Cograph: A Knowledge-Graph Approach to Natural-Language Query over Structured Enterprise Data

Moeen M.
Cograph
moeen@cograph.cloud

April 2026

Abstract

Cograph auto-ingests CSVs into a shared RDF ontology and answers natural-language questions via LLM-generated SPARQL. On a rebuilt held-out benchmark of 26 knowledge graphs and 302 questions spanning healthcare, finance, legal, and scientific/public-sector domains, Cograph achieves 91.4% majority-vote accuracy (276/302, 95% Wilson CI [87.7, 94.1]) with Gemini 3 Flash Preview and 92.1% (278/302, [88.4, 94.6]) with Claude Opus 4.6. We rebuilt the benchmark from v1 (22 KGs, 277 questions, 91.7%) to v2.0 (26 KGs, 302 questions, 91.4%) with a stricter schema-adjacency gate (hard 3-type / 2-relationship overlap rule plus a Jaccard < 0.30 cutoff plus a forward-looking denylist), the removal of seven schema-adjacent clinical-trials KGs, and execution-verified gold questions generated via a tier-balanced Q2Forge v2 pipeline. The headline accuracy stayed essentially flat (91.7% $\rightarrow$ 91.4%, within seed noise), which we read as evidence that the v1 number was not inflated by schema memorization: on a benchmark specifically designed to expose schema leakage, accuracy held. On the identical question set, LLM, and judge, Cograph exceeds three head-to-head baselines: pandas-agent 77.2%, a DAIL-SQL-style text-to-SQL reimplementation 68.5%, and naive RAG 27.8%. The largest structural gap appears on join-heavy questions (tier T3), where Cograph scores 86.3% versus the text-to-SQL reimplementation at 39.7%. All numbers are post-retry following a transient DNS incident; all artifacts (manifest, training pin, adjacency audit, pre-eval checklist, failure analysis) are public.

1 Introduction

Enterprise analyst questions span multiple tables ("which sponsors run the most trials in this condition?"), filter on enumerated values whose exact spelling lives in a data dictionary, and aggregate over result sets. Two classes of existing system address parts of this. Text-to-SQL generates SQL from a schema description, but even strong few-shot variants in the DAIL-SQL style [Gao et al., 2024] must reconstruct entity structure from a flat schema on every query: a "sponsor" is a column value in CSV but a real-world entity in the question. Retrieval-augmented generation over row embeddings [Lewis et al., 2020] works for lookup but is architecturally unable to answer COUNT or AVG questions because it sees only a fraction of matching records. Pandas-code agents [LangChain, 2024] can count and aggregate but rediscover structure at each question.

Cograph takes a third path: auto-ingest CSVs into a typed RDF knowledge graph and then generate SPARQL against the typed graph. This pushes schema discovery to ingest time (one LLM call per CSV, then deterministic row mapping) so the query-time LLM only has to express the question, not rediscover the schema. For enterprise deployments, this matters because new data sources arrive weekly and analyst questions typically cross sources.

Contributions. (1) We design and evaluate Cograph, a production pipeline for CSV-to-SPARQL question answering with a multi-tier anti-cheat example bank. (2) We release a rebuilt held-out benchmark (Holdout v2.0) with 26 knowledge graphs, 302 execution-verified gold questions, and a hard schema-adjacency gate against the training set. (3) We run head-to-head baselines (pandas-agent, DAIL-SQL-style reimplementation, naive RAG, RAG+rerank) on the identical question set, LLM, seeds, and judge. (4) We run a cross-LLM ablation (Opus 4.6, Gemini 3 Flash Preview, Gemini 2.5 Flash Lite) on the identical pipeline, showing a 8.7 pp spread across three capability classes.

2 Approach

Cograph is a CSV-to-SPARQL pipeline structured around a shared multi-tenant RDF ontology stored in Amazon Neptune [Amazon Web Services, 2024]. Three design choices are load-bearing.

Entity-first ontology inference, one LLM call per schema. Schema inference takes column headers, sample rows, and the existing ontology and returns an entity type plus per-column role assignments (type identifier, attribute, relationship) in a single LLM call. Row mapping is then fully deterministic: zero LLM calls per row. Geographic and people columns are forced to relationships in post-processing so that City, State, and Person become first-class entities with their own URIs.

Multi-table ingest. For entity-rich sources spanning multiple CSVs (e.g. CMS Nursing Home Compare, PatentsView, SEC EDGAR 10-K, NIH RePORTER), Cograph uses a client-side SPARQL UPDATE path. Per-source multi-table specs declare cross-file joins explicitly. 22 of the 26 v2.0 holdout KGs use the multi-table path.

SPARQL generation via a few-shot example bank with multi-tier anti-cheat. At query time, the system embeds the question and retrieves up to three working SPARQL examples from past eval runs. Critically, retrieval returns only the *natural-language question* of each matched bank entry plus the relevant ontology template; it does *not* inject the retrieved example's raw SPARQL body or its URIs into the generation prompt. This was a deliberate design decision after an earlier cross-encoder SPARQL-body rerank variant caused URI-copy contamination on mid-tier models. Three gates keep the bank honest:

- `exclude_questions`: the union of v2.0 gold questions is passed to the `/ask` endpoint as an exact-string exclude set (301 entries at runtime after exact-string dedup);
- cross-KG cosine ceiling (0.90) and stricter same-KG ceiling (0.75);
- holdout-v2 KGs are excluded at bank ingestion time.

Cross-domain examples are retained because they teach SPARQL patterns (COUNT, aggregation, relationship traversal) rather than leak answers, and because retrieval injects only the natural-language question, cross-KG contamination of the generation prompt is bounded structurally rather than by a cosine threshold alone.

Execution-verified gold. Q2Forge v2 generates question/SPARQL pairs at four difficulty tiers and stores the gold answer by re-executing the gold SPARQL against Neptune. A deterministic tier classifier does not trust the LLM's self-labelled tier; it re-labels each candidate from the generated SPARQL's syntactic shape (entity-type count, filter presence, join count, aggregation) and rejects tier mismatches. A data-realism probe samples actual attribute values from Neptune per entity type before generation, preventing the LLM from hallucinating filter values.

3 Benchmark Methodology

3.1 Why v2.0 exists

The April 2026 v1 holdout (22 KGs, 277 questions) had three defects that compromised its generalization signal: (i) seven of 22 holdout KGs were clinical-trials variants sharing the schema of clinical-trials KGs used during eval-loop iteration on the example bank; (ii) 56% of questions were healthcare; (iii) the head-to-head baseline comparison was run on a subset, not the full holdout. Holdout v2.0 was designed from scratch to fix all three. Under the harder adjacency gate, plus the removal of the seven schema-adjacent clinical-trials KGs and the addition of legal and scientific/public-sector expansion, the headline moved from 91.7% (22 KGs, 277 questions) to 91.4% (26 KGs, 302 questions). This ∼0.3 pp delta sits well inside the seed-level noise band; we treat the invariance under hardening, not the number itself, as the credibility result.

3.2 Schema-adjacency gate

The v2.0 training set is pinned to 36 KGs (673 examples, underlying file SHA-256 `8eff90fd...`). Every v2.0 holdout KG was gated by: (i) a hard rule ($\geq$3 type overlaps AND $\geq$2 relationship overlaps with matching cardinality against any training KG $\rightarrow$ reject), (ii) Jaccard similarity < 0.30, and (iii) a name-pattern denylist blocking forward-looking insurance/clinical-trials variants. All 26 holdout KGs pass adjacency; the maximum Jaccard is 0.115 (`hrsa-hpsa` vs `cms-hospital-quality`). The per-KG distance table is in the appendix.

3.3 Question generation

Q2Forge v2 generates question/SPARQL pairs per KG at T1 (count/lookup), T2 (filter), T3 (join/relationship traversal), T4 (multi-hop aggregation). A deterministic tier classifier counts entity-type bindings, filters, join predicates, and aggregation operators to assign tier. KGs with fewer than three entity types are capped at T1+T2 (flat-schema cap). The frozen 302-question set spans T1:104, T2:78, T3:68, T4:52.

3.4 Evaluation protocol

We use a 3-seed majority vote (each question asked three times under distinct seeds; the majority-agreed answer is scored). All head-to-head baselines use `google/gemini-3-flash-preview` at $T = 0$, the same retrieval (where applicable), and the same deterministic judge. The judge parses model items, normalizes them (URI path-tail extraction), and verifies $\geq$70% set membership with a minimum of 5 items to prevent single-lucky-match false positives; it falls back to a fast exact-match judge for gold sets with fewer than 20 items. All confidence intervals are Wilson 95% [Newcombe, 1998]. Per-run variance under LLM non-determinism [Google DeepMind, 2025, Anthropic, 2025] is ±5–10 pp on a per-KG basis; 3-seed majority voting reduces aggregate variance but does not eliminate it.

3.5 Freeze and versioning

The v2.0 manifest is frozen at commit `c2933b0` with SHA-256 `f8f4a7fc...`, reproducibly computed by `scripts/compute_manifest_sha.py` (canonical JSON, `sort_keys=True`, `indent=2`, UTF-8, `manifest_sha256` field set to null). The manifest lists 26 ingested holdout KGs; three candidate KGs (FINRA TRACE corporate bonds, NSF awards, a non-prefixed DOE energy research grants

duplicate) are tracked under a separate `deferred_kgs` array with reason codes (bulk-access licensing, swap to NIH RePORTER, duplicate of the prefixed entry).

4 Results

4.1 Primary result

Table 1 shows the headline result with Gemini 3 Flash Preview under 3-seed majority voting. Overall accuracy is 91.4% (276/302, Wilson 95% CI [87.7, 94.1]); seed-level accuracy over 906 calls is 91.4% [89.4, 93.0].

Table 1: Primary Cograph result (Gemini 3 Flash Preview, 3-seed majority vote). Per-tier and per-domain rows are seed-level (n = 3 × per-question count).

Scope	Correct / n	Accuracy	95% Wilson CI
Overall (majority vote)	276/302	91.4%	[87.7, 94.1]
Overall (seed level)	828/906	91.4%	[89.4, 93.0]
T1 (count/lookup)	309/312	99.0%	[97.2, 99.7]
T2 (filter)	216/234	92.3%	[88.2, 95.1]
T3 (join/traverse)	176/204	86.3%	[80.9, 90.3]
T4 (multi-hop agg.)	127/156	81.4%	[74.6, 86.7]
Healthcare	226/246	91.9%	[87.8, 94.7]
Finance	168/177	94.9%	[90.6, 97.3]
Legal	223/252	88.5%	[84.0, 91.9]
Sci./Public	211/231	91.3%	[87.0, 94.3]

4.2 Head-to-head baselines

Table 2 compares Cograph against three head-to-head baselines on the identical 302 questions, identical LLM (`google/gemini-3-flash-preview`), identical seeds, and identical judge. Cograph leads pandas-agent by 14.2 pp, our text-to-SQL reimplementation by 22.9 pp, and naive RAG by 63.6 pp. The largest structural gap is on T3, where Cograph reaches 86.3% and the text-to-SQL reimplementation drops to 39.7% — a 46.6 pp gap under controlled in-harness conditions. This is the most dramatic structural finding, though the comparison is against a reimplementation, not the published DAIL-SQL code (see methodology note and limitations).

Table 2: Head-to-head baselines on the identical 302-question set. All rows use the same LLM, same seeds, same judge.

System	Overall	95% CI	T1	T2	T3	T4
Cograph	**91.4**	[87.7, 94.1]	99.0	92.3	86.3	81.4
Pandas-agent	77.2	[72.1, 81.5]	78.8	78.2	74.0	69.9
Text-to-SQL (DAIL-SQL-style reimpl.)	68.5	[63.1, 73.5]	82.7	84.6	39.7	53.8
Naive RAG (`text-embedding-3-small`, top-10)	27.8	[23.1, 33.1]	41.0	30.8	19.1	7.7

Methodology note — text-to-SQL baseline. Our text-to-SQL row is a DAIL-SQL-style reimplementation inside the Cograph eval harness, not the published DAIL-SQL code [Gao et al., 2024]. The reimplementation performs SQLite conversion of each KG, schema description with

sample rows, few-shot retrieval of similar natural-language questions (same anti-cheat gates as Cograph), 3-attempt self-correction on SQL parse errors, and final answer via Gemini 3 Flash Preview. Published DAIL-SQL uses additional techniques including explicit schema linking, SQL skeleton prediction, and a more sophisticated self-correction chain, and reports substantially higher execution accuracy on Spider 1.0 dev [Yu et al., 2018] (∼86%) and on BIRD [Li et al., 2023] than our in-harness reimplementation reaches here. DAIL-SQL predates Spider 2.0 [Yu et al., 2024], a 2024 follow-up on which no published system is close to Spider 1.0 numbers. We **do not** claim that Cograph beats published state-of-the-art text-to-SQL. We report our reimplementation as a directional in-harness baseline that controls LLM, seeds, judge, and anti-cheat. A direct head-to-head against the published DAIL-SQL code is planned for v2.1 (target Q2 2026).

Methodology note — RAG baseline. We also evaluated RAG with a cross-encoder reranker [Chen et al., 2024] (top-50 retrieve + top-10 rerank). The reranker moved accuracy by −0.3 pp (majority-vote 27.5% vs. naive's 27.8%, within noise) because the bottleneck for retrieval-based systems on this benchmark is not row-ranking quality but structured execution: no fixed k-row window can answer COUNT or AVG queries regardless of which rows are picked. We therefore report naive RAG as the representative retrieval baseline.

4.3 Cross-LLM ablation

Table 3 compares three non-reasoning models of different capability classes on the identical Cograph pipeline [Google DeepMind, 2025, Anthropic, 2025].

Table 3: Cross-LLM ablation (identical pipeline, 302 questions, 3-seed majority vote).

Model	Class	Overall	95% CI	T1	T2	T3	T4
Claude Opus 4.6	frontier	**92.1**	[88.4, 94.6]	98.7	92.7	87.7	84.0
Gemini 3 Flash Preview	mid	**91.4**	[87.7, 94.1]	99.0	92.3	86.3	81.4
Gemini 2.5 Flash Lite	weak	83.4	[78.8, 87.2]	98.7	78.6	77.0	67.9

The spread across three capability tiers is 8.7 pp (majority vote). Notably, this 8.7 pp cross-LLM spread is *smaller* than the 14.2 pp gap between Cograph and the strongest baseline (pandas-agent), which is consistent with the interpretation that accuracy is carried by the structured-query pipeline beyond any single LLM's reasoning. Opus 4.6 outperforms Gemini 3 Flash Preview by 0.7 pp; the two are within each other's 95% CIs and the 0.7 pp point estimate sits well inside the ±5–10 pp single-run noise band. **Opus 4.6 and Gemini 3 Flash Preview should be read as statistically indistinguishable at this sample size; the ordering is not a meaningful frontier advantage.** The spread that large would be unusual if accuracy were entirely model-bound; we read this as consistent with the structured-query pipeline contributing meaningfully beyond the underlying LLM's reasoning, rather than as a causal attribution of accuracy to the pipeline versus the model.

4.4 Per-tier visual comparison

Figure 1 gives a one-glance view of the five systems across the four difficulty tiers.

4.5 DNS incident disclosure

During the post-fix cross-LLM runs, transient DNS resolution failures affected approximately 5% of seed-level calls. Failures concentrated on a single KG (`holdout-v2-cdc-wonder-mortality`),

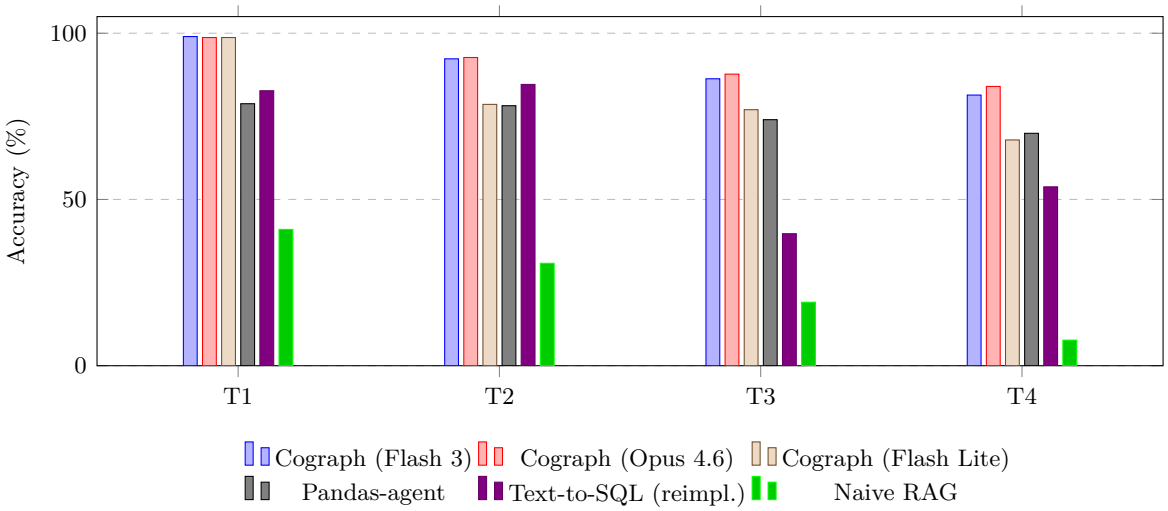

Figure 1: Per-tier accuracy across the five systems (plus two cross-LLM points). Cograph holds T1–T4 evenly across LLM classes; the text-to-SQL reimplementation has the sharpest tier cliff on T3.

where the pre-retry primary run shows 0/13. A surgical retry recovered all affected questions. **All numbers above are post-retry.** The pre-retry snapshot is preserved in the artifact repository for reviewer comparison.

4.6 Latency and cost

A single 3-seed sweep of the 302-question holdout on Gemini 3 Flash Preview takes ∼3,660 s (906 `/ask` calls; ∼4 s per query wall clock) and ∼$1.47 in LLM cost under OpenRouter pricing. The Opus 4.6 cross-LLM ablation costs roughly $20–30 per 3-seed sweep at current Anthropic pricing; Flash Lite is well below the primary.

5 Limitations

- **LLM non-determinism.** Single-run variance on Gemini Flash at $T = 0$ is ±5–10 pp per KG. 3-seed majority vote reduces but does not eliminate this; post-vote run-to-run variance has not been measured.
- **LLM-generated, execution-verified ground truth.** Gold SPARQL is LLM-generated and filtered by successful Neptune execution; the gold answers are `SELECT` result sets over the ingested KG, *not* LLM-judged strings. Execution verification establishes syntactic and runtime validity but not uniqueness or optimality. We cannot fully rule out that the benchmark systematically favors the SPARQL idioms the generator model prefers; the v2.1 mitigation is a human-reviewed 20% slice.
- **Text-to-SQL baseline is our reimplementation.** The 68.5% text-to-SQL number is our DAIL-SQL-style reimplementation, not the published DAIL-SQL code [Gao et al., 2024]. Published DAIL-SQL reports substantially higher execution accuracy on Spider 1.0 dev (∼86%) than our reimplementation here. We do not claim to beat published state-of-the-art text-to-SQL; the T3 cliff is a claim about our in-harness reimplementation under controlled conditions.

A v2.1 head-to-head against the published DAIL-SQL code is planned for Q2 2026.

- **No external benchmark calibration.** Cograph has not yet been evaluated against standardized text-to-SPARQL/SQL benchmarks (Spider 2.0 [Yu et al., 2024], BIRD [Li et al., 2023], KGQA).
- **Structural outliers and partial sources.** `fema-disaster-declarations` is a flat structural outlier; `nih-reporter-non-clinical` has 5 relationships (below the $\geq$8 floor); PatentsView uses a Q1 2020 slice; USPTO uses 434 trademarks due to TSDR rate limits.
- **No calibration or abstention metric.** The pipeline does not currently abstain on unanswerable questions; calibration is a v3 target, abstention a v2.1 target.
- **Entity resolution is unsolved.** Cograph does not currently merge "TX" with "Texas" or "NIH" with "National Institutes of Health."

6 Conclusion

Cograph reaches 91.4% majority-vote accuracy (276/302, 95% CI [87.7, 94.1]) with Gemini 3 Flash Preview on a rebuilt held-out benchmark of 26 knowledge graphs and 302 questions, and 92.1% with Claude Opus 4.6 on the same pipeline. The 0.3 pp v1-to-v2.0 drop under a hardened adjacency gate is well inside the seed-level noise band; we treat this invariance under hardening as the credibility result. On identical questions, LLM, and judge, Cograph exceeds pandas-agent by 14.2 pp, our DAIL-SQL-style text-to-SQL reimplementation by 22.9 pp, and naive RAG by 63.6 pp; the largest structural gap is on join questions (46.6 pp on T3). The 8.7 pp cross-LLM spread is smaller than the 14.2 pp baseline gap, consistent with the structured-query pipeline carrying accuracy beyond any single model. Remaining work: a human-reviewed 20% gold slice, a calibration/abstention metric, and a direct head-to-head against the published DAIL-SQL code. All v2.0 artifacts — manifest, training-set pin, adjacency audit, pre-eval checklist, failure analysis, primary and cross-LLM raw results — are public in the `eval_holdout_v2/` directory of the Cograph repository.

Contact. Feedback welcome: `moeen@cograph.cloud`.

References

Amazon Web Services. Amazon Neptune. https://aws.amazon.com/neptune/, 2024.

Anthropic. Claude opus 4.6 model card. https://www.anthropic.com/claude, 2025.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. BGE reranker v2 m3. https://huggingface.co/BAAI/bge-reranker-v2-m3, 2024.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. Text-to-SQL empowered by large language models: A benchmark evaluation. *Proceedings of the VLDB Endowment*, 17(5), 2024. Preprint arXiv:2308.15363, 2023.

Google DeepMind. Gemini 2.5 and 3 model family. Model card, https://deepmind.google/models/gemini/, 2025.

LangChain. LangChain pandas dataframe agent. https://python.langchain.com/docs/integrations/toolkits/pandas, 2024.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. Can LLM already serve as a database interface? a BIg bench for large-scale database grounded text-to-SQLs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Robert G. Newcombe. Two-sided confidence intervals for the single proportion: comparison of seven methods. *Statistics in Medicine*, 17(8):857–872, 1998.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of EMNLP*, 2018.

Tao Yu et al. Spider 2.0: Evaluating language models on real-world enterprise text-to-SQL workflows. In *Preprint*, 2024.

A Schema-Adjacency Detail

The v2.0 training set is pinned to 36 KGs (673 examples). For each of the 26 holdout KGs we compute type-overlap count, relationship-overlap count, and Jaccard similarity against every training KG, keeping the maximum. All 26 holdout KGs pass the hard rule (<3 type overlaps OR <2 relationship overlaps) and the Jaccard gate (<0.30). The strictest pair is `hrsa-hpsa` vs. `cms-hospital-quality` (6 type overlaps, 0 relationship overlaps, Jaccard 0.115) — it passes because the relationship count is 0. Representative values: `cms-nursing-home-compare` has 2 type overlaps, 0 relationship overlaps, Jaccard 0.051 against its closest training KG. The full 26-row table is in `eval_holdout_v2/closest_training_kg_table.md`. The check is reproducible via `scripts/check_schema_adjacency.py`.

B Q2Forge v2 Pipeline Detail

Q2Forge v2 adds five capabilities over v1: (1) a deterministic tier classifier that re-labels each candidate from its SPARQL shape instead of trusting the LLM's self-label (it counts entity-type bindings, filters, join predicates, and aggregation operators); (2) a data-realism probe that samples actual attribute values from Neptune per entity type before generation, preventing hallucinated filter values; (3) a T3/T4 prompt fix requiring the LLM to type every subject variable with `rdf:type`; (4) a strict T4 classifier rejecting candidates that do not satisfy all three T4 shape criteria (multi-hop join + aggregation + entity-type count); (5) a flat-schema cap for KGs with fewer than three entity types (T1+T2 only).

C Failure Analysis (Methodology Note)

Systematic failure analysis was performed on the post-patch primary run. Eight candidate patterns were identified; six were rejected as overfitting (single-KG or single-domain scope), and two were

kept and fixed as general patterns: (A) *FROM-clause regex repair* — a missing-graph URI normalization path in the SPARQL post-processor; (B) *judge prefix-strip* — a URI local-name matcher that failed on namespace-prefixed gold items. Both fixes apply to multiple KGs and multiple tiers and are not holdout-specific. The methodology rule: a fix is only landed if it generalizes across at least three independent KGs and at least two tiers.

D Anti-Cheat Gates: Walkthrough

The example bank operates under four gates in parallel. (1) `exclude_questions`: at query time, the union of v2.0 gold questions is passed as an exact-string exclude set (301 entries after exact-string dedup). The bank retriever will not return any example whose question string matches. (2) *Cross-KG cosine ceiling* of 0.90 on question embeddings blocks any bank example whose question is too close to the current question, even if it lives in a different KG. (3) *Same-KG cosine ceiling* of 0.75, stricter than the cross-KG rule, blocks near-duplicates within the same KG even if similarity would allow them cross-KG. (4) *Holdout-v2 KG block* at bank ingestion time: `HOLDOUT_V2_KGS` is excluded when the bank is built, so no holdout SPARQL is ever ingested. Retrieval returns only the natural-language question of each matched bank entry plus the relevant ontology template; it never injects the retrieved example's raw SPARQL body or URIs into the generation prompt.