Know what you don't know

Yizhou Li[1*] **Jinyi Xu**[1†] **Qingyi Yang**[1] **Kefan Sun**[2‡]
[1]Academy of Artificial Intelligence and Advanced Technology,
Xi'an Jiaotong-Liverpool University, Suzhou, China
[2]Waversebaan 190, 3050 Oud-Heverlee, Vlaams-Brabant, Belgium
{Yizhou.Li24, Qingyi.Yang24}@student.xjtlu.edu.cn
Jinyi.Xu@xjtlu.edu.cn, sunkefan.pauline@gmail.com

Abstract

Retrieval-augmented generation (RAG) reduces hallucinations by grounding large language models in external knowledge bases. However, most existing methods focus on improving how knowledge is indexed, retrieved and integrated, neglecting a critical source of hallucination: the lack of unknown awareness. Unaware of the knowledge boundaries of the corpus, a model may stitch together fragmented context to hallucinate unsupported relations. To address this, we introduce Substrata, an unknown-aware RAG framework that makes the unknown locatable. Substrata maintains an evolving knowledge graph via Dempster–Shafer evidence fusion, enabling local graph updates instead of costly full graph reconstruction in graph-based RAG. Crucially, we apply a topological analysis method, persistent homology, to detect long-lived structural holes where the corpus fails to close relational gaps within this graph. These holes are converted into retrievable unknown annotations. During inference, Substrata performs hybrid retrieval over both factual chunks and unknown annotations, allowing the generator to access both supporting context and explicit signals of missing contextual support. Under the shared evaluation setup, Substrata achieves the best performance among the six evaluated methods on three public multi-hop QA benchmarks, with an average of 58.5% EM and 68.6% F1 across HotpotQA, MuSiQue, and 2WikiMultiHopQA. On a hallucination-focused domain benchmark, it reaches 95.0% accuracy on Spurious Relational Premise questions while reducing overall token consumption by 10.3× compared with GraphRAG.

1 Introduction

One important motivation behind retrieval-augmented generation (RAG) is to reduce hallucinations in large language models (LLMs) by introducing external and relatively deterministic knowledge sources [17]. Existing RAG studies mainly focus on improving how available knowledge is indexed, retrieved, filtered, and integrated into generation. Representative directions include better indexing strategies [27, 17], query rewriting [21], reranking [24], context compression [14], retrieval-aware generation [1], and graph-based retrieval [5].

Despite their methodological diversity, most existing RAG methods share the same locus of intervention: they treat the knowledge source itself as a given object and intervene mainly in downstream stages. In this sense, conventional RAG primarily answers the question of "what do we know, and how can we retrieve it?" However, hallucination in LLMs is often not merely a failure of retrieval, but a failure of boundary awareness: the model may not know which parts of the currently available knowledge space remain unknown, unsupported, or structurally incomplete. For example, a corpus may contain knowledge chunks about entity X and knowledge chunks about entity Y, yet provide

*Code is available at https://github.com/RicardoLee510520/Know-What-You-Don-t-Know.
†Corresponding author.
‡Kefan Sun was employed by Siemens Industry Software NV.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

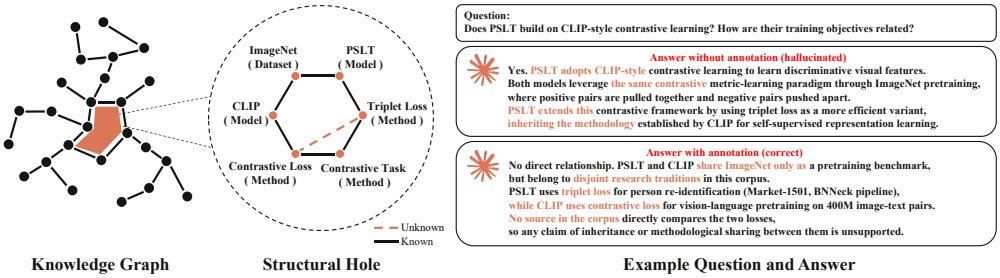

Figure 1: How unknowns affect model generation. The left side shows a knowledge graph, where black nodes and solid lines represent entities and known relations. The orange region highlights a local area containing a structural hole: although several known connections exist, some relations still lack direct contextual support and are therefore marked as "unknown" with orange dashed lines. The right side illustrates this issue through a concrete question-answering example. When the system answers solely based on conceptual proximity, it may easily mistake such unsupported potential relations for real ones, thereby producing hallucinations. In contrast, once the structural hole is explicitly annotated by our framework, the system can recognize that "direct support is missing here" and generate an answer that is more cautious and more faithful to the corpus.

no chunks that a relation exists between X and Y. Existing RAG systems may still retrieve both fragments simultaneously and encourage the generator to cross this relational gap, thereby producing path hallucinations. This phenomenon can also be observed in Figure 1.

We therefore advance a stronger paradigm: the model needs to know not only "what it knows", but also "what it does not know".

To operationalize this paradigm, we design a method for making unknowns locatable. Rather than focusing on downstream optimization of retrieval or generation, our approach intervenes at the level of the upstream knowledge source. We regard the knowledge source not merely as an informative foundation for supporting answer generation, but as a structured space from which missing relationships can be inferred. Specifically, we model the knowledge space as a graph and use topological analysis to detect structural holes, namely positions where the corpus fails to close a relational gap. These structural holes transform otherwise invisible unknowns into structured objects that can be retrieved, read, and processed.

This graph-based formulation, however, introduces the second challenge: how to effectively maintain such boundary information as the corpus evolves. Existing graph-based RAG methods often rely on a static graph construction mechanism, which makes graph construction and maintenance expensive [12, 39, 3, 40]. To address this limitation, we redesign the underlying graph construction mechanism. Instead of treating the graph as a static artifact, our framework maintains an evolving graph that grows with the corpus. New knowledge is incrementally incorporated into the existing graph, and its relations to prior knowledge are updated locally rather than through repeated reconstruction from scratch. This design preserves the unknown-aware capability introduced above while substantially reducing the cost of graph maintenance.

Specifically, we introduce Substrata, an unknown-aware RAG framework built on an evolving knowledge graph. Substrata consists of three coupled components. First, it incrementally incorporates newly arriving documents into the existing graph, avoiding full reconstruction. The new knowledge is clustered into communities and linked to the existing graph at multiple community levels by using Dempster–Shafer Theory (DST) based on graph connectivity and semantic similarity. Second, Substrata detects knowledge holes in the evolving graph through persistent homology. Long-lived topological holes are treated as relational gaps and converted into explicit anti-hallucination annotations that are stored together with ordinary document chunks. Third, at query time, Substrata performs hybrid retrieval over both factual chunks and knowledge-hole annotations, combining vector, BM25, and graph-based rankings. As a result, when a query touches an unsupported region of the knowledge space, Substrata receives not only relevant context but also explicit annotations of missing contextual support, enabling the generator to recognize information gaps rather than hallucinate across them. Under the shared evaluation setup, Substrata consistently achieves the best performance across all three multi-hop QA benchmarks, outperforming both standard retrieval baselines and graph-

based methods. It reaches $69.5\%/84.3\%$ EM/F1 on HotpotQA, $48.4\%/54.6\%$ on MuSiQue, and $57.5\%/66.8\%$ on 2WikiMultiHopQA. On the hallucination-focused domain benchmark, Substrata reaches 95.0% accuracy on Spurious Relational Premise questions, while reducing overall token consumption by $10.3\times$ compared with full graph reconstruction.

In summary, our contributions are as follows:

Evolving graph construction mechanism. Substrata maintains an evolving knowledge graph that incrementally absorbs new knowledge without full graph reconstruction. By using the DST-based fusion mechanism, it links new knowledge to prior knowledge and enables community-level integration, while preserving uncertainty to retain the ability to form new communities.

Unknown detection via persistent homology. We apply persistent homology (PH) to the evolving knowledge graph to identify long-lived topological holes as regions where contextual support fails to close a relational gap. Each detected structural hole is converted into a retrievable "unknown" annotation chunk and stored alongside the source corpus, allowing the generator to consult explicit signals of missing support rather than hallucinate across relational gaps.

Hybrid retrieval with unknown annotations. We design a unified retrieval pipeline that jointly ranks factual document chunks and unknown annotations over a shared retrieval pool. The retriever combines vector similarity, BM25, and our graph-based rankings through rank fusion.

SelfBound SRP Benchmark. We introduce SelfBound, a benchmark comprising 120 high-quality questions within the domain of AI. Beyond standard question categories, SelfBound includes a Spurious Relational Premise category designed to isolate path hallucinations.

2 Preliminary

2.1 Dempster–Shafer Theory of Evidence

Dempster–Shafer theory (DST) is a mathematical framework for uncertainty reasoning and evidence fusion. Unlike standard probability theory, DST does not require all belief to be assigned to singleton hypotheses. Instead, it allows belief mass to be assigned to subsets of the hypothesis space, thereby representing local or global uncertainty that has not yet been clearly attributed. Given a frame of discernment Θ, the mass function satisfies $\sum_{A\subseteq\Theta} m(A) = 1$, where $m(\Theta)$ denotes the mass of the full set of hypotheses, representing uncertainty that cannot be attributed to any specific hypothesis.

In addition, when multiple independent bodies of evidence exist under the same Θ, DST provides combination rules for aggregating them. These rules allow evidence from different sources to be fused into a single combined mass function m^*, thereby producing a more reliable and comprehensive belief estimate. In this paper, we introduce the Least-committed Non-dogmatic Specialization (LNS) rule [29]. The LNS combination of two mass functions m^1 and m^2 is given by

$$\begin{aligned} m^*(A) &= \sum\nolimits_{B\cap C=A} m^1(B)m^2(C), \quad A \subset \Theta, \\ m^*(\Theta) &= m^1(\Theta)m^2(\Theta) + \sum\nolimits_{B\cap C=\emptyset} m^1(B)m^2(C). \end{aligned} \tag{1}$$

Instead of normalising away inter-singleton conflict [28], LNS redistributes such conflict to Θ, thereby preserving uncertainty. This conservative combination rule is particularly suitable for our setting because agreement between graph-connectivity and semantic-similarity evidence strengthens a specific assignment, whereas disagreement is retained as uncertainty rather than being forced into an unsupported merge decision.

2.2 Persistent Homology

Persistent homology (PH) is an algorithmic framework for identifying stable topological structures in complex data, including connected components, loops, and cavities [4, 2]. The basic idea is to represent the data as a simplicial complex K, consisting of vertices, edges, triangles, and higher-dimensional simplices. A filtration function $f : K \to \mathbb{R}$ then orders these simplices by a scale parameter, producing a nested sequence of simplicial complexes: $K_0 \subseteq K_1 \subseteq \cdots \subseteq K_t$, where $K_t = \{\sigma \in K \mid f(\sigma) \leq t\}$. At each scale, homology groups characterise the topological features of the complex. The k-th homology group is defined as $H_k(K) = \ker \partial_k / \operatorname{im} \partial_{k+1}$, where ∂_k is the k-th boundary operator, $\ker \partial_k$ denotes k-dimensional closed structures with no boundary, and $\operatorname{im} \partial_{k+1}$ denotes the part filled by $(k+1)$-dimensional structures. Accordingly, H_0 represents connected

components, H_1 represents loops that cannot be filled, and H_2 represents cavities. Persistent homology further records the scale at which each topological feature is born and dies, yielding a persistence diagram:

$$\mathrm{Dgm}_k = \{(b_i, d_i)\}, \tag{2}$$

where b_i denotes the birth scale of the i-th topological feature and d_i denotes its death scale. Its persistence is given by $\mathrm{pers}_i = d_i - b_i$. A larger persistence value indicates that the feature remains stable across a wider range of scales and is therefore more likely to represent a meaningful structure. In this work, persistent homology is used to detect one-dimensional loops (H_1) in the knowledge graph: sets of entities that are indirectly connected through paths but lack direct evidential support for relational closure. These loops therefore serve as signals of potential structural holes in the corpus.

3 Substrata

The overall workflow of Substrata is shown in Figure 2. Substrata starts from a small set of seed documents. It first extracts entities and relations from the seed documents to construct an initial weighted knowledge graph $G = (V, E, W)$, where nodes V represent entities or concepts, edges E represent relations explicitly stated in the corpus and edge weight w_{ij} is given by the number of chunks independently extracting the relation between v_i and v_j. It then performs Leiden hierarchical clustering [30] to produce L levels of knowledge communities. For each community, an LLM generates a concise summary capped at 150 tokens.

Substrata incrementally integrates local graph $\hat{G}$ derived from new documents into the existing knowledge graph through entity alignment and community fusion. Dempster–Shafer fusion over graph-connectivity and semantic-similarity evidence determines whether a new community should be merged with an existing one or retained independently. After each update, persistent homology detects long-lived holes that indicate potential relational gaps and converts them into unknown annotations. The resulting knowledge base contains original text, the evolving knowledge graph, and unknown annotations. Hybrid retrieval jointly exploits these three sources to support factual question answering and complex reasoning while identifying evidential boundaries and limiting excessive inference along graph paths.

3.1 Evolving graph construction mechanism

After each round produces a new weighted graph $\hat{G}$ and its community structure, Substrata incrementally integrates the new graph into the existing graph G. At level l, the communities in the new graph and the existing graph are denoted as $\hat{G} : \{\hat{C}_1^l, \ldots, \hat{C}_{U_l}^l\}$ and $G : \{C_1^l, \ldots, C_{N_l}^l\}$. In evolving graph construction mechanism, community fusion is performed independently within each hierarchy level using the same procedure. In the following, we take level l as an example. For notational simplicity, we write C_i as shorthand for C_i^l.

Entity alignment. The entities $\hat{V}$ in the new graph are first aligned with the entities V in the existing graph. Specifically, the system first identifies entities with exactly identical names through exact string matching. For unmatched entities, it further computes cosine similarity based on their embedding representations and determines whether they correspond to existing entities according to a predefined threshold τ_{sim}. If an entity cannot be aligned through either exact string matching or semantic similarity matching, it is marked as a genuinely new entity; otherwise, it is subsumed by the matched existing entity, and its associated relationships are redirected to that entity accordingly.

Community fusion. After entity alignment, Substrata performs community fusion level by level. For each new community $\hat{C}_j$ at level l, the system must decide whether it should be merged into one of the existing communities $\{C_1, \ldots, C_N\}$ or created as a new standalone community. Substrata formulates this incremental community assignment problem as an evidence fusion problem under DST [28]. It considers graph connectivity and semantic similarity between communities as two independent sources of evidence for "merge or create" decision. We use $m_j^1(\{C_i\})$ and $m_j^2(\{C_i\})$ to denote the belief masses assigned to the hypothesis that the new community $\hat{C}_j$ should be merged into the existing community C_i, under the graph-connectivity evidence and the semantic-similarity evidence, respectively.

Graph connectivity. We first define the graph connectivity κ_{ij} between a new community $\hat{C}_j$ and an existing community C_i as

$$\kappa_{ij} = \min\left(\frac{q_{ij}}{\tau_{\mathrm{conn}}}, 1\right), \tag{3}$$

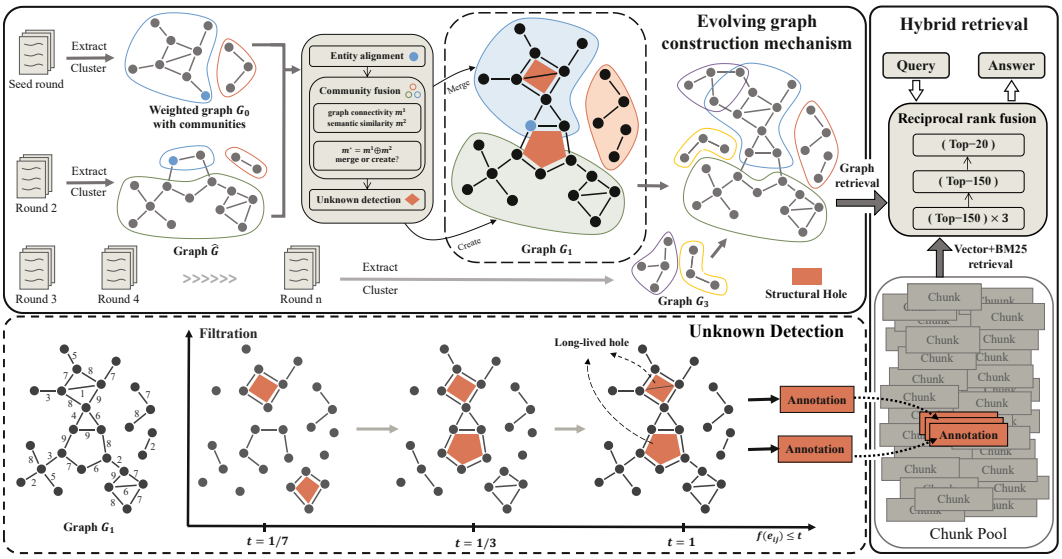

Figure 2: Overview of Substrata. **Upper-left:** The evolving graph construction process starts from a small set of seed documents, converting documents into weighted graphs and hierarchical communities. Through entity alignment and DST-based community fusion, Substrata incrementally builds an evolving knowledge graph from continuously arriving document batches. **Lower-left:** The unknown detection process is applied after each graph update. Substrata constructs an edge-weight filtration, where more strongly supported relations enter the topology earlier, and identify long-lived holes by persistent homology. These structural holes are then converted into unknown annotations and inserted into the chunk pool. **Right:** At query time, Substrata performs hybrid retrieval over the shared chunk pool and fuses vector, BM25, and graph-based rankings through reciprocal rank fusion.

where q_{ij} denotes the total number of such cross-community edges, and τ_{conn} denotes the edge count threshold required to achieve sufficient connectivity.

The mass $m_j^1(\{C_i\})$ increases with the number of edges connecting entities between two communities, representing higher connectivity. The calculation is defined as:

$$m_j^1(\{C_i\}) = \min\left(\kappa_{ij}, \frac{0.95\,\kappa_{ij}}{\sum_{r=1}^{N_l} \kappa_{rj}}\right), \qquad m_j^1(\Theta) = 1 - \sum_{i=1}^{N_l} m_j^1(\{C_i\}), \tag{4}$$

where $m_j^1(\Theta)$ denotes the uncertainty mass assigned by the graph-connectivity evidence for the new community $\hat{C}_j$. A larger value of $m_j^1(\Theta)$ indicates that graph connectivity provides insufficient evidence to confidently assign $\hat{C}_j$ to any specific existing community. In practice, we cap the total mass $\sum_{i=1}^{N_l} m_j^1(\{C_i\})$ at 0.95, so that at least 0.05 mass is reserved for the full-set uncertainty element Θ to prevent overconfident assignments.

Semantic similarity. Semantic similarity is calculated by comparing the cosine similarity between the new community's summary and each existing community's summary, denoted by $\{s_{ij}\}_{i=1}^{N}$. We use the coefficient of variation, $CV = \sigma/\mu$, to measure the relative dispersion of these similarities, where μ and σ denote the mean and standard deviation of cosine similarities , respectively. A larger mean similarity indicates stronger semantic alignment with the existing hierarchy, whereas a larger dispersion indicates greater ambiguity in assigning the new community to a single existing community. Therefore, the mass assignment $m_j^2(\Theta)$ is expressed as follows:

$$m_j^2(\Theta) = \min\big(\max(1 - \mu + \mu \cdot CV, 0), 1\big). \tag{5}$$

When the semantic distance between the new community and existing ones is substantial, or the semantic matching remains unstable, the mass assignment $m_j^2(\Theta)$ increases, leading the system to favor the creation of a new community. The rest mass functions are then defined as

$$m_j^2(\{C_i\}) = \big(1 - m_j^2(\Theta)\big) \cdot \frac{s_{ij}}{\sum_{r=1}^{N_l} s_{rj}}. \tag{6}$$

Merge or create? This stage consists of two steps: first, the system determines whether a new community should be retained; second, for the unretained communities, it decides which existing community they should be merged into.

For a new community $\hat{C}_j$, if both graph connectivity and semantic similarity indicate high uncertainty, that is, if $m_j^1(\Theta)$ and $m_j^2(\Theta)$ both exceed the threshold τ_{merge}, then no existing community provides sufficiently clear evidence to absorb the new community. In this case, the system directly creates $\hat{C}_j$ as a standalone community and skips the subsequent evidence-combination step.

For the rest new communities, the system further determines which existing community they should be merged into. The two mass functions are combined using the LNS rule, defined as

$$\begin{aligned} m_j^*(C_i) &= m_j^1(C_i)\, m_j^2(C_i) + m_j^1(C_i)\, m_j^2(\Theta) + m_j^1(\Theta)\, m_j^2(C_i), \\ m_j^*(\Theta) &= m_j^1(\Theta)\, m_j^2(\Theta) + \sum\nolimits_{r \neq s} m_j^1(C_r)\, m_j^2(C_s), \end{aligned} \tag{7}$$

where $m_j^*(C_i)$ denotes the combined belief mass supporting the hypothesis that the new community $\hat{C}_j$ should be merged into the existing community C_i, while $m_j^*(\Theta)$ denotes the remaining uncertainty after evidence fusion.

Then the system applies the pignistic probability transform [29], which distributes the remaining uncertainty uniformly across all candidate existing communities. This converts belief masses into comparable point probabilities:

$$\mathrm{BetP}_j(C_i) = m_j^*(C_i) + \frac{m_j^*(\Theta)}{N}. \tag{8}$$

The existing community with the highest pignistic probability is then selected as the merge target.

After either merging the new community or creating it as standalone, the system updates the community hierarchy and regenerates the summaries of affected communities. In this way, Substrata incrementally incorporates new communities into the existing multi-level community structure without re-running global clustering from scratch.

3.2 Unknown detection via persistent homology

Structure-hole detection. After each round of graph fusion, Substrata constructs a simplicial complex K over the updated weighted graph G, and uses persistent homology to characterise one-dimensional holes in $H_1(K)$. Non-trivial elements of $H_1(K)$ correspond to closed edge-loops that cannot be filled by any triangle. These are precisely the structural holes we seek to identify: although a set of entities may be indirectly connected through paths in the graph, at least one edge required to support a stable direct relationship is missing.

To detect such structures, we equip G with a filtration. Each entity enters the filtration at $t = 0$, while each edge enters at $f(e_{ij}) = \frac{1}{w_{ij}}$. Thus, relationships supported by larger edge weights, enter the filtration earlier. As edges and higher-dimensional simplices are gradually added, one-dimensional holes may appear and disappear. A hole is born when a set of edges forms a closed loop that has not yet been filled by a triangle, and it dies when the corresponding triangle is added. The persistence diagram $\mathrm{dgm}_1(G)$ records the birth time b_i and death time d_i of the i-th one-dimensional hole and we retain only long-lived holes satisfying

$$d_i - b_i \geq \tau_{\mathrm{persist}}. \tag{9}$$

Such significant holes indicate persistent relational gaps: the involved entities are strongly connected through indirect paths, yet the corpus does not provide sufficient evidence for certain direct relationships among them. They therefore represent not only topological holes in the graph, but also potential unknowns at the boundary of the knowledge source.

Unknown annotation. For each significant hole, the system retrieves the original chunks from which the relevant entities were extracted. The entities and their source chunks are then submitted to an LLM, which verifies against the source text whether any directly stated relationship exists among them and characterises the semantic nature of the structural hole. The LLM produces a structured unknown annotation consisting of four parts: semantic diagnosis, missing-edge analysis, existing-edge analysis, and downstream generation constraints. Representative examples are provided

in Appendix C.5. This report is segmented according to its native sections into chunks and inserted into the same chunk pool as the source-document chunks of the current round, sharing the same index. Downstream retrieval accesses these annotations through their textual content alone, with no special handling required.

3.3 Hybrid retrieval with unknown annotations

At query time, the system retrieves supporting chunks through three independent legs that share a common chunk pool: a vector leg, a BM25 leg, and a graph leg, so their rankings can be fused directly without any alignment step.

The graph leg routes the query through the weighted graph G. It first aligns entities extracted from the query with entities in G. The i-th successfully aligned entity is treated as a query anchor and initiates a one-hop neighbourhood expansion. For the j-th neighbouring entity, the system computes a structural relevance score:

$$g_{ij} = \frac{w_{ij}}{\deg_j}, \tag{10}$$

where $\deg_j$ denotes the total degree of the j-th neighbour, while the anchor itself receives unit mass. For each anchor, the top 25 highest-scoring neighbors are retained. The graph leg then projects entity and edge scores back to chunks through the back-pointers maintained during graph construction. The top chunks are returned, with an empty ranking emitted when no anchor is aligned.

Finally, the candidates returned by the vector, BM25, and graph legs are fused into a single ranked pool. In the reranked configuration, a cross-encoder reranker jointly re-scores each query–chunk pair and selects the final evidence chunks for generation. In the shared-setup evaluation on the public benchmarks, we disable the reranker and pass the top-20 passages from the fused ranking directly to the QA reader. If no query entity can be aligned to the graph, the system automatically falls back to only the vector and BM25 legs.

3.4 SelfBound: benchmarking graph unknowns

SelfBound is a 120-question benchmark designed to evaluate unknown boundary awareness in graph-augmented RAG. It tests whether a system can recognize relational gaps, which is not yet covered by existing RAG evaluations. We refer to this problem type as a Spurious Relational Premise (SRP). SelfBound is built over the academic literature corpus in the domain of AI. It includes four standard question types—Simple with Condition, Comparison, Multi-hop, and False Premise—with 20 questions each, as well as 40 SRP questions. The SRP questions are further divided into four subcategories: contextual grounding verification, entity discrimination, relationship boundary testing, and directional and conditional relationship testing, corresponding respectively to unverifiability hallucination, entity-error hallucination, relation-error hallucination, and faithfulness hallucination caused by logical inconsistency. Each SRP subcategory contains a mixture of confirmable and unconfirmable questions, preventing systems from achieving high scores by uniformly refusing all SRP queries. More detailed descriptions, taxonomy, and the full question set of SelfBound are provided in Appendix C.

4 Experiments

Experimental setup. We evaluate Substrata on three multi-hop question answering benchmarks: HotpotQA [36], MuSiQue [31], and 2WikiMultiHopQA [10], using 1,000 sampled questions per dataset. Table 1 presents the results in two blocks.

The upper block retains the original comparison results with GPT-4o as the QA reader under the original experimental settings [38, 19]. It includes BGE [33, 16], BERT-based reranking [23], GraphRAG [5], and SiReRAG [38].

The lower block reports our evaluation of BM25, a dense retrieval baseline (Dense), RAPTOR [27], LightRAG [7], HippoRAG2 [9], and Substrata under a shared experimental setup. For this comparison, we use qwen3.6-flash as the QA reader and text-embedding-v4 wherever an embedding model is required. All six methods use the same qa-reader-v1 prompt, top-20 retrieved passages, no reranker, and random seed 42. We report answer-level Exact Match (EM) and token-level F1 on each dataset, together with overall EM/F1 and retrieval Recall@20 for the shared-setup comparison. Detailed experimental settings are provided in Appendix B.1.

Table 1: Results (%) on three multi-hop QA benchmarks.

Method	HotpotQA		MuSiQue		2WikiMultiHopQA		Overall		Recall@20
	EM	F1	EM	F1	EM	F1	EM	F1	
Retained results (original settings)									
BGE	47.60	60.36	20.80	30.10	40.10	44.96	–	–	–
Reranking	55.40	70.03	32.00	40.29	53.70	58.44	–	–	–
GraphRAG	31.70	42.74	12.10	20.22	22.50	27.49	–	–	–
SiReRAG	61.70	76.48	40.50	53.08	59.60	67.94	–	–	–
Shared evaluation setup									
BM25	59.0	73.8	24.5	32.6	46.5	54.1	43.3	53.5	75.9
Dense	66.0	81.1	36.0	43.7	48.5	57.1	50.2	60.6	86.8
RAPTOR	66.0	80.4	33.5	42.3	45.0	53.8	48.2	58.8	88.9
LightRAG	62.0	75.8	28.5	37.9	39.5	47.4	43.3	53.7	82.3
HippoRAG2	<u>68.5</u>	<u>83.8</u>	<u>41.0</u>	<u>49.3</u>	<u>55.0</u>	<u>65.0</u>	<u>54.8</u>	<u>66.0</u>	<u>94.0</u>
Substrata	**69.5**	**84.3**	**48.4**	**54.6**	**57.5**	**66.8**	**58.5**	**68.6**	**96.0**

4.1 Performance comparison

Public benchmarks. Table 1 reports results on the three multi-hop QA benchmarks. Within the lower block, the best and second-best results are shown in bold and underlined, respectively. Among the six methods evaluated under the shared setup, Substrata achieves the highest EM and F1 on all three datasets. Compared with HippoRAG2, the second-best method in this block, Substrata improves EM/F1 by 1.0%/0.5% on HotpotQA, 7.4%/5.3% on MuSiQue, and 2.5%/1.8% on 2WikiMultiHopQA. Substrata's consistently strong performance suggests that it improves both simple factual retrieval and complex reasoning: unlike graph-only methods, it preserves direct chunk-level retrieval for precise factual evidence, while its evolving graph construction and unknown detection provide more reliable structural guidance for multi-hop reasoning and help avoid unsupported relational jumps.

Table 2: Accuracy (%) on SelfBound.

Method	SwC	M-hop	Comp	FP	SRP
Substrata	**100.0**	**85.0**	**85.0**	**100.0**	**95.0**
GraphRAG	70.0	60.0	75.0	45.0	65.0
GraphRAG^{+}[4]	95.0	75.0	70.0	85.0	70.0

SelfBound benchmark. We evaluate Substrata's ability to reduce path hallucinations in complex reasoning on the SelfBound benchmark, using GraphRAG as the baseline. For a stronger and fairer comparison, GraphRAG^{+} augments GraphRAG with hybrid vector and BM25 retrieval. As shown in Table 2, Substrata performs strongly across all question types, achieving 100.0% accuracy on False Premise and 95.0% on Spurious Relational Premise, compared with 85.0% and 70.0% for GraphRAG^{+}. These gains arise from compact, low-redundancy graph construction through effective fusion and structural-hole detection that limits unsupported path extension and relation fabrication.

4.2 Ablation study

We further conduct ablation studies on both the graph construction stage and the retrieval stage of Substrata, with the results reported in Table 3. The results show that both components contribute to Substrata's overall performance. Without the unknown-aware graph and its corresponding graph retrieval leg, the system still performs well on simple factual questions, but its accuracy drops noticeably on reasoning-intensive categories such as Multi-hop, Comparison, and SRP. Combining the unknown-aware graph with Vector+BM25 yields the best results across all question types, indicating that Substrata benefits from both precise chunk-level retrieval and hallucination-aware structural guidance.

[4]GraphRAG^{+} denotes GraphRAG with Vector+BM25 retrieval.

Table 3: Component composition and accuracy (%) on SelfBound

Method	Components		Accuracy (%)				
	Unknown-aware graph	Retrieval legs	SwC	M-hop	Comp	FP	SRP
Substrata	×	Vector+BM25	100.0	75.0	75.0	95.0	75.0
	✓	Graph	95.0	75.0	80.0	95.0	85.0
	✓	Graph+Vector+BM25	**100.0**	**85.0**	**85.0**	**100.0**	**95.0**

4.3 Evolving-graph study

Graph growth. Figure 3 shows the graph construction process of Substrata on SelfBound. Each disc represents an L0 community; disc area is proportional to the number of entities it contains, and grey edges indicate inter-community relationships. Orange discs denote new or updated communities in the current round, while blue discs denote communities that are not updated. As shown in Figure 3, Substrata produces a clear hub-and-leaf structure: a dense core surrounded by several smaller peripheral communities. This is a desirable graph structure, indicating that the core knowledge in the corpus is effectively aggregated into a tightly connected central region, while the system can still identify and preserve peripheral communities that are less central to the main topic but genuinely present in the corpus. More details on the evolving graph are provided in Appendix B.3.

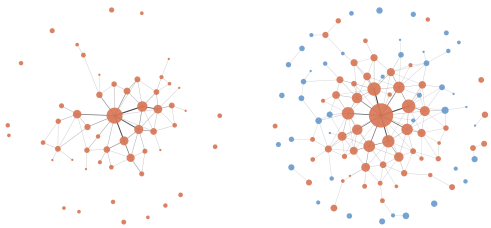

Figure 3: Visualization of the evolving graph.

Token cost. Figure 4 compares the token consumption of Substrata with GraphRAG-style full reconstruction and two incremental graph-based RAG methods, HippoRAG2 and LightRAG. At Round 10, the cost gap between Substrata and full reconstruction reaches 10.3×, with the latter consuming 420.5M tokens compared with 40.7M tokens for Substrata. Although Substrata incurs a slightly higher cost than HippoRAG2, it additionally supports the explicit representation of unknowns through persistent-homology-based structural-hole detection. Detailed cost statistics are reported in the Appendix B.4.

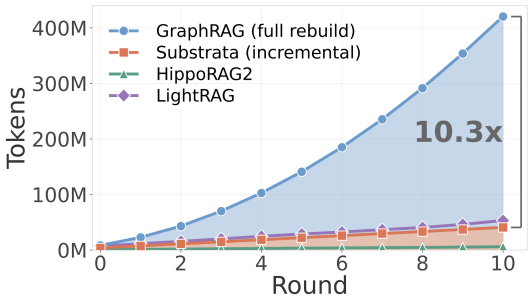

Figure 4: Token cost comparison.

4.4 Robustness to graph construction noise

Graph construction may introduce extraction and alignment errors. To evaluate Substrata's robustness, we inject three types of noise: edge addition, which adds relations between random entity pairs to simulate extraction hallucinations; edge removal, which deletes relations between random entity pairs and all their supporting evidence chunks to simulate missed relations and graph sparsity; and entity replacement, which replaces one endpoint of randomly selected edges with a random entity while retaining the original weights, descriptions, and source-chunk references to simulate entity alignment errors.

We evaluate different noise ratios on SelfBound and public multi-hop QA benchmarks, using $p = 0$ as the reference and averaging three independent random runs per setting. Tables 4 and 5 report results for $p \in \{0\%, 10\%, 20\%\}$. Full results, additionally including 2%, 5%, and 15%, are provided in Appendix B.2.

5 Related Work

Downstream interventions in RAG. Existing RAG studies mainly improve the downstream pipeline while leaving the knowledge source itself unchanged. The first line of work improves the index structure. During indexing, graph-based RAG [5, 19, 38, 20, 7] extracts entities and relations to construct a knowledge graph, thereby organizing the corpus into a graph-structured index for global retrieval and question answering. RAPTOR constructs a hierarchical tree index through corpus-level

Table 4: Robustness of Substrata to graph construction noise on SelfBound. Results are averaged over three runs.

Noise type	p	Detected holes	SwC (%)	M-hop (%)	Comp (%)	FP (%)	SRP (%)
Edge addition	0%	3,839	100.0	85.0	85.0	100.0	95.0
	10%	4,956	100.0	85.0	85.0	98.3	93.3
	20%	6,140	100.0	83.3	85.0	98.3	93.3
Edge removal	0%	3,839	100.0	85.0	85.0	100.0	95.0
	10%	3,793	100.0	83.3	83.3	98.3	91.7
	20%	3,728	100.0	81.7	81.7	96.7	89.2
Entity replacement	0%	3,839	100.0	85.0	85.0	100.0	95.0
	10%	5,482	100.0	83.3	83.3	98.3	90.8
	20%	7,203	100.0	81.7	81.7	96.7	87.5

Table 5: Robustness of Substrata to graph construction noise on public multi-hop QA benchmarks. Results are averaged over three runs.

Dataset	Metric	Edge addition			Edge removal			Entity replacement		
		0%	10%	20%	0%	10%	20%	0%	10%	20%
HotpotQA	EM	69.5	69.5	69.5	69.5	69.4	68.9	69.5	69.2	68.9
	F1	84.3	84.3	84.1	84.3	84.0	83.9	84.3	84.1	84.3
MuSiQue	EM	48.4	47.8	47.4	48.4	46.9	45.9	48.4	46.6	45.6
	F1	54.6	54.1	53.7	54.6	53.2	52.5	54.6	53.1	51.9
2WikiMulti HopQA	EM	57.5	57.3	57.4	57.5	56.7	56.3	57.5	56.7	56.2
	F1	66.8	66.3	66.8	66.8	65.8	65.3	66.8	66.1	65.6
Overall	**EM**	**58.5**	**58.2**	**58.1**	**58.5**	**57.7**	**57.0**	**58.5**	**57.5**	**56.9**
	F1	**68.6**	**68.2**	**68.2**	**68.6**	**67.7**	**67.2**	**68.6**	**67.8**	**67.3**

clustering, enabling retrieval at multiple levels of abstraction [27]. The second line improves retrieval and ranking over a fixed index [24]. Examples include KAG, which combines knowledge graphs with vector retrieval for domain-specific reasoning [18]; HippoRAG, which combines passage–entity indexing with graph-based retrieval and PageRank aggregation [8]; and adaptive retrieval methods that route questions to different retrieval strategies according to query complexity [13]. The third line focuses on query improvement. Rewrite-Retrieve-Read [21] trains a lightweight query rewriter to reformulate the input query before retrieval. HyDE [6] generates a hypothetical document from the query and uses it as the retrieval vector. Finally, generation-side methods improve how models use retrieved information. FLARE [15] actively re-retrieves when the generated content is uncertain; LLMLingua [14] compresses retrieved contexts to shorten the prompt; Self-RAG [1] trains the generator to retrieve and evaluate its own outputs through reflection tokens; and RankRAG [37] instruction-tunes a single model for both context ranking and answer generation.

6 Conclusion

We introduced Substrata, an unknown-aware RAG framework that makes unsupported relational gaps explicit and retrievable. It builds an evolving knowledge graph via DST fusion, detects structural holes with persistent homology, and stores them as unknown annotations alongside document chunks. By exposing both evidence and missing-support signals, Substrata reduces path hallucinations, improves accuracy on public multi-hop QA benchmarks and SelfBound, and lowers graph construction costs by avoiding full reconstruction. These results show that robust RAG requires not only retrieving known knowledge but also recognizing knowledge boundaries. Declaring the unknown is a principle that matters beyond RAG. A current limitation is that natural language enters the graph only through extracted triples. In addition, unknown-relation detection is currently restricted to loop structures to prioritize precision, leaving relational gaps outside such structures insufficiently explored. An important direction for future work is a more faithful structural modeling of natural language, on which richer representations of the unknown can be identified and the idea of marking the unknown can be applied more broadly.

Acknowledgments and Disclosure of Funding

We thank Qi Shao for assistance in running the experiments.This work received no third-party funding. All expenses of this work were borne by Yizhou Li and Jinyi Xu.Kefan Sun is employed by Siemens Industry Software NV; the other authors declare no competing interests.

References

[1] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

[2] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2): 255–308, 2009.

[3] Shengyuan Chen, Chuang Zhou, Zheng Yuan, Qinggang Zhang, Zeyang Cui, Hao Chen, Yilin Xiao, Jiannong Cao, and Xiao Huang. You don't need pre-built graphs for RAG: Retrieval augmented generation with adaptive reasoning structures. In *AAAI Conference on Artificial Intelligence*, 2026.

[4] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, 2002.

[5] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[6] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, 2023.

[7] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10746–10761, 2025.

[8] Bernal J. Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In *Conference on Neural Information Processing Systems*, 2024.

[9] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *International Conference on Machine Learning*, 2025.

[10] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *International Conference on Computational Linguistics*, 2020.

[11] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

[12] Yiqian Huang, Shiqi Zhang, and Xiaokui Xiao. KET-RAG: A cost-efficient multi-granular indexing framework for graph-RAG. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2025.

[13] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2024.

[14] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

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

[16] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2020.

[17] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Conference on Neural Information Processing Systems*, 2020.

[18] Lei Liang, Zhongpu Bo, Zhengke Gui, Zhongshu Zhu, Ling Zhong, Peilong Zhao, Mengshu Sun, Zhiqiang Zhang, Jun Zhou, Wenguang Chen, et al. Kag: Boosting llms in professional domains via knowledge augmented generation. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 334–343, 2025.

[19] Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. HopRAG: Multi-hop reasoning for logic-aware retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1897–1913, 2025.

[20] Yifan Lu, Yigeng Zhou, Jing Li, Yequan Wang, Xuebo Liu, Daojing He, Fangming Liu, and Min Zhang. Knowledge editing with dynamic knowledge graphs for multi-hop question answering. In *AAAI Conference on Artificial Intelligence*, 2025.

[21] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

[22] Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Annual Meeting of the Association for Computational Linguistics*, 2024.

[23] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT, 2020.

[24] Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, 2020.

[25] Xiangyu Peng, Prafulla Kumar Choubey, Caiming Xiong, and Chien-Sheng Wu. Unanswerability evaluation for retrieval augmented generation. In *Annual Meeting of the Association for Computational Linguistics*, 2025.

[26] Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, et al. Ragchecker: A fine-grained framework for diagnosing retrieval-augmented generation. In *Conference on Neural Information Processing Systems*, 2024.

[27] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, 2024.

[28] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[29] Philippe Smets and Robert Kennes. The transferable belief model. *Artificial Intelligence*, 66(2): 191–234, 1994.

[30] Vincent A. Traag, Ludo Waltman, and Nees Jan van Eck. From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233, 2019.

[31] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

[32] Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. When to use graphs in RAG: A comprehensive analysis for graph retrieval-augmented generation. In *International Conference on Learning Representations*, 2026.

[33] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024.

[34] Bingcong Xue and Lei Zou. Knowledge graph quality management: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4969–4988, 2023.

[35] Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, et al. Crag-comprehensive rag benchmark. In *Conference on Neural Information Processing Systems*, 2024.

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

[37] Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. In *Conference on Neural Information Processing Systems*, 2024.

[38] Nan Zhang, Prafulla Kumar Choubey, Alexander Fabbri, Gabriel Bernadett-Shapiro, Rui Zhang, Prasenjit Mitra, Caiming Xiong, and Chien-Sheng Wu. SiReRAG: Indexing similar and related information for multihop reasoning. In *International Conference on Learning Representations*, 2025.

[39] Yingli Zhou, Yaodong Su, Youran Sun, Shu Wang, Taotao Wang, Runyuan He, Yongwei Zhang, Sicong Liang, Xilin Liu, Yuchi Ma, and Yixiang Fang. In-depth analysis of graph-based RAG in a unified framework. *Proceedings of the VLDB Endowment*, 18(13):5623–5637, 2025.

[40] Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. LinearRAG: Linear graph retrieval augmented generation on large-scale corpora. In *International Conference on Learning Representations*, 2026.

A Methodology Interpretation

This section will provide additional methodological interpretations for Substrata in Section 3 of the main text.

A.1 Uncertainty mass assignment from semantic similarity.

The coefficient of variation is defined as $CV = \sigma/\mu$, where μ and σ denote the mean and standard deviation of the cosine similarities between the new community $\hat{C}_j$ and all existing communities $\{C_i\}_{i=1}^{N_l}$, respectively. Unlike the standard deviation, which measures absolute dispersion, the coefficient of variation measures the relative dispersion of semantic similarities with respect to their mean level. Therefore, CV captures how stable or unstable the semantic matching pattern is across candidate existing communities.

The uncertainty mass assigned to the full-set uncertainty element Θ is defined as

$$m_j^2(\Theta) = \min\big(\max(1 - \mu + \mu \cdot CV, 0), 1\big). \tag{11}$$

This expression consists of two components. The term $1 - \mu$ captures the uncertainty caused by low overall semantic similarity. When the average similarity μ is small, the new community is semantically distant from the existing hierarchy, and thus a larger mass should be assigned to Θ. The term $\mu \cdot CV$ captures additional uncertainty caused by instability in the similarity distribution. When the similarity scores are highly dispersed, the semantic correspondence between the new community and existing communities becomes less stable, so the uncertainty mass increases. Multiplying CV by μ ensures that this dispersion-based penalty is mainly activated when there is a meaningful level of semantic relatedness. Finally, the outer clipping operation $\min(\max(\cdot, 0), 1)$ guarantees that $m_j^2(\Theta)$ remains within the valid range of a DST mass function.

A.2 Threshold Settings

Several thresholds are introduced in Substrata. These thresholds are manually specified before running the algorithm and remain fixed within each dataset. In the following experiments, we report the values used for both the SelfBound dataset and the public datasets, together with the meaning of each threshold in Table 6.

Table 6: Thresholds used in Substrata.

Location	**Symbol**	**Value**	**Meaning**
Communities	L	60	Number of hierarchical community levels
Entity alignment	τ_{sim}	0.7	Semantic similarity threshold for entity alignment
Graph connectivity	τ_{conn}	12	Graph-connectivity threshold
Merge or create	τ_{merge}	0.5	Merge decision threshold
Unknown detection	τ_{persist}	0.3	Significant-hole persistence threshold

We set the number of hierarchical community levels to $L = 60$ for Leiden hierarchical clustering. Here, L denotes the number of hierarchy levels, whereas l indexes an individual level.

Threshold sensitivity. We conduct an additional sensitivity analysis on three key components of Substrata: entity alignment, DST-based fusion, and persistent hole detection. Table 7 summarizes the tested parameter ranges, their main effects, and the sensitivity results. Overall, Substrata remains stable under reasonable threshold variations, indicating that its performance does not depend on finely tuned hyperparameters.

Table 7: Representative threshold sensitivity in Substrata. The table summarizes the tested ranges, main effects, and sensitivity results for entity alignment, DST-based fusion, and persistent hole detection.

Component	Parameter	Tested Range	Main Effect	Sensitivity Result
Entity Alignment	τ_{sim}	0.60–0.80	Controls entity merging strictness	Higher threshold produces more conservative merging and more entities; graph construction remains stable in this range
DST-based Fusion	τ_{conn}	4–48	Controls community connection decision	New-community creation changes smoothly without abrupt transitions
DST-based Fusion	τ_{merge}	0.25–0.75	Controls evidence allocation during merging	Stable within practical range; instability only occurs at extreme values
Persistent Hole Detection	$\tau_{persist}$	0.25–0.80	Filters insignificant topological holes	Retained finite holes change from 270 to 258 (-4.4%), indicating stable hole detection

B Experimental Supplements

The main text provides a compact summary of the experimental setting and key results. This appendix gives a more detailed account of the experimental setup and graph construction process. Specifically, we first describe the corpus, incremental rounds, model configurations, and evaluation protocol in greater detail. We then present the complete results from five supplementary experiments: robustness to noise and precision validation of detected holes, the growth information of the evolving graph, computational consumption, and retrieval recall of unknown annotations.

B.1 Detailed experiment setting

Public benchmark. We evaluate Substrata on three public multi-hop question answering benchmarks: HotpotQA [36], MuSiQue [31], and 2WikiMultiHopQA [10]. Following the evaluation protocol of SiReRAG [38] and HopRAG [19], we sample 1,000 questions from each development set. For each dataset, we construct a shared retrieval corpus by pooling the supporting and distractor passages associated with the sampled questions, following HippoRAG [8]. After deduplication, each corpus contains approximately 9K passages. For each query in Substrata, the three retrieval legs each return their top-150 candidates; these are merged and deduplicated by Reciprocal Rank Fusion (RRF), producing a fused ranking of length between 150 and 450. For the shared-setup evaluation in the lower block of Table 1, the top-20 passages from the fused ranking are passed directly to the QA reader without reranking.

Table 1 separates the retained results under the original settings from our shared-setup evaluation. The upper block includes BGE [33, 16], BERT-based reranking [23], GraphRAG [5], and SiReRAG [38], retaining their original comparison results with GPT-4o as the QA reader. The lower block evaluates BM25, Dense, RAPTOR [27], LightRAG [7], HippoRAG2 [9], and Substrata under a shared setup. For the lower block, we use qwen3.6-flash as the QA reader and text-embedding-v4 wherever embeddings are required. All six methods use the same qa-reader-v1 prompt, top-20 retrieved passages, no reranker, and random seed 42. Evaluation follows the official protocols and reports answer-level EM and token-level F1 for each dataset, together with overall EM/F1 and retrieval Recall@20.

Domain-specific benchmark. We construct SelfBound, a 120-question domain-specific benchmark over the same academic corpus used to build Substrata's knowledge graph. All gold-standard answers are verified against the source papers. Since knowledge graph quality directly affects downstream answer reliability [34], we also conduct sampled verification of the extracted triples, which yields approximately 96.5% triple-level accuracy.

SelfBound contains five question types. Four are adapted from the Comprehensive RAG Benchmark [35]: Simple with Condition, Comparison, Multi-hop, and False Premise, with 20 questions each. We further introduce Spurious Relational Premise (SRP), a new 40-question category designed to capture a failure mode not covered by existing RAG benchmarks [25]: all relevant entities are present and retrievable, but their graph connections are too weak to support the relational inference implied by the question.

The SRP questions are divided into four subcategories, each corresponding to a hallucination type in the established taxonomy [11]: Evidential Grounding Verification, Entity Discrimination, Relationship Boundary Testing, and Directional and Conditional Relationship Testing, with 10 questions each. Each subcategory mixes $40\% - -60\%$ confirmable and $40\% - -60\%$ unconfirmable questions, preventing systems from obtaining high scores by uniformly refusing all SRP queries.

Evaluation protocol. Following GraphRAG [5] and GraphRAG-Bench [32], we evaluate responses using binary correctness judgments [26, 35]. Each response is independently assessed by human experts and an LLM judge, with disagreements resolved through discussion. To reduce randomness, each question is submitted three times to each system, and the majority label is used as the final score.

B.2 Robustness to Noise and Precision Validation of Detected Holes

This section evaluates whether Substrata remains reliable when the constructed graph contains extraction errors and whether the structural holes detected on a noise-free graph correspond to genuine knowledge gaps. We examine these two questions through controlled noise injection and independent annotation validation.

Robustness to graph construction noise To evaluate the robustness of Substrata against graph construction noise, we conduct experiments with three types of noise injection. Edge addition

simulates hallucinated entity/relation extraction: existing relations remain unchanged, and a new relation is added between a randomly selected entity and another randomly sampled entity. Edge removal simulates graph sparsity and relation extraction errors: we randomly select entity pairs and remove all supporting evidence associated with the corresponding relation, including all source evidence chunks. Entity replacement simulates entity alignment mistakes: we randomly select a proportion p of edges and replace one endpoint with a randomly sampled entity, while keeping the original relation weight, description, and source-chunk references unchanged.

We evaluate each noise type at injection ratios of 2%, 5%, 10%, 15%, and 20%, with $p = 0$ as the reference. For each noise ratio, the experiment is randomly repeated three times, and the reported results are averaged over the three runs. Table 8 reports the SelfBound results, including detected holes and accuracy for the five question categories. Table 9 reports EM and F1 on the public benchmarks.

Table 8: Robustness of Substrata to three types of graph construction noise on SelfBound.

	(A) Edge addition						(B) Edge removal						(C) Entity replacement					
p	Detected holes	SwC (%)	M-hop (%)	Comp (%)	FP (%)	SRP (%)	Detected holes	SwC (%)	M-hop (%)	Comp (%)	FP (%)	SRP (%)	Detected holes	SwC (%)	M-hop (%)	Comp (%)	FP (%)	SRP (%)
0%	3,839	100.0	85.0	85.0	100.0	95.0	3,839	100.0	85.0	85.0	100.0	95.0	3,839	100.0	85.0	85.0	100.0	95.0
2%	4,056	100.0	85.0	85.0	98.3	94.2	3,828	100.0	83.3	85.0	100.0	94.2	4,192	100.0	85.0	85.0	100.0	94.2
5%	4,382	100.0	85.0	85.0	98.3	94.2	3,829	100.0	83.3	83.3	98.3	92.5	4,653	100.0	85.0	85.0	100.0	91.7
10%	4,956	100.0	85.0	85.0	98.3	93.3	3,793	100.0	83.3	83.3	98.3	91.7	5,482	100.0	83.3	83.3	98.3	90.8
15%	5,544	100.0	83.3	85.0	98.3	93.3	3,771	100.0	83.3	83.3	98.3	90.0	6,291	100.0	81.7	81.7	96.7	89.2
20%	6,140	100.0	83.3	85.0	98.3	93.3	3,728	100.0	81.7	81.7	96.7	89.2	7,203	100.0	81.7	81.7	96.7	87.5

Table 9: Robustness of Substrata to three types of graph construction noise on public multi-hop QA benchmarks.

Dataset	Metric	(A) Edge addition						(B) Edge removal						(C) Entity replacement					
		0%	2%	5%	10%	15%	20%	0%	2%	5%	10%	15%	20%	0%	2%	5%	10%	15%	20%
HotpotQA	EM	69.5	69.5	69.3	69.5	69.5	69.5	69.5	69.5	69.2	69.4	69.5	68.9	69.5	69.4	69.3	69.2	68.9	68.9
	F1	84.3	84.3	84.3	84.3	84.3	84.1	84.3	84.3	84.1	84.0	84.1	83.9	84.3	84.0	84.1	84.1	83.9	84.3
MuSiQue	EM	48.4	48.3	48.1	47.8	47.3	47.4	48.4	48.0	47.7	46.9	46.6	45.9	48.4	47.9	47.4	46.6	46.0	45.6
	F1	54.6	54.5	54.2	54.1	53.9	53.7	54.6	54.1	53.8	53.2	53.2	52.5	54.6	54.2	53.7	53.1	52.5	51.9
2WikiMulti	EM	57.5	57.5	57.4	57.3	57.3	57.4	57.5	57.5	57.1	56.7	56.7	56.3	57.5	57.0	56.9	56.7	56.3	56.2
HopQA	F1	66.8	66.8	66.5	66.3	66.6	66.8	66.8	66.7	66.4	65.8	65.8	65.3	66.8	66.5	66.6	66.1	65.4	65.6
Overall	**EM**	**58.5**	**58.4**	**58.3**	**58.2**	**58.0**	**58.1**	**58.5**	**58.3**	**58.0**	**57.7**	**57.6**	**57.0**	**58.5**	**58.1**	**57.9**	**57.5**	**57.1**	**56.9**
	F1	**68.6**	**68.5**	**68.3**	**68.2**	**68.3**	**68.2**	**68.6**	**68.4**	**68.1**	**67.7**	**67.7**	**67.2**	**68.6**	**68.2**	**68.1**	**67.8**	**67.3**	**67.3**

Substrata's robustness is supported by three complementary designs. Relation weights are derived from independently extracted supporting chunks. Detected holes are verified against original source chunks when generating unknown annotations. These annotations serve as auxiliary retrieval signals and are jointly retrieved with original evidence chunks for further generator assessment.

Independent annotation validation To evaluate unknown-annotation quality, we randomly sample 40 detected holes (approximately 1%) from the noise-free SelfBound graph. Three independent evaluators, GPT-5.6, Claude Opus 4.8, and a domain expert, classify each annotation as a valid unknown (A) or an invalid case (B). Each evaluator is provided with the generated unknown annotation and the complete source documents from which the knowledge graph is constructed.

Table 10: Evaluator-specific validity of unknown annotations on SelfBound.

Evaluator	Holes	Valid Unknowns(A)	Invalid(B)	Precision
GPT-5.6	40	28	12	70%
Claude Opus 4.8	40	33	7	82.5%
Human	40	39	1	97.5%

A valid unknown has meaningful and correctly identified entities, a semantically plausible missing relation, and consistency with the original corpus evidence. It corresponds to a relational gap that is not explicitly supported by the source corpus. Invalid cases include relations incorrectly labeled as unknown despite corpus support, entity or relation misinterpretations, source-evidence misreading, and unsupported facts or semantic conclusions. After a case is judged invalid, its local subgraph and associated source chunks are used for error attribution. Local graph information is used for error attribution after invalidity is determined, rather than for deciding annotation validity.

Table 10 presents the overall precision results. The domain expert labels 39 of 40 annotations as valid (97.5%), while GPT-5.6 and Claude Opus 4.8 label 28 (70%) and 33 (82.5%) as valid, respectively. Table 11 summarizes the error categories assigned by the LLM evaluators. Counts refer to each evaluator's own invalid set (12 cases for GPT-5.6 and 7 for Claude Opus 4.8). GPT-5.6 and Claude Opus 4.8 agree on 31 of the 40 validity judgments and disagree on 9.

Table 11: Error-category counts among annotations judged invalid by GPT-5.6 and Claude Opus 4.8.

ID	Error Category	GPT-5.6	Claude Opus 4.8
B1	Incorrect Unknown Identification	8	5
B2	Unsupported Semantic Diagnosis	0	1
B3	Entity Misinterpretation	0	1
B4	Relation Misjudgment	0	0
B5	Source Evidence Misreading	4	0
B6	Other False Positive	0	0
	Total Invalid Annotations	12	7

B.3 Round-by-Round Growth of the Evolving Graph

This subsection examines the structural evolution of Substrata's knowledge graph across ten incremental growth rounds, the round-by-round behaviour of the DST fusion mechanism, and the accumulation of persistent-homology-based structural hole annotations. Table 12 presents the evolution of Substrata across ten incremental rounds. The left panel reports graph size and hierarchical community counts, where $|V|$ and $|E|$ denote the numbers of entities and relationships, respectively, and L0/L1/L2 represent community counts at different hierarchy levels. The middle panel summarizes DST fusion decisions as *merge/new*, indicating how many incoming communities are merged into existing communities or retained as new standalone communities at each level. The right panel reports the cumulative number of distinct structural-hole warnings and the number of community reports containing at least one hole annotation. It reports the underlying quantitative summary across all eleven snapshots, and Figure 5 visualises the evolving L0 community structure panel-by-panel.

Table 12: Evolution of the knowledge graph and structural-hole annotations across ten incremental rounds.

	Graph Growth						DST merge / new			Structural Holes	
Round	Papers	$\|V\|$	$\|E\|$	L0	L1	L2	L0	L1	L2	distinct warnings	annotated communities
0	15	1,839	2,786	18	112	28	—	—	—	393	16
1	25	4,311	5,513	25	168	31	23/7	76/56	2/3	779	8
2	34	4,894	7,402	25	173	31	15/0	63/5	—	1103	9
3	44	5,547	10,089	27	177	31	16/2	74/4	—	1483	10
4	54	6,109	12,090	30	188	31	21/3	67/11	—	1752	12
5	63	7,121	15,491	31	202	31	20/1	92/14	—	2279	13
6	71	7,365	16,845	31	204	31	11/0	49/2	—	2446	13
7	80	7,763	18,893	32	211	31	14/1	63/7	—	2649	13
8	90	8,071	20,374	32	217	31	12/0	52/6	—	2828	13
9	100	9,004	23,316	34	250	31	19/2	91/33	—	3246	17
10	110	10,140	26,660	35	262	33	20/1	107/12	1/2	3839	19

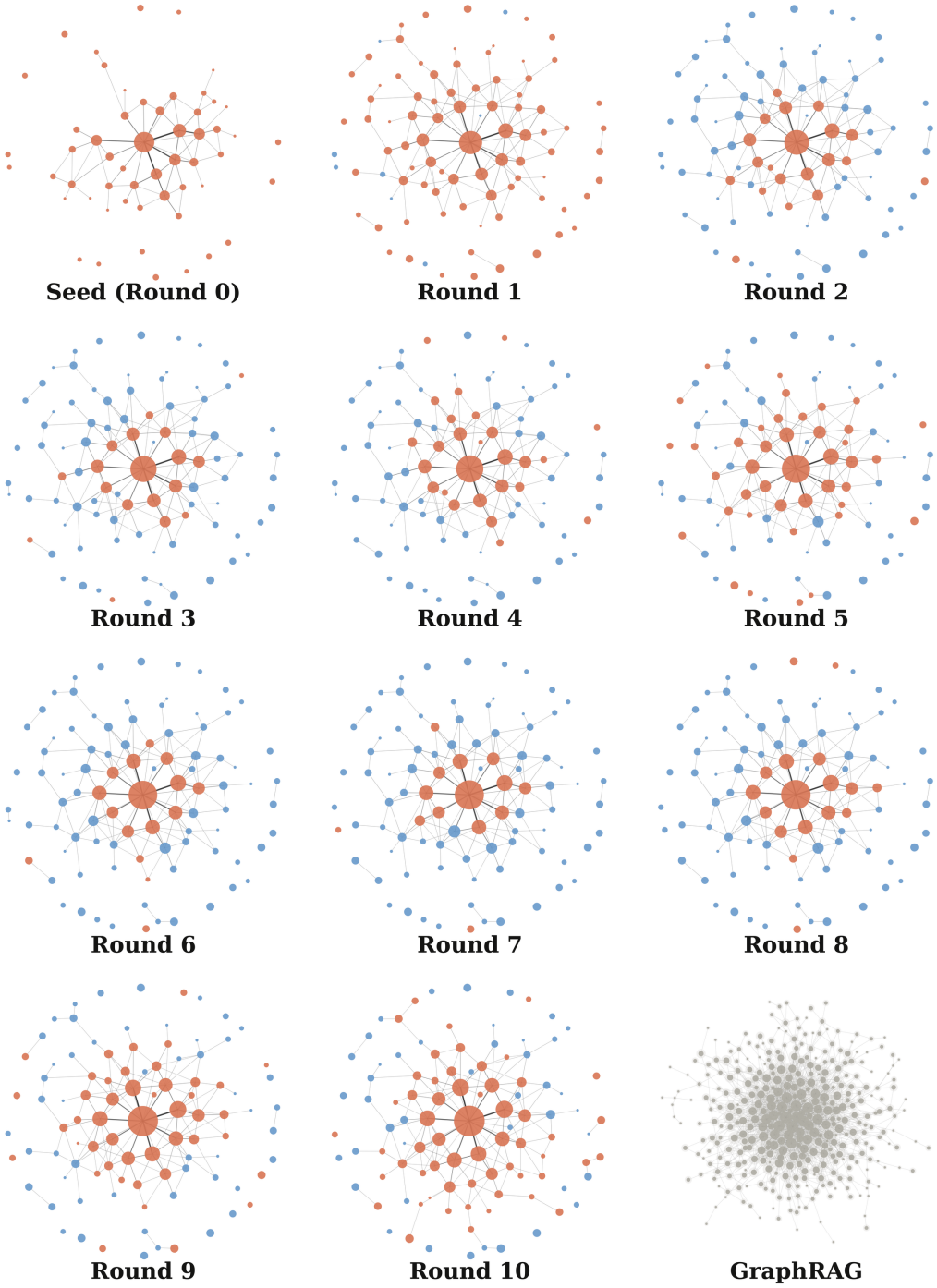

Figure 5: Per-round visualisation of the evolving knowledge graph at all eleven snapshots (Rounds 0–10).

B.4 Computational consumption

This section evaluates the computational cost of Substrata from two complementary perspectives. First, we examine the computational scalability of persistent homology as the knowledge graph grows across incremental rounds. Second, we compare the cumulative token consumption of Substrata with three graph-based RAG baselines, namely GraphRAG, HippoRAG2, and LightRAG. Together, these experiments assess the computational overhead introduced by structural-hole detection and the efficiency of the overall incremental graph construction pipeline.

Persistent homology runtime. We profile the computational scalability of persistent homology during incremental graph growth. Table 13 reports the numbers of vertices, edges, and detected structural holes, together with the H_1 pipeline time at each round. The complete H_1 pipeline requires only 1.0 second on a single CPU thread in Round 10, with a total cost of approximately 6.5 seconds over all 11 incremental rounds. The runtime increases smoothly as the graph grows, indicating that persistent homology is not the computational bottleneck in Substrata.

Table 13: Persistent homology profiling during incremental graph growth. Each row reports the graph size, detected structural holes, and measured H_1 pipeline time at the end of one round. The Round 10 pipeline takes 1.0 second on a single CPU thread; the reported times sum to approximately 6.5 seconds over all 11 rounds.

Round	Vertices	Edges	Detected Structural Holes	H1 Pipeline Time (s)
0	1,839	2,786	393	0.2
1	4,311	5,513	779	0.3
2	4,894	7,402	1,103	0.3
3	5,547	10,089	1,483	0.5
4	6,109	12,090	1,752	0.5
5	7,121	15,491	2,279	0.6
6	7,365	16,845	2,446	0.7
7	7,763	18,893	2,649	0.7
8	8,071	20,374	2,828	0.8
9	9,004	23,316	3,246	0.9
10	10,140	26,660	3,839	1.0

Token-cost comparison via graph-based method. Table 14 reports LLM tokens, embedding tokens, and total token consumption.

For GraphRAG, LLM tokens dominate the construction cost, accounting for more than 90% of the total expenditure at every round. At Round 10, GraphRAG with full reconstruction consumes 391.0M LLM tokens, compared with 34.0M for Substrata, resulting in an 11.5× difference. When embedding tokens are included, GraphRAG consumes 420.5M tokens and Substrata consumes 40.7M, corresponding to a 10.3× difference. The embedding-token difference is smaller at 4.4× because Substrata must still embed all newly added text chunks, whereas its LLM-based entity extraction and community-report generation are restricted to new documents and locally affected subgraphs.

Table 14: Cumulative token consumption of four graph-based RAG methods during graph construction. Values are reported in millions of tokens.

	LLM Tokens				**Embedding Tokens**				**Overall Tokens**			
Round	G-RAG	Sub.	Hippo	Light	G-RAG	Sub.	Hippo	Light	G-RAG	Sub.	Hippo	Light
0	7.8	3.1	0.6	5.9	0.6	0.6	0.1	0.9	8.4	3.7	0.7	6.9
2	40.0	9.3	1.4	13.4	3.1	1.8	0.3	2.1	43.1	11.1	1.7	15.5
4	95.2	15.5	2.3	21.3	7.5	3.0	0.5	3.3	102.7	18.5	2.8	24.7
6	171.8	21.7	3.1	28.1	13.3	4.3	0.7	4.4	185.1	25.9	3.7	32.4
8	271.0	27.8	3.9	35.1	20.6	5.5	0.8	5.5	291.5	33.3	4.7	40.6
10	391.0	34.0	4.9	45.8	29.5	6.7	1.0	7.4	420.5	40.7	5.9	53.2

B.5 Retrieval of unknown annotations

We evaluate whether unknown annotations can be retrieved from a mixed corpus on the SelfBound Spurious Relational Premise (SRP) subset described in Appendix C. We repeat this retrieval experiment three times and report both metrics averaged over the three runs. As shown in Table 15, hybrid retrieval combining Dense and BM25 achieves 75.0% annotation Recall@20, with 89.0% of questions retrieving at least one relevant annotation. The successful recall of structural-hole annotations does not depend on additional projection layers or manually designed query formats, but emerges from the complementary strengths of standard lexical and semantic retrieval.

Table 15: Retrieval of unknown annotations on the SelfBound Spurious Relational Premise subset. Results are averaged over three runs.

Retrieval Component	Annotation Recall@20	Questions with $\geq$1 Annotation Retrieved
Hybrid Retrieval (Dense + BM25)	75.0%	89.0%

C SelfBound Benchmark: Design Rationale and Full Question Set

This appendix details the construction of the SelfBound benchmark used in Section 4.1 of the main text. It complements the brief description in Section 3.4 with the full motivation, the hallucination-taxonomy mapping behind the Spurious Relational Premise category, and the complete 120-question set. To support reproduction and external evaluation, we release the questions and gold-standard answers together with the paper.

C.1 Motivation

Existing RAG benchmarks evaluate whether a system answers correctly when the evidence is present, or refuses correctly when the evidence is absent. Neither setting captures the failure mode that a graph-augmented retriever is most prone to: all relevant entities are present in the graph, relevant text chunks can be retrieved, but the connections between them are absent or too weak to support the relational inference implied by the question. We call this configuration a Spurious Relational Premise (SRP). Existing RAG evaluations do not cover this failure mode. The six unanswerability categories of Peng et al. [25] address question-level problems (underspecified, nonsensical, modality-limited, safety-concerned) and knowledge-base coverage problems (out-of-database, false presupposition), but not structural insufficiency in the graph. Generation-side faithfulness evaluation [22, 26] checks whether the generator is faithful to its retrieved context, but cannot detect when that context is structurally misleading. SRP targets the upstream problem: structural deficiencies in the knowledge graph that cause topologically proximate but evidentially unsupported entity pairs to produce misleading retrieval results.

C.2 Benchmark composition

SelfBound is a 120-question benchmark built over the corpus from which Substrata's knowledge graph is constructed, with gold-standard answers verified against source papers. The question taxonomy adapts four types from the Comprehensive RAG Benchmark [35], with 20 questions each. Simple with Condition retrieves a single fact under a filtering constraint, such as a specific metric reported by a given model under a specified dataset and configuration. Comparison requires the system to retrieve key attributes of two methods or architectures and produce a contrastive judgment along a specified dimension. Multi-hop chains reasoning steps across multiple papers along design-inheritance, contrast, or evolution relationships to derive the final answer. False Premise embeds a fabricated attribution or factual claim in the question, such as ascribing a design from one paper to another, and requires the system to detect and reject the false premise. The fifth type, Spurious Relational Premise (40 questions), is the contribution this benchmark centres on.

C.3 The SRP taxonomy

The 40 SRP questions are organised into four subcategories, each targeting a specific hallucination failure mode from the established taxonomy [11].

- **(A) Evidential Grounding Verification (10 questions).** Tests whether the system fabricates claims that have no explicit evidential support in the corpus, corresponding to factual fabrication (unverifiability hallucination).
- **(B) Entity Discrimination (10 questions).** Tests whether the system confuses similar-named entities and misattributes their properties, corresponding to factual contradiction (entity-error hallucination).
- **(C) Relationship Boundary Testing (10 questions).** Tests whether the system upgrades co-occurrence into causal or derivation relationships, corresponding to factual contradiction (relation-error hallucination).
- **(D) Directional and Conditional Relationship Testing (10 questions).** Tests whether the system respects asymmetric and context-dependent relationships, corresponding to faithfulness hallucination caused by logical inconsistency.

Each subcategory contains a deliberate mixture of 40–60% confirmable questions, where the relational premise is actually supported by the corpus, and 40–60% unconfirmable questions, preventing a trivial strategy of uniformly refusing all SRP queries.

C.4 Full question set

The four standard categories (Q1–Q4) and the SRP category (Q5) jointly constitute the 120-question SelfBound benchmark.

Q1: Simple with Condition (20 questions)

Q1. In the Transformer paper ("Attention Is All You Need"), what BLEU score did the Transformer (big) model achieve on the WMT 2014 English-to-German translation task?

Q2. In ALBERT, what self-supervised loss was proposed to replace BERT's Next Sentence Prediction (NSP) loss for modeling inter-sentence coherence?

Q3. In the ILSVRC-2012 image classification competition, what top-5 test set error rate did the winning entry (an ensemble of 7 CNNs) achieve, as reported in the AlexNet paper?

Q4. In the word2vec paper, when comparing architectures trained on the same data with 640-dimensional word vectors (Table 3), what was the semantic accuracy (%) of the Skip-gram model on the Semantic-Syntactic Word Relationship test set?

Q5. According to the Swin Transformer paper, what mIoU did the Swin-L model (with ImageNet-22K pre-training) achieve on the ADE20K val set for semantic segmentation?

Q6. In the original GAN paper (Goodfellow et al., 2014), what value of k (the number of steps to update the discriminator per generator update) was used in the experiments?

Q7. In the EfficientNet paper, what ImageNet top-1 accuracy did EfficientNet-B7 achieve?

Q8. In the CLIP paper, what top-1 accuracy did the best CLIP model achieve on ImageNet in the zero-shot setting?

Q9. In DenseNet, how are feature maps from all preceding layers combined as input to each layer — by element-wise summation (as in ResNet) or by concatenation?

Q10. According to the LLaMA paper, which model with only 13B parameters outperforms GPT-3 (175B) on most benchmarks despite being $10\times$ smaller?

Q11. In VGGNet, what is the convolution filter size used throughout the entire network (all convolutional layers)?

Q12. According to the Object Detection Survey, how many parameters (in millions) does GoogLeNet have, making it far more parameter-efficient than VGGNet (134M)?

Q13. In the Batch Normalization paper, the authors showed that applying BN to a state-of-the-art image classification model could match its performance using what fraction of the original training steps?

Q14. In the ConvNeXt paper, what ImageNet top-1 accuracy did ConvNeXt-XL achieve when pre-trained on ImageNet-22K and fine-tuned at 384^2 resolution?

Q15. In the Vision Transformer (ViT) paper, how does ViT handle 2D images as input to the standard Transformer encoder?

Q16. In BERT's Masked Language Model (MLM) pre-training task, what percentage of all WordPiece tokens in each sequence are randomly masked?

Q17. In the ResNet paper, what top-5 error rate did the ensemble of deep residual nets achieve on the ImageNet test set, winning 1st place in the ILSVRC 2015 classification task?

Q18. In the GPT-3 paper, what accuracy (%) did GPT-3 achieve on the TriviaQA benchmark in the few-shot (closed-book) setting?

Q19. In the LSTM paper, what fundamental problem of conventional recurrent network training (BPTT/RTRL) does LSTM specifically address?

Q20. According to the original MS COCO paper, how many common object categories does the MS COCO dataset contain in total?

Q2: Comparison (20 questions)

Q1. How do BERT and GPT (OpenAI GPT) differ in their pre-training objectives, and what consequence does this difference have for the directionality of context modeling?

Q2. What is the fundamental difference between GloVe and word2vec (Skip-Gram/CBOW) in how they leverage corpus statistics for learning word embeddings, and which performs better on the word analogy task under controlled conditions?

Q3. How do Swin Transformer and the original ViT differ in their self-attention computation strategy, and what are the implications for computational complexity with respect to image size?

Q4. How do GANs and DDPMs (Denoising Diffusion Probabilistic Models) differ in their training paradigm and generation mechanism?

Q5. How do YOLO (one-stage) and Faster R-CNN (two-stage) differ in their detection pipeline design, and what are the resulting trade-offs in speed and accuracy?

Q6. How do DenseNet and ResNet differ in their inter-layer connectivity patterns, and what impact does this have on parameter efficiency?

Q7. How does CBAM differ from SENet in the scope of its attention mechanism, and does the addition of spatial attention lead to performance improvement?

Q8. How do CLIP and MoCo differ in their contrastive learning supervision signals, and what downstream capabilities does this difference enable?

Q9. How does ConvNeXt compare to Swin Transformer in terms of architectural design philosophy, and can a pure ConvNet match a hierarchical vision Transformer in performance?

Q10. What parameter reduction techniques does ALBERT introduce compared to BERT, and how does ALBERT's inter-sentence pre-training loss differ from BERT's?

Q11. In a vision backbone that combines convolution and self-attention, which mechanism is more suitable for the shallow (early) stages versus the deep (late) stages of the network?

Q12. Compared to SSD's approach of building feature pyramids only from higher layers, does FPN's top-down pathway with lateral connections uniformly improve detection across all object scales?

Q13. Between attention (as used in DeiT) and simple pooling (as used in PoolFormer), which is the better token mixer for vision tasks?

Q14. Between CLIP (contrastive language-image pre-training) and MoCo (momentum contrastive visual learning), which produces better representations for downstream vision tasks?

Q15. Between global self-attention (as in ViT) and local window-based self-attention (as in Swin Transformer), which design is superior for vision tasks?

Q16. Between Batch Normalization (BN) and Layer Normalization (LN), which is more suitable for training deep vision models?

Q17. Between GPT-3's few-shot in-context learning and BERT's fine-tuning approach, which achieves better performance on NLP downstream tasks?

Q18. Is EfficientNet's compound scaling method always superior to single-dimension scaling (scaling only depth, width, or resolution)?

Q19. Between training an object detector from scratch with a proposal-free (one-stage) framework like SSD/YOLO and a proposal-based (two-stage) framework like Faster R-CNN, which approach is more effective?

Q20. Between using featurized image pyramids (multi-scale testing with image rescaling) and FPN's single-scale feature pyramid for multi-scale object detection, which approach is better?

Q3: Multi-hop (20 questions)

Q1. ConvNeXt claims a pure ConvNet can match Swin Transformer on dense prediction tasks like detection and segmentation. But what was the core architectural problem with the original ViT that Swin Transformer solved to become a general-purpose backbone, and which of Swin's solutions did ConvNeXt actually borrow versus replace with convolution equivalents?

Q2. Focal loss was specifically designed to address the massive foreground-background class imbalance caused by dense anchor boxes in one-stage detectors. CornerNet is anchor-free — so why does it still benefit from focal loss?

Q3. DeepLabv3+ combines atrous spatial pyramid pooling with an encoder-decoder structure to achieve strong semantic segmentation. What earlier architectural insight about combining coarse and fine features did it build upon, and why was Xception — originally designed for classification — a particularly suitable backbone choice for this segmentation model?

Q4. MoCo demonstrated that unsupervised pre-training can surpass supervised ImageNet pre-training on 7 detection and segmentation tasks. Given that contrastive self-supervised methods learn representations through instance discrimination without any multi-scale objective, what architectural component in the downstream detection framework was likely critical for bridging the gap between single-scale pre-trained features and multi-scale detection requirements?

Q5. ShuffleNet v2 argues that optimizing for FLOPs alone is misleading and proposes practical guidelines like balanced channel widths, avoiding excessive group convolutions, and minimizing element-wise operations. When ConvNeXt later adopted depthwise convolutions from MobileNet to modernize ResNet, how did it navigate the efficiency concerns that ShuffleNet v2 had identified about such operations?

Q6. SENet won ILSVRC 2017 by introducing channel attention through a "squeeze-and-excitation" mechanism that uses dimensionality reduction with two FC layers. Was this dimensionality reduction design actually necessary, and is there evidence from the project's papers that it might even be harmful?

Q7. DenseNet achieves strong parameter efficiency by connecting each layer to every subsequent layer via concatenation, enabling extensive feature reuse. Does ShuffleNet v2's "channel split" design achieve a similar feature reuse pattern, and if so, how does it avoid DenseNet's well-known memory inefficiency problem?

Q8. Mask R-CNN introduced RoIAlign to fix the spatial misalignment caused by quantization in RoIPool, which proved critical for pixel-accurate mask prediction. Cascade R-CNN independently identified a different kind of "mismatch" problem in detection. Are these two mismatch problems related, and could they compound each other in a system that does both detection and segmentation?

Q9. CLIP achieves impressive zero-shot classification by aligning image and text embeddings through contrastive learning on 400 million image-text pairs. CLIP-Adapter claims to improve on CLIP's downstream performance with simple feature adapters rather than prompt tuning. But from a representation learning perspective, why might CLIP's contrastive pre-training produce features that are particularly amenable to lightweight adaptation, compared to representations from standard supervised ImageNet pre-training?

Q10. Batch Normalization was proposed to address "internal covariate shift" and enabled training much deeper networks with higher learning rates. The Pre-activation ResNet paper later argued that the placement of BN relative to weight layers and activation functions is critical for very deep networks. What specific evidence from the Pre-activation ResNet work validates or extends BN's original claims about stabilizing training?

Q11. Res2Net claims to expose a new "scale" dimension that is more effective than the existing dimensions of depth, width, and cardinality. But GoogLeNet already used parallel filters with different kernel sizes to capture multi-scale information. What is fundamentally different about Res2Net's approach, and why does the Res2Net paper argue GoogLeNet's strategy was limited?

Q12. DSOD demonstrated that object detectors can be trained from scratch without ImageNet pre-training, using a DenseNet-based backbone. Why was DenseNet's architecture particularly well-suited for this from-scratch training scenario, while other backbone architectures like VGGNet failed?

Q13. CBAM extends SENet's channel-only attention by adding a sequential spatial attention module. For image classification, channel attention alone already provides consistent gains. What evidence from the detection literature suggests that spatial attention should matter more for localization-sensitive tasks like object detection than for classification?

Q14. VGGNet championed stacking simple 3×3 convolutions for depth, while GoogLeNet used complex multi-branch Inception modules with heterogeneous filter sizes for width. These seem like opposing design philosophies. How did ResNeXt manage to unify both ideas into a single, simple framework, and what evidence shows its approach is superior to both predecessors?

Q15. SSD was one of the first detectors to use a ConvNet's built-in feature hierarchy for multi-scale detection, yet FPN superseded it as the standard detection neck. What specific design flaw in SSD's approach to multi-scale features did FPN identify and fix, and how did RetinaNet's results validate FPN's solution?

Q16. Deformable convolution claims to be a generalization of atrous (dilated) convolution. DeepLabv3+ relies heavily on atrous convolution for semantic segmentation. If deformable convolution truly generalizes atrous convolution, why doesn't DeepLabv3+ simply use deformable convolution everywhere instead?

Q17. EfficientNet's compound scaling method proportionally scales width, depth, and resolution together using fixed coefficients. ConvNeXt, on the other hand, started from ResNet-50 and "modernized" it step-by-step toward Swin Transformer design at a fixed FLOP budget. Are these two scaling philosophies compatible, and does one subsume the other?

Q18. PANet added a bottom-up path augmentation on top of FPN to enhance low-level localization signals. But FPN already combines low-level and high-level features through lateral connections. What specific information propagation problem does PANet's extra bottom-up path solve that FPN's existing lateral connections cannot?

Q19. Word2vec showed that simple models trained on massive text data can learn surprisingly rich semantic structure (e.g., "King $-$ Man $+$ Woman $\approx$ Queen"). CLIP later learned joint visual-linguistic representations from 400 million image-text pairs. Is CLIP's contrastive learning fundamentally doing the same thing as word2vec but in a multi-modal space, or is there a deeper architectural difference?

Q20. Inception v3 factorized 5×5 convolutions into two consecutive 3×3 convolutions and even asymmetric $n \times 1$ and $1 \times n$ pairs to reduce computation while maintaining receptive field. ConvNeXt later moved in the opposite direction, using large 7×7 depthwise convolutions. Why did ConvNeXt reverse what seemed like an established design principle, and what made large kernels viable again?

Q4: False Premise (20 questions)

Q1. Given that ECA-Net improves upon SENet's channel attention by simultaneously employing both global average pooling and global max pooling for feature aggregation before applying its efficient 1D convolution, how does this dual-pooling aggregation strategy interact with the avoidance of dimensionality reduction to produce ECA-Net's performance gains over the original SE block?

Q2. Swin Transformer achieves cross-window connections by learning input-dependent offset values to adaptively determine the window displacement between consecutive self-attention layers, adopting an approach closely related to the offset learning paradigm in Deformable Convolutional Networks. How does this learned adaptive shifting compare to a naive sliding window approach in terms of computational efficiency and modeling power?

Q3. In ConvNeXt, the authors conducted a systematic exploration of depthwise convolution kernel sizes and demonstrated that classification accuracy continued to improve consistently as the kernel size was enlarged from 3×3 up to 11×11, after which it plateaued. Given this finding, what was the rationale for ultimately selecting 7×7 as the default kernel size in ConvNeXt?

Q4. SENet won first place in the ILSVRC 2017 classification competition by introducing the concept of "cardinality" as a new dimension that controls the number of aggregated transformations in each building block, complemented by its Squeeze-and-Excitation mechanism for channel recalibration. How do these two innovations jointly contribute to SENet's state-of-the-art performance?

Q5. In the Res2Net module, the feature map subsets produced after the initial 1×1 convolution are processed through independent parallel 3×3 convolutions, and their outputs are concatenated to form multi-scale representations with varying receptive field sizes. How does this parallel multi-branch design differ from the split-transform-merge strategy used in Inception modules?

Q6. In the Feature Pyramid Network (FPN), the top-down pathway generates higher-resolution feature maps by upsampling coarser maps and merging them with the corresponding bottom-up feature maps through channel-wise concatenation followed by a 1×1 convolution for dimensionality reduction. How does this concatenation-based fusion strategy help bridge the semantic gap between low-level and high-level features?

Q7. In Denoising Diffusion Probabilistic Models (DDPM), the forward process variance schedule $\beta_1, \ldots, \beta_T$ is jointly learned with the reverse process parameters during training, allowing the model to adaptively determine the optimal noise injection rate at each timestep. How does this learned variance schedule contribute to the quality of generated samples compared to using a fixed schedule?

Q8. Swin Transformer follows the positional encoding strategy introduced in ViT by adding learnable 1D absolute position embeddings to patch tokens at the input stage. Given this shared design, how does the hierarchical windowed self-attention in Swin Transformer compensate for the absence of 2D-aware positional information?

Q9. In the study of pre-activation Residual Networks, He et al. investigated multiple arrangements of activation functions within residual units and found that the ReLU-only pre-activation variant — where only ReLU (without Batch Normalization) is placed before the weight layers — achieved the best results on deep networks such as the 1001-layer ResNet. What properties of the ReLU-only design make it superior to the full pre-activation variant for extremely deep networks?

Q10. Denoising Diffusion Probabilistic Models (DDPM) were proposed as a direct improvement to the GAN training framework, replacing the adversarial minimax objective between generator and discriminator with a more stable variational bound optimization. How does DDPM's reformulation of the GAN objective into a denoising score matching loss resolve the well-known training instability issues of adversarial training?

Q11. In Xception, the author found that inserting a ReLU non-linearity between the depthwise convolution and the pointwise convolution in each separable convolution layer improves both convergence speed and final classification accuracy, mirroring the benefits of non-linearities observed in standard Inception modules. Given this finding, why did Xception choose ReLU over ELU as the intermediate activation function?

Q12. The word2vec Skip-gram model learns word embeddings by first constructing a global word-word co-occurrence count matrix from the entire corpus, and then factorizing it using a weighted least-squares objective that down-weights extremely frequent co-occurrences. How does this matrix factorization approach enable Skip-gram to capture both semantic and syntactic word relationships so effectively?

Q13. In MoCo, the ablation study on the momentum coefficient m for the key encoder update rule $\theta_k \leftarrow m\theta_k + (1-m)\theta_q$ showed that a relatively small momentum value of $m = 0.9$ yielded the best performance, as it allows the key encoder to adapt quickly to the evolving query encoder. Why does this fast adaptation produce more discriminative representations compared to using larger momentum values?

Q14. The original LSTM architecture proposed by Hochreiter and Schmidhuber (1997) introduced three multiplicative gate units — the input gate, the forget gate, and the output gate — to control information flow through the memory cell's Constant Error Carousel (CEC). How do these three gates coordinate to solve the vanishing gradient problem in recurrent neural networks?

Q15. In GoogLeNet's Inception module, cross-channel correlations and spatial correlations are decoupled by first performing depthwise spatial convolutions independently over each input channel and then applying 1×1 pointwise convolutions to project the results into a new channel space. How does this depthwise-first ordering in the Inception module contribute to GoogLeNet's computational efficiency compared to standard convolutions?

Q16. BERT introduced two pre-training objectives: Masked Language Modeling (MLM) and Sentence Order Prediction (SOP), where SOP is a self-supervised loss that requires the model to distinguish whether two consecutive text segments have been swapped. How does the combination of MLM and SOP enable BERT to learn fine-grained inter-sentence coherence for downstream tasks like Natural Language Inference?

Q17. In the deeper ResNet variants (ResNet-50/101/152), each residual function F uses a bottleneck design consisting of three stacked layers: a 3×3 convolution for spatial feature extraction, followed by a 1×1 convolution for channel reduction, and another 1×1 convolution for restoring the output dimension. What is the computational advantage of this bottleneck design over the two-layer 3×3 blocks used in the shallower ResNets?

Q18. In PoolFormer, the authors adopted standard Layer Normalization as the default normalization strategy after finding that it outperformed both Batch Normalization and Group Normalization in their ablation study. Given that PoolFormer's token mixer is a simple non-parametric pooling operation, why does standard Layer Normalization pair particularly well with this parameter-free token mixing approach?

Q19. CLIP jointly trains an image encoder and a text encoder using a contrastive objective on 400 million image-text pairs. To maintain representational consistency across the massive training dataset, CLIP adopts a momentum-updated text encoder similar to MoCo's key encoder design, where the text encoder's parameters are updated as an exponential moving average of the image encoder. How does this asymmetric momentum-based design improve CLIP's training stability compared to end-to-end contrastive approaches?

Q20. In BERT's Masked Language Modeling (MLM) pre-training, when a token position is selected for prediction, the token is replaced with the [MASK] symbol 90% of the time and left unchanged 10% of the time, in order to reduce the mismatch between pre-training and fine-tuning. How effective is this strategy at bridging the pre-train/fine-tune discrepancy compared to always using the [MASK] token?

Q5: Spurious Relational Premise (40 questions)

The 40 SRP questions are organised into the four subcategories described in Section C.3. Each subcategory contains 10 questions.

Sub-category A: Evidential Grounding Verification.

Q1. According to the corpus, is there direct textual evidence that CLIP was evaluated on the MS COCO dataset for object detection or image captioning tasks?

Q2. Does the corpus provide evidence that ResNet was used as a backbone for semantic segmentation evaluation on the ADE20K dataset?

Q3. According to the corpus, is there evidence that Kaiming He is explicitly credited as an author or creator of ResNet within the source passages of any paper?

Q4. Does the corpus contain any passage that documents ImageNet being used as a pre-training dataset for the CLIP model?

Q5. According to the corpus, is there textual evidence that ViT (Vision Transformer) was directly evaluated on the COCO object detection benchmark in the ViT 2021 paper itself?

Q6. According to the corpus, what role does LVIS play, and is there any passage that describes its annotation structure, category count, or relationship to the COCO dataset in detail?

Q7. The corpus mentions GhostNet in the context of efficient architectures. Is there any passage that reports a specific ImageNet top-1 accuracy figure for GhostNet?

Q8. Does any paper in the corpus explicitly state that the PASCAL Visual Object Challenge (VOC) was designed to complement the Caltech-101 dataset, or vice versa?

Q9. According to the corpus, is there evidence that SceneParse150 uses ResNet as a baseline architecture in its benchmark evaluation?

Q10. The corpus mentions Kinetics-400, Kinetics-600, and Kinetics-700 across different papers. Is there any passage that explicitly describes the differences between these three dataset versions in terms of category count or video count?

Sub-category B: Entity Discrimination.

Q1. ConvFormer (from MetaFormer Baselines 2023) and Conv2Former (from Conv2Former 2024) have confusingly similar names. According to the corpus, are they the same architecture or different? What are their respective design principles?

Q2. ResNet, ResNeXt, and ResNeSt all share the "Res" prefix and are part of the residual network family. According to the corpus, what is the key architectural innovation that distinguishes each from the others?

Q3. Swin Transformer (2021) and Swin Transformer V2 (2022) share the same name root. According to the corpus, what specific architectural modifications were introduced in V2 that are absent from V1?

Q4. MViTv2 (Multiscale Vision Transformer v2) and MPViT (Multi-Path Vision Transformer) have similar abbreviations. According to the corpus, are these the same model? What evidence distinguishes them?

Q5. The corpus mentions multiple researchers named "Zhou" across different papers. CLIP-Adapter 2023 cites a "Zhou et al. 2022" in the context of CoOp, From Show to Tell 2022 mentions a "Zhou et al." in image captioning, and MetaFormer Baselines 2023 lists "Pan Zhou" and "Yichen Zhou" as co-authors. According to the corpus, are these the same person or different people?

Q6. The surname "He" appears across multiple papers in the corpus. PReLU 2015 and Pre-act ResNet 2016 both list "Kaiming He" as first author, while ResNeSt 2022 lists a co-author named "Tong He", and MoCo 2020 is authored by "Kaiming He" as well. According to the corpus, are these the same person, and what distinct contributions does each paper attribute to its "He" author?

Q7. Multiple researchers named "Wang" appear across the corpus. HRNet 2020 lists "Jingdong Wang" (Microsoft Research) and "Xinggang Wang" (Huazhong University), while Conv2Former 2024 is authored by "Xinchao Wang" (MetaFormer Baselines 2023 also lists this Wang), and Swin Transformer V2 2022 does not list any Wang author. Are any of these "Wang" researchers the same person?

Q8. The corpus mentions "ImageNet", "ImageNet-1K", "ImageNet-5K", "ImageNet-21K", and "ImageNet-22K". According to the corpus, how are these variants related, and do they differ in scale or purpose?

Q9. The corpus mentions both "COCO" (or "MS COCO") and "COCO-Stuff". According to the corpus, are these the same dataset? Do they share the same annotation team or the same annotation scope?

Q10. The corpus references "Kinetics-400", "Kinetics-600", and "Kinetics-700" in different papers. MViTv2 2022 reports results on all three, while GCNet 2020 describes Kinetics-400 in detail. Are models' accuracy figures comparable across these three variants, or do they represent fundamentally different evaluation conditions?

Sub-category C: Relationship Boundary Testing.

Q1. Both SENet (2018) and the Swin Transformer (2021) report ImageNet classification results and are evaluated on image classification benchmarks. Does the corpus establish any direct methodological relationship between SENet's channel attention mechanism and Swin Transformer's shifted window self-attention?

Q2. MoCo (2020) and CLIP (2021) both use contrastive learning objectives and both report results involving ImageNet. Does the corpus establish that one is derived from or directly builds upon the other?

Q3. ConvNeXt (2022) and EfficientNet (2019) both report ImageNet-1K top-1 accuracy and both are CNN-based architectures. Does their shared benchmark usage imply they were developed by related research teams or share a direct architectural lineage?

Q4. PVT (2021) and Swin Transformer (2021) are both hierarchical vision Transformers evaluated on both ImageNet classification and COCO object detection. Does the corpus establish that they share a direct architectural derivation or collaborative relationship?

Q5. CBAM (2018) and Non-local Networks (2018) both use attention mechanisms and are both evaluated on COCO for object detection. Does this shared mechanism category and benchmark imply that one builds upon the other?

Q6. ShuffleNet (2018) and ConvNeXt (2022) both use depthwise convolutions in their architectures. Does this shared mechanism imply that ConvNeXt inherits from or is methodologically related to ShuffleNet?

Q7. Both HRNet (2020) and FPN (2017) produce multi-scale feature representations and are used for object detection on COCO. Does the corpus establish that HRNet's multi-resolution design is derived from FPN's top-down feature pyramid?

Q8. FasterRCNN 2016 and Swin Transformer 2021 were both published in venues under the IEEE umbrella (TPAMI and ICCV respectively). Does this shared publication venue imply any collaborative or organizational relationship between their research teams?

Q9. ACmix (2022) argues that self-attention and convolution share underlying 1×1 convolution operations. Conformer (2021) fuses CNN and Transformer branches in a dual-branch architecture. Both combine attention and convolution. Does the corpus establish that ACmix's decomposition approach is derived from Conformer's fusion approach?

Q10. GCNet (2020) and CBAM (2018) are both evaluated on MS COCO for object detection and both propose attention mechanisms for CNNs. GCNet's paper explicitly discusses SENet and Non-local Networks. Does this imply that GCNet is a variant of CBAM?

Sub-category D: Directional and Conditional Relationship Testing.

Q1. Swin Transformer (2021) evaluates on the same ImageNet benchmark that AlexNet won the ILSVRC competition on in 2012. Does this mean Swin Transformer participated in or is connected to the ILSVRC-2012 competition?

Q2. EfficientNet (2019) places AlexNet (2012), GoogLeNet (2014), and ResNet (2016) in a historical lineage of CNN accuracy progression. Does this mean AlexNet's paper itself discusses or acknowledges GoogLeNet and ResNet as successors?

Q3. The PASCAL VOC dataset (2009 paper) predates the COCO dataset (2014 paper). FasterRCNN 2016 reports results on both PASCAL VOC and COCO, and shows that a COCO-trained model achieves 76.1% mAP on VOC 2007 without any VOC fine-tuning. Does this mean the PASCAL VOC paper itself documents this COCO-to-VOC transfer relationship?

Q4. Caltech-101 appears in both AlexNet 2012 (as a historical small-scale dataset) and CLIP-Adapter 2023 (as a modern few-shot evaluation benchmark). Do these two temporal contexts represent the same use of Caltech-101, or has its role changed?

Q5. MetaFormer Baselines 2023 explicitly builds upon and cites PoolFormer 2022, treating PoolFormer as a predecessor baseline. Does PoolFormer 2022 in turn reference or acknowledge MetaFormer Baselines 2023?

Q6. PVT 2021 cites and compares against ViT, showing that ViT-Small/32 achieves only 31.7 AP on COCO val2017 for object detection, while PVT-Small achieves 40.4 AP. Does ViT 2021 in turn cite PVT or acknowledge its detection superiority?

Q7. Conv2Former 2024 compares against SENet 2018, noting that "adding a Sigmoid function as done in SENet decreases the performance by more than 0.5%." Does SENet 2018 acknowledge or discuss Conv2Former's convolutional modulation approach?

Q8. ResNet achieves strong results on ImageNet image classification. The corpus also documents ResNet as a backbone for COCO object detection (e.g., in FasterRCNN 2016) and ADE20K semantic segmentation (e.g., in CoT 2022). Does ResNet's strong classification performance on ImageNet guarantee equally strong performance relative to newer architectures on detection and segmentation tasks?

Q9. CLIP 2021 reports that zero-shot CLIP outperforms ResNet-50 features by 14.5% on Kinetics-700 for action recognition. MViTv2 2022 reports 79.4% top-1 accuracy on Kinetics-700 with supervised fine-tuning. Can these two results be directly compared to conclude which model is better on Kinetics-700?

Q10. MoCo 2020 demonstrates that self-supervised pre-training on ImageNet-1M can match or outperform supervised pre-training for COCO object detection and several other downstream tasks. Does this mean MoCo's advantage generalizes to all downstream tasks in the corpus?

C.5 Essential structural holes

Structural Hole 1: Contrastive vs. Triplet metric learning

Persistence: $+\infty$ **Entities:** CONTRASTIVE TASK, CONTRASTIVE LOSS, CLIP, IMAGENET, PSLT, TRIPLET LOSS

Semantic diagnosis. This structural hole is essential because it crosses two metric-learning loss families that are connected through ImageNet but never directly compared in any single paper of the corpus. The six entities partition cleanly into two distinct lineages. The first lineage covers contrastive losses in self-supervised and multi-modal settings, where contrastive objectives in the InfoNCE form drive ImageNet pre-training (MoCo) or contrastive language–image pre-training (CLIP). The second lineage covers triplet losses in re-identification and fine-grained matching, where PSLT is first pre-trained on ImageNet for general visual features and then fine-tuned with triplet loss and BNNeck for person re-identification, in the closed tradition that traces back to FaceNet and the Market-1501 re-ID literature. The two loss families share the same underlying principle — pull positive pairs together, push negative pairs apart in feature space, with triplet loss being approximately equivalent to contrastive loss with a margin and a single negative.

Why the gap is essential. No chunk in the corpus discusses contrastive and triplet losses comparatively. They are mentioned in entirely separate contexts: contrastive-lineage chunks discuss ImageNet pre-training, zero-shot transfer, and multi-modal alignment; triplet-lineage chunks discuss re-ID benchmarks, BNNeck, and identity-preserving features. No paper in the 110-paper corpus says "triplet and contrastive losses are formal alternatives" or proposes one as a replacement for the other. The conceptual link is therefore absent in the source.

Downstream constraints. If a query asks how PSLT relates to CLIP and retrieval surfaces only the contrastive-lineage chunks, the generator has no signal that PSLT lives in a different research tradition; it may then fabricate a comparison ("PSLT applies CLIP-style contrastive learning") or claim influence ("PSLT inherits from CLIP"). The annotation chunk warns the generator that contrastive loss (CLIP / MoCo / self-supervised lineage) and triplet loss (PSLT / re-ID lineage) converge on the same metric-learning principle but are studied in disjoint research traditions in this corpus. Do not fabricate inheritance, competitive comparison, or shared methodology between models from these two lineages unless explicitly supported by the text.

Structural Hole 2: Multi-path scheme vs. Grouped convolution

Persistence: $+\infty$ **Entities:** GPU UTILIZATION, MULTI-PATH SCHEME, MVIT-T, IMAGENET, RESNET-101, GROUPED CONVOLUTION

Semantic diagnosis. This structural hole connects two architectural micro-design choices — transformer multi-path schemes and ConvNet grouped convolution — that have been independently observed to produce suboptimal GPU utilisation. Unlike Structural Hole 1, the corresponding link does live in the corpus: the MPViT paper explicitly draws the parallel between its own multi-path scheme and grouped convolution, attributing both to the same hardware-level mechanism (GPU context switching, kernel synchronisation). However, this parallel was extracted as two separate edges (MULTI-PATH SCHEME $\rightarrow$ GPU UTILIZATION and GROUPED CONVOLUTION $\rightarrow$ GPU UTILIZATION) rather than as a direct edge between the two methods. As a result, the structural hole has to traverse the long path through MVIT-T $\rightarrow$ IMAGENET $\rightarrow$ RESNET-101 (the architectural / benchmark backbone detour) instead of closing via the chord that the source actually supports.

Why the gap behaves differently from Structural Hole 1. Real-world retrieval that surfaces the MPViT chunk will already see the explicit parallel observation, so a thinking model has all the evidence needed to correctly synthesise the connection between MPViT's multi-path scheme and ResNeXt's grouped convolution. The structural hole is therefore better described as an *internal extraction inconsistency* than as a genuine relational gap.

Downstream constraints. The annotation flags this structural hole as a candidate for re-extraction: the chunk that originated both edges should produce a direct MULTI-PATH SCHEME $\leftrightarrow$ GROUPED CONVOLUTION edge with the relationship description "share the same GPU-utilisation bottleneck mechanism". Until that chord is added to the graph, the annotation instructs the generator to retain caution but not to refuse the synthesis when the MPViT chunk is itself in context.

Structural Hole 3: Small-object-detection methodology family

Persistence: $+\infty$ **Entities:** RFLA, FASTER R-CNN, OBJECT DETECTION, SMALL OBJECTS, SAMPLE-ORIENTED METHODS, OPTIMIZED LABEL ASSIGNMENT

Semantic diagnosis. The six entities form a coherent methodological narrative: RFLA is a label-assignment technique that improves Faster R-CNN's object-detection performance for small objects by treating it as a sample-oriented optimisation problem under the optimised-label-assignment family. All six entities live in the same L0 community at the macro level (C13, "YOLO and Fast R-CNN Object Detection Systems"), yet Leiden's L1 partition splits them across four sub-communities. The structural hole is essential because every potential chord — e.g. a direct edge from OBJECT DETECTION to OPTIMIZED LABEL ASSIGNMENT, or from SMALL OBJECTS to FASTER R-CNN — is absent in the underlying entity graph despite the obvious conceptual proximity.

Why this differs from a cross-domain gap. Unlike Structural Holes 1 and 2, this hole does not bridge two research traditions. It fragments a single methodological topic across multiple Leiden sub-clusters because the chain of "is-a" / "applies-to" / "is-improved-by" relationships does not contain short-cuts. The structural hole is therefore better described as a *Leiden granularity artefact* that signals tightly-knit research breadth rather than a genuine cross-paradigm gap.

Downstream constraints. The annotation flags this hole as low-risk for hallucination: a generator that retrieves any subset of these entities is safe to synthesise their methodological relationship, since the chain is well-grounded in the corpus. The constraint is conservative: do not invent specific quantitative claims (e.g. "RFLA improves Faster R-CNN by X mAP on COCO") unless directly supported by retrieved chunks.

Structural Hole 4: Transformer-compression technique family

Persistence: $+\infty$ **Entities:** ASH, EFFICIENT TRANSFORMER, TRANSFORMER, BERT, ALBERT, DECOMPOSITION

Semantic diagnosis. The hole traces a path through several Transformer compression and efficiency strategies. Three distinct compression mechanisms appear: ALBERT's cross-layer parameter sharing and factorised embedding (the DECOMPOSITION edge), generic efficient-Transformer architectures (sparse attention, low-rank approximations), and ASH's attention-head sharing. All target the same goal — reducing the parameter or compute footprint of Transformer / BERT-style models — but the corpus does not contain a single paper that systematically compares them on a common axis.

Why the gap is essential. The closing chords would be direct comparisons between ALBERT and ASH, or between decomposition-based methods and attention-sharing methods. No such chord exists in the corpus: each compression technique is documented independently with its own ablation against an uncompressed baseline, but cross-technique comparisons are absent.

Downstream constraints. The annotation warns the generator against claiming a quantitative or qualitative ranking among compression techniques in this family ("ASH outperforms ALBERT", "decomposition is more effective than parameter sharing") unless retrieved chunks contain such a comparison. Independent claims about each technique relative to its own baseline remain valid.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in the Conclusion (Section 6).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: We do not provide theoretical claims in this paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have detailed the system pipeline, experimental settings in Section 3, Section 4, and Appendix B.1 to ensure the reproducibility of our results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and data will be released upon acceptance of the paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We report the experimental settings and details in Section 4 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: For each question, we submit it three times to each system and report results based on majority voting. We do not report error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Our experiments do not require special hardware, and can be run on most laptops with production models. They require only an API key from a model provider-this is clear from the paper. We note that the overall research project consumed additional tokens beyond those reported, attributable to preliminary and failed experiments rather than to the method itself.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research is very ethical, and does not breach any of the NeurIPS ethics guidelines.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper presents foundational research on methodology for knowledge graph construction. The work does not introduce deployed systems or applications with a direct path to societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work does not release pretrained models, scraped datasets, or other assets with a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All existing assets used in this paper are properly credited, and their licenses and terms of use are respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets introduced in this paper are released with sufficient documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve research with human subjects, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are used in standard components of the RAG framework, including entity and relation extraction, answer generation, and auxiliary answer evaluation. These uses follow common practice in existing RAG systems and are not the original methodological contribution of this work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.