a

[2] https://www.genomicsengland.co.uk/initiatives/100000-genomes-project/rare-disease
[3] https://clinicaltrials.gov/

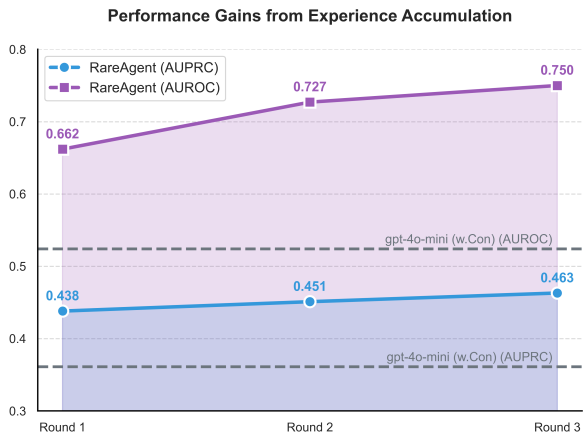

Figure 2: The cumulative effect of multiple self-evolution rounds on agent performance.

distinct and monotonic improvement, with AUPRC rising from 0.438 to 0.463 and AUROC increasing from 0.662 to 0.750. This steady upward trend validates that our self-evolutionary loop enables the system to effectively accumulate experience from prior investigations, thereby enhancing its long-horizon predictive accuracy.

4.3 RQ2: General biomedical reasoning

To assess whether the benefits observed on rare-disease repurposing transfer to broader biomedical reasoning, we evaluate on BioHopR, a PrimeKG-derived multi-hop benchmark. We follow the original evaluation protocol and compare against strong general and medical LLM baselines.

(1) **Main results are shown in Table 2**. RareAgent attains 33.87% on 1-hop precision and 17.68% on 2-hop precision. On 2-hop, RareAgent exceeds all baselines and indicates stronger multi-entity chain integration. On 1-hop, RareAgent using 4o-mini surpasses gpt-4o. These trends are consistent with RareAgent's design, which constructs and refines evidence graphs through adversarial debate and self-evolution, particularly beneficial when reasoning must traverse longer causal chains. (2) **Relation and answer-set analysis are in Figure 3**. Under BioHopR's answer-split evaluation, RareAgent achieves 56.49% / 37.22%(Single/Multi) on 1-hop, and 21.07% / 13.66% on 2-hop. We attribute this pattern to T-EGraph's emphasis on assembling mechanistically coherent paths and the PI's directives that encourage disjoint evidence routes, which together improve coverage in stricter Multi settings.

4.4 RQ3: Ablations of key components

We perform ablations to quantify the contribution of (i) the adversarial debate among Proponent/Skeptic, (ii) the self-evolution with textual feedback and transferable heuristics, and (iii) the termination condition. We maintain the RQ1 data, splits, and metrics. Results are shown in Table 3.

Method	Prec_HOP1 (%)	Prec_HOP2 (%)	BOTH_COR (%)	BOTH_WR (%)↓
Llama-3.1-8B	0.12	0.05	0.00	99.76
HuatuoGPT-o1-70B	0.16	0.00	0.00	99.93
HuatuoGPT-o1-8B	0.20	0.04	0.00	99.54
UltraMedical-8B	13.75	5.21	2.28	82.33
Llama-3.3-70B	25.58	9.58	4.94	68.33
Llama-3.1-70B	26.38	9.47	4.93	65.64
gpt-4o-mini	28.11	14.57	6.54	64.69
gpt-4o	32.88	14.57	7.86	57.96
RareAgent	**33.87**	**17.68**	**9.74**	**56.52**

Table 2: **BioHopR main results.** *Prec_HOP1/2*: precision on 1-hop/2-hop tasks; *BOTH_COR/WR*: fraction where both hops are correct/wrong.

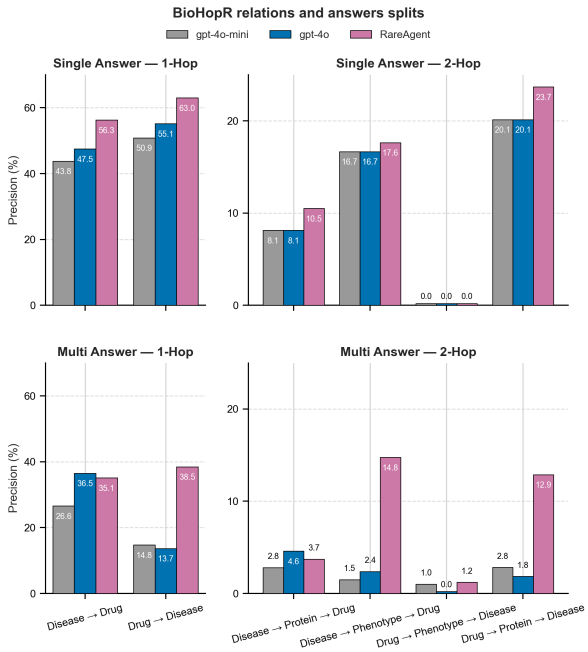

Figure 3: **BioHopR relations and answers splits.** *Single* asks for one correct answer per hop; *Multi* asks for all correct answers.

(1) **Self-evolution is decisive**. Removing textual feedback causes the most significant drops: AUPRC -31.1% and AUROC -16.1%. Disabling heuristic transfer also degrades performance. These results indicate that post-hoc critique and distilled priors drive most of the ranking gains. (2) **Adversarial debate matters**. Collapsing the debate to a single-agent setting yields AUPRC -7.2%, AUROC -3.9%, while removing only the Skeptic produces a clear hit on top-K (P@10 -14.3%, R@10 -10.7%). (3) **PI interruption/termination helps**. When the PI does not interrupt intermediate rounds (i.e., scores only at the end), ranking quality falls. This supports the design choice of convergence-based termination and supervised interruption to curb un-

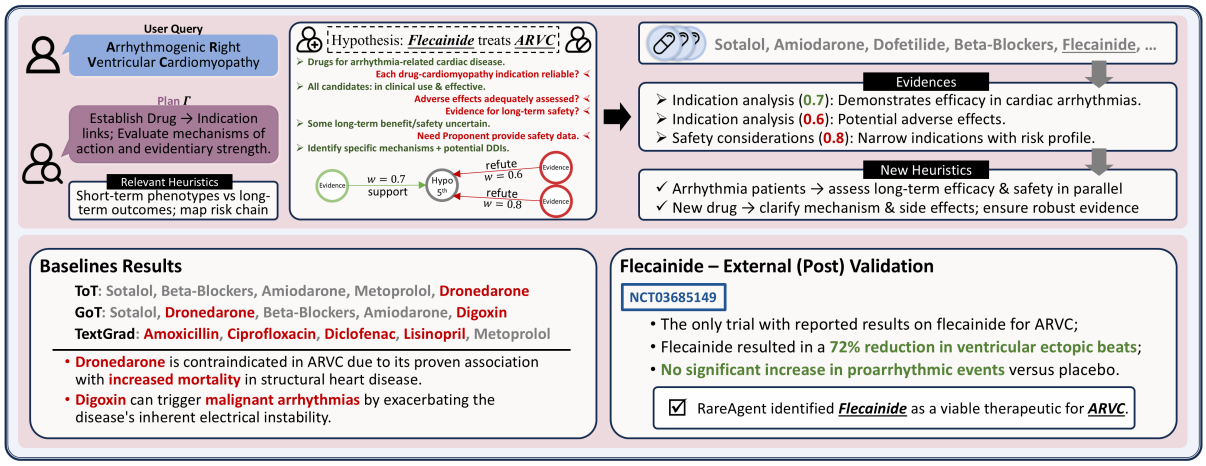

Figure 4: Case study. **RareAgent** identified ***Flecainide*** as a candidate therapy for ***ARVC***, whereas the baseline found no novel therapy and proposed drugs with severe safety risks. The sole flecainide-ARVC trial supports this conclusion, and the **RareAgent** did not access that trial record during its operation.

Configuration	AUPRC	AUROC	P@10	R@10
RareAgent	0.4381	0.6623	0.0412	0.2745
-Debate (single-agent)	0.4067	0.6364	0.0412	0.2353
-Skeptic (no critique)	0.4314	0.6570	0.0353	0.2451
-PI interrupts (final-only)	0.4145	0.6364	0.0412	0.2647
-Textual feedback	0.3018	0.5556	0.0412	0.2647
-Heuristic transfer	0.3877	0.6518	0.0353	0.2353

Table 3: **Ablation study.** We report the same evaluation protocol as in table 1.

productive exploration and consolidate strong evidence paths more effectively.

4.5 RQ4: Case Study: Auditable Evidence Trail for ARVC (Flecainide)

We use Arrhythmogenic Right Ventricular Cardiomyopathy (ARVC) as a case study to demonstrate RareAgent's auditable reasoning. ARVC's combination of rarity (approx. 1/5000 to 1/2000), lack of approved drugs, and high clinical severity (a leading cause of sudden cardiac death in the young) (Shah et al., 2025) makes it a representative stress test for drug repurposing. As shown in Figure 4, RareAgent elevates the hypothesis that *"Flecainide treats ARVC"* and subjects it to debate, where refute edges immediately contest the proposal based on safety risks. This process distills new safety-oriented heuristics that explicitly penalize the types of dangerous candidates proposed by baseline methods, such as the contraindicated *Dronedarone* and high-risk *Digoxin*. This mechanism allows RareAgent to prune unsafe reasoning paths, a crucial capability that distinguishes it from other reasoning frameworks. The only trial with reported results for this indication (NCT03685149)[4] confirms that flecainide results in a 72% reduction in ventricular ectopic beats, with no significant increase in proarrhythmic events, compared to placebo. Thus, RareAgent correctly identifies Flecainide as a viable therapeutic for ARVC, demonstrating coherence with real-world clinical evidence.

5 Conclusions

In this work, we introduced RareAgent, a self-evolving multi-agent framework that reframes drug repurposing for rare diseases from passive pattern recognition to active, evidence-seeking reasoning. Our approach is tailored for the sparse-zero bipartite setting, where the absence of prior drug-disease associations critically limits traditional methods. RareAgent's novelty lies in three mechanisms: an adversarial debate to autonomously construct a task-specific evidence graph; a self-evolutionary loop that refines agent policies via automated textual feedback; and the distillation of successful reasoning trajectories into transferable heuristics. Together, these features enable the system to ground hypotheses in auditable evidence trails and accumulate domain-specific intuition to accelerate future investigations. Comprehensive experiments demonstrate that RareAgent significantly outperforms both domain-specific GNN baselines and large language models in rare disease drug repurposing, with strong generalization in broader biomedical reasoning tasks. Ultimately, by producing verifiable and interpretable results, RareAgent offers a robust solution for complex scientific discovery in high-stakes, data-limited fields.

[4] https://clinicaltrials.gov/study/NCT03685149

Limitations

While RareAgent demonstrates a significant advancement in drug repurposing for rare diseases, we acknowledge several limitations that warrant future investigation. The performance of RareAgent is fundamentally dependent on the capabilities of the underlying LLMs that power the agents. Although our framework is designed to be model-agnostic, the quality of the generated evidence and reasoning is directly tied to the LLM's factuality and susceptibility to hallucination. Furthermore, the mechanism for distilling and transferring reasoning heuristics, while effective for accelerating convergence on related tasks, carries a risk of overfitting. Heuristics derived from one class of diseases may not generalize effectively to novel queries involving fundamentally different biological pathways. Future work should explore methods to assess the generalizability of these learned heuristics and adapt them to more distant problem domains.

References

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690. ArXiv:2308.09687 [cs].

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578. Publisher: Nature Publishing Group.

Payal Chandak, Kexin Huang, and Marinka Zitnik. 2023. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67. Publisher: Nature Publishing Group.

Valentina Cipriani, Letizia Vestito, Emma F. Magavern, Julius O. B. Jacobsen, Gavin Arno, Elijah R. Behr, Katherine A. Benson, Marta Bertoli, Detlef Bockenhauer, Michael R. Bowl, Kate Burley, Li F. Chan, Patrick Chinnery, Peter J. Conlon, Marcos A. Costa, Alice E. Davidson, Sally J. Dawson, Elhussein A. E. Elhassan, Sarah E. Flanagan, and 42 others. 2025. Rare disease gene association discovery in the 100,000 Genomes Project. *Nature*, pages 1–9. Publisher: Nature Publishing Group.

Jeffrey L. Cummings, Yadi Zhou, Alexandra Van Stone, Davis Cammann, Reina Tonegawa-Kuji, Jorge Fonseca, and Feixiong Cheng. 2025. Drug repurposing for Alzheimer's disease and other neurodegenerative disorders. *Nature Communications*, 16(1):1755. Publisher: Nature Publishing Group.

Jesus de la Fuente, Guillermo Serrano, Uxía Veleiro, Mikel Casals, Laura Vera, Marija Pizurica, Nuria Gómez-Cebrián, Leonor Puchades-Carrasco, Antonio Pineda-Lucena, Idoia Ochoa, Silve Vicent, Olivier Gevaert, and Mikel Hernaez. 2025. Towards a more inductive world for drug repurposing approaches. *Nature Machine Intelligence*, 7(3):495–508. Publisher: Nature Publishing Group.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-Referential Self-Improvement Via Prompt Evolution. *arXiv preprint*. ArXiv:2309.16797 [cs].

Daniel Greene, Daniela Pirri, Karen Frudd, Ege Sackey, Mohammed Al-Owain, Arnaud P. J. Giese, Khushnooda Ramzan, Sehar Riaz, Itaru Yamanaka, Nele Boeckx, Chantal Thys, Bruce D. Gelb, Paul Brennan, Verity Hartill, Julie Harvengt, Tomoki Kosho, Sahar Mansour, Mitsuo Masuno, Takako Ohata, and 14 others. 2023. Genetic association analysis of 77,539 genomes reveals rare disease etiologies. *Nature Medicine*, 29(3):679–688. Publisher: Nature Publishing Group.

Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio E Baranzini. 2017. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife*, 6:e26726. Publisher: eLife Sciences Publications, Ltd.

Kexin Huang, Payal Chandak, Qianwen Wang, Shreyas Havaldar, Akhil Vaid, Jure Leskovec, Girish N. Nadkarni, Benjamin S. Glicksberg, Nils Gehlenborg, and Marinka Zitnik. 2024. A foundation model for clinician-centered drug repurposing. *Nature Medicine*, 30(12):3601–3613. Publisher: Nature Publishing Group.

Rui Huang and Ping Li. 2023. Hub-hub connections matter: Improving edge dropout to relieve oversmoothing in graph neural networks. *Knowledge-Based Systems*, 270:110556.

Vassilis N. Ioannidis, Xiang Song, Saurav Manchanda, Mufei Li, Xiaoqin Pan, Da Zheng, Xia Ning, Xiangxiang Zeng, and George Karypis. 2020. Drkg - drug repurposing knowledge graph for covid-19. `https://github.com/gnn4dr/DRKG/`.

Adarsh Jamadandi, Celia Rubio-Madrigal, and Rebekka Burkholz. 2024. Spectral Graph Pruning Against Over-Squashing and Over-Smoothing. *Advances in Neural Information Processing Systems*, 37:10348–10379.

Chao Jiang, Victoria Ngo, Richard Chapman, Yue Yu, Hongfang Liu, Guoqian Jiang, and Nansu Zong.

2022. Deep Denoising of Raw Biomedical Knowledge Graph From COVID-19 Literature, LitCovid, and Pubtator: Framework Development and Validation. *Journal of Medical Internet Research*, 24(7):e38584.

Yunsoo Kim, Yusuf Abdulle, and Honghan Wu. 2025. BioHopR: A Benchmark for Multi-Hop, Multi-Answer Reasoning in Biomedical Domain. *arXiv preprint*. ArXiv:2505.22240 [cs].

Wenjun Li, Wanjun Ma, Mengyun Yang, and Xiwei Tang. 2024. Drug repurposing based on the DTD-GNN graph neural network: revealing the relationships among drugs, targets and diseases. *BMC Genomics*, 25(1):584.

Zongren Li, Qin Zhong, Jing Yang, Yongjie Duan, Wenjun Wang, Chengkun Wu, and Kunlun He. 2022. DeepKG: an end-to-end deep learning-based workflow for biomedical knowledge graph extraction, optimization and applications. *Bioinformatics*, 38(5):1477–1479.

Hui Liu, Wenhao Zhang, Bo Zou, Jinxian Wang, Yuanyuan Deng, and Lei Deng. 2020. DrugCombDB: a comprehensive database of drug combinations toward the discovery of combinatorial therapy. *Nucleic Acids Research*, 48(D1):D871–D881.

Ruoqi Liu, Lai Wei, and Ping Zhang. 2021. A deep learning framework for drug repurposing via emulating clinical trials on real-world patient data. *Nature Machine Intelligence*, 3(1):68–75. Publisher: Nature Publishing Group.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. *arXiv preprint*. ArXiv:2408.06292 [cs].

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535. Publisher: Nature Publishing Group.

Tengfei Ma, Yujie Chen, Wen Tao, Dashun Zheng, Xuan Lin, Patrick Cheong-Iao Pang, Yiping Liu, Yijun Wang, Longyue Wang, Bosheng Song, Xiangxiang Zeng, and Philip S. Yu. 2024. Learning to Denoise Biomedical Knowledge Graph for Robust Molecular Interaction Prediction. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):8682–8694.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative Refinement with Self-Feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Pablo Perdomo-Quinteiro and Alberto Belmonte-Hernández. 2024. Knowledge Graphs for drug repurposing: a review of databases and methods. *Briefings in Bioinformatics*, 25(6):bbae461.

Sandy N. Shah, Krishna Kishore Umapathi, Maria S. Horenstein, and Tony I. Oliver. 2025. Arrhythmogenic Right Ventricular Cardiomyopathy. In *StatPearls*. StatPearls Publishing, Treasure Island (FL).

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-Consistency Improves Chain of Thought Reasoning in Language Models.

Siqi Wei, Christo Sasi, Jelle Piepenbrock, Martijn A. Huynen, and Peter A.C. 'T Hoen. 2025. The use of knowledge graphs for drug repurposing: From classical machine learning algorithms to graph neural networks. *Computers in Biology and Medicine*, 196:110873.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. 2025. The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search. *arXiv preprint*. ArXiv:2504.08066 [cs].

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *Advances in Neural Information Processing Systems*, 36:11809–11822.

Weijian Yu, Jie Yang, and Dingqi Yang. 2024. Robust Link Prediction over Noisy Hyper-Relational Knowledge Graphs via Active Learning. In *Proceedings of the ACM Web Conference 2024*, pages 2282–2293, Singapore Singapore. ACM.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. 2025. Optimizing generative AI by backpropagating language model feedback. *Nature*, 639(8055):609–616. Publisher: Nature Publishing Group.

Rui Zhang, Dimitar Hristovski, Dalton Schutte, Andrej Kastrin, Marcelo Fiszman, and Halil Kilicoglu. 2021. Drug repurposing for COVID-19 via knowledge graph completion. *Journal of Biomedical Informatics*, 115:103696.

Marinka Zitnik, Monica Agrawal, and Jure Leskovec. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466.

A Appendix

A.1 Explorer: Encoding and Scoring Details

Entity Encoding. For every entity $e \in \mathcal{E}$, we construct a textual representation T_e by concatenating its ontology and descriptive text. A pre-trained sentence encoder $\text{Enc}(\cdot)$ produces a fixed feature embedding $\mathbf{x}_e = \text{Enc}(T_e) \in \mathbb{R}^{d_{in}}$, yielding a matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{E}| \times d_{in}}$. These features remain frozen in subsequent training.

Relation-aware Projection. To bridge static descriptions and relational contexts, we employ type-specific projections ($\text{Proj}_{\text{drug}}$, $\text{Proj}_{\text{disease}}$) that map $\mathbf{x}_e$ to a complex embedding $(\mathbf{h}_e^r, \mathbf{h}_e^i) \in \mathbb{R}^d \times \mathbb{R}^d$. Each relation $r \in \mathcal{R}$ is represented as $(\mathbf{r}^r, \mathbf{r}^i)$.

Scoring Function. Given a triplet (q, r, c), plausibility is estimated via a complex bilinear interaction:

$$\begin{aligned} f(q, r, c) = \langle \mathbf{h}_q^r, \mathbf{r}^r, \mathbf{h}_c^r \rangle + \langle \mathbf{h}_q^i, \mathbf{r}^r, \mathbf{h}_c^i \rangle \\ + \langle \mathbf{h}_q^r, \mathbf{r}^i, \mathbf{h}_c^i \rangle - \langle \mathbf{h}_q^i, \mathbf{r}^i, \mathbf{h}_c^r \rangle. \end{aligned} \tag{6}$$

Hypothesis Set Construction. For a query $Q = (q, r_{\text{target}})$, the Explorer scores all valid candidates c with $f(q, r_{\text{target}}, c)$ and selects the top-k to initialize the hypothesis set, which seeds the T-EGraph $G_E^{(0)}$ used in the refinement phase.

A.2 Core Agent Prompts

The following are *functional summaries* of the core agents' prompts used in RAREAGENT. They clarify roles, I/O contracts, and hard constraints.

A.2.1 PI (Principal Investigator)

Role Evaluate and rank hypotheses, plan/ revise graph-building & debate tasks, optionally request seed regeneration, and produce final reports/heuristics.

Modes

- `init`: Specify number of rounds, scoring weights, stopping criteria, and the overall plan.
- `score`: Score and rank each hypothesis using the current T-EGraph snapshot.
- `revise`: Issue graph actions to Proponent/Skeptic and set debate foci; optionally request Explorer to regenerate seeds (request only, no names).
- `report_and_evolve`: Produce a concise report, credit assignment, prompt patches, and distilled heuristics.

Common input (schematic)

```
{
  "mode": "init|score|revise|report_and_evolve",
  "query": {"entity":"string","relation":"string
    ↪ "},
  "hypotheses": [{"id":"H1","candidate":{"name":
    ↪ "string"}}],
  "tegraph_snapshot": {"nodes":[...],"edges"
    ↪ :[...],"round_index":1},
  "history": {"round":1,"last_scores":[...]},
  "thresholds": {"stop_delta":0.03,"
    ↪ saturation_ratio":0.65},
  "seed_context": {"target_type":"drug|disease",
    ↪ "seed_history":[["..."]]}
}
```

Typical outputs

- **`init`**:

```
{"plan":{"rounds":2,"stopping":{"
    ↪ delta_threshold":0.03}}}
```

- **`score`**:

```
{
  "scoring_summary":[{"hypothesis_id":"H1","
    ↪ score":0.68}],
  "ranking":["H1","H2"],
  "delta_since_last_round":0.04,
  "stop_decision":{"should_stop":false}
}
```

- **`revise`**:

```
{
  "revisions":[
    {
      "hypothesis_id":"H1",
      "graph_actions":[
        {"type":"add_mechanism_link","assignee
    ↪ ":"Proponent"}
      ],
      "debate_focus":["mechanistic closure &
    ↪ safety hotspots"]
    }
  ],
  "seed_request":{"should_regenerate":false,"
    ↪ reason":"example"}
}
```

- **`report_and_evolve`**:

```
{
  "final_recommendations":[{"hypothesis_id":"
    ↪ H1","score":0.75}],
  "prompt_patches":[{"role":"Proponent","patch
    ↪ ":"example"}]
}
```

Scoring signals Mechanism fit, class prior, PK/PD feasibility, indication plausibility, safety risk, structural graph bonus (disjoint paths/connectivity), conflict penalty.

Hard constraints

- PI does *not* directly modify the graph; it emits instructions for other agents.

- Score every input hypothesis. Seed regeneration can only be *requested*, not generated here.

A.2.2 Proponent (Support Builder)

Role Build *supportive* mechanistic chains and graph updates for a given hypothesis, or execute PI-assigned graph actions.

Modes

- `build_chain`: Complete supportive paths based on common sense and heuristics.
- `execute_actions`: Execute `graph_actions` (assignee=Proponent).

Input (schematic)

```
{
  "mode":"build_chain|execute_actions",
  "query":{"entity":"string","relation":"string"
    ↪ },
  "hypothesis":{"id":"H1","candidate":{"name":"
    ↪ string"}},
  "tegraph_snapshot":{"nodes":[...],"edges"
    ↪ :[...]},
  "graph_actions":[{"type":"add_mechanism_link",
    ↪ "assignee":"Proponent"}],
  "constraints":{"require_disjoint_paths":2}
}
```

Output (schematic)

```
{
  "graph_updates":{
    "add_nodes":[
      {"id":"n1","type":"Target","label":"..."},
      {"id":"n2","type":"Pathway","label":"..."}
    ],
    "add_edges":[
      {"source":"drugX","target":"n1","relation":
    ↪ "acts_on","weight":0.8,"rationale":"brief
    ↪ "}
    ],
    "merge":[{"keep":"n2","remove":"n2_dup"}]
  },
  "subconclusions":[{"id":"C1","text":"closure
    ↪ formed","confidence":"medium"}],
  "uncertainties":["e.g., BBB unknown"]
}
```

Key constraints

- Aim for $\geq$1–2 *disjoint* paths: Drug $\rightarrow$ Target/-Pathway $\rightarrow$ Phenotype/Disease.
- Each new edge has a weight $[0, 1]$ and a one-line rationale; merge synonyms/duplicates.
- Do not build refutation/safety chains (that is Skeptic's role).

A.2.3 Skeptic (Counter-Builder)

Role Construct *refutation/risk* chains and conflict hotspots; or execute PI-assigned risk actions.

Modes

- `build_counterchain`: Extend refutation/risk mechanisms.
- `execute_actions`: Execute `graph_actions` (assignee=Skeptic).

Input (schematic)

```
{
  "mode":"build_counterchain|execute_actions",
  "query":{"entity":"string","relation":"string"
    ↪ },
  "hypothesis":{"id":"H1","candidate":{"name":"
    ↪ string"}},
  "tegraph_snapshot":{"nodes":[...],"edges"
    ↪ :[...]},
  "graph_actions":[{"type":"stress_test_safety",
    ↪ "assignee":"Skeptic"}]
}
```

Output (schematic)

```
{
  "graph_updates":{
    "add_nodes":[{"id":"k1","type":"Pathway","
    ↪ label":"RiskPathway"}],
    "add_edges":[
      {"source":"drugX","target":"k1","relation":
    ↪ "involved_in","weight":0.7,"rationale":"
    ↪ brief"},
      {"source":"k1","target":"H1","relation":"
    ↪ refutes","weight":0.85,"rationale":"
    ↪ safety conflict"}
    ],
    "conflict_hotspots":[
      {"topic":"supine_hypertension","pro_nodes"
    ↪ :["..."],"con_nodes":["..."]}
    ]
  },
  "counterclaims":[{"id":"K1","text":"
    ↪ directional conflict/insufficient PK","
    ↪ confidence":"medium"}],
  "falsification_tests":["minimal falsifiable
    ↪ checks..."]
}
```

Key constraints

- Prioritize safety/contraindication, mechanistic contradiction, PK/PD barriers, and phenotype-vs-outcome gaps.
- Use `refutes/contradicts` edges to the hypothesis/claims; annotate conflict hotspots.
- Control node/edge growth.

A.2.4 Shared Constraints (All Agents)

- Return a single UTF-8 JSON object (no Markdown or prose).
- No fabricated sources; no clinical advice; mark uncertainty when evidence is thin.
- Only Proponent/Skeptic update the T-EGraph; PI never edits the graph directly.
- Field names follow the schemas above; missing optional fields must be tolerated.