MOTIF-DRIVEN, LLM-FREE BENCHMARK CONSTRUCTION FOR QUESTION ANSWERING OVER IT KNOWLEDGE GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

As operational environments become increasingly heterogeneous, organisations are adopting graph-based representations to integrate knowledge across incidents, services, configurations, dependencies and remediation procedures. This creates a need for reliable benchmarks that can evaluate question answering over the resulting operational knowledge graphs. Such evaluation has a fundamental ground-truth problem: the answer is a structured graph object whose correctness depends on the graph's schema and instances, yet benchmark references are commonly produced by language models. This makes it possible for a reference to contain plausible but unsupported relations, making disagreements between a system and its reference difficult to diagnose. We present a graph-first approach to benchmark construction in which ground truth is computed, rather than generated. Starting from structurally informative patterns in a knowledge graph, we compile deterministic graph operations into reference subgraphs and only then generate natural-language questions that describe those references. Language models therefore determine how a question is phrased, but not what constitutes its answer. The same grounding mechanism identifies questions that cannot be answered from the graph and separates schema-level, instance-level, and grounding ambiguities.

1 INTRODUCTION

In IT organisations, there is a need to integrate operational knowledge that is otherwise fragmented across incident records, runbooks, post-mortems, service dependencies, and configuration-management repositories such as Ansible and Chef. Graph-based representations can unify these heterogeneous artifacts into a structured representation in which incidents are connected to symptoms, runbooks to the services and playbooks they govern, and problems to their corresponding resolutions, enabling graph-based retrieval and operational question answering. As these graphs become a foundation for AI-assisted operations, a critical challenge emerges: how to reliably evaluate question-answering systems over operational knowledge graphs. In this paper, we build a Context Knowledge Graph (CKG) as the graph-based representation of operational knowledge. The CKG we instantiate the method on is constructed from ITBench (Jha et al., 2025) incident scenarios and comprises 553 nodes over 15 node types and $1{,}090$ edges over 17 relation types.

Questions over a CKG often require structured answers. 'Which incidents have no automated remediation?" requires an exact node set satisfying a gap condition; 'How is a Kafka fault in the checkout service resolved?" requires an ordered procedure spanning runbook steps and automation tasks; and "Which services are affected if this namespace goes down?" requires a dependency subgraph. Evaluation therefore requires a reference subgraph rather than a reference string, and that subgraph must respect the graph's actual schema and relations. In an operational CKG, different sources may contribute different relation structures, so a reference that conflates them or asserts a relation absent from the graph cannot be used reliably for evaluation.

A common approach is to use a language model to generate both questions and reference answers. In operational settings, this can be risky: if a reference contains an unsupported recommendation, it may be mistaken for a valid action by downstream automation.We take a different approach. References are compiled directly from graph operations over a fixed snapshot, questions are generated only after

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

their supporting structures have been identified, and unanswerable questions are kept as explicit outcomes. We make the following contributions:

1. **Executable reference construction for operational CKG QA** (§4). We define four question families and derive their references directly from graph structure, using generation only where natural-language output is required.
2. **Evidence-bounded generation and abstention** (§4.6; §4.4). Generated content is constrained to the compiled reference, while unsupported questions are retained as explicit out-of-schema, missing-instance, or ambiguous-grounding cases.
3. **Graph-derived question generation** (§5). We mine informative graph structures, compile their references first, and generate questions afterwards, separating question formulation from ground-truth construction.
4. **Validation and evaluation** (§6; §7). We provide intrinsic validation and measures for unsupported relation claims, and instantiate the framework on a production operational CKG with released artifacts.

System-level evaluation against the resulting benchmark is reported in §7.

2 Limitations of Existing Ground-Truth Practice

Existing work relevant to our setting falls into several related lines. First, GraphRAG systems and recent GraphRAG benchmarks study retrieval and reasoning over graph-structured evidence (Edge et al., 2025; Xiang et al., 2025; He et al., 2024). These works establish the value of graph-based retrieval, but do not directly address how reliable, graph-grounded reference subgraphs should be constructed for an evolving operational CKG.

Second, executable KGQA datasets such as QALD, WebQuestionsSP (Yih et al., 2016), ComplexWebQuestions (Talmor & Berant, 2018), GrailQA (Gu et al., 2021),GrailQA++ Dutt et al. (2023), and MetaQA (Zhang et al., 2018) derive answers through executable or structured representations. This provides the grounding principle we adopt, but these resources are primarily human-authored and designed for relatively stable, open-domain knowledge bases. They do not directly cover operational gap, negation, impact, and prescriptive questions or snapshot-aware regeneration.

Third, recent work has moved toward automatically generating graph-grounded QA data Steinmetz & Sattler (2021). Query-graph-based question generation, KG-QAGen,UniKgQa, M^3GQA, SynthKGQA/GTSQA, and KGQAGen explore structured question generation, multi-hop reasoning, and graph-grounded references (Wang, 2022; Zhang et al., 2026; Peng et al., 2025; Cattaneo et al., 2026; Banerjee et al., 2024; Jiang et al., 2023). These efforts are particularly close to ours, but our focus is on operational CKGs where reference construction must additionally support type-specific compilation, explicit abstention, provenance, version binding, and validation under graph mutation.

Finally, reference-free RAG evaluation frameworks such as RAGAS (Es et al., 2024), ARES (Saad-Falcon et al., 2024) , Zhou et al. (2026) evaluate properties such as context relevance and answer faithfulness without requiring a reference answer. They are complementary to our approach, but cannot establish whether the evidence retrieved by a system is the correct graph evidence for the question. Operational benchmarks such as ITBench (Jha et al., 2025) further demonstrate the need for evaluation in real-world IT automation settings, but do not provide the graph-derived reference construction considered here.

Prior work provides important components of graph-based retrieval, executable KGQA, automated question generation, and reference-free evaluation. Our focus is their combination into a reproducible ground-truth construction process for evolving operational context knowledge graphs, with explicit handling of abstention, provenance, and graph change.

3 Formal Setting

We represent the operational CKG at version t as a typed, attributed multigraph

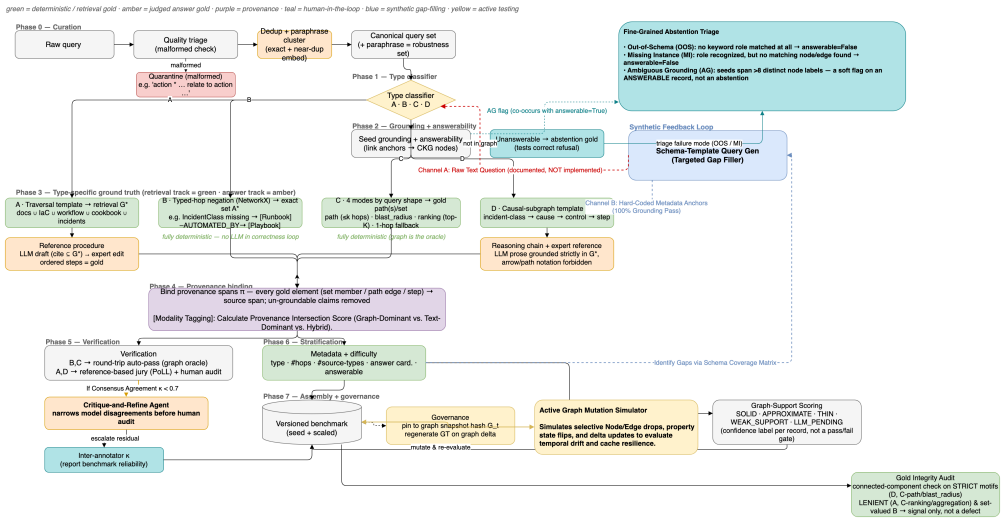


Figure 1: The proposed construction pipeline over a pinned graph snapshot. Deterministic stages compile references by graph traversal; generative stages are confined to answer wording and are admitted only under the grounding constraint of §4.6.

$$G_t = (V, E, \ell, \rho),$$

where V denotes nodes, E typed directed edges, ℓ node labels from the ontology, and ρ node and edge attributes, including textual properties. Each snapshot is identified by a content hash $h(G_t)$, allowing references to be tied to the exact graph version from which they were constructed.

Given a natural-language question q, the construction produces a reference record

$$\Gamma(q; G_t) = \langle \tau, G^*, A^*, \pi, \mu, \sigma \rangle,$$

where τ is the question family, $G^* \subseteq G_t$ is the reference subgraph, A^* is an optional natural-language answer, π records provenance, μ identifies the evidence modality, and σ contains stratification and trust metadata.

The construction proceeds through eight stages: curation, typing, grounding, compilation, provenance binding, verification, stratification, and assembly. The complete stage definitions and their intermediate representations are given in Appendix B.

4 Proposed Method: Deterministic Ground-Truth Construction

4.1 Design Principles

The construction is guided by three principles. First, the graph is the source of truth whenever an answer can be determined from its structure; such answers are computed through graph operations rather than generated by a language model. Second, generation is used only when natural-language formulation is necessary. In these cases, the reference subgraph is still constructed deterministically, and generated text is accepted only if it satisfies the grounding constraints. Third, the benchmark makes reference reliability explicit by assigning each item a trust level based on the evidence supporting it.

As shown in Figure 1, the pipeline proceeds from question curation and typing to grounding and type-specific reference compilation. The resulting reference is then linked to its provenance, verified, stratified by evidence and reliability, and assembled into a benchmark tied to a specific graph snapshot.

4.2 CURATION AND TYPING

We first group duplicate and near-duplicate questions using embedding similarity. One question is retained as the canonical version, while the others are kept as paraphrases for testing invariance to surface form. Malformed questions are quarantined rather than silently removed, keeping curation decisions auditable.

Each canonical question is assigned to one of four families, reflecting the form of its reference: **(A)** procedural questions require an ordered procedure; **(B)** gap or negation questions return the nodes that violate an expected structural condition; **(C)** relational and impact questions return relevant paths or affected entities; and **(D)** causal or prescriptive questions require a causal subgraph and associated reasoning chain.

Routing uses ordered rules with an optional model-based fallback. Because the reference is constructed by the selected family-specific procedure, routing errors can be detected during validation and corrected. Routing therefore determines how a question is processed, not what constitutes a correct answer.

4.3 GROUNDING AND ANSWERABILITY

Grounding links the entities and concepts in a question to the CKG using three signals: *role grounding*, which maps mentioned categories to ontology labels; *lexical grounding*, which searches node names and stored properties; and *semantic grounding*, which retrieves sufficiently similar nodes.

Let $S(q) \subseteq V$ be the grounded seed nodes, $R(q)$ the target roles, and $\Pi(q)$ the property-level matches. We define

$$\mathrm{ground}(q) = [R(q) \neq \varnothing \ \vee \ S(q) \neq \varnothing \ \vee \ \Pi(q) \neq \varnothing] \,,$$

and consider a question answerable when the grounding also provides evidence required by its compilation procedure:

$$\mathrm{ans}(q) = [\mathrm{ground}(q) \wedge \mathrm{evidence}(q, G_t) \neq \varnothing] \,.$$

The distinction matters for set-valued questions: "which incidents lack a playbook" need not name an individual incident, since role grounding can identify the relevant class and the graph can determine the set. Questions that cannot be grounded or supported are retained and passed to the abstention mechanism rather than discarded. Detailed grounding definitions are given in Appendix B.

4.4 UNANSWERABILITY AS A REFERENCE OUTCOME

When **ans(q) = false** , the construction produces an explicit abstention rather than forcing an answer. We distinguish three cases: *out of schema* when the requested concept is absent from the ontology; *missing instance* when the concept exists but the requested entity is absent from the current snapshot; and *ambiguous grounding* when multiple competing anchors remain unresolved.

The abstention reason is recorded as

$$\mathrm{abstain}(q) = \begin{cases} \mathrm{OOS} & \text{concept} \notin \mathcal{L}, \\ \mathrm{MI} & \text{concept} \in \mathcal{L}, \ \text{instance} \notin G_t, \\ \mathrm{AG} & \text{multiple competing anchors remain.} \end{cases}$$

In each case, $\tau = \textsc{Abstain}$, while the question and its metadata remain in the benchmark. This allows evaluation to distinguish a correct abstention from an unsupported answer.

4.5 TYPE-SPECIFIC COMPILATION

The reference construction depends on the question family, since each family requires a different form of graph evidence.

Procedural questions (A). We construct a bounded typed neighbourhood around the grounded anchors. The subgraph is deterministic; a language model is used only to express it as an ordered procedure, subject to the grounding constraint in §4.6.

Gap and negation questions (B). We define the expected structural condition and return the nodes that violate it. For example, "What type of incidents still lack a playbook?" is compiled as

$$\{\, v : \ell(v) = \texttt{IncidentClass},\ \nexists\, v \rightsquigarrow \texttt{AnsiblePlaybook} \,\}.$$

Membership is determined independently by bounded typed reachability, making the result deterministic and independent of evaluation order.

Proposition 1 (Soundness and exactness of $\mathcal{O}_B$)**.** *For a label $\lambda \in \mathcal{L}$ and bounded typed reachability predicate φ,*

$$\mathcal{O}_B(\lambda, \varphi) = \{v \in V : \ell(v) = \lambda,\ \neg\varphi(v)\}$$

is complete, sound, and independent of evaluation order.

The proof is given in Appendix D.

Relational and impact questions (C). Relational questions are answered by bounded path enumeration, while impact questions identify nodes connected to an affected component through dependency relations. The traversal determines the reference; sparse evidence is retained and reflected in the trust label rather than supplemented with unsupported links (§4.9).

Causal and prescriptive questions (D). We construct a causal subgraph linking symptom, problem, and resolution, optionally including the automation that implements it. The subgraph is deterministic, while the language model only expresses the resulting reasoning chain. Compound questions compose these procedures; for example, a question about resolutions for incidents lacking automation first computes the family B gap set and then restricts the causal expansion to that set.

4.6 Bounding the generative component

Where generation is used, it is constrained by the compiled reference. Let $\mathcal{E}(x)$ denote the entities mentioned in a generated string x, and G^* the corresponding reference subgraph. We define grounding precision as

$$\mathrm{gp}(x) = \frac{|\{e \in \mathcal{E}(x) : e \in G^*\}|}{|\mathcal{E}(x)|}.$$

A generated answer is accepted only if $\mathrm{gp}(x) \geq \gamma$. With $\gamma = 1$, every entity mentioned must occur in G^*; unsupported components or procedural steps therefore cause the draft to be rejected and regenerated. The same constraint is applied to generated questions in §5. This keeps generation responsible for wording rather than determining the underlying evidence.

4.7 Provenance and retrieval modality

Each reference element, including set members, path edges, and procedural steps, is linked to its source evidence; elements without supporting provenance are removed. References are also tagged as relying on graph structure, node text, or both. This allows later evaluation to distinguish failures in graph traversal from failures in interpreting node text.

4.8 Verification

Verification follows the compilation procedure. For families B and C, the reference is re-computed from the question and pinned snapshot; the result must be identical, with any discrepancy indicating non-determinism. For families A and D, a panel of model judges evaluates whether the generated

wording is supported by the reference subgraph. Low agreement triggers a critique-and-refine step, after which the revised wording must again satisfy the grounding constraint before human audit is considered.

Property 1 (Reproducibility under re-execution)**.** *For $\tau \in \{B, C\}$, the compiled reference G^* is a deterministic function of the question q and the pinned snapshot G_t. Re-executing the compilation procedure on (q, G_t) therefore returns the same G^*.*

The proof is given in Appendix D.

4.9 Stratification, trust labelling and governance

Each reference records its family, hop depth, contributing modalities, answer cardinality, answerability, and difficulty bin, together with a trust label: SOLID for deterministic and verified references, APPROXIMATE for lexical or semantic grounding, THIN for sparse supporting evidence, and WEAK when the graph lacks the structure required by the question.

Coverage over node-role and question-family combinations is recorded in a schema coverage matrix. The benchmark is pinned to the graph snapshot's content hash, allowing affected references to be regenerated when the graph changes. Empty coverage cells are passed to a synthetic feedback loop that generates schema-templated questions through either the normal typing and grounding pipeline or direct metadata anchoring for guaranteed grounding.

5 Proposed Method: Graph-Derived Question Generation

A benchmark needs questions as well as reference answers. Manual authoring is difficult to scale and may miss structurally demanding cases, while generating questions first and finding supporting evidence afterwards can introduce a mismatch between the question and its reference. We therefore reverse the process: mine an informative graph structure, compile its reference, and generate the question only afterwards. The reference is thus fixed before question generation and cannot be determined by the generated wording.

Motif mining. We mine structures suited to each question family: violation sets for gap questions, symptom–problem relations and dependency closures for relational questions, symptom–problem–resolution chains for causal questions, and runbook hubs linking symptoms, resolutions, and incident classes for procedural questions.

For path-based questions, we favour structures passing through nodes with high betweenness centrality,

$$\mathrm{bc}(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}},$$

where σ_{st} is the number of shortest paths between s and t, and $\sigma_{st}(v)$ the number passing through v. This favours questions whose answers require graph traversal rather than simple lexical matching.

Grounded synthesis. Given a mined structure and its compiled reference, a question is generated either from a template containing the relevant node names or through a constrained model rewrite. The grounding constraint in §4.6 requires every named entity to occur in the reference subgraph. Templates satisfy this at $\gamma = 1$ by construction; model rewrites are explicitly gated.

In our instantiation, the gate rejected 40 of 64 candidate rewrites and retained 24 grounded question–reference pairs. The surviving questions use an operational register, while their references remain fixed to the previously compiled subgraphs.

6 Instantiation and Intrinsic Validation

We validate the construction using three complementary checks: reproducibility under re-execution, sensitivity to graph changes, and coverage. These tests evaluate the reference construction itself and

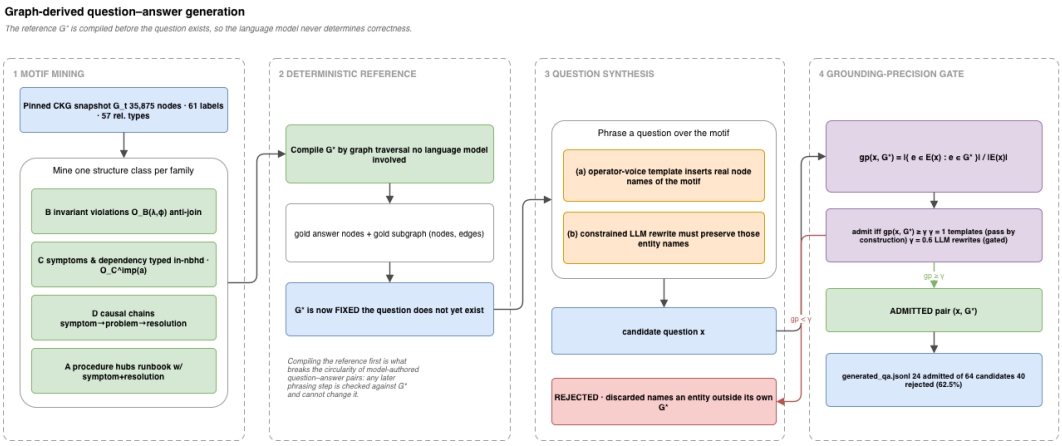


Figure 2: Overview of the graph-derived question generation pipeline. Structures mined from the knowledge graph are compiled into references and then converted into grounded question–answer pairs through templates or constrained rewriting.

Table 1: Summary of the instantiated benchmark on the production IT CKG. The snapshot and construction parameters are given in Appendix I.

Quantity	**Value**
References (seed + synthetic)	18 (14+4)
Answerable questions retained	14/14
Families A / B / C / D	5/6/2/1
Difficulty (hard / medium / easy)	13/1/4
Trust (SOLID / APPROXIMATE / THIN / WEAK)	6/7/2/3
B, C re-execution agreement	0.83
Node labels covered	39
Uncovered (role × family) cells	29

do not require human re-answering. The reported instantiation and all construction parameters are given in Appendix I.

Re-execution. For deterministic families B and C, re-running compilation on the same question and pinned snapshot should produce the same reference. We obtain an agreement of 0.83. The remaining differences are concentrated in items with sparse supporting evidence, which are marked THIN or WEAK.

Graph-mutation sensitivity. We perturb the graph and measure the resulting change in each reference using the symmetric difference between the pre- and post-mutation references. Node and edge deletion produce mean changes of 0.28 and 0.21, respectively. This confirms that references respond to changes in the graph structure on which they depend; the complete mutation protocol and results are provided in Appendix I.

Coverage. The instantiated benchmark covers 39 node labels and leaves 29 (role × family) combinations uncovered. These gaps are recorded explicitly and passed to the synthetic feedback loop described in §4.9. Reporting these uncovered regions makes the scope of the benchmark explicit rather than implying coverage that has not been tested.

7 BASELINE EXPERIMENTS

We evaluate four retrieval paradigms on the 33 benchmark queries: Zero-Shot LLM (parametric knowledge only), Vanilla RAG (BM25 or dense retrieval over the operational document corpus),

Table 2: All baseline configurations (n=33, n_{recall}=22).

Model	Paradigm	Hits↑	Recall↑	ECS↑	TGA-F1↑	GHP↓
Zero-Shot						
GPT-4o	zero-shot	0.591	0.131	0.262	0.051	0.919
GPT-5	zero-shot	0.682	0.133	0.290	0.033	0.953
Gemini 2.5 Pro	zero-shot	0.682	0.137	0.288	0.034	0.960
Claude Opus 4	zero-shot	0.636	0.136	0.234	0.040	0.933
Claude Sonnet 4.5	zero-shot	0.682	0.134	0.204	0.048	0.918
Vanilla RAG						
BM25	vanilla RAG	0.636	0.141	0.174	0.064	0.872
Dense	vanilla RAG	0.636	0.136	0.204	0.071	0.867
Text-to-Cypher						
GPT-4o	T2C	0.591	0.126	0.154	0.043	0.925
GPT-5	T2C	0.682	0.181	0.213	0.054	0.896
Gemini 2.5 Pro	T2C	0.545	0.141	0.168	0.046	0.890
Claude Opus 4	T2C	0.727	0.167	0.319	0.075	0.886
Claude Sonnet 4.5	T2C	**0.727**	**0.190**†	0.249	0.065	0.886
Think-on-Graph						
GPT-4o (ToG)	ToG	0.682	0.153†	**0.292**	**0.077**	0.890

Full 95% bootstrap CIs are in Appendix J. Bold marks the best value per column.
GHP↓: lower is better. † paired bootstrap significant vs. GPT-4o Zero-Shot ($p<0.001$).

Table 3: Text-to-Cypher execution behaviour ($n = 33$ queries, max 5 turns per query).

Model	Cypher Calls	Failed Calls	Error Rate	5-Turn Limit Hit
GPT-4o	80	0	0.0%	7/33 (21%)
GPT-5	168	36	21.4%	25/33 (76%)
Gemini 2.5 Pro	133	0	0.0%	20/33 (61%)
Claude Opus 4	383	3	0.8%	31/33 (94%)
Claude Sonnet 4.5	281	1	0.4%	30/33 (91%)

Text-to-Cypher (up to five interactive Cypher tool-call turns against the live Neo4j graph) and Think-on-Graph (ToG) (Wu et al., 2026) (beam-search traversal over the CKG before generation). All results are single generations at temperature 0. We report five metrics:

- **EM Hits**: binary indicator that at least one gold entity name appears verbatim in the answer.
- **EM Recall**: fraction of gold entity names recovered verbatim, computed over the 22 answerable queries; the 11 unanswerable items are excluded.
- **ECS** (Entity Coverage Score): fraction of gold answer-node names mentioned in the answer text (recall-oriented).
- **TGA-F1** (Text–Graph Alignment F1): mean of entity F1 (harmonic mean of mention precision and ECS) and relation F1 (harmonic mean of extracted-triple precision and recall against gold subgraph edges).
- **GHP** (Graph Hallucination Penalty): proportion of answer entity mentions and extracted relation triples not supported by the gold subgraph; lower is better.

ECS, TGA-F1 and GHP are text-surface measures that allow comparison across paradigms returning no structured output. We report 95% bootstrap confidence intervals (CIs) over the 22 per-query recall values in Appendix J; at this sample size a single query shifts the mean by ± 0.045, so independent-interval overlap cannot establish pairwise significance and we supplement with paired bootstrap tests where specific comparisons of interest arise.

Paradigm effects: Table 2 consolidates all configurations. No configuration reaches EM Recall of 0.20 or TGA-F1 above 0.077, which is expected given that correct answers require traversal across

multiple labelled edges rather than a simple keyword match (§5). The paradigms separate along how much graph-grounded content a system commits to. Zero-shot answers name plausible entities from generic parametric knowledge and achieve moderate Hits (0.591–0.682) but carry the highest GHP values (0.918–0.960), because parametric knowledge rarely aligns with this specific graph's content. Vanilla RAG improves TGA-F1 slightly (0.064–0.071 vs. 0.051 for zero-shot GPT-4o) but ECS drops from 0.262 to 0.174–0.204, because document retrieval anchors the model to a local chunk neighbourhood at the cost of the broader entity coverage it would otherwise draw from memory. Text-to-Cypher with GPT-4o returns the weakest recall (0.126): it writes clean Cypher but hits the five-turn limit on 7 of 33 queries before assembling the full gold set, and the retrieved rows provide little context for a discursive answer. ToG achieves the best ECS and TGA-F1 of any configuration (0.292 and 0.077), and a paired bootstrap confirms its recall advantage over zero-shot GPT-4o is real: mean recall difference $+0.022$, 95% CI $[0.009, 0.036]$, $p < 0.001$. Outside zero-shot, GHP clusters tightly between 0.867 and 0.925. The floor effect of prose extraction picking up asserted relations keeps even the best paradigm above 0.86, so the composition of mentions matters more than their count.

Model effects: Under zero-shot all five models sit in the narrow band EM Hits 0.591–0.682 with near-zero TGA-F1, confirming that difficulty comes from graph structure and not from any single provider. ECS varies more (0.204–0.290), pointing to differences in what each model's training data emphasises. With Text-to-Cypher, point estimates spread more widely. Claude Opus 4 and Claude Sonnet 4.5 reach the highest Hits of any configuration (0.727), with Recall of 0.167 and 0.190 respectively, the latter the best recall in the study, pointing to more effective multi-turn Cypher revision. GPT-5 reaches a Hits of 0.682 and a strong Recall of 0.181 but at cost: 21.4% syntax errors and the five-turn limit hit on 76% of queries (Table 3). Gemini 2.5 Pro produces clean Cypher but hits the five-turn limit on 61% of queries, and its Hits (0.545) trail GPT-4o's despite a comparable Recall (0.141 vs. 0.126), indicating failures concentrated on queries where it never recovers any gold entity rather than partial failures spread uniformly. A paired bootstrap between Claude Sonnet 4.5 Text-to-Cypher and GPT-4o Zero-Shot confirms real improvement from graph access: mean recall difference $+0.059$, 95% CI $[0.007, 0.154]$, $p < 0.001$. All other pairwise comparisons are suggestive rankings rather than established orderings at this sample size.

8 Conclusion

We have argued that ground truth for question answering over operational knowledge graphs should be compiled from the graph rather than elicited from a language model, and we have given a method that does so: exact set compilation for gap questions, bounded traversal for relational and impact questions, deterministic reference subgraphs for procedural and causal questions with generation confined to surface wording under an explicit grounding constraint, and unanswerability treated as a reference outcome in its own right. We have proposed a question generation method that mines structure, compiles its reference and only then phrases a question, so that generation cannot determine correctness. We have validated the resulting references intrinsically—by re-execution, by an active mutation probe, and by coverage—and proposed two measures directed at the assertion of relations absent from the graph. The benchmark is pinned to a snapshot and regenerates as the graph evolves, which we consider the appropriate posture for evaluating systems that operate over a living operational graph.

Reproducibility statement

The complete pipeline is described in §4 and §5, with formal stage-level specifications in Appendix B and all thresholds and instantiation parameters in Appendix I. The compilation oracles for question families B and C (§4.5) are deterministic functions of the question and a graph snapshot; any practitioner with access to a CKG conforming to the schema described in §1 can apply the method and reproduce references of the same kind. For families A and D the generative step is gated by the grounding constraint in §4.6 and the judge-panel verification procedure in §4, both of which are fully specified.

ETHICS STATEMENT

We affirm that we have read and comply with the ICLR Code of Ethics. The benchmark is constructed from operational IT infrastructure metadata derived from ITBench (Jha et al., 2025) incident scenarios and contains no personal data and no information about human subjects. The supplementary material consists of graph-derived reference records; no raw graph data is included.

AI USE STATEMENT

Generative AI tools were used to assist in drafting and editing portions of this manuscript. All AI-assisted text was reviewed and verified by the authors. We take responsibility for the final content of this work, including all claims and artifacts produced with the aid of generative AI.

REFERENCES

Somnath Banerjee, Amruit Sahoo, Sayan Layek, Avik Dutta, Rima Hazra, and Animesh Mukherjee. Context matters: Pushing the boundaries of open-ended answer generation with graph-structured knowledge context. In Franck Dernoncourt, Daniel Preoţiuc-Pietro, and Anastasia Shimorina (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 290–302, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.23. URL https://aclanthology.org/2024.emnlp-industry.23/.

Alberto Cattaneo, Carlo Luschi, and Daniel Justus. Ground-truth subgraphs for better training and evaluation of knowledge graph augmented LLMs, 2026. URL https://openreview.net/forum?id=Sp6znUhP1n.

Ritam Dutt, Sopan Khosla, Vinayshekhar Bannihatti Kumar, and Rashmi Gangadharaiah. GrailQA++: A challenging zero-shot benchmark for knowledge base question answering. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi (eds.), *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 897–909, Nusa Dua, Bali, November 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.ijcnlp-main.58. URL https://aclanthology.org/2023.ijcnlp-main.58/.

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, 2025. URL https://arxiv.org/abs/2404.16130.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. RAGAs: Automated evaluation of retrieval augmented generation. In Nikolaos Aletras and Orphee De Clercq (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 150–158, St. Julians, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-demo.16. URL https://aclanthology.org/2024.eacl-demo.16/.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. Beyond i.i.d.: Three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, WWW '21, pp. 3477–3488, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3449992. URL https://doi.org/10.1145/3442381.3449992.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=MPJ3oXtTZl.

Saurabh Jha, Rohan Arora, Yuji Watanabe, Takumi Yanagawa, Yinfang Chen, Jackson Clark, Bhavya Bhavya, Mudit Verma, Harshit Kumar, Hirokuni Kitahara, Noah Zheutlin, Saki Takano, Divya

Pathak, Felix George, Xinbo Wu, Bekir O. Turkkan, Gerard Vanloo, Michael Nidd, Ting Dai, Oishik Chatterjee, Pranjal Gupta, Suranjana Samanta, Pooja Aggarwal, Rong Lee, Pavankumar Murali, Jae wook Ahn, Debanjana Kar, Ameet Rahane, Carlos Fonseca, Amit Paradkar, Yu Deng, Pratibha Moogi, Prateeti Mohapatra, Naoki Abe, Chandrasekhar Narayanaswami, Tianyin Xu, Lav R. Varshney, Ruchi Mahindru, Anca Sailer, Laura Shwartz, Daby Sow, Nicholas C. M. Fuller, and Ruchir Puri. Itbench: Evaluating ai agents across diverse real-world it automation tasks, 2025. URL https://arxiv.org/abs/2502.05352.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. UniKGQA: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Z63RvyAZ2Vh.

Boci Peng, Yongchao Liu, Xiaohe Bo, Jiaxin Guo, Yun Zhu, Xuanbo Fan, Chuntao Hong, and Yan Zhang. M³GQA: A multi-entity multi-hop multi-setting graph question answering benchmark. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30594–30620, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1478. URL https://aclanthology.org/2025.acl-long.1478/.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. ARES: An automated evaluation framework for retrieval-augmented generation systems. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 338–354, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.20. URL https://aclanthology.org/2024.naacl-long.20/.

Nadine Steinmetz and Kai-Uwe Sattler. What is in the kgqa benchmark datasets? survey on challenges in datasets for question answering on knowledge graphs. *Journal on Data Semantics*, 10, 12 2021. doi: 10.1007/s13740-021-00128-9.

Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 641–651, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1059. URL https://aclanthology.org/N18-1059/.

Zimu Wang. Generating complex questions from knowledge graphs with query graphs. In *2022 IEEE 10th International Conference on Information, Communication and Networks (ICICN)*, pp. 606–613, 2022. doi: 10.1109/ICICN56848.2022.10006514.

Xiaojun Wu, Cehao Yang, Xueyuan Lin, Chengjin Xu, Xuhui Jiang, Yuanliang Sun, Hui Xiong, Jia Li, and Jian Guo. Think-on-graph 3.0: Efficient and adaptive llm reasoning on heterogeneous graphs via multi-agent dual-evolving context retrieval, 2026. URL https://arxiv.org/abs/2509.21710.

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. *arXiv preprint arXiv:2506.05690*, 2025.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 201–206, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2033. URL https://aclanthology.org/P16-2033/.

Liangliang Zhang, Zhuorui Jiang, Hongliang Chi, Haoyang Chen, Mohammed ElKoumy, Fali Wang, Qiong Wu, Zhengyi Zhou, Shirui Pan, Suhang Wang, and Yao Ma. Diagnosing and addressing

pitfalls in KG-RAG datasets: Toward more reliable benchmarking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=Vd5JXiX073.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. Variational reasoning for question answering with knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Dongzhuoran Zhou, Yuqicheng Zhu, Xiaxia Wang, Hongkuan Zhou, Yuan He, Jiaoyan Chen, Steffen Staab, and Evgeny Kharlamov. What breaks knowledge graph based RAG? benchmarking and empirical insights into reasoning under incomplete knowledge. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2522–2538, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.114. URL https://aclanthology.org/2026.eacl-long.114/.

Table 4: Notation used throughout the paper.

Symbol	Meaning
$G_t = (V, E, \ell, \rho)$	pinned snapshot (Def. **??**)
$\mathcal{L},\ \mathcal{R}$	node labels; relation types
$\mathcal{R}_{\mathrm{dep}}$	dependency relation subset
$h(G_t)$	content hash of the snapshot
$q,\ Q$	question; canonical question set
$P(q)$	paraphrase set of q
τ	family $\in \{A, B, C, D, \textsc{Abstain}\}$
$\Gamma(q)$	reference record for q
G^*	reference (gold) subgraph
A^*	reference answer string
$\pi,\ \mu,\ \sigma$	provenance; modality; stratification
Φ_i	stage operator i
$\mathcal{O}_\tau$	compilation oracle for family τ
$N^r_{\rightarrow},\ N^r_{\leftarrow}$	typed out-/in-neighbours
$\Lambda(r)$	label set of role r
$S(q), R(q), \Pi(q)$	seeds; target roles; predicate hits
$\mathrm{gp}(\cdot),\ \gamma$	grounding precision; threshold
$\theta_d,\ \theta_\ell$	dedup; semantic grounding floor
$k,\ h$	path bound; neighbourhood radius
δ	reference drift under mutation
$\mathrm{bc}(v)$	betweenness centrality of v
$\hat{G}, \hat{A}, \mathrm{Asrt}$	system subgraph; answer; assertions
$m,\ \kappa_0$	judge panel size; escalation threshold

A Notation

B Stage Operators in Full

This appendix gives the formal specification of the eight-stage construction introduced in Section 3. The main paper describes the overall pipeline; here we specify the inputs, operations, and outputs of each stage in more detail.

B.1 Operational CKG Snapshot

The operational CKG at version t is represented as

$$G_t = (V, E, \ell, \rho),$$

where V is the set of nodes,

$$E \subseteq V \times V \times \mathcal{R}$$

is the set of typed directed edges, and

$$\ell : V \to \mathcal{L}$$

assigns each node a label from the ontology $\mathcal{L}$. The mapping ρ stores attributes associated with nodes and edges, including free text contained in node properties.

The snapshot is identified by its content hash $h(G_t)$. This identifier binds each generated reference to the graph version used during its construction and allows affected references to be regenerated when the underlying graph changes.

B.2 Reference Record

For a question q and snapshot G_t, the construction returns

$$\Gamma(q; G_t) = \langle \tau, G^*, A^*, \pi, \mu, \sigma \rangle.$$

The components are defined as follows:

- $\tau \in \{A, B, C, D\} \cup \{\textsc{Abstain}\}$ specifies the question family.
- $G^* \subseteq G_t$ is the reference subgraph selected during compilation.
- A^* is an optional natural language answer constructed from the selected evidence.
- π records the provenance linking the reference to its supporting graph elements and textual evidence.
- $\mu \in \{\textsc{graph}, \textsc{text}, \textsc{hybrid}\}$ records the retrieval modality.
- σ contains the stratification metadata associated with the reference, including its trust label.

B.3 Stage Operators

The complete construction is expressed as

$$\Gamma = \Phi_7 \circ \Phi_6 \circ \Phi_5 \circ \Phi_4 \circ \Phi_3 \circ \Phi_2 \circ \Phi_1 \circ \Phi_0.$$

Φ_0: Question curation. Given raw questions Q_{raw} with sentence embeddings $\mathbf{e}_i$, we form clusters using cosine similarity,

$$q_i \sim q_j \iff \cos(\mathbf{e}_i, \mathbf{e}_j) \geq \theta_d.$$

One canonical representative is retained from each cluster, while the remaining questions are stored in the paraphrase set $P(q)$. Preserving these paraphrases makes it possible to evaluate sensitivity to surface form separately from answer accuracy.

A question is placed in quarantine if it is degenerate, such as being empty or containing no content token, or if it exhibits template leakage, where an automated question-generation template leaves a placeholder in the resulting question. The output of this stage is the curated question set Q, the corresponding paraphrase sets $\{P(q)\}_{q \in Q}$, and a quarantine log.

Φ_1: Typing. A router

$$\tau : Q \to \{A, B, C, D\}$$

assigns each question to a question family using an ordered rule set

$$\mathcal{P} = [(\tau_1, \varrho_1), \ldots, (\tau_n, \varrho_n)].$$

The first rule whose pattern ϱ_i matches the question determines its type. If no rule matches, an optional model-based fallback may be used, with A as the default.

The ordering of the rules is important because some questions contain patterns associated with more than one family. More specific structures are therefore considered first. In particular, impact patterns such as *affected*, *blast radius*, and *goes down* precede prescriptive patterns such as *should we* and *prevent*; these precede gap patterns such as *lack*, *without*, and *missing*; and procedural patterns such as *how do I* are considered subsequently.

Φ_2: Grounding. Grounding connects the question to the entities and evidence available in the CKG. Let $\Lambda(r) \subseteq \mathcal{L}$ denote the label set associated with role r. We use both lexical and embedding-based retrieval:

$$S(q) = \mathrm{ft}(q) \cup \mathrm{nn}_{\theta_\ell}(q),$$

where $\mathrm{ft}(q)$ denotes full-text matches over node names and stored properties, and $\mathrm{nn}_{\theta_\ell}(q)$ denotes embedding neighbours above the similarity floor θ_ℓ.

The lexical index includes the node properties $\rho(v)$, rather than only node names. This is important because operational information such as region identifiers, exposure indicators, and configuration values may occur only in property values.

The candidate roles are given by

$$R(q) = \{r : \Lambda(r) \cap \ell(V) \neq \varnothing\}.$$

The resulting grounding is passed to the answer and triage procedures defined in Sections 4.3–4.4. If grounding does not provide sufficient evidence, abstention is recorded on the reference rather than removing the question from the benchmark.

Φ_3: Compilation. The grounded entities, relations, and textual evidence are compiled into the candidate reference representation. For each question, this stage constructs the relevant subgraph G^* and, where applicable, the associated answer A^*.

The compilation procedure is family dependent because different question types require different reference structures. The resulting candidate is passed to provenance binding and subsequent verification.

Φ_4: Provenance binding and modality. For each element $x \in G^*$, including selected set members, path edges, or procedural steps, $\pi(x)$ records the source span from which x derives. Elements for which no supporting source can be identified,

$$\pi(x) = \bot,$$

are removed from G^*. This ensures that the resulting reference remains attributable to identifiable evidence.

The retrieval modality is recorded as

$$\mu = \begin{cases} \textsc{graph}, & \text{if support is topological only,} \\ \textsc{text}, & \text{if support comes from property text only,} \\ \textsc{hybrid}, & \text{if both contribute.} \end{cases}$$

Φ_5: Verification. Verification depends on the question family. For $\tau \in \{B, C\}$, the compiled reference is re-executed and required to satisfy

$$G^*_{\mathrm{re}} = G^*.$$

This round-trip condition checks whether the reference can be recovered from the underlying construction.

For $\tau \in \{A, D\}$, a panel of m judges provides judgements $j_1, \ldots, j_m$ concerning the faithfulness of A^* to G^*. Agreement is measured using κ. When

$$\kappa < \kappa_0,$$

the item is passed to a critique and refinement step using the disagreement information. The revised answer is required to satisfy

$$\mathrm{gp}(A^*, G^*) \geq \gamma.$$

Items that remain below the required agreement threshold after refinement are flagged for human audit rather than being admitted automatically.

Φ_6: Stratification. The stratification stage produces the metadata record σ. This includes the question family, graph hop depth, number of contributing modalities, answer cardinality $|A^*|$, answerability, difficulty bin, and trust label.

The detailed trust-label definitions and assignment criteria are provided in Appendix F.

Φ_7: Assembly. The final stage assembles the verified components into the reference record and associates the result with the snapshot identifier $h(G_t)$.

The same version binding also supports incremental regeneration. When a graph update changes only part of the snapshot, references whose subgraphs do not intersect the changed region can be retained. Only references satisfying

$$G^* \cap \Delta G \neq \varnothing$$

need to be recompiled, where ΔG denotes the changed region of the graph.

B.4 Intermediate Representations

Let X_i denote the representation after stage i. The construction can then be written as

$$X_{i+1} = \Phi_i(X_i),$$

with the initial representation derived from the curated question and the relevant snapshot, and the final representation given by

$$X_8 = \Gamma(q; G_t).$$

This formulation makes explicit that the stages operate sequentially while allowing each stage to retain the metadata required by later verification, provenance, and stratification procedures.

C Compilation Oracles in Full

$\mathcal{O}_B$: negation and structural invariants. As defined in §4.5. The invariants instantiated in this work, with the violation counts obtained on the instantiated graph, are: incident classes lacking an automated playbook (64); symptom classes not diagnosed by any runbook; problem clusters with no resolution; and incident classes whose runbooks document no resolution. Each is expressed as a φ of the same shape—an existential over a typed in-neighbourhood, optionally composed with a second typed hop—so that adding a new invariant requires specifying only (λ, φ).

$\mathcal{O}_C$: pathfinding and impact. For grounded endpoints s, t and bound k,

$$\mathcal{O}_C(s, t, k) = \{\, p \text{ simple} : s \rightsquigarrow t,\ |p| \leq k \,\},$$

and for impact questions with anchor a,

$$\mathcal{O}_C^{\mathrm{imp}}(a) = \{\, v \in V : v \rightsquigarrow a \text{ via } \mathcal{R}_{\mathrm{dep}},\ \text{within } k \,\}.$$

G^* is the union of the returned paths, or the affected set together with the edges realising the closure. The bound k is required for tractability and is reported with the benchmark, since an unbounded enumeration is not a well-defined reference on a cyclic graph.

$\mathcal{O}_A$**: procedural.** $G^* = \mathrm{Nbhd}_h(S(q))$, the bounded typed h-hop neighbourhood of the seeds, capped at a maximum cardinality to keep the reference reviewable. A^* is drafted over G^* subject to $\mathrm{gp}(A^*, G^*) \geq \gamma$.

$\mathcal{O}_D$**: causal.** G^* instantiates the typed template

$$\text{SYMPTOM} \rightarrow \text{PROBLEM} \rightarrow \text{RESOLUTION} \; (\rightarrow \text{AUTOMATION})$$

rooted at the grounded anchor and restricted to causal relations. The compound form is $\mathcal{O}_{D \circ B}$: $\mathcal{O}_D$ evaluated with anchors restricted to $\mathcal{O}_B(\lambda, \varphi)$, which is how questions of the form "what resolutions have worked for incidents that lack automation" are compiled without any generative step determining the anchor set.

D PROOFS

Proposition 1 (soundness and exactness of $\mathcal{O}_B$). Let $V_\lambda = \{v \in V : \ell(v) = \lambda\}$, which is finite since V is finite. The predicate φ is a bounded typed reachability condition and is therefore decidable at each $v \in V_\lambda$ by exploring at most $|E|$ edges to the stated depth. Define the indicator $\chi(v) = 1$ if and only if $\neg\varphi(v)$. Then by construction $\mathcal{O}_B(\lambda, \varphi) = \{v \in V_\lambda : \chi(v) = 1\}$.

Completeness. Suppose v violates φ and $\ell(v) = \lambda$. Then $\chi(v) = 1$ and $v \in V_\lambda$, hence $v \in \mathcal{O}_B$; no violator is omitted.

Soundness. Suppose $v \in \mathcal{O}_B$. Then $\chi(v) = 1$, i.e. $\neg\varphi(v)$, so v is a violator; no non-violator is included.

Order independence. $\chi(v)$ is a function of (v, G_t) alone and does not depend on any other member of V_λ nor on the state of the enumeration. The result is therefore invariant to the order in which V_λ is traversed. □

Property 1 (reproducibility). $\mathcal{O}_B$ and $\mathcal{O}_C$ are defined by set-builders over (q, G_t) containing no stochastic component, no sampling step, and no dependence on external state. For fixed (q, G_t) the returned set is therefore unique, and re-execution yields an identical G^*. □

Proposition 2 (joint necessity of the two measures). Consider four behaviours.

(i) Complete. $\hat{G} = G^*$ and $\hat{A} \subseteq V(\hat{G})$ give $F_1 = 1$ and $\mathrm{grd} = 1$, hence $\mathrm{GAS} = 1$; $\mathrm{Asrt} \subseteq E(G_t)$ gives $\mathrm{HBR} = 0$, hence $\mathrm{PF} = 1$.

(ii) Partial. $\hat{G} \subsetneq G^*$ gives $F_1 < 1$, hence $\mathrm{GAS} < 1$; asserting only existing relations gives $\mathrm{PF} = 1$.

(iii) Universally abstaining. $\hat{G} = \varnothing$ gives $F_1 = 0$, hence $\mathrm{GAS} = 0$; $\mathrm{Asrt} = \varnothing$ gives $\mathrm{HBR} = 0$ by the convention of Appendix K, hence $\mathrm{PF} = 1$.

(iv) Asserting invented relations. With $\hat{A} \not\subseteq V(\hat{G})$ we obtain $\mathrm{grd} = 0$ and hence $\mathrm{GAS} = 0$; invented edges give $\mathrm{HBR} > 0$ and hence $\mathrm{PF} < 1$.

Behaviours (iii) and (iv) share $\mathrm{GAS} = 0$ and are separated only by PF; behaviours (i) and (ii) share $\mathrm{PF} = 1$ and are separated only by GAS. Hence no single measure separates all four, and the pair is necessary. □

E MEASURES IN FULL

Conventions at the boundary. The measures are defined on possibly empty sets, and the conventions matter for the abstention track. We set $\mathrm{HBR} = 0$ when $\mathrm{Asrt} = \varnothing$ (asserting nothing cannot be unfaithful), $\mathrm{grd} = 0$ when $\hat{A} \neq \varnothing$ but $V(\hat{G}) = \varnothing$ (an answer with no retrieved support is ungrounded), and $F_1 = 1$ when both $\hat{G}$ and G^* are empty (correct abstention on an abstention reference). Without the first convention a correctly abstaining system would be indistinguishable from a maximally unfaithful one; without the third, correct refusal could not score.

Edge identity. Relations are compared as typed and direction-insensitive triples: an asserted (u, v, r) matches an existing edge if either $(u, v, r) \in E(G_t)$ or $(v, u, r) \in E(G_t)$. Self-loops are retained rather than collapsed, since an operational graph legitimately contains self-referential service relations, and collapsing them would spuriously report a hallucinated bridge.

Aggregation. Both measures are reported as an unweighted mean over questions, and additionally stratified by family and by trust label. The stratification by trust label is the important one: averaging a SOLID reference and a WEAK reference into a single figure attributes to the system a failure that is a property of the graph.

F TRUST LABEL ASSIGNMENT

A reference is labelled SOLID when $\tau \in \{B, C\}$, verification passed, and the answer is non-empty; APPROXIMATE when grounding relied on lexical or semantic matching rather than role grounding, or when compilation was a neighbourhood aggregation; THIN when the supporting evidence has cardinality at most two; and WEAK when the structure the question requires is absent from G_t—for example when no relation type in $\mathcal{R}$ expresses the queried property, or when the required property values were truncated during ingestion. Where more than one condition holds, the weaker label is assigned. The labels are computed from the compiled reference and the graph, not assigned by hand, so they are reproducible alongside the reference itself.

G SCHEMA COVERAGE MATRIX

For roles r ranging over a designated role set and families τ, the coverage matrix is

$$C[r, \tau] = \big|\{\, \Gamma(q) : \tau(q) = \tau,\ \exists v \in G^* \text{ with } \ell(v) \in \Lambda(r) \,\}\big|,$$

and cells with $C[r, \tau] = 0$ are emitted as gap targets for the synthetic feedback loop. The relation spine is covered analogously, by counting references whose G^* contains at least one edge of each type in a designated spine $\mathcal{R}_{\text{spine}} \subseteq \mathcal{R}$. In the instantiated benchmark 39 node labels and 11 spine relation types are exercised, and 29 cells remain empty; these are reported in the released artifact rather than omitted.

H QUESTION GENERATION: MOTIF FAMILIES

Mined structures. For family B, violation sets returned by $\mathcal{O}_B$. For family C, two structures: the symptoms co-occurring with a given problem, obtained as the typed in-neighbourhood of a problem node under the indication relation; and the dependency closure of a component, obtained as $\mathcal{O}_C^{\text{imp}}$. For family D, symptom–problem–resolution chains obtained from $\mathcal{O}_D$, optionally extended to the implementing automation. For family A, runbook hubs that simultaneously relate to a symptom, a resolution and an incident class, since only these support an end-to-end procedure.

Synthesis and gating. A question is produced either by a template that inserts the actual node names of the mined structure, or by a constrained model rewrite required to preserve those names. Admission requires $\text{gp}(x, G_M^*) \geq \gamma$. Template phrasings satisfy this at $\gamma = 1$ by construction, since the names they insert are drawn from G_M^* itself; model rewrites are gated at a lower threshold and rejected on any invented entity. Applying the gate at $\gamma = 1$ to templates and at a relaxed threshold to rewrites is deliberate: the former cannot fail except through an implementation error, so a failure there is diagnostic, whereas the latter is where invention actually occurs.

Yield. On the instantiated graph, 64 candidate rewrites produced 24 admitted pairs, a rejection rate of 62.5%, distributed over families B, C and D. We report the rejection rate rather than only the yield because it is the quantity that evidences the gate is operative.

I INSTANTIATION DETAILS

This appendix reports the configuration and detailed statistics for the instantiation described in §6.

I.1 CONFIGURATION

The reported instantiation uses duplicate-clustering threshold $\theta_d = 0.93$, semantic-grounding floor $\theta_\ell = 0.45$, path bound $k = 4$, and gap-reachability depth 3. Procedural references use neighbourhood radius $h = 1$ with a cardinality cap of 25, while causal references use radius 2 with a cap of 20.

The judge panel contains $m = 3$ judges with escalation threshold $\kappa_0 = 0.7$. The grounding-precision threshold is $\gamma = 1$ for template-based phrasings and $\gamma = 0.6$ for model rewrites. The mutation probe uses one run per mutation class with a deletion fraction of 0.05.

I.2 BENCHMARK STATISTICS

The instantiated benchmark contains 18 references: 14 seed references and 4 synthetic references. All $14/14$ answerable questions are retained. The family distribution over A/B/C/D is $5/6/2/1$, and the difficulty distribution over hard/medium/easy is $13/1/4$. The trust distribution over SOLID/APPROXIMATE/THIN/WEAK is $6/7/2/3$.

I.3 GRAPH-MUTATION PROBE

For each mutation class, reference change is measured using

$$\delta = \frac{\left|G^*_{\text{pre}} \triangle G^*_{\text{post}}\right|}{\left|G^*_{\text{pre}} \cup G^*_{\text{post}}\right|}.$$

The mean changes are 0.28 for node deletion, 0.21 for edge deletion, 0.20 for property flips, and 0.17 for incremental updates. These results provide the detailed mutation analysis underlying the summary reported in §6.

J BOOTSTRAP CONFIDENCE INTERVALS

All CIs are 95% bootstrap CIs computed over n_{recall}=22 per-query recall values (the 11 unanswerable queries contribute no recall observation and are excluded).

Table 5: Per-configuration EM Recall with 95% bootstrap CIs.

Model	Paradigm	Recall	95% CI
Zero-Shot			
GPT-4o	zero-shot	0.131	[0.036, 0.263]
GPT-5	zero-shot	0.133	[0.038, 0.264]
Gemini 2.5 Pro	zero-shot	0.137	[0.043, 0.268]
Claude Opus 4	zero-shot	0.136	[0.041, 0.266]
Claude Sonnet 4.5	zero-shot	0.134	[0.040, 0.265]
Vanilla RAG			
BM25	vanilla RAG	0.141	[0.043, 0.273]
Dense	vanilla RAG	0.136	[0.040, 0.268]
Text-to-Cypher			
GPT-4o	T2C	0.126	[0.030, 0.258]
GPT-5	T2C	0.181	[0.055, 0.325]
Gemini 2.5 Pro	T2C	0.141	[0.034, 0.275]
Claude Opus 4	T2C	0.167	[0.067, 0.296]
Claude Sonnet 4.5	T2C	0.190	[0.066, 0.338]
Think-on-Graph			
GPT-4o (ToG)	ToG	0.153	[0.056, 0.282]

All independent intervals overlap. Two paired bootstrap tests clear significance: ToG vs. GPT-4o Zero-Shot (Δ=+0.022, CI $[0.009, 0.036]$, p<0.001) and Claude Sonnet 4.5 T2C vs. GPT-4o

Zero-Shot (Δ=+0.059, CI $[0.007, 0.154]$, p<0.001). All remaining pairwise differences are within single-query noise.

K Formal Evaluation Measures

This appendix gives the formal definitions and validation of the evaluation measures introduced in §**??**.

K.1 Metric definitions

Let a system return a retrieved subgraph $\hat{G}$, a set of answer nodes $\hat{A}$, and a set Asrt of asserted relations. The *grounded answer score* is defined as

$$\mathrm{GAS} = \mathrm{HM}\left(F_1(\hat{G}, G^*), \mathrm{grd}(\hat{A}, \hat{G})\right),$$

where

$$\mathrm{grd}(\hat{A}, \hat{G}) = \frac{|\hat{A} \cap V(\hat{G})|}{|\hat{A}|}.$$

Thus, GAS combines agreement between the retrieved and reference subgraphs with the extent to which the system's answer is supported by its own retrieved evidence. The harmonic mean penalises cases where either component is low.

For empty predicted answer sets, grd is defined using the abstention status of the reference: a correctly empty answer receives full grounding credit, while an empty answer for a non-abstention reference receives zero.

We define the *hallucinated bridge rate* as

$$\mathrm{HBR} = \frac{|\{e \in \mathrm{Asrt} : e \notin E(G_t)\}|}{|\mathrm{Asrt}|},$$

and provenance faithfulness as

$$\mathrm{PF} = 1 - \mathrm{HBR}.$$

PF therefore measures the proportion of asserted relations that are supported by the graph. For abstention references, where the correct behaviour is $\mathrm{Asrt} = \varnothing$, the measure additionally captures whether the system introduces unsupported relations.

K.2 Complementarity of the measures

The two measures are designed to distinguish different failure modes. A system that retrieves only part of the required evidence but makes no unsupported assertions can retain high PF while receiving a lower GAS. Conversely, a system can retrieve the correct neighbourhood while introducing relations that are absent from the graph, reducing PF.

Proposition 2 (Joint necessity)**.** *No single measure separates the four canonical behaviours considered here. A universally abstaining system and a system asserting invented relations can obtain the same GAS but differ in PF, while a partially retrieving system and a complete system can obtain the same PF but differ in GAS. Both measures are therefore required to distinguish these behaviours.*

The proof is given in Appendix D.

Table 6: Synthetic probes illustrating the complementary behaviour of the proposed measures. These probes isolate failure modes rather than represent evaluated systems.

Reference behaviour	**GAS**	**PF**	**HBR**
Complete	1.00	1.00	0.00
Partial (60%)	0.70	1.00	0.00
Asserting invented	0.00	0.42	0.58
Universally abstaining	0.00	1.00	0.00