GeoKG-Bench: Evaluating LLMs for Geospatial Domain-Specific Knowledge Graph Query Generation

Anonymous ACL submission

Abstract

We study the problem of translating natural language (NL) questions into Nebula Graph Query Language (nGQL) using large language models (LLMs). To systematically evaluate robustness, we introduce a benchmark of 105 NL–nGQL pairs covering single-hop and multi-hop queries, as well as easy and hard domain-specific anomaly questions that require implicit domain knowledge and semantic reasoning. In particular, maritime queries such as identifying loitering or rendezvous events cannot be answered through literal keyword filtering and instead require reasoning over domain-defined conditions and graph structure. We evaluate LLMs under on two categories of NL questions: (i) knowledge-graph schema-dependent direct questions, and (ii) domain-concept anomaly event questions. Performance is measured using token-level schema linking accuracy, constraint/filter match accuracy, projection precision, and execution accuracy. Our results show that while existing LLMs generate accurate nGQL queries for simple questions, their performance degrades significantly on domain-specific questions, highlighting fundamental limitations in domain-aware reasoning for reliable property-graph query generation.

1 Introduction

Knowledge graphs (KGs) are increasingly used in real-world systems to support complex analytical and decision-making tasks in domains such as maritime monitoring, cybersecurity, and supply-chain intelligence. These domains rely on specialized schemas, spatiotemporal reasoning, and implicit operational semantics, and KG-based systems are typically embedded in larger investigative pipelines where correctness and interpretability are critical. However, interacting with KGs remains challenging, as writing correct graph queries requires detailed knowledge of the schema, domain semantics, and query language constructs. Large Language Models (LLMs) offer a promising natural-language interface for KG-based systems and are increasingly explored as controllers in multi-stage, agentic pipelines. In such settings, LLMs must translate natural-language questions into executable graph queries, making query generation a foundational step whose errors directly affect downstream analysis. This task is particularly challenging for property-graph query languages such as NebulaGraph's nGQL, which are highly schema-dependent and rely on multi-hop traversal, temporal constraints, and domain-specific assumptions. Moreover, operational and regulatory constraints in sensitive domains often require on-premise deployment, limiting the applicability of proprietary cloud-based models.

This motivates a systematic evaluation of deployable LLMs for property-graph query generation. While prior work has focused primarily on relational databases, existing benchmarks do not capture domain concepts that motivate graph-based modeling in practice, where behaviors such as loitering or rendezvous must be inferred from spatiotemporal patterns across multiple nodes and relations. **Our contributions** in this paper are **(1) We benchmark** LLMs on NL-to-nGQL generation for both schema-dependent simple queries and domain-specific complex queries. **(2) We curate** a dataset of natural language questions paired with gold-standard nGQL queries that encode domain semantics through traversal patterns, temporal filters, and spatial constraints. **(3) Evaluating open-source models** across syntactic validity, schema adherence, execution accuracy, and answer fidelity, we show that although many generated queries are syntactically valid, they often fail to correctly operationalize domain concepts.

While this study evaluates query generation in isolation, it is explicitly motivated by the broader goal of building reliable, in-premise multi-agent deep-search systems over knowledge graphs.

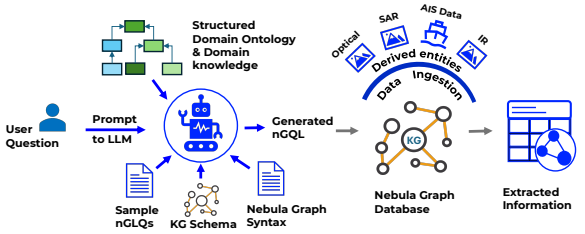

Figure 1: End-to-end pipeline for NL to nGQL query generation over a maritime KG. The NebulaGraph KG ingests raw multimodal data sources (AIS and satellite imagery) and derived entities, such as AIS tracks, inference runs, detections etc. A user's NL query is provided to an LLM conditioned on the KG schema, nGQL syntax, and simple representative example queries, which generates an executable nGQL query that is executed on the graph to retrieve structured results.

2 Benchmark Setup

2.1 Related Work

NL to structured query generation has been studied in the context of relational databases, with the SPIDER benchmark (Yu et al., 2018) establishing text-to-SQL as a core semantic parsing task under unseen schemas. Recent surveys highlight rapid advances driven by LLMs, including prompt engineering, fine-tuning, execution-based evaluation, and error analysis (Liu et al., 2025). Subsequent work identifies schema grounding as a central challenge and proposes context-aware schema retrieval and linking mechanisms to improve query generation accuracy (Nahid et al., 2025). Beyond one-shot generation, agentic and multi-step approaches combine schema-aware prompting, task decomposition, and iterative debugging, demonstrating improved execution accuracy, particularly for smaller open-source models (Phan et al., 2025; Ahmed et al., 2025). Compared to text-to-SQL, NL to KG query generation remains less standardized and more sensitive to traversal logic and schema constraints. Recent studies explore LLM-based generation of graph queries such as Cypher and SPARQL using few-shot prompting and retrieval-augmented schema support to reduce hallucinations and improve validity (Munir and Aldini, 2024; Emonet et al., 2024), with emerging work extending to multi-turn, context-dependent graph querying (Liang et al., 2025).

Gap: Domain concepts and operational semantics: Existing approaches improve query generation through schema exposure, examples, retrieval, or agentic refinement, and evaluate performance using exact match or execution accuracy. However, they typically treat user queries as self-contained and do not explicitly incorporate domain concept definitions (e.g., the conditions defining loitering or dark vessel behavior) as first-class inputs. Consequently, current benchmarks offer limited insight into how operational domain knowledge affects an LLM's ability to generate correct domain-grounded property-graph queries. Our benchmark addresses this gap by evaluating NL-to-nGQL generation on both schema dependant direct questions and domain-specific compelx questions.

2.2 Knowledge Graph Schema and Data Generation

We construct a multi-modal maritime KG using NebulaGraph, grounded in a structured domain ontology that captures vessels, AIS tracks, satellite media, inference runs, detections, ports, and operational regions. The schema encodes explicit relationships between AIS-derived motion patterns and satellite-derived visual observations, enabling joint reasoning over movement, time, and physical presence. Detailed KG schema is in the appendix.

To support controlled experimentation and evaluation, we generate synthetic yet domain-consistent data samples for the year 2025. AIS tracks, satellite media footprints and detection events are simulated according to realistic spatial, temporal and kinematic constraints, while preserving the relational structure imposed by the ontology. This approach allows us to scale the dataset while ensuring semantic validity and reproducibility.

The synthetic dataset includes six ports, three maritime regions, and eight vessel classes spanning four vessel types, as summarized in Table 1. In total we have generated 26.4k vessel detections across 17.5k synthetic satellite media wkt's, 156k AIS tracks for 30 vessels for 365 days. These generated samples includes normal as well as anomaly events. We randomly generate domain-specific maritime anomaly events monthly, includes loitering, rendezvous, impossible transit, and dark vessel.

2.3 Natural Language Questions and Ground-Truth nGQL

Using the constructed KG, we curated a benchmark of 105 NL–to–nGQL query pairs. Each NL question is manually translated into an executable nGQL query, which serves as the ground truth. The benchmark includes a balanced mix of single-hop

Port (UN/LOCODE)	Jebel Ali (AEJEA), Fujairah (AEFJR), Muscat (OMMCT), Duqm (OMDQM), Dammam (SADMM), Doha (QADOH)
Maritime Region	Hormuz Strait, Arabian Gulf, Oman Coast
Vessel Class	Large Tanker, Product Tanker, Bulk or General Cargo, Container Ship, Fishing Vessel, Naval or Coastguard, Dhow, Small Boat
Vessel Type	Tanker, Cargo vessel, Government vessel, Regional vessel

Table 1: Domain categories and geographic coverage used in the synthetic data generation process.

and multi-hop direct schema-level queries, easy and hard questions, and domain-specific anomaly queries that require semantic interpretation beyond schema-level reasoning. A substantial portion of the benchmark focuses on maritime operational concepts such as Loitering (16 Qns), Rendezvous(13 Qns), Dark Vessel Behavior(17 Qns) and Impossible Transit(13 Qns) events. These concepts are not explicitly represented as flags or attributes in the KG; instead, they are implicitly defined through combinations of spatio-temporal patterns, cross-modal inconsistencies, and domain-specific constraints. For example, **loitering** denotes prolonged low-speed movement within a confined area outside normal port or anchorage activity; a **rendezvous** refers to two or more vessels remaining in close proximity outside designated ports; a **dark vessel** is identified through the absence of AIS signals despite satellite detection; and an **impossible transit** indicates physically infeasible vessel movement suggestive of spoofing or data anomalies.

Because such domain concepts are not natively encoded in system-level KGs, generating correct queries requires an understanding of their operational definitions and the ability to translate them into appropriate graph traversals and filtering conditions. This makes domain-concept queries fundamentally more challenging than generic direct KG schema-level queries. Accordingly, we manually construct ground-truth nGQL queries for both schema-level and domain-specific hard questions, ensuring that each query faithfully encodes the expert-defined semantics. All NL–nGQL pairs are independently verified by the domain experts to ensure correctness and consistency.

2.4 LLM Prompt Construction

To evaluate LLMs under consistent and informative conditions, we design a structured system prompt that provides all necessary context for query generation. It includes: (i) NebulaGraph schema, (ii) nGQL syntax specification, (iii) a set of explicit rules aimed at preventing common errors (e.g., misuse of tag-property access, invalid traversals), (iv) two representative example simple nGQL queries, and (v) formal definitions of domain concepts.

2.5 Evaluation Setup

For evaluation, natural language questions are sampled from the benchmark and provided to each LLM using a fixed system prompt. Models generate a single nGQL query per question without execution feedback or iterative refinement. Generated queries are compared against manually written ground-truth nGQL using token-based metrics that capture partial correctness and structural alignment rather than exact string matching. We use the following evaluation metrics:

Schema Linking Accuracy (SLAcc). Token-level F1 score measuring the correctness of MATCH patterns, including vertex tags, edge types, and traversal direction. Incorrect edge direction yields zero credit for the corresponding edge tokens.

Constraint / Filter Match Accuracy (CMAcc). Token-level F1 score evaluating the correctness of predicates in the WHERE clause, including attributes, operators, values, and logical connectors, independent of predicate order.

Projection Precision (Proj.). Token-level precision of the RETURN clause, computed as the ratio of correctly predicted output tokens to the total predicted tokens.

Execution Accuracy. The proportion of generated nGQL queries that execute successfully on the KG without syntax or schema errors; a query is considered correct only if the result row count matches that of the ground-truth query.

3 Experimental Analysis and Discussion

We divide benchmark questions into simple schema-level queries and domain-specific anomaly queries (e.g., loitering, rendezvous, dark vessel behavior, and impossible transit). We evaluate multiple open-source LLMs and summarize the results in Table 2. The results show that while existing LLMs perform well on schema-driven queries, their performance degrades substantially on domain-specific queries.

This performance gap indicates that current LLMs struggle to generate correct nGQL queries

Model	Simple questions				Impossible Transit				Dark Vessel Behavior				Loitering Event				Rendezvous Event			
	SLAcc	CMAcc	Proj.	ExeAcc	SLAcc	CMAcc	Proj.	ExeAcc	SLAcc	CMAcc	Proj.	ExeAcc	SLAcc	CMAcc	Proj.	ExeAcc	SLAcc	CMAcc	Proj.	ExeAcc
CodeLlamma: 7B	0.6073	0.3370	0.5248	0.1304	0.0803	0.0769	0.0000	0.0000	0.3008	0.1986	0.2916	0.0625	0.3056	0.0367	0.0655	0.0000	0.1367	0.0890	0.0912	0.00
CodeLlamma: 13B	0.7366	0.3388	0.8152	0.1956	0.0512	0.0000	0.0000	0.0000	0.4349	0.2574	0.5755	0.0815	0.0967	0.0840	0.0343	0.0588	0.0370	0.0705	0.0523	0.00
CodeLlamma: 34B	0.7666	0.4373	0.7898	0.3043	0.0085	0.089	0.0090	0.069	0.4210	0.3109	0.5625	0.4375	0.1579	0.0228	0.0235	0.0000	0.1168	0.0849	0.0384	0.0769
Deepseek-coder: 6.7B	0.6926	0.3801	0.6737	0.3260	0.085	0.079	0.0068	0.0153	0.3037	0.2476	0.3958	0.3125	0.2029	0.0588	0.0970	0.0467	0.0843	0.0822	0.00	0.00
Deepseek-coder: 33B	0.7410	0.4319	0.7427	0.3695	0.0701	0.1043	0.0602	0.0769	0.3980	0.4638	0.5424	0.3750	0.2506	0.2208	0.0490	0.0000	0.0786	0.0901	0.0476	0.00
Falcon3:3B	0.6001	0.2173	0.4140	0.1938	0.0911	0.0109	0.0384	0.0000	0.3654	0.1254	0.2120	0.128	0.1296	0.062	0.0677	0.0588	0.1267	0.0862	0.0865	0.0469
Devstral:24B	0.7624	0.5434	0.7673	0.3509	0.2264	0.3076	0.2538	0.1491	0.2882	0.2068	0.6145	0.4382	0.4536	0.2884	0.3501	0.2354	0.1247	0.1869	0.1681	0.0356
DeepCoder: 1.5B	0.7153	0.1102	0.4695	0.0384	0.0046	0.0000	0.0000	0.0000	0.3858	0.1630	0.3377	0.0625	0.0082	0.0117	0.0215	0.0000	0.0213	0.01	0.03	0.0769
DeepCoder: 14B	0.7321	0.1286	0.4923	0.1092	0.1851	0.0803	0.0564	0.0769	0.4730	0.2072	0.4852	0.25	0.1363	0.539	0.0927	0.0588	0.1945	0.2516	0.0692	0.0200
Granite-code: 20B	0.7242	0.3108	0.7065	0.1739	0.1695	0.1732	0.2179	0.1686	28.34	0.1838	0.1875	0.3125	0.1356	0.1176	0.0756	0.0743	0.0878	0.0896	0.0000	0.0000
Qwen2.5-coder:3B	0.6827	0.4176	0.7122	0.2857	0.0952	0.0420	0.0192	0.0230	0.4006	0.2701	0.1895	0.1250	0.2174	0.0573	0.2098	0.0588	0.1133	0.0653	0.1410	0.0752
Qwen2.5-coder:14B	0.7239	0.3716	0.6612	0.4285	0.1695	0.1732	0.2179	0.3076	0.4539	0.2775	0.3467	0.375	0.4357	0.2549	0.3585	0.2352	0.1435	0.0943	0.1230	0.1538

Table 2: Comparison of LLMs on the NL-to-nGQL benchmark on simple schema-level query and domain-specific complex query categories. SLAcc and CMAcc denote token-level F1 scores for schema linking and constraint matching, respectively; Proj. denotes token-level projection precision; ExeAcc denotes execution accuracy.

Que: List out dark vessel candidates detected within 125 nautical miles area of Fujairah port.

GT nGQL:

```
MATCH (p:PORT), (d:DETECTION)
WITH p, d, ST_Distance(p.PORT.port_wkt, d.DETECTION.bbox) AS dist

WHERE p.PORT.name == "Fujairah" AND d.DETECTION.no_ais == true AND dist <= 185200

RETURN p.PORT.name AS port_name, d.DETECTION.detection_id AS detection_id,
d.DETECTION.detection_time AS detection_time, d.DETECTION.vessel_class AS vessel_class,
d.DETECTION.vessel_type AS vessel_type, d.DETECTION.confidence AS confidence,
d.DETECTION.no_ais AS no_ais, dist
ORDER BY dist DESC;
```

Que: Count region-wise occurance of Rendezvous events in the year 2025.

GT nGQL:

```
MATCH (v1:VESSEL)-[:VESSEL_HAS_AIS_TRACK]->(t1:AIS_TRACK)-[:AIS_TRACK_IN_REGION]->(mr:MARITIME_REGION)
WHERE NOT (t1)-[:DURING_PORT_CALL]->(:PORT_CALL)
  AND t1.AIS_TRACK.start_time_utc.year == 2025 AND t1.AIS_TRACK.avg_speed_kn <= 3.0
WITH t1, v1, mr
MATCH (v2:VESSEL)-[:VESSEL_HAS_AIS_TRACK]->(t2:AIS_TRACK)-[:AIS_TRACK_IN_REGION]->(r2:MARITIME_REGION)
WHERE NOT (t2)-[:DURING_PORT_CALL]->(:PORT_CALL)
  AND t2.AIS_TRACK.avg_speed_kn <= 3.0 AND t2.AIS_TRACK.start_time_utc.year == 2025
  AND v2.VESSEL.vessel_id != v1.VESSEL.vessel_id AND timestamp(t1.AIS_TRACK.start_time_utc) <
timestamp(t2.AIS_TRACK.end_time_utc) AND timestamp(t2.AIS_TRACK.start_time_utc) <
timestamp(t1.AIS_TRACK.end_time_utc) AND ST_Distance(t1.AIS_TRACK.start_pt, t2.AIS_TRACK.start_pt) < 500

RETURN mr.MARITIME_REGION.name as maritime_region, count(v1) as suspisious_vessels;
```

Figure 2: Sample domain-concept questions and their manually annotated nGQLs.

when questions require grounding operational domain concepts into precise graph constraints and filtering logic. In particular, models frequently fail to translate expert-defined semantics, such as spatio-temporal persistence, cross-modal inconsistencies, or physical feasibility constraints into executable query patterns. This trend is consistent across all evaluated models, highlighting a fundamental limitation of existing LLMs in handling domain-grounded property-graph query generation. Notably, in some LLM families, larger models achieve higher schema linking and filter accuracy yet exhibit near-zero execution accuracy compared to smaller variants. Manual analysis shows that these models over-generate syntactically valid but logically irrelevant schema patterns and filters, leading to executable queries that return incorrect results.

An exception is observed for dark vessel detection, where models outperform other domain-specific categories due to the presence of an explicit `no_ais` attribute in the `DETECTION` node. In contrast, for anomaly types lacking such explicit indicators, models fail to infer the required constraints. Finally, larger models tend to over-specify queries by adding unnecessary traversals or non-existent filters, resulting in hallucinations and reduced execution performance.

4 Conclusion and Future Work

We introduced a benchmark for evaluating LLMs on natural language to nGQL query generation over a multimodal maritime knowledge graph. Our experiments show that while existing LLMs perform well on simple schema-level queries, they consistently fail on domain-specific questions requiring operational reasoning, such as loitering or dark vessel detection. These failures arise from an inability to translate expert-defined domain concepts into precise graph constraints. Beyond methodological evaluation, this work is motivated by the practical goal of supporting real-world system development in geospatial and maritime domains, where rigorous benchmarking is essential for building reliable and deployable KG-driven analytical systems. Future work will focus on integrating domain knowledge more explicitly into LLM-based query generation through structured concept representations, retrieval-augmented prompting, and execution-aware validation.

5 Limitations

This work evaluates NL-to-nGQL query generation in isolation and does not consider multi-turn or interactive querying. Although grounded in a realistic maritime knowledge graph, the benchmark relies on synthetic data and expert-defined concepts, which may not fully capture real-world variability. Our evaluation is limited to English questions and a fixed domain, and does not assess generalization across domains, schemas, or graph query languages.

References

Fahim Ahmed, Md Mubtasim Ahasan, Jahir Sadik Monon, Muntasir Wahed, M Ashraful Amin, AKM Rahman, and Amin Ahsan Ali. 2025. Bappa: Benchmarking agents, plans, and pipelines for automated text-to-sql generation. *arXiv preprint arXiv:2511.04153*.

Vincent Emonet, Jerven Bolleman, Severine Duvaud, Tarcisio Mendes de Farias, and Ana Claudia Sima. 2024. Llm-based sparql query generation from natural language over federated knowledge graphs. *arXiv preprint arXiv:2410.06062*.

Yuanyuan Liang, Lei Pan, Tingyu Xie, Yunshi Lan, and Weining Qian. 2025. Multi-turn natural language to graph query language translation. *arXiv preprint arXiv:2508.01871*.

Xinyu Liu, Shuyu Shen, Boyan Li, Peixian Ma, Runzhi Jiang, Yuxin Zhang, Ju Fan, Guoliang Li, Nan Tang, and Yuyu Luo. 2025. A survey of text-to-sql in the era of llms: Where are we, and where are we going? *IEEE Transactions on Knowledge and Data Engineering*.

Siraj Munir and Alessandro Aldini. 2024. Towards evaluating large language models for graph query generation. In *International Conference on Computational Science and Computational Intelligence*, pages 30–45. Springer.

Md Mahadi Hasan Nahid, Davood Rafiei, Weiwei Zhang, and Yong Zhang. 2025. Rethinking schema linking: A context-aware bidirectional retrieval approach for text-to-sql. *arXiv preprint arXiv:2510.14296*.

Xuan-Quang Phan, Tan-Ha Mai, Thai-Duy Dinh, Minh-Thuan Nguyen, and Lam-Son Lê. 2025. Askdb: An llm agent for natural language interaction with relational databases. *arXiv preprint arXiv:2511.16131*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, and 1 others. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv preprint arXiv:1809.08887*.