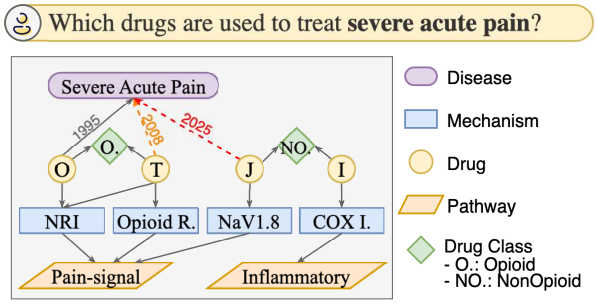
# Assessing LLMs for Serendipity Discovery in Knowledge Graphs: A Case for Drug Repurposing

**Mengying Wang*, Chenhui Ma*, Ao Jiao*, Tuo Liang, Pengjun Lu, Shrinidhi Hegde, Yu Yin, Evren Gurkan-Cavusoglu, Yinghui Wu**

Case Western Reserve University, Cleveland, OH, USA
{mxw767, cxm590, axj770, txl859, pxl465, sxh1426, yxy1421, exg44, yxw1650}@case.edu

## Abstract

Large Language Models (LLMs) have greatly advanced knowledge graph question answering (KGQA), yet existing systems are typically optimized for returning highly relevant but predictable answers. A missing yet desired capacity is to exploit LLMs to suggest surprise and novel ("serendipitious") answers. In this paper, we formally define the serendipity-aware KGQA task and propose the SerenQA framework to evaluate LLMs' ability to uncover unexpected insights in scientific KGQA tasks. SerenQA includes a rigorous serendipity metric based on relevance, novelty, and surprise, along with an expert-annotated benchmark derived from the Clinical Knowledge Graph, focused on drug repurposing. Additionally, it features a structured evaluation pipeline encompassing three subtasks: knowledge retrieval, subgraph reasoning, and serendipity exploration. Our experiments reveal that while state-of-the-art LLMs perform well on retrieval, they still struggle to identify genuinely surprising and valuable discoveries, underscoring a significant room for future improvements. Our curated resources and extended version are released at: https://cwru-db-group.github.io/serenQA.




| Drugs | Appro. Year | Relevance | Novelty | Surprise |
|---|---|---|---|---|
| Oxycodone | 1995 | ✓ | | |
| Tapentadol | 2008 | ✓ | ✓ | |
| Journavx | 2025 | ✓ | ✓ | ✓ |
| Ibuprofen | \ | | | ✓ |

Figure 1: Suggesting Drugs that treat Severe Acute Pain: A Serendipitous case of Journavx.

## 1 Introduction

Large language models (LLMs) are rapidly advancing the bridge between natural language understanding and effective question answering. Significant efforts, such as domain-specific fine-tuning, prompt engineering, and Retrieval-Augmented Generation (RAG), have enabled LLMs to leverage external knowledge bases to produce highly relevant and precise answers tailored to specialized research questions (Le et al. 2024). However, these systems often focus on returning information already familiar to experts, missing the crucial scientific capacity to uncover surprising connections that inspire new research directions (Song et al. 2023).

"*Serendipity*", the art of luck and beneficial discovery, arises from both unexpected findings and the skill to recognize novel applications of such discoveries in various domains, serving as a catalyst for genuine scientific breakthroughs. While serendipity has been studied in web search (Huang et al. 2018) and recommender systems (Tokutake and Okamoto 2024), it remains largely unexplored in scientific question answering. Empowering LLMs with the ability to discover new knowledge from existing, valuable knowledge bases is thus a critical step towards true LLM-empowered scientific discovery.

**Example 1:** Fig. 1 illustrates a KGQA task to find drugs that can treat severe acute pain. There are four possible answers. (1) Opioids *e.g.,* **Oxycodone**, a well-known drug with recognized mechanism on targeting the $\mu$-opioid receptor within the pain-signaling pathway. (2) **Tapentadol** (2008) expanded this paradigm by adding a dual mechanism, hence with increased novelty for the question. (3) **Journavx**, a first non-opioid analgesic for severe acute pain (FDA 2025) approved by FDA in 2025. Journavx acts through a novel mechanism, selectively inhibiting NaV1.8 sodium channels in peripheral pain-sensing neurons. Surprisingly, with this paradigm shift and different targets, it remains relevant by sharing the broader pain-signal pathway context with opioids. Hence it is a "serendipitous" result in the KGQA search, in terms of relevance, novelty, and an unexpected answer, which may inspire new medical research directions. (4) **Ibuprofen**, in contrast, works through the classical inflammatory COX inhibition pathway, targeting

*These authors contributed equally.

mild-to-moderate pain and thus showing low embedding relevance and novelty, while suggesting Ibuprofen for severe acute pain would still be surprising. □

"*Can LLMs, while enhanced by domain KGs, suggest serendipitous answers for domain sciences?"* This paper makes a first step to investigate the potential of LLMs to surface serendipitous discoveries within scientific KGQA, with a focus on drug repurposing, which is a cornerstone of medical research. We address three core research questions:

- (**RQ1**): How may "serendipity" be characterized and quantitatively measured for scientific KGQA tasks?
- (**RQ2**): What roles could LLMs play for serendipity discovery in domain science KGQA?
- (**RQ3**): How to evaluate state-of-the-art LLMs, and what are their performances in serendipity discovery?

To this end, we introduce the SerenQA framework designed to systematically evaluate the ability of LLMs to uncover serendipitous answers within the context of KGQA. It includes three core components (shown in Fig. 2):

- **Serendipity Metric (**RNS): A rigorous, graph-based measure capturing **R**elevance, **N**ovelty, and **S**urpriseness in KGQA answers, justified by an *axiomatic* analysis that clarifies the trade-offs and properties.
- **Serendipity-aware Benchmark**: An expert-annotated KGQA dataset for drug repurposing, based on the Clinic Knowledge Graph (Santos et al. 2022). It features curated question-answer pairs and explicit serendipity annotations for fine-grained evaluation.
- **Assessment Pipeline**: A principled and reproducible three-phase workflow that systematically evaluates LLMs' roles in serendipitous discovery. It decomposes the task into knowledge retrieval, reasoning, and exploratory search, providing insights into model capabilities and limitations in scientific knowledge discovery.

We performed extensive experiments with various LLMs across different scales, demonstrating that while frontier models excel in knowledge retrieval tasks, nearly all models struggle significantly in serendipity exploration, highlighting inherent challenges and opportunities in this area.

**Related works**. We categorize related works as follows.

*Serendipity-Driven Knowledge Exploration* Serendipity, defined as an unexpected yet valuable discovery, has emerged as a crucial goal in recommender systems and knowledge exploration (Bordino, Mejova, and Lalmas 2013). Recent studies have leveraged LLMs to generate and evaluate serendipitous recommendations through advanced prompt engineering (Fu and Niu 2024) or by aligning model outputs with human preferences (Xi et al. 2025). Notably, existing approaches primarily rely on subjective human annotation, LLM self-evaluation, or comparisons against benchmark groundtruths for serendipity evaluation. In contrast, we propose a graph-based serendipity measure (RNS), which transforms the knowledge graph (KG) into a probability matrix (Dehmer and Mowshowitz 2011), enabling an information-theoretic quantification of various subjective aspects of serendipity, resulting in a more rigorous evaluation.

*LLM-Augmented Novelty Detection*. LLMs are increasingly seen as creative partners that can accelerate scientific discovery across disciplines (AI4Science and Quantum 2023). By mining vast knowledge and generating hypotheses, LLMs can propose novel research ideas or unexpected connections that human experts might overlook (Si, Yang, and Hashimoto 2025). Despite these efforts, the community still lacks a more comprehensive understanding and benchmark datasets specifically designed to assess serendipitous discoveries. To address this gap, we present a drug repurposing KGQA dataset which enables a systematic and objective assessment of serendipitous knowledge exploration.

SerenQA is the first reproducible and extensible framework for advancing serendipity discovery in drug repurposing. We advocate its broader application to facilitate new research opportunities in scientific KGQA tasks.

## 2 Serendipitous Assessment with KGQA

Below, we define relevant concepts and core notations:

### 2.1 Serendipity-aware KGQA

SerenQA performs LLM assessment by processing a pipeline of *serendipity-aware* KGQA. Given a natural language (NL) question $Q$, a large language model $\mathcal{L}$, a directed, multigraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of entities with size $V = |\mathcal{V}|$, and $\mathcal{E}$ is the set of relations with size $E = |\mathcal{E}|$, a serendipity-aware KGQA system returns an answer set as an ordered partition $\mathcal{A} = (\mathcal{A}_e, \mathcal{A}_s)$, where:

- $\mathcal{A}_e$: the **existing answer set**, containing answers explicitly supported by facts in $\mathcal{G}$;
- $\mathcal{A}_s$: the **serendipity answer set**, containing answers that are relevant but extend beyond direct explicit knowledge, revealing novel and unexpected relationships in $\mathcal{G}$.

such that $\mathcal{A}_e \cup \mathcal{A}_s \subseteq \mathcal{V}$ and $\mathcal{A}_e \cap \mathcal{A}_s = \emptyset$. We define $|\mathcal{A}| = |\mathcal{A}_e \cup \mathcal{A}_s|$ as the total size of the answer set.

This serendipity-aware setup is motivated by the real-world scientific discovery process, which frequently involves uncovering not only established knowledge ($\mathcal{A}_e$) but also insightful and surprising associations ($\mathcal{A}_s$), potentially leading to innovative research opportunities, such as novel drug repurposing. Knowledge graphs are particularly suitable for this task due to their structured representation of interconnected entities and relations, enabling systematic exploration of indirect and surprising relationships.

### 2.2 Graph-specified Serendipity Formulation

To rigorously quantify serendipity, we define a graph-based serendipity measure (RNS), which quantifies how effectively a serendipity answer set $\mathcal{A}_s$ for a given question $Q$ provides relevant yet novel and surprising insights beyond the explicit answer set $\mathcal{A}_e$. Intuitively, serendipity is a composite experience, encompassing multiple dimensions simultaneously (Niu and Abbas 2017). Formally, we define the RNS score as a weighted combination of three perspectives: relevance, novelty, and surprise, which can be flexibly adjusted to suit user preferences. Given an answer set $\mathcal{A} = (\mathcal{A}_e, \mathcal{A}_s)$, the serendipity score is computed as:

$$\mathsf{RNS}(\mathcal{A}_e, \mathcal{A}_s) = \alpha\, R(\mathcal{A}_e, \mathcal{A}_s) + \beta\, N(\mathcal{A}_e, \mathcal{A}_s) + \gamma\, S(\mathcal{A}_e, \mathcal{A}_s)$$

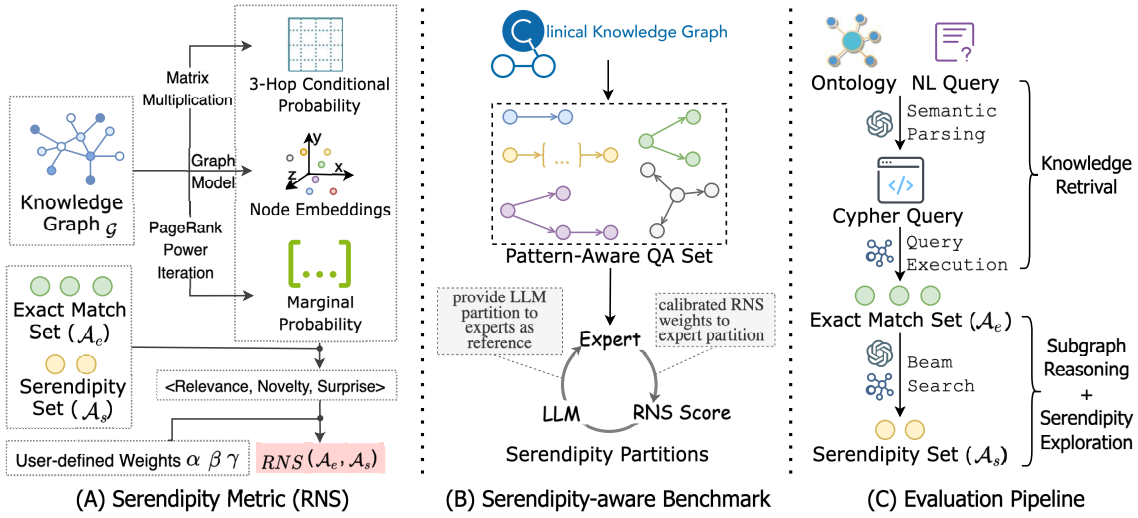



Figure 2: SerenQA Framework. **(A)**: Computing RNS score for partition $(\mathcal{A}_e, \mathcal{A}_s)$ form $\mathcal{G}$; (Sec. 3). **(B)**: Constructing SerenQA dataset from ClinicalKG; (Sec. 4). **(C)**: For an NL query, our pipeline retrieves $\mathcal{A}_e$ from $\mathcal{G}$ and explores $\mathcal{A}_s$ from $\mathcal{A}_e$ with beam search. (Sec. 5).

- $R$ (**Relative Relevance**): context similarity of $\mathcal{A}_e$ and $\mathcal{A}_s$;
- $N$ (**Relative Novelty**): new information in $\mathcal{A}_s$ beyond $\mathcal{A}_e$;
- $S$ (**Relative Surprise**): unpredictability of $\mathcal{A}_s$ given $\mathcal{A}_e$.

The weights $\alpha, \beta, \gamma$ can be tuned to user preference; recommended defaults are fit to expert evaluations. Details of the metric and its computation are described in Sec 3.

In the following sections, we detail how the SerenQA framework establishes a unified benchmark, dataset, and evaluation protocols specifically designed to assess LLM capabilities in serendipitous knowledge discovery tasks, particularly in the critical area of drug repurposing.

## 3 Serendipity Quantification

Quantifying serendipity is inherently challenging due to its abstract and subjective nature. As discussed in Sec 1, existing methods often rely heavily on subjective human annotations or LLM-generated evaluations, which suffer from limitations like poor interpretability, scalability issues, and potential biases. To overcome these, we introduce an information-theoretic approach enabling *scalable*, *interpretable*, and *reproducible* serendipity evaluations.

### 3.1 Serendipity: A characterization

To align with human intuition about "Serendipity" while allowing for rigorous quantification, as introduced in Sec 2.2, we specifically decompose it into three complementary dimensions: *Relevance*, *Novelty*, and *Surprise*. For an answer set $\mathcal{A} = (\mathcal{A}_e, \mathcal{A}_s)$ to a query $Q$, we define the **Serendipity Score** (RNS) as a weighted combination of the relative measures between $\mathcal{A}_s$ and $\mathcal{A}_e$, with user-configurable weights to accommodate different preferences or application scenarios. Each dimension is adapted to well-established information-theoretic measures, as described below:

**Relative Relevance**. We compute relative Relevance ($R$) as the average normalized Euclidean distance ($d(\cdot)$) between the GCN embeddings of entities in $\mathcal{A}_s$ and $\mathcal{A}_e$:

$$R(\mathcal{A}_e, \mathcal{A}_s) = -\frac{\sum_{i \in \mathcal{A}_s, j \in \mathcal{A}_e} d(n_i, n_j)}{|\mathcal{A}_s||\mathcal{A}_e|}$$

where $n_i$ (resp. $n_j$) refers to the embedding of the entity $i \in A_s$ (resp. $j \in \mathcal{A}_e$). A larger distance reflects greater contextual difference, indicating $\mathcal{A}_s$ belongs to more distinct clusters in $\mathcal{G}$ and may diverge from the core context of $Q$.

**Relative Novelty**. Relative Novelty ($N$) is derived from a mutual-information-based score between the existing and serendipity sets. For a partition $(\mathcal{A}_e, \mathcal{A}_s)$, we define $N(\mathcal{A}_e, \mathcal{A}_s)$ = 1- $MI(\mathcal{A}_e, \mathcal{A}_s)$, where $MI(\mathcal{A}_s, \mathcal{A}_e)$ measures the shared amount of information between $\mathcal{A}_s$ and $\mathcal{A}_e$, and is given by:

$$MI(\mathcal{A}_e, \mathcal{A}_s) = \sum_{i \in \mathcal{A}_e} P(i) \sum_{j \in \mathcal{A}_s} P(j|i) \log \frac{P(j|i)}{P(j)}$$

A higher N score indicates $\mathcal{A}_s$ are less redundant given $\mathcal{A}_e$.

**Relative Surpriseness**. Relative Surprise ($S$) is quantified via Jensen–Shannon divergence (JSD) between entity distributions $P_s$ and $P_e$, which are the accumulated probability distributions over entities in $\mathcal{A}_s$ and $\mathcal{A}_e$, respectively:

$$S(\mathcal{A}_e, \mathcal{A}_s) = \frac{1}{2}(D_{KL}(P_s \| P_{Mix}) + D_{KL}(P_e \| P_{Mix}))$$

where $D_{KL}(\cdot\|\cdot)$ is the Kullback–Leibler divergence (Kullback 1951), and $P_{Mix} = \frac{1}{2}(P_s + P_e)$.

Given $\mathcal{A}_e$, a *higher* RNS indicates a "more" serendipitous set $\mathcal{A}_s$ with greater diverse, novel and surprise entities that cannot be inferred from $\mathcal{A}_e$, as exemplified by "Journavx", the first non-opioid analgesic for severe acute pain (Exp. 1).

### 3.2 Cost-effective Graph Probabilistic Modeling

Cost-effective graph probabilistic models ($P(\cdot)$) is crucial for efficient RNS computation. We present the detailed models, justified by an axiomization analysis.

**3-Hop Conditional Probability**. Serendipitous findings may from indirect, multi-hop connections. Thus, we consider a multi-hop conditional probability matrix $M$ that aggregates transition probabilities across both direct and indirect relations to capture a global probabilistic propagation. Empirically, 99% of serendipity answers in our datasets are reachable from existing answers within three hops, prompting our analysis to up to 3-hop neighbors of entities in $\mathcal{G}$.

Given graph $\mathcal{G}$, we initialize $M$ as a weighted matrix $M$, with $M_{ij}$ the number of links from node $i$ to $j$. We normalize $M$ to obtain the one-hop transition probabilities that ensures row-stochasticity: $P_1(j|i) = \frac{M_{ij}}{\sum_{k\in\mathcal{E}} M_{ik}}$. The $k$-hop conditional probability matrix $P_k$ is computed as:

$$P_k = \sum_{h=1}^{k} \alpha_h P_1^h, \quad \alpha_h = \frac{h}{\sum_{h=1}^{k} h}$$

where $P_1^h$ represents the probability of reaching a node in $h$ hops, and weights $\alpha_h$ increases for larger $h$ to prioritize longer connections. We can justify that $P_k$ consistently satisfies the necessary constraints of a transition matrix:

- *Non-negativity*: $(P_k)_{ij} \geq 0$ for all $(i, j)$,
- *Row-Stochastic Property*: $\sum_j (P_k)_{ij} = 1$ for all $i$.

*Cost Analysis.* Constructing $M$ takes $\mathcal{O}(V^2)$ for dense graphs. Traditional $P_3$ computation[1] via graph traversal is $\mathcal{O}(V^4)$. We employ Divide-and-Conquer optimized matrix multiplication (Strassen 1969) and parallel computation with $t$ processors, reducing the cost to $\mathcal{O}(V^{\log_2^7}/t)$.

**Marginal Probability**. The marginal probability $\mathbf{P}(i)$ quantifies steady-state node probabilities at node $i$ under the law of total probability: $\mathbf{P}(i) = \sum_j P_3(i|j)\mathbf{P}(j)$. This leads to the linear system representation:

$$(I - P_3^T)\mathbf{P} = 0, \quad \sum_i \mathbf{P}(i) = 1$$

which can be solved by matrix inversion in $\mathcal{O}(V^3)$ time. To further reduce the cost, we approximate the computation with a PageRank-style damped iteration:

$$\mathbf{P}_{t+1} = \lambda P_3^T \mathbf{P}_t + (1 - \lambda)\mathbf{P}_0$$

where $\mathbf{P}_0$ is an initial probability distribution, set uniformly as $\frac{1}{V}$, ensuring convergence even on disconnected graphs. This reduces the cost in $\mathcal{O}(V^2 \log V)$ time.

We remark that the probabilistic matrices are computed "once for all" and are shared for multiple queries, and readily adapt to different domain graphs.

Further analyses are included in the Appendix C.

**Axiomization Analysis**. We further justify that RNS is a proper serendipity measure for KGQA tasks through the following axiomatic analysis. For any query and a corresponding retrieved, fixed existing set $\mathcal{A}_e$, consider an optimization process that finds an optimal serendipitous set $\mathcal{A}_s^*$ with at most $K$ entities, *i.e.,* $A_s^* = \arg\max_{|A_s|\leq K} \mathsf{RNS}(\mathcal{A}_e, \mathcal{A}_s)$. We can show that RNS satisfies the following properties:

- (**Scale invariance**). $\mathcal{A}_s^*$ remains to maximize RNS even if $R$, $N$ or $S$ are scaled by a constant. This ensures the invarance of $\mathcal{A}_s^*$ under RNS measure regardless of how the user preference ($\alpha$, $\beta$, $\gamma$) changes.
- (**Consistency**). Making the $R, N, S$ larger (resp. smaller) for any entities in $\mathcal{A}_e$ (resp. $\mathcal{A}_s$) does not change the ranking of entities in $\mathcal{A}_s^*$ in terms of RNS.
- (**Non-monotonicity**). $\mathsf{RNS}(\mathcal{A}_e, \mathcal{A}_s) \not\leq \mathsf{RNS}(\mathcal{A}_e, \mathcal{A}'_s)$ if $|\mathcal{A}_s| \leq |\mathcal{A}'_s|$. Indeed, larger answer sets do *not* necessarily indicate that they are more "serendipitous" in practice.
- (**Independence**). RNS is only determined by the embeddings of entities from $\mathcal{A}_s \cup \mathcal{A}_e$. No information from entities not seen in $\mathcal{A}$ can affect the serendipitous of $\mathcal{A}_e$. This justifies RNS for serendipity in a pragmatic "semi-closed world" assumption, striking a balance between a challenging open-world analysis ($\mathcal{A}_s$ can be infinite) and a rigorous, overkilling closed world ($\mathcal{A}_s = \emptyset$) setting.

# 4 Serendipity-aware Benchmark

The proposed RNS measure enables quantitative assessment of serendipity within any answer set $(\mathcal{A}_e, \mathcal{A}_s)$ derived from a graph $\mathcal{G}$. Yet scoring alone is insufficient: assessing cornerstone steps such as retrieving and reasoning demands a benchmark with authoritative groundtruth serendipity answer set. We therefore introduce a drug-repurposing benchmark that supports both standard KGQA tasks and serendipity-aware evaluations, giving the fine-grained supervision required for end-to-end assessment.

## 4.1 QA Set Construction

Our benchmark is built upon the Clinical Knowledge Graph (CKG) (Santos et al. 2022), a widely recognized biomedical resource containing extensive data on drug, gene, and disease interactions. Our focus is on drug repurposing, which is a critical research task aimed at identifying novel therapeutic uses of existing drugs (Pushpakom et al. 2019).

Our dataset supports typical KGQA tasks through a contextualized query scenario that consists of standardized configuration including *expert-verified*, scientifically meaningful NL queries, their structured graph (Cypher) counterparts with query components that are explicitly annotated with their semantics, and grounded and validated answer sets. Unlike its peer NL-only benchmark datasets in KGQA, it couples each NL query to a distinct, validated "ground truth", structured graph query, thereby reducing ambiguity and mitigating possible semantic redundancy. It also explicitly annotates graph patterns, such as multi-hop and intersection queries, to reflect realistic query complexities in scientific inquiry. Dataset statistics are summarized in Table 1. We present details of graph queries in Appendix A.

## 4.2 Answer Set Construction

To reliably establish ground-truth serendipity sets, we start with the latest version of Clinic KG, denoted as $\mathcal{G}_c$. For each query $Q$, we initially obtain its complete candidate answer set $\mathcal{A}_c$ from $\mathcal{G}_c$. We then partition it into an existing set $\mathcal{A}_e$ and a serendipity set $\mathcal{A}_s$ , with $\mathcal{A}_e \cap \mathcal{A}_s = \emptyset$ and $\mathcal{A}_e \cup \mathcal{A}_s = \mathcal{A}_c$. We apply three distinct partitioning strategies:

[1]While we make a case for 3-hop queries here, our discussion readily extends to $k$-hop queries for $k \geq 3$.

| Statistic | Value |
|---|---|
| Number of Distinct Queries | 1529 |
| Number of Relations in $\mathcal{G}$ ($E$) | 201,704,256 |
| Number of Entities in $\mathcal{G}$ ($V$) | 15,430,157 |
| Number of Graph Pattern Types | 9 |
| Avg. Answer Set Size ($\lvert\mathcal{A}\rvert$ per query) | 4.04 |
| Number of Experts for NL Query Verification | 4 |
| Number of Experts for Serendipity Annotation | 6 |

Table 1: Dataset Statistics of SerenQA Benchmark.

**LLM Ensemble**. Following established practices, we prompt four state-of-the-art LLMs to assign a "serendipity score" to each candidate answer. For every query, entities in the complete answer set $\mathcal{A}_c$ are ranked by their average LLM score; the top 20% are collected as the serendipity set $\mathcal{A}_s$, and the reminder form $\mathcal{A}_e$.

**Expert Crowdsourced**. We engaged a team of 6 domain experts (three physicians, one pharmaceutical scientist, and two trained medical model annotators) via an online questionnaire (DrugKG Questionnaire 2025). They were requested to refine the rankings from LLMs. The questionnaire is accepting continuous responses from human experts.

**RNS Guided**. With the justified RNS metric (Sec.3) we treat serendipity partitioning as:

$$\max_{\mathcal{A}_e,\mathcal{A}_s} \mathsf{RNS}(\mathcal{A}_e,\mathcal{A}_s), \quad \text{s.t. } |\mathcal{A}_s| = b, b = \max(1, \lfloor 0.2|\mathcal{A}_c| \rfloor)$$

Starting from an initial partition, we apply the greedy-swap algorithm in Algorithm 1 to (approximately) compute an optimal answer set $\mathcal{A}_s$ in $\mathcal{A}_c$. The algorithm iteratively swaps entity pairs between $\mathcal{A}_e$ and $\mathcal{A}_s$ that yield the greatest improvement in a marginal gain of RNS, continuing until no further improvement is possible. Each iteration has a complexity of $\mathcal{O}(|\mathcal{A}|^2)$. We found in our tests that $\mathcal{A}_e$ usually contains a few entities (on average 4; see Table 1), And the algorithm is quite fast in practice. During that, we calibrated the RNS weights to align with the expert-crowdsourced partitions for consistency and fair assessment.

For each partitioning result, we construct $\mathcal{G}$ by removing selected edges from $\mathcal{G}_c$, ensuring that for each query $Q$, entities in $\mathcal{A}_e$ remain derivable from $\mathcal{G}$, while entities in $\mathcal{A}_s$ become inaccessible. This creates a controlled evaluation environment aligned with problem definitions (Sec. 2).

Algorithm 1: Greedy Swap for RNS –Guided Optimization

**Input**: initial partition $(\mathcal{A}_e^0, \mathcal{A}_s^0)$;
pre-computed probability matrices $P_3$, $\mathbf{P}$ for graph $\mathcal{G}$
**Output**: optimized partition $(\mathcal{A}_e, \mathcal{A}_s)$

**set** $(\mathcal{A}_e, \mathcal{A}_s) := (\mathcal{A}_e^0, \mathcal{A}_s^0)$, $\tau = \mathsf{RNS}(\mathcal{A}_e, \mathcal{A}_s)$
**while true do**
  **set** $\Delta_{\max} := 0$; $(i^*, j^*) := \mathsf{null}$
  **for** $i \in \mathcal{A}_s$ **do**
    **for** $j \in \mathcal{A}_e$ **do**
      $\mathcal{A}'_s := (\mathcal{A}_s \backslash \{i\}) \cup \{j\}$, $\mathcal{A}'_e := (\mathcal{A}_e \backslash \{j\}) \cup \{i\}$
      $\Delta := \mathsf{RNS}(\mathcal{A}'_e, \mathcal{A}'_s) - \tau$
      **if** $\Delta > \Delta_{\max}$ **then**
        $\Delta_{\max} := \Delta$; $(i^*, j^*) := (i, j)$
      **end if**
    **end for**
  **end for**
  **if** $\Delta_{\max} = 0$ **then break**;
  **end if**
  $\mathcal{A}_s := (\mathcal{A}_s \backslash \{i^*\}) \cup \{j^*\}$, $\mathcal{A}_e := (\mathcal{A}_e \backslash \{j^*\}) \cup \{i^*\}$
  $\tau := \tau + \Delta_{\max}$
**end while**
**return** $(\mathcal{A}_e, \mathcal{A}_s)$

## 5 Evaluation Pipeline

We next introduce our evaluation pipeline (Fig. 2(C)), which systematically evaluates the serendipity discovery capabilities of LLMs using our curated serendipity-aware benchmark. The pipeline is modularized into three highly correlated tasks, each of which independently measures a specific, "cornerstone" aspect of an LLM's role and performance on serendipity discovery in scientific KGQA tasks.

**Knowledge Retrieval**. In this task, LLM translates an NL question $Q$ into a Cypher query $C$ to retrieve an answer set $\mathcal{A}_e$ from the knowledge graph $\mathcal{G}$. The performances are evaluated by comparing the accuracies of the retrieved answer set $\mathcal{A}_e$ against the ground truth. Additionally, the performances across different query patterns (such as one-hop, two-hop, and intersection queries) are compared to evaluate the LLM's capability to handle varying levels of query complexity and structural diversity.

**Subgraph Reasoning**. This task evaluates the LLM's capability to interpret and concisely summarize the retrieved answer of a graph-structured query $C$ in a knowledge graph (as a subgraph) into domain-aware natural language answers. The generated summaries provide essential contextual support for subsequent serendipity exploration tasks, requiring nuanced biomedical understanding and logical reasoning.

**Serendipity Exploration**. This third (final) task evaluates the LLMs' proactive ability to uncover serendipity entities $\mathcal{A}_s$ through an LLM-guided beam search from $\mathcal{A}_e$. Given a beam width $w$, we prompt LLM to select the top-$w$ nodes at each step from the candidate list as the next target nodes based on criteria such as supporting evidence, interaction strength, biological effect direction, and their expression level. The model further determines whether to continue exploration based on relevance and potential novelty. This task assesses the LLM's ability to use biomedical knowledge and contextual search to effectively navigate serendipitous discovery while balancing depth and breadth in exploration. We remark that the serendipity set $\mathcal{A}_s$ produced in this section is the pipeline's output at evaluation time; in contrast, the $\mathcal{A}_s$ defined in Sec. 4 is the benchmark ground-truth constructed for scoring. More details are provided in Appendix D.

## 6 Experiments

### 6.1 Experiment Setting

We conduct experiments using the benchmark introduced in Sec. 4, and evaluated LLMs across multiple scales, from frontier models with billions of parameters to smaller vari-

| Model | One-Hop | | | Two-Hop | | | Multiple(3+)-Hop | | | Intersection | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Hit(%) | F1(%) | Exe.(%) | Hit(%) | F1(%) | Exe.(%) | Hit(%) | F1(%) | Exe.(%) | Hit(%) | F1(%) | Exe.(%) |
| DeepSeek-V3 | **20.45** | **78.71** | 72.88 | 3.46 | 10.71 | 9.86 | 1.97 | 6.22 | 6.55 | 2.64 | 7.15 | 8.03 |
| GPT-4o | 19.71 | 77.16 | 60.17 | 2.08 | 6.36 | 7.89 | 1.40 | 4.20 | 4.85 | 1.56 | 4.65 | 5.21 |
| Claude-3.5-Haiku | 13.28 | 48.54 | 48.73 | 9.78 | 39.01 | 32.89 | **4.43** | 8.64 | **14.08** | 1.38 | 3.90 | 4.66 |
| Llama-3.3-70B | 19.28 | 70.67 | 74.58 | **16.63** | **44.34** | **56.57** | 2.98 | **10.16** | 11.89 | **4.80** | **9.60** | 16.05 |
| DeepSeek-R1-70B | 19.87 | 69.07 | **80.08** | 12.03 | 37.00 | 43.42 | 2.97 | 8.06 | 13.11 | 3.49 | 6.16 | **16.46** |
| Med42-V2-70B | 18.34 | 69.43 | 69.92 | 5.92 | 19.12 | 19.74 | 0.23 | 0.51 | 1.21 | 0.08 | 0.13 | 0.68 |
| Qwen3-32B | 0.37 | 1.27 | 1.27 | 0.16 | 0.65 | 0.65 | 0.24 | 0.36 | 0.48 | 0.00 | 0.00 | 0.00 |
| DeepSeek-R1-32B | 17.90 | 65.23 | 68.22 | 3.06 | 5.72 | 7.24 | 1.87 | 4.50 | 5.58 | 0.79 | 1.84 | 3.16 |
| Qwen3-8B | 10.07 | 37.24 | 39.83 | 0.98 | 2.87 | 3.95 | 0.90 | 2.01 | 4.85 | 1.58 | 1.91 | 5.62 |
| DeepSeek-R1-8B | 1.27 | 3.41 | 5.51 | 0.00 | 0.00 | 0.00 | 0.04 | 0.24 | 0.24 | 0.00 | 0.00 | 0.00 |
| Med42-V2-8B | 8.11 | 23.90 | 49.15 | 1.05 | 3.31 | 3.97 | 1.71 | 4.07 | 4.12 | 0.04 | 0.13 | 0.14 |
| Qwen3-1.7B | 0.84 | 3.72 | 11.86 | 0.65 | 1.98 | 3.29 | 0.00 | 0.00 | 0.24 | 1.08 | 1.56 | 2.74 |
| DeepSeek-R1-1.5B | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

Table 2: Knowledge Retrieval ($T1$), Best scores are **bolded**, second best are underlined

ants ( $1B$ parameters). Experimental results are presented in Tables 2–3, including three evaluation tasks within our pipeline: $T1$ (Knowledge Retrieval), $T2$ (Subgraph Reasoning) and $T_3$ (Serendipity Exploration).

**Evaluation metrics**. Table 2 ($T1$) reports *F1 scores*, *Executability* (percentage of error-free queries), and *Hit Rate*($|\mathcal{A}_e \cap \mathcal{A}'_e|/|\mathcal{A}_e|$), categorized by query patterns; and Table 3 ($T2$, $T3$) reports their performances on three ground-truth partitions (LLM-Ensemble, Expert-Crowdsourced, RNS-Guided). During beam search (beam width 30, maximum depth 3), we employ one-shot learning by providing a single query with detailed ground-truth serendipity paths in the prompt, helping models understand exploration paths. In addition, $T2$ and $T3$ are measured with (a) Subgraph Reasoning:*Faithful.* (1–5, LLM-judged, factual accuracy of summaries); *Compre.* (1–5, LLM-judged, coverage of key graph elements); *SerenCov* (0–1, fraction of serendipity paths explicitly mentioned). (b) Serendipity Exploration: *Relevance* (1–5, LLM-judged alignment with groundtruth entities); *TypeMatch* (0–1, the fraction of predicted entity types that match the ground truth types); and *SerenHit* (0–1, match rate with groundtruth serendipity set).

**Experiment Environment** We depoly our system on 5 x AWS c6a.24xlarge on-demand instances for distributed computation and 5 x c6a.xlarge instances as relation storage nodes, each node runs Ubuntu 22.04 with Docker and Redis 7.2, using mounted dump.rdb as readonly data source. The system supports 500 concurrent LLM reasoning tasks across distributed nodes via asyncio.

## 6.2 Task Analysis

We next analyze experimental results task-by-task.

**Task 1: Knowledge Retrieval**. The results in Table 2 show that larger models (e.g., DeepSeek-V3, GPT-4o) consistently excel in simpler one-hop retrieval (F1 $\approx 78\%$), yet both exhibit performance degradation for more complex multi-hop queries (F1 drops to $< 10\%$ for queries with $3+$ hops). Smaller models are less accurate in coping with both simpler and more complex queries, reflecting limitations in reasoning depth and broader coverage of the biomedical context. Notably, the two 70B models (Llama-3.3-70B, DeepSeek-R1-70B) achieve better performances, which may be due to their more up-to-date training datasets.

**Task 2: Subgraph Reasoning**. In Table 3 (upper), Mixtral-8×7B achieves (surprisingly) high Serendipity Coverage ($60\%+$) despite moderate scores in Faithfulness and Comprehensiveness (2-3 out of 5). This interestingly indicates that summarization approaches yield broader serendipitous path coverage but risk factual inaccuracies. In contrast, larger models (*e.g.,*Llama-3.3-70B) achieve higher Faithfulness and Comprehensiveness but lower "SerenCov", suggesting a consistent trade-off that their richer pre-trained knowledge produces more precise, yet narrower summaries.

**Task 3: Serendipity Exploration**. The rows labeled "w.o. summary" evaluate performance without subgraph summaries, isolating the effect of providing chain-of-thought guidance. For almost all models, removing the summary improved performance on all three metrics. One possible reason for this is that the model may introduce hallucinations during the summary process, which can influence the exploration path, as proven by Table 3 (upper), many models did not achieve the desired score in subgraph reasoning.

## 6.3 In-Depth Discussion

**Model scale vs. Serendipity.** As shown in the tables, larger models generally perform better in retrieval and exploration tasks. However, for subgraph summarization and reasoning (denoted as $T2$), there is significant variance and no obvious correlation with model size. This may suggest that retrieval and exploration benefit more from the model's inherent knowledge, which larger models excel at, while summarization and reasoning do not follow the same trend.

**Partition Sensitivity.** Fig. 3 displays triangle plots of Pearson Correlations for TypeMatch, SerenCov, and SerenHit, with each triangle representing one metric. The corners de-

| Models | LLM Ensemble | | | Expert Crowdsourced | | | RNS Guided | | |
|---|---|---|---|---|---|---|---|---|---|
| | Faithful. | Compre. | SerenCov | Faithful. | Compre. | SerenCov | Faithful. | Compre. | SerenCov |
| DeepSeek-V3 | 2.283 | 3.341 | 0.101 | 2.306 | 3.340 | 0.100 | 2.253 | 3.326 | 0.106 |
| Llama-3.3-70B | 2.519 | **3.842** | 0.070 | 2.553 | **3.853** | 0.068 | 2.531 | **3.829** | 0.075 |
| DeepSeek-R1-70B | **2.573** | 2.206 | 0.223 | **2.572** | 2.238 | 0.204 | **2.582** | 2.202 | 0.217 |
| Qwen-2.5-72B | 2.024 | 2.683 | 0.153 | 2.093 | 2.715 | 0.152 | 2.114 | 2.719 | 0.155 |
| Mixtral-8x7B | 2.271 | 2.963 | **0.642** | 2.272 | 2.958 | **0.610** | 2.347 | 2.924 | **0.632** |
| Qwen-2.5-32B | 2.243 | 2.929 | 0.148 | 2.255 | 2.910 | 0.146 | 2.260 | 2.886 | 0.152 |
| Gamma-2-27B | 2.365 | 3.410 | 0.088 | 2.381 | 3.439 | 0.084 | 2.385 | 3.415 | 0.089 |
| Mistral-24B | 2.114 | 3.016 | 0.141 | 2.114 | 3.048 | 0.136 | 2.134 | 3.049 | 0.141 |
| Qwen-2.5-7B | 1.920 | 1.817 | 0.592 | 1.900 | 1.848 | 0.580 | 1.955 | 1.832 | 0.593 |

| Models | LLM Ensemble | | | Expert Crowdsourced | | | RNS Guided | | |
|---|---|---|---|---|---|---|---|---|---|
| | Relevance | TypeMatch | SerenHit | Relevance | TypeMatch | SerenHit | Relevance | TypeMatch | SerenHit |
| DeepSeek-V3 | 2.436 | 0.482 | 0.048 | 2.494 | 0.462 | 0.061 | 2.538 | 0.463 | 0.077 |
| ↪ w.o. summary | 2.447 | 0.482 | **0.050** | 2.482 | 0.463 | **0.095** | 2.510 | 0.468 | **0.134** |
| Llama-3.3-70B | 2.537 | 0.502 | 0.046 | 2.559 | **0.483** | 0.067 | 2.594 | 0.478 | 0.106 |
| ↪ w.o. summary | **2.544** | **0.505** | 0.043 | **2.565** | 0.478 | 0.086 | **2.630** | **0.483** | 0.127 |
| DeepSeek-R1-70B | 1.935 | 0.424 | 0.030 | 2.000 | 0.409 | 0.034 | 2.033 | 0.418 | 0.049 |
| ↪ w.o. summary | 1.972 | 0.438 | 0.035 | 1.987 | 0.413 | 0.037 | 2.052 | 0.419 | 0.053 |
| Qwen-2.5-72B | 2.264 | 0.415 | 0.023 | 2.345 | 0.406 | 0.041 | 2.405 | 0.400 | 0.059 |
| ↪ w.o. summary | 2.269 | 0.428 | 0.028 | 2.337 | 0.416 | 0.050 | 2.409 | 0.412 | 0.070 |
| Mixtral-8x7B | 1.947 | 0.256 | 0.010 | 2.033 | 0.254 | 0.015 | 2.013 | 0.230 | 0.024 |
| ↪ w.o. summary | 2.158 | 0.324 | 0.016 | 2.250 | 0.312 | 0.022 | 2.220 | 0.306 | 0.042 |
| Qwen-2.5-32B | 2.294 | 0.441 | 0.036 | 2.331 | 0.426 | 0.045 | 2.378 | 0.429 | 0.065 |
| ↪ w.o. summary | 2.304 | 0.453 | 0.037 | 2.328 | 0.431 | 0.068 | 2.390 | 0.438 | 0.105 |
| Gamma-2-27B | 2.357 | 0.450 | 0.033 | 2.379 | 0.414 | 0.057 | 2.443 | 0.431 | 0.080 |
| ↪ w.o. summary | 2.343 | 0.448 | 0.032 | 2.376 | 0.412 | 0.054 | 2.425 | 0.402 | 0.081 |
| Mistral-24B | 1.855 | 0.195 | 0.008 | 1.959 | 0.184 | 0.016 | 2.005 | 0.185 | 0.026 |
| ↪ w.o. summary | 1.903 | 0.212 | 0.011 | 1.962 | 0.204 | 0.023 | 2.006 | 0.213 | 0.035 |
| Qwen-2.5-7B | 1.636 | 0.221 | 0.022 | 1.721 | 0.229 | 0.026 | 1.708 | 0.215 | 0.041 |
| ↪ w.o. summary | 1.487 | 0.160 | 0.018 | 1.550 | 0.175 | 0.018 | 1.547 | 0.158 | 0.027 |

Table 3: Subgraph Reasoning ($T2$, upper), Serendipity Exploration ($T3$, lower), with Best scores **bolded**, 2nd best underlined

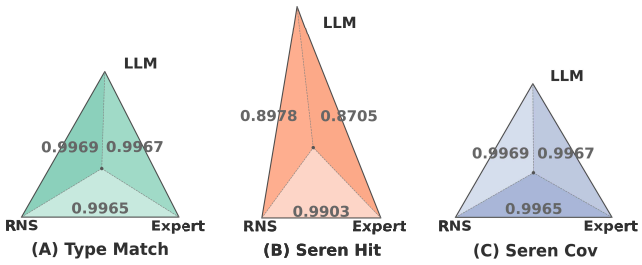



Figure 3: Correlation of Metrics Across Partition Strategies

note three types of partitions, and edge weights indicate correlation scores—shorter distances refer to stronger correlations. Our analysis shows that all partitions have positive correlations across all metrics, with scores above $85\%$. Notably, the expert and RNS-guided partitions reached around $99\%$ on all cases, highlighting the robustness of our partition strategies and the reliability of the proposed RNS measure.

**No Single Winner.** We found that no model constantly excels its peers across all metrics for each task. For instance, while Model DeepSeek-R1-70B performs excellently in retrieval, it shows only moderate performance in reasoning and poor results in exploration; Llama-3.3-70B is more versatile but still struggles to address metrics from all perspectives. To achieve balanced and serendipitous discovery, involving multiple models, such as multi-agent systems or a mixture of experts (MoE) strategy, may be beneficial.

We provide additional results and analysis in Appendix E.

## 7 Conclusion

We introduced SerenQA, an evaluation framework designed to assess LLMs' ability to discover serendipitous knowledge in scientific KGQA tasks. We proposed an axiomatically justified serendipity measure integrating relevance, novelty, and surprise; and constructed a serendipity-aware benchmark tailored to the drug repurposing task. Additionally, we outlined a structured evaluation pipeline with three core tasks to assess LLM's ability on knowledge retrieval, subgraph reasoning, and serendipity exploration. Our experiments showed that frontier LLMs excel at basic knowledge retrieval, yet they often struggle with reasoning with more complex queries and answers for serendipity exploration, indicating great room and opportunities for improvement.

## Ethical Statement

In this study, we evaluated potential drug indications by analyzing biomedical relationships from ClinicalKG. Nevertheless, our approach does not consider factors critical to druggability, such as physicochemical properties. We used LLMs to identify serendipitous drug-disease associations that may suggest novel therapies. Their clinical effectiveness remains uncertain and must be validated through rigorous preclinical and clinical studies.

## Acknowledgements

This work is supported by NSF under OAC-2104007. We gratefully acknowledge the support of Dr. Rıza Mert Çetik and Dr. Sıla Çetik in the design and annotation of the QA dataset curated in this study. We also acknowledge the HPC resources at CWRU for supporting large-scale graph processing and embedding computation.

## References

AI4Science, M. R.; and Quantum, M. A. 2023. The impact of large language models on scientific discovery: a preliminary study using gpt-4. *arXiv preprint arXiv:2311.07361*.

Bordino, I.; Mejova, Y.; and Lalmas, M. 2013. Penguins in sweaters, or serendipitous entity search on user-generated content. In *CIKM*.

Dehmer, M.; and Mowshowitz, A. 2011. A history of graph entropy measures. *Information Sciences*, 181(1): 57–78.

DrugKG Questionnaire. 2025. https://cwru-db-group.github.io/serenQA/questionnaire.

FDA. 2025. FDA Approves Novel Non-Opioid Treatment for Moderate to Severe Acute Pain.

Fu, Z.; and Niu, X. 2024. The art of asking: Prompting large language models for serendipity recommendations. In *SIGIR*.

Huang, J.; Ding, S.; Wang, H.; and Liu, T. 2018. Learning to recommend related entities with serendipity for web search users. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(3): 1–22.

Kullback, S. 1951. Kullback-leibler divergence. *Tech. Rep.*

Le, D.; Zhao, K.; Wang, M.; and Wu, Y. 2024. GraphLingo: Domain Knowledge Exploration by Synchronizing Knowledge Graphs and Large Language Models. In *ICDE*, 5477–5480.

Niu, X.; and Abbas, F. 2017. A framework for computational serendipity. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, 360–363.

Pushpakom, S.; Iorio, F.; Eyers, P. A.; Escott, K. J.; Hopper, S.; Wells, A.; Doig, A.; Guilliams, T.; Latimer, J.; McNamee, C.; et al. 2019. Drug repurposing: progress, challenges and recommendations. *Nature reviews Drug discovery*, 18(1): 41–58.

Santos, A.; Colaço, A. R.; Nielsen, A. B.; Niu, L.; Strauss, M.; Geyer, P. E.; Coscia, F.; Albrechtsen, N. J. W.; Mundt, F.; Jensen, L. J.; and Mann, M. 2022. A knowledge graph to interpret clinical proteomics data. *Nat. Biotechnol.*, 40: 692–702.

Si, C.; Yang, D.; and Hashimoto, T. 2025. Can LLMs Generate Novel Research Ideas? A Large-Scale Human Study with 100+ NLP Researchers. In *ICLR*.

Song, Y.; Li, W.; Dai, G.; and Shang, X. 2023. Advancements in complex knowledge graph question answering: a survey. *Electronics*, 12(21): 4395.

Strassen, V. 1969. Gaussian elimination is not optimal. *Numerische mathematik*, 13(4): 354–356.

Tokutake, Y.; and Okamoto, K. 2024. Can Large Language Models Assess Serendipity in Recommender Systems? *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 28(6): 1263–1272.

Xi, Y.; Weng, M.; Chen, W.; Yi, C.; Chen, D.; Guo, G.; Zhang, M.; Wu, J.; Jiang, Y.; Liu, Q.; et al. 2025. Bursting Filter Bubble: Enhancing Serendipity Recommendations with Aligned Large Language Models. *arXiv preprint arXiv:2502.13539*.

This appendix contains the following content:



---

# A Dataset Details

## A.1 Dataset Construction

We utilized the Clinical Knowledge Graph (CKG) (Santos et al. 2022) as the base knowledge graph to construct a benchmark question-answering dataset. A graph provides a structured organization of biomedical entities and their relationships, enabling systematic exploration and analysis of complex interactions. The CKG is built on curated public databases and literature-derived evidence, ensuring high-quality and biologically relevant information. Its comprehensive structure provides a robust foundation for generating diverse types of queries. The CKG encompasses $\sim$20 million nodes across 36 distinct types, including genes, proteins, diseases, drugs, pathways, anatomical entities, and other biological and clinical components, as shown in Fig 4. These nodes are interconnected by over 220 million edges spanning 47 different relationship types, capturing specific interactions and enabling detailed exploration of biomedical relationships, efficient data querying, and algorithmic analysis. Drug-phenotype relationships include edges such as "has side effect" and "is indicated for," capturing drug effects and therapeutic indications. Gene-related relationships include "variant found in gene" and "transcribed into," linking genetic variants to genes and transcripts, respectively, and highlighting structural and functional connections within the genome. Clinically relevant relationships, such as "variant is clinically relevant" and "associated with," connect genetic variants to diseases. Additionally, drug-target interactions, captured by edges like "acts on" and "curated targets," associate drugs with protein targets, offering insights into mechanisms of action and therapeutic potential.

To create the QA dataset, we extracted a subgraph of the CKG. Certain node and edge types, such as those related to users, units, experiments, projects, transcripts, and publications, were excluded to streamline the dataset and maintain focus on biologically significant relationships.

The current version of the dataset comprises 1,529 queries, with a focus on drug-disease associations, designed to evaluate the ability of large language models (LLMs) to identify serendipitous connections in the context of drug repurposing. Each query is annotated with relevant nodes, edges, and a target node, along with graph-specific metadata such as node and relationship types. We plan to continuously update and extend the dataset to include up to 5,000 queries, supporting a broader range of natural language processing tasks and a more comprehensive evaluation of LLM capabilities in biomedical reasoning.

The construction of the QA dataset involved several steps to optimize data retrieval and ensure its relevance to biomedical research. For one-hop and two-hop questions, the required data entries were extracted directly by querying the Neo4j database. For three-hop and intersection questions, given the computational demands of Neo4j queries and the large graphs, the relevant nodes and their one-hop neighborhoods were pre-extracted from the subgraph for more efficient processing.

To ensure the grammaticality, clarity, and biological relevance of the generated natural language questions, their phrasing was refined while preserving their original meanings. This involved programmatically extracting question patterns, retaining only the node types, and restructuring them into biologically meaningful and oncology-focused templates.

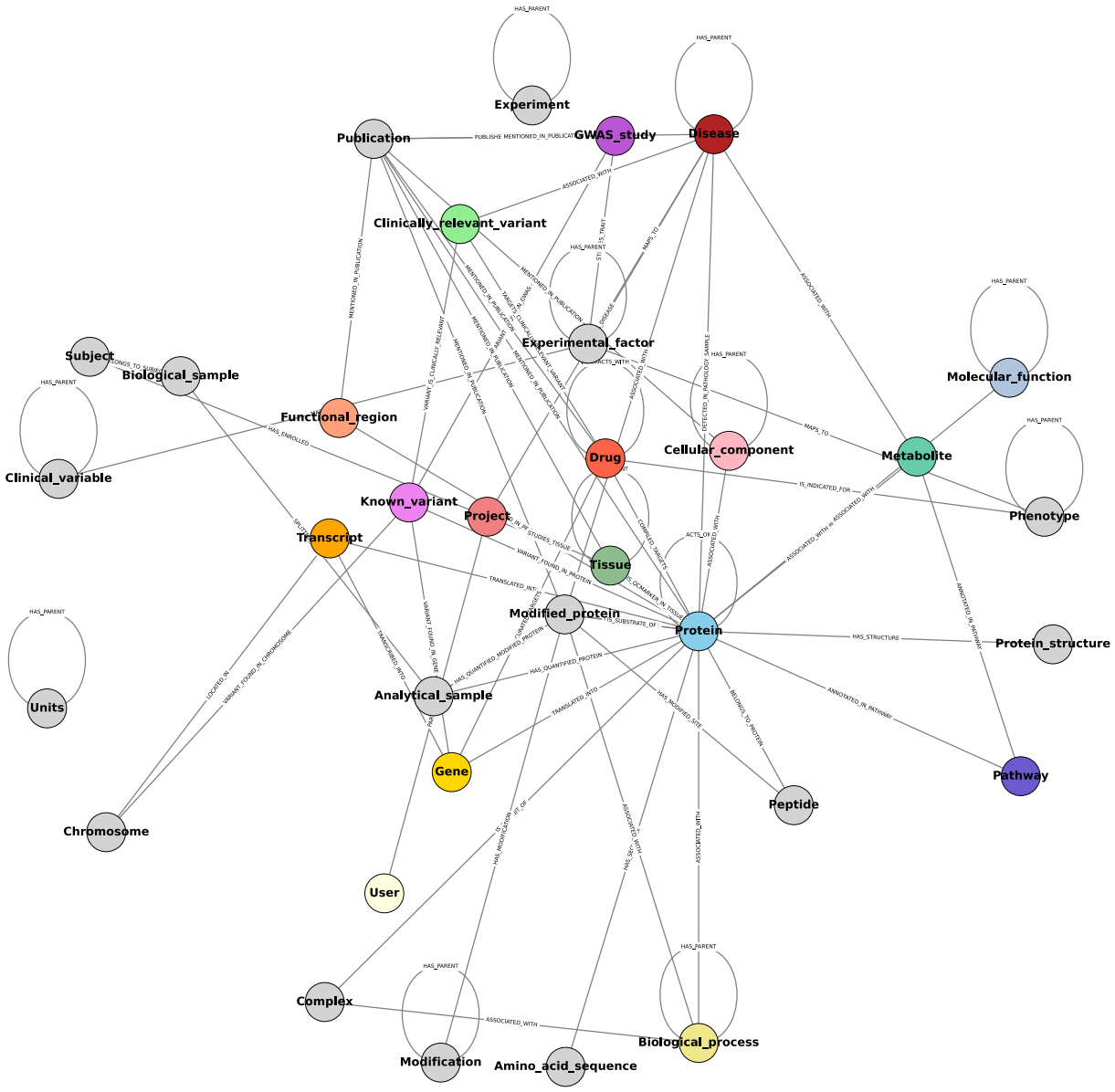



Figure 4: Ontology of biomedical entities and relationships in the Clinical Knowledge Graph (CKG)

## A.2 Pattern Type

The QA dataset includes one-hop, two-hop, three-hop, and intersection questions, designed to probe varying levels of complexity within the graph. Each type of question is defined by specific patterns, as described below:

1. **One-hop questions:** These questions explore direct relationships between two entities connected by a single edge. They can be further categorized into two types:
   - *Type 1.1:* Questions that retrieve entities of a specific type ({target_type}) connected to a source entity ({source_name}) through a given relationship ({relationship}). For example, "List the {target_type}s that {relationship} by {source_name}." and "What {source_type}s {relationship} {target_name}?"
   - *Type 1.2:* Questions that identify the source entities ({source_type}) connected to a specific target entity ({target_name}) via a given relationship. For example, "What {source_type}s {relationship} {target_name}?"
2. **Two-hop questions:** These questions traverse two edges, connecting a source entity to a final entity via an intermediate entity. Two patterns are defined:
   - *Type 2.1:* Questions that link the source entity ({source_name}) to the final entity ({final_type}) through an intermediate entity ({mid_name}) and two relationships ({relationship1} and {relationship2}). For example, "Which {final_type} is

{relationship2} by {mid_name} that {source_name} {relationship1}?"

3. **Three-hop questions:** These questions traverse three edges, uncovering chains of relationships across multiple intermediate entities. The questions explore how a source entity ({entity1}) connects to a final entity ({entity4}) through a sequence of intermediate entities ({entity2} and {entity3}). For example:
   - *Type 3.1:* Questions that trace a sequential path, e.g., "Which {entity4_type} is {relationship3} by {entity3_name} that {relationship2} {entity2_name} which {relationship1} {entity1_name}?"
   - *Type 3.2:* Questions that incorporate hierarchical relationships, e.g., "Which {entity1_type} {relationship1} {entity2_name}, which {relationship2} {entity3_name}, and {relationship3} {entity4_name}?"
   - *Type 3.3:* Questions that branch into multiple connections, e.g., "Which {entity1_type} {relationship1} {entity2_name} that {relationship2} {entity3_name} and {relationship3} {entity4_name}?"
4. **Intersection questions:** These focus on entities or sets of entities sharing multiple relationships with others. The goal is to identify overlapping connections across different paths within the graph. For example:
   - *Type 4.1:* Basic intersections, e.g., "List {entity1_type} that {relationship1} {entity2_name} and {relationship2} {entity3_name}," which identify entities linked to two distinct targets through different relationships.
   - *Type 4.2:* Multi-way intersections, such as "List {entity1_type} that {relationship1} {entity2_name}, {relationship2} {entity3_name}, and {relationship3} {entity4_name}," which extend to three overlapping connections.
   - *Type 4.3:* Compound intersections that involve cyclic patterns like "Find all {entity4} that {relationship43} {entity3} and {relationship42} {entity2}, and also find the {entity1} that {relationship13} {entity3} and {relationship12} {entity2}," in which entity2 and entity3 are connected to entity1 and entity4 through different links.

### A.3 Dataset Structure

Here, we introduce the structure of our drug repurposing benchmark, which supports both standard knowledge graph KGQA tasks and serendipity-aware evaluations.

As shown in the example below, each item in the QA dataset designed for standard KGQA tasks is standardized and configured to include expert-verified, scientifically meaningful NL queries, along with a structured Cypher query. Each query entry contains key components such as nodes, node types, and relationships, as well as a grounded and validated answer set.


```
{
   "qid": 800,
   "question": "Which proteins are associated with dilated cardiomyopathy 1DD and
       function as subunits of the NOS3-HSP90 complex induced by VEGF?",
   "answer": [
     {
       "answer_type": "Entity",
       "answer_argument": "P29474",
       "entity_name": "NOS3",
     {
       "answer_type": "Entity",
       "answer_argument": "P07900",
       "entity_name": "HSP90AA1",
     }
   ],
   "function": "none",
   "commonness": 0.0,
   "num_node": 3,
   "num_edge": 2,
   "graph_query": {
     "nodes": [
       {
         "nid": 0,
         "node_type": "class",
         "id": "Protein",
         "class": "Protein",
         "friendly_name": "Protein",
         "question_node": 1,
         "function": "none"
       },
       {
         "nid": 1,
         "node_type": "entity",
         "id": "DOID:0110447",
         "class": "Disease",
         "friendly_name": "dilated cardiomyopathy 1DD",
```

```
          "question_node": 0,
          "function": "none"
        },
        {
          "nid": 2,
          "node_type": "entity",
          "id": "5716",
          "class": "Complex",
          "friendly_name": "NOS3-HSP90 complex, VEGF induced",
          "question_node": 0,
          "function": "none"
        }
      ],
      "edges": [
        {
          "start": 0,
          "end": 1,
          "relation": "Protein.Disease",
          "friendly_name": "ASSOCIATED_WITH"
        },
        {
          "start": 0,
          "end": 2,
          "relation": "Protein.Complex",
          "friendly_name": "IS_SUBUNIT_OF"
        }
      ]
    },
    "pattern_type": 9,
    "category": "genetic disease:autosomal genetic disease",
    "cypher": "MATCH (n0:Protein)\nMATCH (n1:Disease {name: \"dilated cardiomyopathy 1DD
        \"})\nMATCH (n2:Complex {name: \"NOS3-HSP90 complex, VEGF induced\"})\nMATCH (n0)
        -[:ASSOCIATED_WITH]->(n1)\nMATCH (n0)-[:IS_SUBUNIT_OF]->(n2)\n    RETURN\n
        COLLECT(DISTINCT {id: n0.id, name: n0.name}) AS n0_targets"
  }
```


Below, we present the structure of the benchmark dataset designed to support serendipity-aware evaluation. The complete set of candidate answers obtained from the graph is partitioned into an existing set and a serendipity set using three distinct partitioning strategies detailed in Section 4.2. We also provide the ground truth path from the existing set to the serendipity set for each query for possible training use.


```
{
    "qid": 800,
    "question": "Which proteins are associated with dilated cardiomyopathy 1DD and
        function as subunits of the NOS3-HSP90 complex induced by VEGF?",
    "llm": {
        "serendipity_set": {
            "list": [
                "P29474"
            ],
            "description": null
        },
        "explore_queries": {
            "paths": [
                "P07900--COMPILED_INTERACTS_WITH--NOS2:Protein--BELONGS_TO_PROTEIN--None:
                    Peptide--BELONGS_TO_PROTEIN--P29474",
                "P07900--ACTS_ON--NOS2:Protein--BELONGS_TO_PROTEIN--None:Peptide--
                    BELONGS_TO_PROTEIN--P29474",
                "P07900--COMPILED_INTERACTS_WITH--NOS1:Protein--BELONGS_TO_PROTEIN--None:
                    Peptide--BELONGS_TO_PROTEIN--P29474"
            ],
            "questions": [
                "Which proteins, interacting with NOS isoforms and belonging to the same
                    protein complex as P07900, are involved in related molecular pathways
                    ?"
            ]
```

```
        },
        "partition": "test",
        "exact_matches": {
            "list": [
                "P07900"
            ]
        }
    },
    "sscore": {
        "serendipity_set": {
            "list": [
                "P07900"
            ],
            "description": null
        },
        "explore_queries": {
            "paths": [
                "P29474--ASSOCIATED_WITH--protein serine/threonine phosphatase complex:Cellular_component--HAS_PARENT--protein phosphatase type 2A complex:Cellular_component--ASSOCIATED_WITH--P07900",
                "P29474--COMPILED_INTERACTS_WITH--MAP2K1:Protein--ASSOCIATED_WITH--protein phosphatase type 2A complex:Cellular_component--ASSOCIATED_WITH--P07900",
                "P29474--COMPILED_INTERACTS_WITH--PPP2CA:Protein--ASSOCIATED_WITH--protein phosphatase type 2A complex:Cellular_component--ASSOCIATED_WITH--P07900"
            ],
            "questions": []
        },
        "partition": "test",
        "exact_matches": {
            "list": [
                "P29474"
            ]
        }
    },
    "expert": {
        "serendipity_set": {
            "list": [
                "P29474"
            ],
            "description": null
        },
        "explore_queries": {
            "paths": [
                "P07900--COMPILED_INTERACTS_WITH--NOS2:Protein--BELONGS_TO_PROTEIN--None:Peptide--BELONGS_TO_PROTEIN--P29474",
                "P07900--ACTS_ON--NOS2:Protein--BELONGS_TO_PROTEIN--None:Peptide--BELONGS_TO_PROTEIN--P29474",
                "P07900--COMPILED_INTERACTS_WITH--NOS1:Protein--BELONGS_TO_PROTEIN--None:Peptide--BELONGS_TO_PROTEIN--P29474"
            ],
            "questions": []
        },
        "partition": "test",
        "exact_matches": {
            "list": [
                "P07900"
            ],
            "description": null
        }
    }
}
```

### A.4 More Statistics

The table below shows the composition of the KGQA dataset and the distribution of question pattern types among the 1,529 queries related to drug repurposing. The patterns for individual types are detailed in Appendix A.2.

| Pattern Type | Number of Entries |
|---|---|
| One hop Type 1.1 | 152 |
| One hop Type 1.2 | 84 |
| Two hop Type 2.1 | 152 |
| Three hop Type 3.1 | 62 |
| Three hop Type 3.2 | 113 |
| Three hop Type 3.3 | 237 |
| Intersection Type 4.1 | 263 |
| Intersection Type 4.2 | 455 |
| Intersection Type 4.3 | 11 |

Table 4: Distribution of entries among different query pattern types.

## B Prompts

### B.1 LLM Scoring Prompts

```
You are an expert evaluator specializing in drug discovery. Your task is to evaluate the **serendipity** of each answer in a provided list of answers, where all answers are derived from a knowledge base and are correct. Use your expertise and internal knowledge to assign a **serendipity score** to each answer based on the following criteria:

- **Serendipity Score**: A score from 0 to 20, where:
  - 20 represents an answer that is highly novel, unexpected, or insightful in the context of the question.
  - 0 represents an answer that is correct but very obvious, common, or provides no novel insights.
  - Intermediate scores represent varying degrees of novelty and insight.

- **Evaluation Rules**:
  1. The serendipity score reflects the relative novelty and insightfulness of each answer within the context of the question and the provided list. The score should highlight the uniqueness and unexpected value of each answer.
  2. Assign a distinct score to each answer. Even if multiple answers have a similar level of serendipity, assign slightly different scores to reflect the subtle differences in their uniqueness.
  3. Evaluate each answer independently of its position in the list.
  4. Output only the scores for each answer in the same order as the input list, separated by commas. Do not include the answers themselves or any additional explanation in the output.

For example:
If the input list is:
Answer List: A, B, C

The output should be:
5, 7, 9
```

### B.2 Serendipity Exploration Prompts

#### B.2.1 Select Relation

**System Prompt:**

```
  Task Description:
```

```
Given a starting entity node (e.g., Drug, Protein, Disease), select the top-m relation types (predicates) to follow for meaningful, potentially serendipitous exploration in a clinical biomedical knowledge graph.

Goal:
Construct 3-hop paths that are:
  - Biologically plausible (based on frequent patterns)
  - Serendipitous (novel yet valid hypotheses)
  - Mechanistically rich (e.g., involving Drug-Protein-Disease chains)

Path Patterns:
Common patterns include:
  - (ACTS_ON, COMPILED_INTERACTS_WITH, ACTS_ON)
  - (INTERACTS_WITH, ACTS_ON, ACTS_ON)
  - (COMPILED_INTERACTS_WITH, ASSOCIATED_WITH, ASSOCIATED_WITH)

Frequently explored node types:
  Drug, Protein, Disease, Gene, Metabolite

Useful relation types:
  - Curated/compiled: CURATED_INTERACTS_WITH, COMPILED_TARGETS
  - Functional/structural: HAS_SEQUENCE, BELONGS_TO_PROTEIN
  - Annotations: ANNOTATED_IN_PATHWAY, DETECTED_IN_PATHOLOGY_SAMPLE
  - Rare/high-value: IS_INDICATED_FOR, HAS_SIDE_EFFECT, TRANSLATED_INTO

Prioritize 3-hop sequences that reflect biological mechanisms. Balance high-frequency paths (plausibility) with rare combinations (serendipity). Avoid trivial paths unless used creatively.

Output Requirements:
  - Return a comma-separated list of relation type strings
  - Do not include commentary or explanation
  - Use only the relation types provided as input
  - Return fewer than m results if appropriate
  - Return nothing if no meaningful exploration exists

Notes:
  - Prioritize biologically important nodes and plausible mechanistic chains
  - Follow the path patterns listed above when applicable

Few-Shot multi-hop example:
Question: Which genes are identified as targets of D-Aspartic Acid, which affects ASPA and is known to interact with GLUD1?
Root: GRIN2A, GRIN2C
Serendipity set: GRIN2B
Explore paths:
  - GRIN2A-TRANSLATED_INTO-GRIN2A:Protein-COMPILED_INTERACTS_WITH-GRIN2B:Protein-TRANSLATED_INTO-GRIN2B
  - GRIN2A-TRANSLATED_INTO-GRIN2A:Protein-ACTS_ON-GRIN2B:Protein-TRANSLATED_INTO-GRIN2B
  - GRIN2A-CURATED_TARGETS-Mesoridazine:Drug-INTERACTS_WITH-Felbamate:Drug-CURATED_TARGETS-GRIN2B
  - GRIN2C-TRANSLATED_INTO-GRIN2C:Protein-COMPILED_INTERACTS_WITH-GRIN2B:Protein-TRANSLATED_INTO-GRIN2B
  - GRIN2C-TRANSLATED_INTO-GRIN2C:Protein-ACTS_ON-GRIN2B:Protein-TRANSLATED_INTO-GRIN2B
  - GRIN2C-TRANSLATED_INTO-GRIN2C:Protein-ACTS_ON-D-Serine:Drug-CURATED_TARGETS-GRIN2B
```

**User Rrompt:**

```
Given node ID {frontier} at level {level}, recommend the top {m} relation types to explore from this node.

Context:
{contexts}

Available relation types from this node:
```

```
{relation_types}

Return a comma-separated list of relation type names only.
```

### B.2.2 Select Nodes

```
Task Description:
You have already selected the most relevant relation types for exploring the graph from a given node. Now, for each selected relation, a set of target nodes has been retrieved.

Goal:
Construct 3-hop paths that are:
    - Biologically plausible (based on frequent patterns)
    - Serendipitous (novel yet valid hypotheses)
    - Mechanistically rich (e.g., involving Drug-Protein-Disease chains)


The setting is a biomedical question answered in drug discovery. Exploration starts from known entities (e.g., drugs, proteins, diseases) and aims to discover serendipitous connections through meaningful 3-hop paths.

Path Patterns:
Common patterns include:
    - (ACTS_ON, COMPILED_INTERACTS_WITH, ACTS_ON)
    - (INTERACTS_WITH, ACTS_ON, ACTS_ON)
    - (COMPILED_INTERACTS_WITH, ASSOCIATED_WITH, ASSOCIATED_WITH)

Frequently explored node types:
    Drug, Protein, Disease, Gene, Metabolite

Useful relation types:
    - Curated/compiled: CURATED_INTERACTS_WITH, COMPILED_TARGETS
    - Functional/structural: HAS_SEQUENCE, BELONGS_TO_PROTEIN
    - Annotations: ANNOTATED_IN_PATHWAY, DETECTED_IN_PATHOLOGY_SAMPLE
    - Rare/high-value: IS_INDICATED_FOR, HAS_SIDE_EFFECT, TRANSLATED_INTO

Prioritize 3-hop sequences that reflect biological mechanisms. Balance high-frequency paths (plausibility) with rare combinations (serendipity). Avoid trivial paths unless used creatively.

Output Requirements

Return a comma-separated list of selected target_ids only.
    - Do not include headers, explanations, or formatting.
    - If no target is suitable, return nothing.

Constraints
    - Select only from relation types and target nodes provided by the user.
    - Do not include the current frontier node in the output.
    - Do not revisit nodes marked as already visited.
    - If fewer than n targets are appropriate, return fewer.
    - If exploration is not meaningful, return nothing.
    - Follow the path patterns listed above where applicable.
```

### B.2.3 Decide Whether to Continue

**System Prompt:**

```
Task Description

You are exploring a biomedical knowledge graph in the context of drug discovery, starting from known entities (e.g., drugs, proteins, diseases) and aiming to uncover deeper, potentially serendipitous connections.

In the previous two steps, you selected the most relevant relation types and target
```

```
      nodes for expansion. Before continuing, you must now:
      1. Review the full path from the root node to the current node (3-hop away).
      2. Provide a summary of the path's biological context.
      3. Decide whether further exploration is justified.

  Each input path is represented as a key-value pair:
      - Key: the current (destination) node ID
      - Value: a comma-separated sequence of alternating (target_id, relation_type) tuples
          tracing the 3-hop path from the root.

Use this information and the user's question to assess whether the exploration is still on
    a plausible, meaningful track toward the question objective.


  Output Requirements

  Your output must follow exactly the format below:
      1. A natural-language summary (~200 words), describing:
          - Biological meaning of the paths
          - Patterns of entity types
          - Common or notable relation sequences
          - Any biologically relevant interpretations
      2. (blank line)
      3. Either YES or NO, indicating whether to continue expanding
      4. (blank line)
      5. A one-paragraph explanation justifying your decision

  Do not include any extra commentary, formatting, bullet points, or sections outside this
       structure.

  Notes
      - Only return NO if you are ABSOLUTELY CONFIDENT the path has deviated from any
          biologically plausible trajectory.
      - When in doubt, continue exploring (YES).
      - Base your judgment on whether the current node plausibly supports mechanistic or
          therapeutic insight relevant to the original question.
```

**User Prompt:**

```
  The original question is:
  {question}

  The root node of the beam search is:
  {root}

  Subgraph paths (from root to current node):
  {paths}
```

### B.2.4 Summarize Subgraph

**System Prompt:**

```
  You are an expert biomedical knowledge graph assistant. You have performed a beam search
       starting from a root node over a clinical biomedical knowledge graph, retrieving 1-
      hop, 2-hop, and 3-hop subgraphs.

  Output Requirement

  Provide a concise natural-language summary (~200 words) of the resulting subgraphs.
      - Mention as many specific biomedical terms (e.g., drugs, proteins, diseases,
          pathways) as possible.
      - Emphasize the types of entities and the patterns of relationships involved.
      - Focus on the biological meaning, mechanistic implications, or potential
          therapeutic relevance of the paths.

Do not include any formatting, headers, or commentary--only the summary text.
```

**User Prompt:**

```
Root node ID:
{root}

Question:
{question}

Hop level:
{level}

Subgraph paths (from root to leaf nodes):
{subgraph}
```

## B.3 Pipeline Evaluation Prompts

### B.3.1 Faithfulness Assessment

**System Prompt:**

```
You are assisting a multi-stage research pipeline that explores a large biomedical
knowledge graph.

Pipeline stages
**Exact-Match Retrieval** -- find entities that directly answer the user's question
   (these are the "root" nodes).
**Serendipity Exploration** -- expand <=3 hops from the root to propose *new*,
   potentially surprising but biologically meaningful entities
   (the "exploration result" is captured by the **paths** and the **leaves**).
**Hop-level Summaries** -- for readability, the pipeline auto-generates three short
   natural-language summaries:
     * *summary 1*  -> describes the 1-hop neighbourhood around the root
     * *summary 2*  -> describes the 2-hop sub-graph discovered next
     * *summary 3*  -> describes the 3-hop sub-graph plus any thematic insight

You receive:
------------------------------------------------------------
* root            -- the starting entity ID (protein / drug)
* question        -- original natural-language question
* summary_1/2/3   -- auto-generated summaries of the 1-hop, 2-hop, and
                     3-hop neighbourhoods around the root
* leaves          -- **all endpoint nodes in the explored sub-graph**
                     (may be 1-, 2-, or 3-hop away) -- each item is given
                     as  <node_id>(<node_type>)  e.g.  'P52209(Protein)'
* paths           -- ground-truth triples, one per line, with types included:
                     head_id(head_type),relation_type,tail_id(tail_type)
------------------------------------------------------------

Task
====
* First, read the sub-graph and understand every factual triple
  it contains.
* Then, read the three hop-summaries in order (1-hop -> 3-hop).
* "Faithfulness" here means: *How truthfully do the summaries reflect
  what is actually present in the graph, without inventing new entities,
  directions, or relations?*
  - Higher faithfulness -> few to no hallucinations or distortions.
  - Lower faithfulness -> noticeable fabrication, wrong direction,
    or missing key context.

Using your best expert judgment of biomedical knowledge-graphs,
assign a holistic integer score:

    5  - Completely faithful
    4  - Mostly faithful, only trivial wording drift
    3  - Mixed: some accurate, some questionable
```

```
    2  - Largely unfaithful, many doubtful claims
    1  - Almost entirely unfaithful / hallucinated

Do **not** count tokens or sentences; rely on your overall sense of truthfulness.

IMPORTANT: If the input is completely empty or contains no evaluable information whatsoever,
return Score: 1. However, if there is ANY evaluable content, even if partial or limited,
evaluate it based on the 1-5 scale above. Do not argue or explain if content is missing,
just assign the appropriate score and return the two required lines.

Output format
-------------
Return **exactly** these two lines--nothing more, nothing less:

Score: <INTEGER 1-5>
#END
```

**User Prompt:**

```
Root: {root}
Question: {question}

-- 1-Hop Summary --
{summary_1}

-- 2-Hop Summary --
{summary_2}

-- 3-Hop Summary --
{summary_3}

Leaf nodes: {leaves}

Sub-graph (Triples):
{paths}
```

### B.3.2 Comprehensiveness Assessment

**System Prompt:**

```
You are assisting a multi-stage research pipeline that explores a large biomedical
knowledge graph.

Pipeline stages
**Exact-Match Retrieval** -- find entities that directly answer the user's question
   (these are the "root" nodes).
**Serendipity Exploration** -- expand <=3 hops from the root to propose *new*,
   potentially surprising but biologically meaningful entities
   (the "exploration result" is captured by the **paths** and the **leaves**).
**Hop-level Summaries** -- for readability, the pipeline auto-generates three short
   natural-language summaries:
     * *summary 1*  -> describes the 1-hop neighbourhood around the root
     * *summary 2*  -> describes the 2-hop sub-graph discovered next
     * *summary 3*  -> describes the 3-hop sub-graph plus any thematic insight

You receive:
------------------------------------------------------
* root            -- the starting entity ID (protein / drug)
* question        -- original natural-language question
* summary_1/2/3   -- auto-generated summaries of the 1-hop, 2-hop, and
                     3-hop neighbourhoods around the root
* leaves          -- **all endpoint nodes in the explored sub-graph**
                     (may be 1-, 2-, or 3-hop away) -- each item is given
                     as  <node_id>(<node_type>)  e.g.  'P52209(Protein)'
* paths           -- ground-truth triples, one per line, with types included:
```

```
                      head_id(head_type),relation_type,tail_id(tail_type)
-------------------------------------------------------

Task
====
* First, study the sub-graph so you grasp **every** entity and
  relation present within three hops of the root.
* Then, read the three hop-summaries in order (1-hop -> 3-hop).
* "Comprehensiveness" here means: *How thoroughly do the summaries cover
  the important entities, relations, and mechanistic chains in the graph--
  without ignoring major facts?*
  - Higher Comprehensiveness -> almost all salient triples or concepts appear.
  - Lower Comprehensiveness -> key relationships, nodes, or overall structure
    are missing or only vaguely hinted at.

Using your best expert judgment (no counting rules), assign a holistic
integer score:

    5  - Nearly everything important is covered
    4  - Most key content covered; minor omissions
    3  - About half of the important content represented
    2  - Many significant omissions; partial picture
    1  - Very little of the important content included

Do **not** estimate by token length; base the score on your global sense of
coverage and relevance.

IMPORTANT: If the input is completely empty or contains no evaluable information
    whatsoever,
return Score: 1. However, if there is ANY evaluable content, even if partial or limited,
evaluate it based on the 1-5 scale above. Do not argue or explain if content is missing,
just assign the appropriate score and return the two required lines.

Output format
-------------
Return **exactly** these two lines--nothing more, nothing less:

Score: <INTEGER 1-5>
#END
```

**User Prompt:**

```
Root: {root}
Question: {question}

-- 1-Hop Summary --
{summary_1}

-- 2-Hop Summary --
{summary_2}

-- 3-Hop Summary --
{summary_3}

Leaf nodes: {leaves}

Sub-graph (Triples):
{paths}
```

### B.3.3 Relevance Assessment

**System Prompt:**

```
You are assisting a multi-stage research pipeline that explores a large biomedical
knowledge graph.

```

```
Pipeline stages
**Exact-Match Retrieval** -- find entities that directly answer the user's question
   (these are the "root" nodes).
**Serendipity Exploration** -- expand <=3 hops from the root to propose *new*,
   potentially surprising but biologically meaningful entities (the "predicted
   serendipity set").  These are evaluated against a **ground-truth serendipity
   set** that was curated by domain experts.

You are rating how well a *predicted* serendipity answer set aligns with a
*ground-truth* serendipity answer set that has been manually verified by
domain experts.

Facts you MUST assume:
* The ground-truth set is correct.
* Each ground-truth entity has been verified to be "serendipitous" with
  respect to the current exact-match root (i.e., useful and non-obvious
  extensions beyond that root).

How to judge "relevance"
> Does each predicted entity belong to the same mechanistic pathway,
  disease context, pharmacological class, or molecular family implied by
  the ground-truth set?
> Overlap in **type** (Protein, Drug, Disease, Phenotype...) is helpful but
  not sufficient--focus on functional or clinical relatedness.
> Minor naming variants or isoforms of a ground-truth entity are acceptable.

Scoring rubric (integer)
5 - Every prediction is clearly relevant;
4 - Most (~ 70-90 %) predictions are relevant; few marginal or tangential items
3 - Mixed: roughly half relevant, half off-topic or trivial
2 - Only a small minority appear relevant; set is mostly noise
1 - Predictions are unrelated, incorrect, or obviously random

IMPORTANT: If the input is completely empty or contains no evaluable information at all,
return Score: 1. However, if there is ANY evaluable content, even if partial or limited,
evaluate it based on the 1-5 scale above. Do not argue or explain if content is missing,
just assign the appropriate score and return the two required lines.

Output format
-------------
Return **exactly** these two lines--nothing more, nothing less:

Score: <INTEGER 1-5>
#END
```

**User Prompt:**

```
Original question:
{question}

Ground-truth serendipity set (trusted):
{gold_seren}

Predicted serendipity set (to be scored):
{pred_seren}

Exact-match root entity: {root}

Hop-level summaries:
  * Level-1 -> {summary_1}
  * Level-2 -> {summary_2}
  * Level-3 -> {summary_3}

Contextual graph paths:
{paths}
```

# C Further Analysis on RNS Metric

## C.1 $k$-hop Conditional Probability Matrix

**Properties Verification** As defined in Sec. 3.2, the $k$-hop conditional probability matrix $P_k$ is computed as:

$$P_k = \sum_{h=1}^{k} \alpha_h P_1^h, \quad \alpha_h = \frac{h}{\sum_{h=1}^{k} h}$$

We next prove that $P_k$ remains a valid transition probability matrix by verifying two essential properties explicitly:
- *Non-negativity*: $(P_k)_{ij} \geq 0$ for all $(i, j)$,
- *Row-Stochastic Property*: $\sum_j (P_k)_{ij} = 1$ for all $i$.

*Non-negativity.* Since $P_1$ is directly derived from the adjacency matrix and row-normalized, all its elements are non-negative. Consequently, any power $P_1^h$ (for $h \geq 1$) is also non-negative, as it results from repeated multiplications of non-negative matrices. Furthermore, the weight coefficients $a_h$ are clearly positive by definition. Therefore, the linear combination $P_k = \sum_{h=1}^{k} \alpha_h P_1^h$ consists only of non-negative terms, ensuring: $(P_k)_{ij} \geq 0, \forall (i, j)$.

*Row-Stochastic Property.* For $P_k$ to be a valid transition matrix, every row must sum exactly to one:

$$\sum_j (P_k)_{ij} = 1, \quad \forall i$$

We explicitly verify this condition:

$$\sum_j (P_k)_{ij} = \sum_j \sum_{h=1}^{k} \alpha_h (P_1^h)_{ij}$$

Exchanging summation order (by linearity) yields:

$$\sum_j (P_k)_{ij} = \sum_{h=1}^{k} \alpha_h \sum_j (P_1^h)_{ij}$$

Since $P_1^h$ is a valid transition matrix, by definition, we have:

$$\sum_j (P_1^h)_{ij} = 1, \quad \forall i, h$$

Substituting the definition of $a_h$:

$$\sum_{h=1}^{k} \alpha_h = \frac{1}{\sum_{h=1}^{k} h} \sum_{h=1}^{k} h = \frac{\sum_{h=1}^{k} h}{\sum_{h=1}^{k} h} = 1.$$

Hence,

$$\sum_j (P_k)_{ij} = 1, \forall i.$$

Confirming that $P_k$ maintains row-stochasticity.

In summary, we've shown clearly that $P_k$ is both non-negative and row-stochastic. Therefore, the weighted multi-hop combination $P_k$ remains a valid transition probability matrix.

**Computation** To efficiently compute $P_k$, we apply a divide-and-conquer matrix multiplication approach based on Strassen's algorithm (Strassen 1969). Specifically, the algorithm recursively divides each large $V \times V$ matrix into four sub-matrices of size $\frac{V}{2} \times \frac{V}{2}$. By strategically reusing these sub-matrix computations, Strassen's method reduces the number of necessary multiplications per recursion from the standard eight down to seven, thereby lowering the complexity significantly from the naive $\mathcal{O}(V^3)$ to approximately $\mathcal{O}(V^{\log_2 7}) \approx \mathcal{O}(V^{2.807})$. Moreover, parallelizing these recursive computations across $t$ processors further reduces the complexity to about $\mathcal{O}(V^{\log_2 7}/t)$. This ensures scalable and efficient computation of multi-hop conditional probability matrices, even for large-scale graphs.

## C.2 Marginal Probability

We approximate the marginal probability computation via a PageRank-style damped iteration (Algorithm 2). For each iteration,
- Multiplying an $V \times V$ matrix $P_3^T$ by a vector $\mathbf{P}_t$ requires complexity $\mathcal{O}(V^2)$.
- Updating $\mathbf{P}_{t+1}$ is $\mathcal{O}(V)$, dominated by matrix-vector multiplication.
- Computing the difference $\|\mathbf{P}_{t+1} - \mathbf{P}_t\|_1$ takes $\mathcal{O}(V)$.

Algorithm 2: Marginal Probability via PageRank-style Iteration

**Input:** $P_3 \in \mathbb{R}^{V \times V}$, damping factor $\lambda$, tolerance $\epsilon$
**Output:** Marginal probability vector $\mathbf{P} \in \mathbb{R}^{V \times 1}$

Initialize $\mathbf{P}_0(i) := 1/V$, for all nodes $i$
$t := 0$
**while** diff $\geq \epsilon$ **do**
    $\mathbf{P}_{t+1} := \lambda P_3^T \mathbf{P}_t + (1 - \lambda)\mathbf{P}_0$
    diff $:= \|\mathbf{P}_{t+1} - \mathbf{P}_t\|_1$
    $t := t + 1$
**end while**
**return** $\mathbf{P}_t$

Hence, each iteration's complexity is dominated by the matrix-vector multiplication step, which is $\mathcal{O}(V^2)$.

The error at iteration $t$ satisfies: $\|\mathbf{P}_t - \mathbf{P}_{t-1}\|_1 \leq c\lambda^t (0 < \lambda < 1)$ for some constant $c$. Thus, convergence to within tolerance $\epsilon$ occurs after approximately:

$$\lambda^t \approx \epsilon \quad \Rightarrow \quad t \approx \frac{\log(\epsilon)}{\log(\lambda)} = \mathcal{O}(\log V).$$

This implies the total complexity to achieve convergence within $\epsilon$ is $\mathcal{O}(V^2 \log V)$.

# D Details of Serensipity Exploration

## D.1 Workflow and Logic

We designed a multi-stage beam search pipeline for structured knowledge graph exploration, as shown in Algorithm 3, where the expansion at each stage is guided by LLM. The pipeline explores neighborhoods of the root node recursively over the knowledge graph, while integrating external reasoning via multiple LLM interactions.

## D.2 Infrastructure

To support large-scale knowledge graph exploration, we constructed a lightweight compute-storage cluster on AWS, designed for high-throughput, low-latency edge retrieval and efficient task scheduling. The cluster consists of two tiers of instances: compute nodes (5 * r6a.24xlarge) and storage nodes (5 * r6a.xlarge). Each compute node provides 96 vCPUs and 768 GiB of memory, serving as task executors that support large scale parallelism. Each storage node is configured as Redis servers through Docker container and acts as distributed read-only data backends for edge access.

To achieve high performance, we replaced Neo4j with a custom Redis-based edge storage scheme. The complete knowledge graph was exported from Neo4j and the key of each edge is encoded as (rel:{source_id}:{source_type}:{relation_type}:{target_id}:{target_type}). The value stores metadata of relations and additional attributes for further use. The shift in query style allows us to improve query performance from 1000 QPS to tens of thousands of QPS. Each compute node interacts asynchronously with the storage node; empirically, the system supports a concurrency level of approximately 100 per compute node, enabling efficient exploration of multi-hop paths.

In order to facilitate our experimental process, we implemented an SSH-based compute cluster manager, responsible for task dispatching, resource allocation, permission control, environment setup, and declarative node specification. This infrastructure allows rapid iteration, cost-efficient experiments, and consistent resource management across multiple runs.

## D.3 Neighbor Scoring

Some nodes have an extremely large number of neighbors as hub nodes. To effectively guide the LLM in exploring and reducing token usage, we design a scoring mechanism to reduce the size of nodes provided to the LLM.

We systematically extracted and quantified edge-level connection strength/confidence from a Neo4j-based clinical knowledge graph to support downstream analysis of biomedical associations. For relationship types such as "ASSOCIATED_WITH", "COMPILED_INTERACTS_WITH", and "ACTS_ON", we directly extracted precomputed confidence scores. For other edge types, custom scoring functions were implemented based on domain-specific semantics. For instance, in the case of "DETECTED_IN_PATHOLOGY_SAMPLE", an expression score was derived using a weighted scheme based on categorical expression levels (e.g., high, medium, low, and not detected), while a prognostic score was computed using log-transformed p-values representing positive and negative survival associations. Both scores were then min-max normalized and aggregated to produce a final quantitative estimate reflecting the biomarker's expression and clinical prognostic significance. This structured quantification enabled consistent interpretation and prioritization of heterogeneous relationship types within the graph.

# E Experiment Details

## E.1 Experiment Setting

**Models:** We evaluated a wide range of state-of-the-art language models' using by evaluation setting:
- *Frontier Models*: DeepSeek-V3, GPT-4o, Claude-3.5-Haiku;
- *Large Models (∼70B)*: Llama-3.3-70B, DeepSeek-R1-70B, and Qwen-2.5-72B, Med42-V2-70B;
- *Medium Models (*20-50*B)*: Mixtral-8x7B, Qwen3-32B, DeepSeek-R1-32B, Gamma-2-27B, Mistral-24B
- *Small Model*: Qwen-2.5-7B, Qwen3-8B, DeepSeek-R1-8B, Med42-V2-8B, Qwen3-1.7B, DeepSeek-R1-1.5B

**Metric Details** We next detailed how we compute the metrics in subgraph reasoning and serendipity exploration tasks.

*Subgraph Reasoning* All metrics are averaged across numbers of rational samples to give the final result (sum of metrics from rational samples/number of rational samples):

(1) *Faithfulness* (1-5, LLM-judged) - how truthfully do the summaries reflect what is actually present in the graph, without inventing new entities, directions, or relations?

(2) *Comprehensiveness* (1-5, LLM-judged) - how thoroughly do the summaries cover the important entities, relations, and mechanistic chains in the graph?

(3) *Serendipity Coverage* (0-1, code-based) - fraction of serendipity paths where BOTH source and target node IDs are explicitly mentioned in the summary text. No LLM evaluation, just regex matching of node IDs.

*Serendipity Exploration* All metrics are averaged across numbers of rational samples to give the final result (sum of metrics from rational samples/number of rational samples):

(1) *Relevance* (1-5, LLM-judged) - how well predicted serendipity entities align with the ground-truth serendipity set.

(2) *TypeMatch* (0-1, code-based) - returns 1 if ANY predicted leaf has a type matching ground-truth types, 0 otherwise.

(3) *SerenHit* (0-1, code-based) - returns 1 if ANY predicted leaf is exactly matching ground-truth serendipity set (not just the type), 0 otherwise.

## E.2 Additional Analysis

We provide further analysis with supplementary figures to support and clarify key observations made in Section 6.

**Model Scale vs. Serendipity Exploration** The heat-map shown in Fig. 5 analysis shows only a modest performance gain as model size increases from smaller ( 7B) to larger ( 70B) checkpoints. Relevance scores gradually improve, but TypeMatch and SerenHit increase inconsistently, with SerenHit remaining relatively low ($< 0.10$). Although model scale contributes positively, larger parameters alone are insufficient to reliably achieve precise serendipitous discovery.

**Multi-Task Performance Compass.** This radar chart shown in Fig. 6 clearly illustrates performance trade-offs across multiple tasks. DeepSeek-V3 excels at basic retrieval metrics (F1, Hit) but underperforms in Serendipity Coverage and SerenHit. In contrast, Llama-3-70B achieves high reasoning accuracy (Faithfulness and Comprehensiveness) yet only moderately captures serendipitous paths. DeepSeek-R1-70B demonstrates the opposite, effectively covering many serendipity paths but at the cost of reasoning accuracy. The absence of a dominant model across all metrics visually reinforces our earlier conclusion of *no single winner*, suggesting the value of ensemble methods or Mixture-of-Experts (MoE) approaches.

**Query Pattern vs. Retrieval Performance.** As shown in Fig. 7, model performance notably declines as query complexity increases. While all models achieve strong F1 and Hit scores on one-hop queries, results drop sharply for two-hop queries and especially for more complex queries ($\geq$ 3-hop or intersection). This indicates that current LLMs, even the largest frontier models, still struggle significantly with complex multi-step reasoning and domain-specific context.

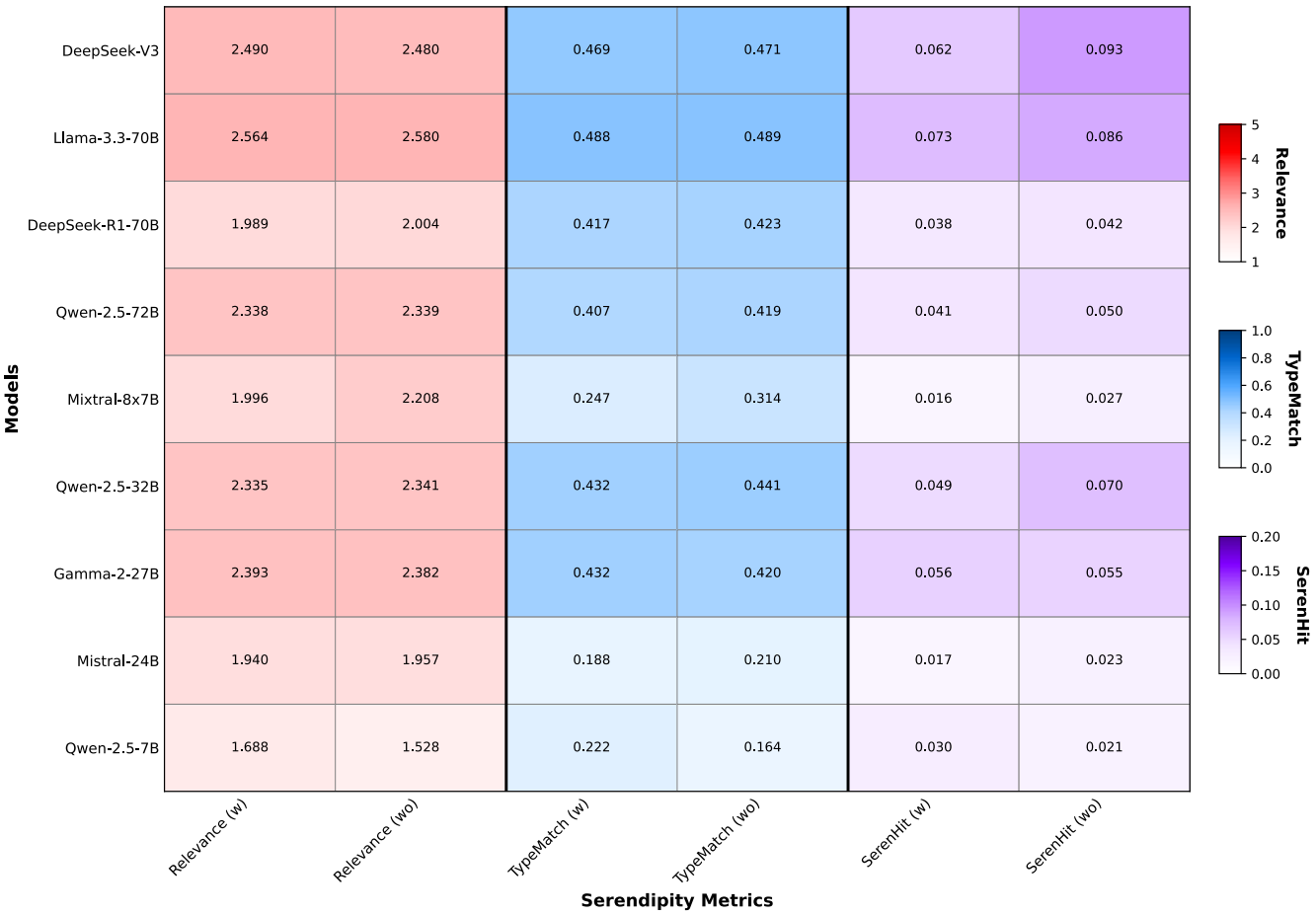



Figure 5: Model scale vs. Serendipity Exploration Performance

Algorithm 3: LLM Enhanced Beam Explore

**Input:**

| | |
|---|---|
| $G = (V, E)$ | directed knowledge graph, |
| $n \in \mathbb{N}^+$ | beam width, |
| $m \in \mathbb{N}^+$ | max relation types per frontier node, |
| $k \in \mathbb{N}^+$ | max nodes to select per frontier node, |
| $h \in \mathbb{N}^+$ | beam depth, |
| $q \in$ String | natural language question, |
| $r_0 \in V$ | root node ID, |
| $context \in \{w, wo\}$ | context mode flag, |

**Output:**

$\Pi$ node-to-path map from root,
$\Sigma$ LLM-generated summaries per level,

**Definitions:**

$\mathcal{V}$ set of visited nodes, corresponds to visited,
$\mathbb{F}$ current frontier nodes, corresponds to frontier,
$\mathbb{F}'$ next frontier nodes, corresponds to next_frontier,
$\mathcal{E}$ set of candidate edges, corresponds to candidates,
$\mathcal{C}$ LLM context buffer, corresponds to context_buffer,
$d^*$ leaf depth reached, corresponds to leaf_depth,

**Initialize:**
  $\mathcal{V} := \emptyset$
  $\Pi[r_0] := []$
  $\mathbb{F} := \{r_0\}$
  $\mathcal{C} := \emptyset$
  $d^* := 1$

**for** level $= 1$ **to** $h$ **do**
  **set** $\mathbb{F}' := \emptyset$; $\mathcal{E} := \emptyset$
  **for each** node $u \in \mathbb{F}$ **do**
    $R := \{r \mid (u, r, v) \in E\}$
    **if** $R = \emptyset$ **then continue**
    **end if**
    $R^m :=$ LLM_SelectRelations$(q, u, R, m, \text{level}, \mathcal{C})$
    $C := \{(u, r, v) \in E \mid r \in R^m\}$
    $\mathcal{E} := \mathcal{E} \cup C$
  **end for**
  **if** $\mathcal{E} = \emptyset$ **then break**
  **end if**
  **for each** edge $e \in \mathcal{E}$ **do**
    **try** $e$.score := Score$(e)$ **except** $e$.score $:= -1$
  **end for**
  **if** $\forall e \in \mathcal{E},\ e$.score $= -1$ **then**
    $\mathcal{E} :=$ UniformSample$(\mathcal{E}, \min(k, |\mathcal{E}|))$
  **else**
    $\mathcal{E} :=$ FilterTopKByScore$(\mathcal{E}, k)$
  **end if**
  $V^d :=$ LLM_SelectNodes$(q, u, \mathcal{E}, n, \text{level}, \mathcal{C})$
  **for each** $(u, r, v) \in \mathcal{E}$ **where** $v \in V^d$ **do**
    $\Pi[v] := \Pi[u] \parallel (u, r, v)$
    $\mathbb{F}' := \mathbb{F}' \cup \{v\}$
  **end for**
  $\mathcal{V} := \mathcal{V} \cup V^d$
  $\mathcal{C}[\text{level}] :=$ LLM_DescribePaths$(q, r_0, \Pi)$
  decision := LLM_ShouldContinue$(q, r_0, \Pi)$
  **if** decision $=$ no **then break**
  **end if**
  $\mathbb{F} := \mathbb{F}'$
  $d^* :=$ level
**end for**
**for** $l = 1$ **to** $d^*$ **do**
  $\Sigma[l] :=$ LLM_Summarize($\Pi$ of depth $l$)
**end for**
**return** $(\Pi, \Sigma)$

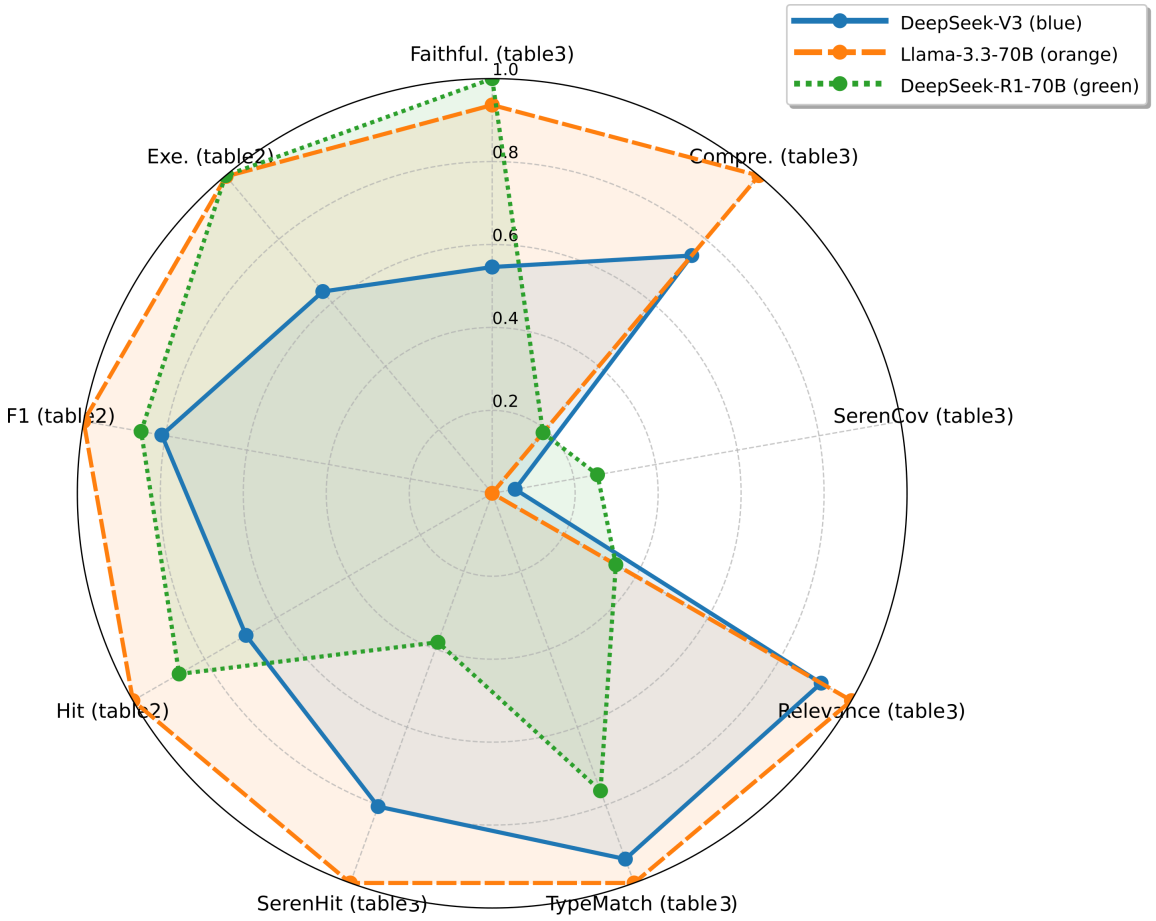



Figure 6: Multi-step Performance Radar Chart

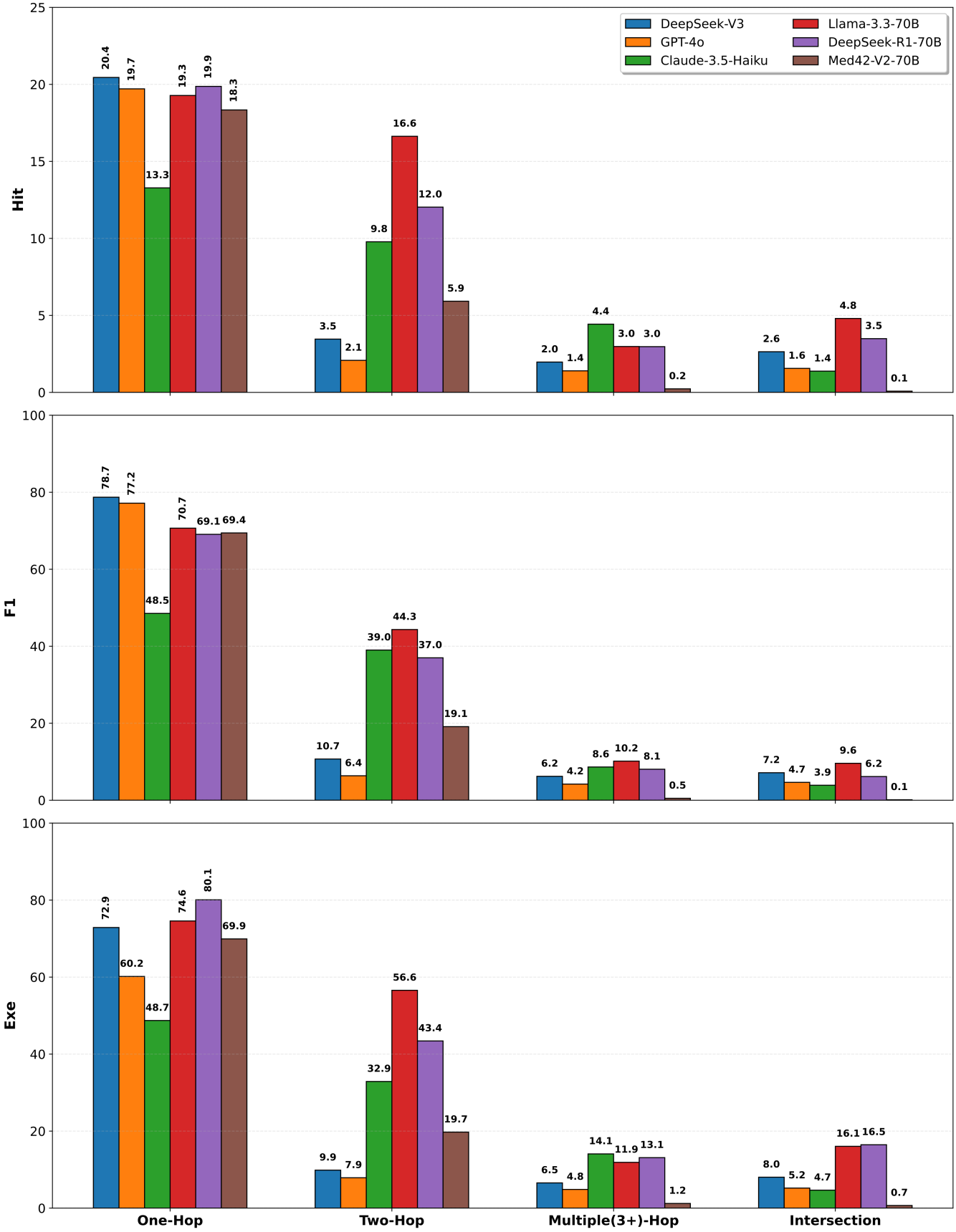



Figure 7: Query Pattern vs. Retrieval Performance