Leveraging LLM-GNN Integration for Open-World Question Answering over Knowledge Graphs

Hussein Abdallah[1], Ibrahim Abdelaziz[2], Panos Kalnis[3] and Essam Mansour[1]
[1]Concordia University, [2]IBM, [3]KAUST
hussein.abdallah@mail.concordia.ca, ibrahim.abdelaziz1@ibm.com,
panos.kalnis@kaust.edu.sa, essam.mansour@concordia.ca

Abstract

Open-world Question Answering (OW-QA) over knowledge graphs (KGs) aims to answer questions over incomplete or evolving KGs. Traditional KGQA assumes a closed world where answers must exist in the KG, limiting real-world applicability. In contrast, open-world QA requires inferring missing knowledge based on graph structure and context. Large language models (LLMs) excel at language understanding but lack structured reasoning. Graph neural networks (GNNs) model graph topology but struggle with semantic interpretation. Existing systems integrate LLMs with GNNs or graph retrievers. Some support open-world QA but rely on structural embeddings without semantic grounding. Most assume observed paths or complete graphs, making them unreliable under missing links or multi-hop reasoning. We present GLOW, a hybrid system that combines a pre-trained GNN and an LLM for open-world KGQA. The GNN predicts top-k candidate answers from the graph structure. These, along with relevant KG facts, are serialized into a structured prompt (e.g., triples and candidates) to guide the LLM's reasoning. This enables joint reasoning over symbolic and semantic signals, without relying on retrieval or fine-tuning. To evaluate generalization, we introduce GLOW-BENCH, a 1,000-question benchmark over incomplete KGs across diverse domains. GLOW outperforms existing LLM–GNN systems on standard benchmarks and GLOW-BENCH, achieving up to 53.3% and an average 38% improvement. GitHub code and data are available here.

1 Introduction

Open-World Question Answering (OW-QA) over knowledge graphs (KGs) aims to answer questions when relevant facts are missing or the KG is incomplete. This challenge arises in real-world domains like biomedicine, scientific research, and finance, where knowledge is often evolving, implicit, or incomplete (Lu and Yang, 2022; Wang et al., 2024). For example, systems may need to infer undocumented drug interactions, latent collaborations, or causal relationships. In such cases, answers are not explicitly stored in the KG and must be predicted based on graph structure, entity semantics, and cues in the question.

Traditional KGQA methods assume a *closed-world* setting, where all facts are known and retrievable. This limits their use in dynamic or incomplete environments. Open-world QA instead requires reasoning over both observed and missing information. Unlike standard retrieval or link prediction, it must integrate symbolic language with structural graph signals. This motivates hybrid approaches that combine the semantic flexibility of Large Language Models (LLMs) with the relational reasoning of Graph Neural Networks (GNNs).

Most existing KGQA systems rely on structured or dense retrieval to find explicit paths between question and answer entities. Methods like G-Retriever (He et al., 2024), GNN-RAG (Mavromatis and Karypis, 2024), RoG (Luo et al., 2024), and ToG (Sun et al., 2024) retrieve graph fragments using semantic similarity. GCR (Luo et al., 2025) improves scalability but still assumes a complete graph. These methods often fail under open-world conditions, where answer paths are missing or poorly aligned with the question. This results in noisy retrieval and low accuracy. OW-QA instead requires *predictive reasoning* to infer plausible answers beyond observed facts. AskGNN (Hu et al., 2024) addresses this by using GNNs to guide LLM inference, but relies heavily on structural embeddings and lacks semantic grounding. This limits its ability to handle complex or multi-hop questions.

To address this challenge, we propose GLOW, a novel hybrid approach that synergistically combines the strengths of both LLMs and GNNs. Our method uses a pre-trained GNN to predict the top-k possible answers based on the graph

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 575–592

Table 1: Average accuracy (%) on OW-QA benchmarks by reasoning depth. GLOW-GN leads on both 1- and 2-hop questions, while GCR drops sharply under OWA. All use Qwen3-8B; scores for existing datasets are averaged over arxiv2023, ogbn-arxiv, and ogbn-products.

Method	Existing Datasets	GLOW-Bench (ours)	
	1-Hop	**1-Hop**	**2-Hop**
LLM_{Only}	25.7	30.0	18.9
GCR	7.8	34.1	15.3
GoG	44.6	48.3	29.5
AskGNN	53.7	79.4	34.3
GLOW-GN	**71.7**	**83.3**	**42.4**

topology. These answers, along with a serialized subgraph of KG facts relevant to the question, are then incorporated into the LLM prompt using a controlled and structured format (e.g., relational triples, top-ranked candidate answers), enabling the LLM to reason jointly over language and structure. Rather than relying on retrieval or pretraining alone, our model dynamically bridges gaps in the KG by augmenting the LLM with on-the-fly, GNN-driven prompts.

We evaluate GLOW on the AskGNN (Hu et al., 2024) open-world QA benchmark, which is based on ogbn-arxiv, ogbn-products, and arxiv2023 datasets. AskGNN used these datasets to reflect realistic KG incompleteness and avoid the closed-world assumption in existing benchmarks, such as WebQSP or CWQ (Luo et al., 2025). To test generalization, we introduce GLOW-BENCH, a new benchmark of 1,000 natural language questions across diverse domains. Each question requires reasoning over incomplete KGs, with the correct answer deliberately removed. Unlike AskGNN's dataset, which covers only single-hop questions in less diverse domains, GLOW-BENCH includes 1- and 2-hop questions across multiple KGs.

We compare our method against state-of-the-art baselines, including AskGNN (Hu et al., 2024), GCR (Luo et al., 2025), a KG-grounded QA pipeline, GoG (Xu et al., 2024),KGQA over incomplete KG, and LLMs such as GPT-4o-mini (OpenAI et al., 2024) and Qwen3-8B (Yang et al., 2025). As shown in Table 1, LLMs perform poorly on deeper reasoning tasks, often failing to retrieve or organize relevant knowledge. AskGNN performs better but is limited by its structural focus and lack of semantic flexibility. GCR struggles under incomplete KGs, often hallucinating answer paths or returning incorrect answers. GoG hallucinates answer path generation as it overlooks the underlying KG schema and structural constraints, resulting in semantically inconsistent or invalid paths. Our benchmark highlights these issues and shows the need for models that can reason jointly over symbolic and structural signals. In summary, our contributions are:

- We propose GLOW, a novel OW-KGQA system that combines a GNN and an LLM to jointly reason over structured and unstructured knowledge.
- We present GLOW-BENCH, a 1,000-question benchmark for open-world KGQA with multi-hop reasoning over incomplete, cross-domain KGs.
- We demonstrate that GLOW outperforms state-of-the-art LLM–GNN QA and KGQA systems across standard benchmarks and GLOW-BENCH, with up to 53.3% and an average of 38% improvement.

2 Related Work

Our work connects to research in KG completion, LLM-based QA, GNN–LLM hybrid models, and open-world KGQA.

Knowledge Graph Completion. Embedding-based methods (TransE, RotatE, ComplEx) (Rossi et al., 2021a) and recent semantic models (StructurE, HopfE, DensE) (Ge et al., 2024) infer missing links via latent representations. Rule-based systems (AnyBURL (Rossi et al., 2021b), SAFRAN (Ott et al., 2021)) generalize KG patterns with logical rules. While useful for KG augmentation, these approaches operate independently of QA. CBR-iKB applies case-based reasoning with KGEs but is computationally expensive and limited to transductive settings. In contrast, GLOW integrates KG completion into QA through GNN-guided prompting.

LLMs for KGQA. Fine-tuning LLMs on KG triples can improve domain adaptation (Wang et al., 2021; Shu et al., 2024; Jiang et al., 2024), but requires extensive training and risks catastrophic forgetting (Xia et al., 2024; Zhao et al., 2021). Our approach avoids fine-tuning by injecting both graph-structured data and GNN outputs into LLM prompts, enabling semantic and structural reasoning without modifying LLM weights.

GNN–LLM Hybrid Models. GNN-based QA systems (Yasunaga et al., 2021; Zhu et al., 2023; Abdallah et al., 2024) support multi-hop reasoning, but often ignore language semantics. Retrieval-augmented methods, such as

GNN-RAG (Mavromatis and Karypis, 2024), G-Retriever (He et al., 2024), STaRK (Wu et al., 2024), and RoG (Luo et al., 2024), embed graph fragments for LLMs to reason over. However, they assume complete graphs and rely on retrieving full answer paths, which breaks under missing links (Zhou et al., 2025). Our method differs by using a GNN to predict candidate answers and relevant subgraphs, which are serialized into prompts for LLM reasoning.

Open-World QA with In-Context Learning. AskGNN (Hu et al., 2024) enhances retrieval using GNN-based Structure-Enhanced Retrieval (SE-Retriever) to select in-context examples. However, it relies on joint LLM–GNN training using open-weight LLMs and scales poorly with the model and graph size. It may also bias predictions toward dominant classes. GLOW avoids these issues by using lightweight GNNs for candidate generation and prompt construction, without requiring fine-tuning, and generalizing to various LLMs. GoG (Xu et al., 2024) addresses KGQA over incomplete KGs by deliberately removing randomly answer path predicates. While effective, this approach cannot guarantee the complete elimination of all answer paths and their associated edges, and generates on the fly LLM-based triples that do not conform to the KG schema, hence causing answer hallucination.

Dense KGQA via Path Retrieval. Methods like G-Retriever, GNN-RAG, ToG (Sun et al., 2024), and RoG (Luo et al., 2024) retrieve semantically similar paths, assuming the answer exists in the KG. GCR (Luo et al., 2025) improves on RoG by fine-tuning LLMs to extract answers from retrieved paths. However, these systems fail in incomplete KG settings where the answer path is missing from the KG (Zhou et al., 2025). Unlike them, GLOW supports predictive reasoning by prompting the LLM with GNN-predicted candidates and structured facts, even when no complete path exists.

Benchmarks. Existing KGQA benchmarks (e.g., WebQuestionsSP (Yih et al., 2016), LC-QuAD (Trivedi et al., 2017), MetaQA (Zhang et al., 2018), STaRK (Wu et al., 2024)) assume closed-world settings with guaranteed answer paths. AskGNN (Hu et al., 2024) introduced open-world benchmarks but is limited to single-hop reasoning in narrow domains. We introduce GLOW-BENCH, a benchmark of 1,000 open-world questions requiring single- and multi-hop reasoning across diverse KGs, where gold answers are explicitly removed to test generalization under incompleteness.

Algorithm 1 GETPROMPT: Generate GLOW Prompt from Input OWA Question over KG

Require: Q:User input question, $GLOW_v$: System variation
1: **function** GETPROMPT(Q,$GLOW_v$)
2: $Q_n, Q_e, KG \leftarrow$ entityExtraction(Q) ▷ extract the question's node and edge
3: $KG_{Sc} \leftarrow$ getKGSchema(KG) ▷ Load KG schema
4: $v_t, e_t \leftarrow$ ER-Linking(Q, KG_{Sc}, Q_n, Q_e) ▷ Link the question's node and edge crossponding KG's node and edge URIs
5: $\mathcal{RC}_q \leftarrow$ TextToSPARQL(KG_{Sc}, v_t) ▷ generate $\mathcal{RC}$ equivalent SPARQL query
6: $\mathcal{RC}_{Triples} \leftarrow$ execSPARQL($\mathcal{RC}_q, KG$) ▷ Execute the $\mathcal{RC}$ SPARQL Query
7: $L \leftarrow$ getPossibleLabels(v_t, e_t) ▷ Extract the set of possible labels
8: $RC \leftarrow$ RC-Serialization($\mathcal{RC}_{Triples}$) ▷ Serialize the $\mathcal{RC}$ triples into text.
9: $GNN_{Ans} \leftarrow$ GNNPredict(v_t, e_t) ▷ predict the top-K GNN answers for v_t and e_t
10: $P \leftarrow$ getPrompt($Q, v_t, e_t, L, \mathcal{RC}, GNN_{Ans}, GLOW_v$) ▷ generate the GLOW's variation prompt
11: **return** P
12: **end function**

OW-KGQA vs. GNN Node Classification. Our task fundamentally differs from GNN node classification (NC); it takes natural language questions as input and infers answer nodes using KG structure and semantics, whereas NC operates on graph inputs with fixed labels and no linguistic reasoning.

3 The GLOW Approach

GLOW[1] is a hybrid approach for open-world QA on KGs, combining graph-based reasoning with LLMs. Beyond *In-Context Learning* (ICL), GLOW introduces *In-Structure Learning*, where textual and KG signals jointly guide reasoning. A pre-trained GNN predicts top-k candidates and retrieves a relevant KG subgraph based on the question's entity. These are serialized into a structured prompt with relational triples and GNN predictions, enabling the LLM to reason over linguistic and structural cues without additional fine-tuning.

3.1 GLOW Pipeline Overview

We develop three GLOW variants to explore different ways of integrating structured signals: GLOW-G (graph context), GLOW-N (GNN predictions), and GLOW-GN (combined). The full architecture is shown in Figure 1 and Algorithm 1. The pipeline proceeds in four stages: Question Understanding & Linking, Retrieval, Augmentation, and Generation. Each contributes build a GNN-guided LLM prompt.

[1] **G**raph-**L**LM for **O**pen-**W**orld QA

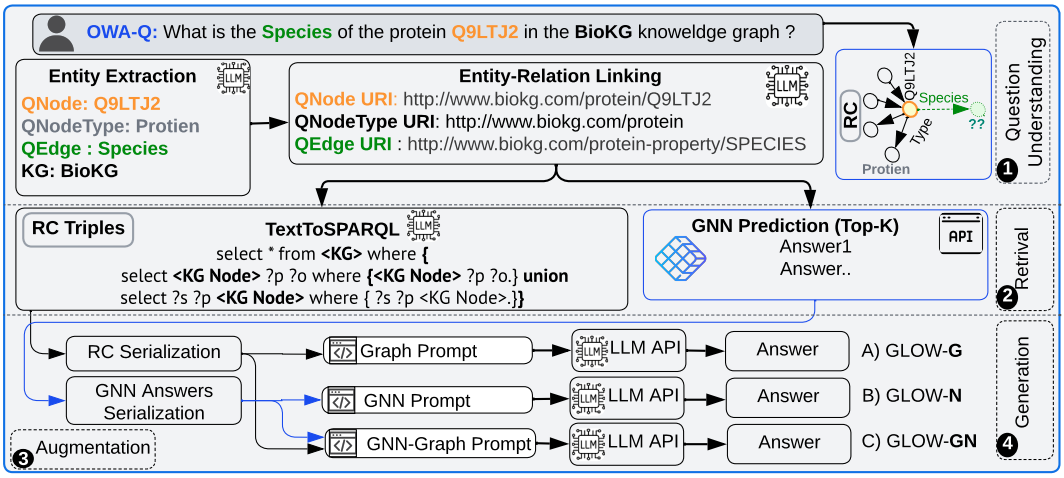

Figure 1: Overview of GLOW's four stages: question understanding, retrieval, augmentation, and generation. GLOW builds GNN-guided prompts using KG context and/or GNN predictions, with three variants: (a) KG facts (GLOW-G), (b) GNN predictions (GLOW-N), and (c) both (GLOW-GN).

Question Understanding & Linking: Given a question (e.g., "What is the **species** of the protein **Q9LTJ2** from **BioKG**"?), we extract the main entity node (Q_n), its type, the target relation (Q_e), and the KG name at algorithm 1 lines 2 and 3. The linking step at line 4 maps these elements to KG schema types using an LLM-based prompt (see Appendix A.2) over RDF metadata. For example, species maps to predicate URI http://www.biokg.com/property/SPECIES, denoted e_t, and protein maps to a schema node type URI. We resolve the question entity to node v_t via a SPARQL query over name or label fields.

Unlike exact matching, this step uses semantic signals from attributes (name, label, description, URI) and schema-aware prompts to identify the node and relation types robustly. For two-hop questions, the second edge is parsed and appended to the SPARQL query. GLOW generalizes to multi-hop reasoning through extended semantic parsing.

KG Retrieval: LLMs often lack domain-specific facts (e.g., in BioKG) to answer questions about v_t. However, neighboring KG nodes provide rich textual and structural context ($\mathcal{RC}$), which aids in inferring the missing label along e_t. The possible labels L for edge e_t are retrieved via SPARQL and provided to constrain generation at lines 5,6 and 7. We retrieve 1-hop triples connected to v_t via a SPARQL query generated at line 5 (*Text-To-SPARQL*) that accounts for schema and namespace details and serialized into triples format at line 8.

Instead of retrieving top-K similar question entity examples like AskGNN (Hu et al., 2024), which may be biased toward dominant GNN classes, we train a GNN NC model per question pattern (e.g., Protein→Species). These GNNs are trained independently of the LLM and queried via an API at inference time to return top-k candidate answers at line 9. See the GNN technical training/inference details in appendix A.4. These predictions guide the LLM to correlate graph-derived candidates with the textual semantics in $\mathcal{RC}$, improving robustness. Unlike AskGNN, if the GNN underperforms due to poor structure, the LLM can still rely on $\mathcal{RC}$. This decouples model performance from GNN reliability.

Augmentation: Each prompt contains the question Q, node v_t, edge e_t, and possible answers L, optionally augmented with $\mathcal{RC}$ and GNN predictions at line 10. The prompt takes the form:

$$\hat{Y} = f(Q, v_t, e_t, L, \mathcal{RC}, GNN_{Ans}) \quad (1)$$

Variant's Examples are provided in Appendix A.1.

Answer Generation: The LLM predicts a label for each node. Predictions are evaluated using an LLM-as-a-judge module described in §4.

3.2 GLOW Pipeline Variants

We study how structured prompts influence QA using three *In-Structure Learning* variants.

Graph-Context Prompt(GLOW-G): This variant adds the neighborhood $\mathcal{RC}$ of v_t, serialized into text via verbalization strategies (Baek et al., 2023) as shown in Figure 1.A. While this provides

grounded semantic context, it may inflate prompt size if neighborhoods are large. See Figure 1 *Text-To-SPARQL* query. AskGNN $\mathcal{RC}$ comprises the top-k similar question nodes as ICL examples, whereas GLOW-G $\mathcal{RC}$ includes a subgraph of attributes and neighboring nodes connected to the question node and serialized to offer contextual grounding to the LLM for effective reasoning.

A GNN-Guided Prompt (GLOW-N): As shown in Figure 1.B, this variant injects top-k GNN predictions for node v_t as soft guidance. It provides structural signals without needing full KG serialization, but depends on GNN accuracy. We train a GNN model for each question pattern using GraphSAINT. The training subgraphs are extracted using KGTOSA (Abdallah et al., 2024), excluding benchmark nodes. This improves scalability and yields diverse, task-specific subgraphs. At inference time, the GNN model returns top-k candidates via API. See GNN details in the appendix.

Hybrid Graph-GNN Prompt (GLOW-GN): This variant combines GLOW-G and GLOW-N by injecting both $\mathcal{RC}$ and GNN predictions. As shown in Figure 1.C, it enables the LLM to reason jointly over semantic and structural cues. If GNN confidence is low, the LLM can still rely on the verbalized KG context.

4 An Open-World Benchmark for KGQA

We present GLOW-Bench, a benchmark for multi-hop reasoning across diverse domains. It includes 25 open-world question templates based on four real-world KGs. Each template spans one of four dimensions: reasoning depth, knowledge domain, target entity type, and multiple-choice answer count.Table 2 summarizes the templates by (see Appendix A.3 for details): 1) *Reasoning Hops (RH):* 1–2 steps from target to answer; 2) *Target Entity Type:* ranging from general (e.g., people, works) to domain-specific (e.g., drugs, proteins); 3) *Knowledge Domain (KD):* Generic (G, YAGO4 (Tanon et al., 2020)), Entertainment (E, LinkedMDB (Hassanzadeh and Consens, 2009)), and Domain-Specific (DS, BioKG (Walsh et al., 2020), CrunchBase (Färber et al., 2018)); 4) *Multiple Choice Count (MCC):* 2–32+ candidates, one correct.

While GLOW-Bench builds on existing KGs, it introduces a new benchmark for OW-QA and multi-hop reasoning, which current KGQA datasets lack. All questions are designed with answers *absent* from the KG, enabling realistic evaluation under incompleteness, and are grounded in real KGs but formulated for OW-QA.

Task Formulation: Each template defines a node classification task. For example, template #1 classifies drugs by structure (Organic vs. Non-Organic) using BioKG. Given a target node v_t, the KG context is retrieved while excluding the gold answer (if present) from $\mathcal{RC}$. The model selects the correct answer from the candidate set.

Answer Evaluation: Evaluating LLM predictions is challenging due to linguistic variation, making exact matching insufficient. We adopt the LLM-as-a-Judge framework (Gu et al., 2024), where an auxiliary LLM compares outputs to gold answers in two modes: 1) *Exact Match (EM):* Identical or semantically equivalent terms (e.g., *Actor* vs. *Film Star*); 2) *Hierarchical Match (HM):* Synonyms or subtypes (e.g., *Athlete* and *Player*). We use GPT-4o-mini as the evaluation judge (Tan et al., 2025).

5 Experiments

5.1 Experimental Setup

Datasets and Metrics: In addition to our GLOW-Bench, we also evaluate GLOW on three existing datasets (Hu et al., 2024); arxiv2023, ogbn-arxiv, and ogbn-products. These datasets are originally node classification datasets, but AskGNN (Hu et al., 2024) adopted it for OW-QA, where each dataset is converted into a QA dataset using predefined question templates. In all experiments, we report the average across two runs.

Baselines: We compare GLOW against recent methods in 3 categories: *LLM-Only*, *Open-world KGQA*, and *closed-world KGQA*.

LLM: we use commercial models like GPT-4o-Mini (OpenAI et al., 2024) and DeepSeek-V3 (DeepSeek-AI et al., 2025), open-weight models including Qwen3-8B (Yang et al., 2025), DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI, 2025), and IBM Granite 3.3-8B-Instruct (Saon et al., 2025).

Open-world KGQA: *AskGNN* (Hu et al., 2024) is an approach that integrates LLMs and GNNs for QA over homogeneous graph datasets. We adapt *AskGNN* for the OW-QA setting over KGs and evaluate its performance on more challenging benchmark datasets. Recently *GoG* (Xu et al., 2024) has been developed as a KGQA system designed for incomplete KGs where the answer path is intentionally partially removed.

Table 2: Summary of Open-World Question Template (OW-QT) across four KG-based features: (1) Knowledge Domain (KD): Domain-Specific (DS), Entertainment (E), or Generic (G); (2) Target Entity Type (ET); (3) Reasoning Hops (RH); and (4) Multiple Choice Count (MCC).

KG	KD	#OW-QT	Target ET(s)	RH	#Class	MCC	Label Types
YAGO4	G	10	Person & Creative Work	1–2	3–102	2–32+	Nationality, Publisher, Occupation
BioKG	DS	6	Drug & Protein	1–2	2–29	2–32	Kingdom, Class, SPECIES, R.Keyword
LinkedMDB	E	5	Film	1–2	7–39	4–32+	Language, Producer, Genre
CrunchBase	DS	4	Investor	1	6–14	4–16	Country, InvestRegion, Company

Table 3: **Exact Match Accuracy (%) of GLOW vs. baselines across four OWA-QA datasets.** GLOW-GN consistently outperforms AskGNN, GCR, GoG, and LLM-only setups across all LLMs and datasets, showing strong generalization and effective use of both textual and structural signals, even with smaller LLMs.

	LLM-Model	Dataset	RH	LLM-Only	AskGNN	GCR	GoG	GLOW-G	GLOW-N	GLOW-GN
Open-Weight LLMs	Qwen3-8B (Yang et al., 2025)	Arxiv2023	1	31	62	4	53	64	66	**81**
		ogbn-arxiv	1	12	70	14	57	71	60	**77**
		ogbn-product	1	34	29	5	44	45	51	**57**
		GLOW-Bench	1	30	79	34	48	60	80	83
		GLOW-Bench	2	19	34	15	29	**44**	37	42
	DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI, 2025)	Arxiv2023	1	24	48	7	28	55	59	67
		ogbn-arxiv	1	13	50	15	35	30	61	63
		ogbn-product	1	27	19	12	44	38	43	51
		GLOW-Bench	1	21	75	36	20	58	78	**81**
		GLOW-Bench	2	12	31	16	13	27	32	**39**
	Granite-3.3-8B-Instruct (Saon et al., 2025)	Arxiv2023	1	14	63	14	53	32	63	77
		ogbn-arxiv	1	5	66	21	56	31	55	**73**
		ogbn-product	1	22	30	13	47	45	52	54
		GLOW-Bench	1	24	79	47	42	58	80	**82**
		GLOW-Bench	2	14	35	19	27	39	32	**42**
Commercial LLMs	GPT-4o-mini (OpenAI et al., 2024)	Arxiv2023	1	35	N/A	N/A	41	59	48	62
		ogbn-arxiv	1	50	N/A	N/A	58	63	55	**67**
		ogbn-product	1	37	N/A	N/A	48	43	51	**59**
		GLOW-Bench	1	29	N/A	N/A	57	68	78	**82**
		GLOW-Bench	2	28	N/A	N/A	31	**53**	35	49
	DeepSeek-V3 (DeepSeek-AI, 2024)	Arxiv2023	1	17	N/A	N/A	44	57	61	73
		ogbn-arxiv	1	58	N/A	N/A	61	62	63	**65**
		ogbn-product	1	36	N/A	N/A	51	45	50	**57**
		GLOW-Bench	1	34	N/A	N/A	60	69	78	**84**
		GLOW-Bench	2	30	N/A	N/A	47	**54**	34	53

Closed-world KGQA: *GCR* (Luo et al., 2025) is a method designed for scalable KGQA over large KGs. It outperforms *RoG* (Luo et al., 2024) via fine-tuning LLMs to extract answers from retrieved answer paths.

Evaluation Setup: GNN training was performed on an Ubuntu VM with dual 32-core Intel Xeon 2.4GHz CPU, 250GB RAM, and V100D-8C 16G vGPU. The GNN models were trained for the node classification tasks using GraphSAINT (Zeng et al., 2020) and ShaDowGNN (Zeng et al., 2021) with the task-oriented sampling method in (Abdallah et al., 2024). KGs were hosted on Virtuoso 07.20.32 with default settings. The GCR 8B models are fine-tuned using Colab A100 GPUs with 40G of VRAM.

5.2 Experimental Results

Benchmark Results: As shown in Table 3, GLOW consistently outperforms all baselines across LLMs and datasets, showing strong generalization and effective use of textual and structural cues. Unlike AskGNN, which depends on GNN-based ICL examples, GLOW retrieves the question's entity context and combines GNN predictions with neighborhood text for better performance. For example, using Qwen3-8B, GLOW-GN outperforms AskGNN by 18% on average across the AskGNN datasets, and by 4% and 8% on 1-hop and 2-hop GLOW-Bench questions, respectively. It also surpasses GCR by (64%, 49%, and 27%) and GoG by (20%, 35%, and 13%) . Similar trends hold across other open-weight (e.g., DeepSeek-R1-Distill-Qwen-7B, Granite-3.3-8B-Instruct) and commercial (e.g.,

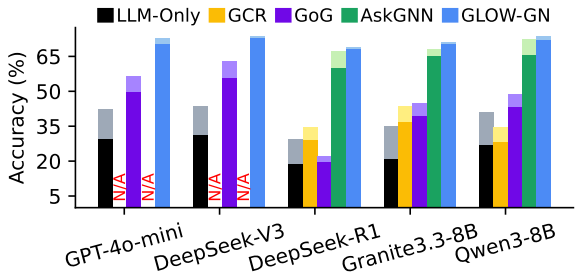

Figure 2: Average Exact and Hierarchical Match Accuracy (%) across datasets using different methods and LLMs. Dark bars show Exact Match; lighter segments show added Hierarchical Match gains. GCR and AskGNN are inapplicable to commercial LLMs.

GPT-4o-mini, DeepSeek-V3) LLMs.

AskGNN's performance varies and declines when either the GNN or LLM underperforms, showing its reliance on both components. GCR depends entirely on the presence of correct answer paths; when missing, it generates incorrect paths and answers. GoG relies entirely on generating missing paths, which are not constrained by the underlying KG schema, often leading to the creation of misleading or semantically invalid answer paths. In contrast, GLOW remains robust even with smaller LLMs, thanks to effective retrieval and contextualization. More detailed results per template and LLM are shown in Appendix A.5. For generic questions like Person $\rightarrow$ Nationality, textual attributes (e.g., name, residence, education) often suffice for LLMs to infer the answer. But for domain-specific tasks (e.g., Protein $\rightarrow$ Family or Protein $\rightarrow$ Species), textual cues are limited. Structurally similar nodes help the GNN yield better predictions.

Combining both inputs in GLOW-GN consistently boosts accuracy—up to 26 points on ogbn-arxiv and ogbn-products, and 6 points on GLOW-Bench templates like creative-work$\rightarrow$country. GLOW-G performs best on 2-hop GLOW-Bench with GPT-4o-mini and DeepSeek-V3 due to weaker GNN results on dense subgraphs. For example, in creative-work$\rightarrow$genre, the GNN reached only 22% vs. 33% by GLOW-G. In contrast, for drug$\rightarrow$class, GNN accuracy was higher (68% vs. 45%).

Exact vs. Hierarchical Match Accuracy: LLMs paraphrase answers or return semantically related concepts rather than producing exact matches. For instance, the occupation "Singer" may be returned in place of "Artist", its superclass—potentially acceptable in some contexts. Figure 2 analyzes this phenomenon by comparing Hierarchical-Match and Exact-Match accuracies, where GPT-4o-mini is used as a judge for the Hierarchical-Match.

The LLM_{Only} pipeline, which relies purely on pretraining without structural grounding, is particularly prone to generating such approximate answers. On average, its Hierarchical-Match accuracy exceeds Exact-Match accuracy by 12.5%, reflecting this tendency. This gap narrows to 3.3% with AskGNN, which supplements the LLM with contextual graph signals but still lacks fine-grained control over output specificity. In contrast, GLOW-GN demonstrates minimal reliance on hierarchical leeway—showing only a 1.1% gain—indicating its robustness in steering the LLM toward precise answers. The integration of textual and structural semantics helps the LLM disambiguate fine-grained targets, improving accuracy and consistency across datasets and LLM architectures.

5.3 Effect of Domain, Graph, and Question

Knowledge Domain (KD): We analyze model performance across question domains using GLOW-Bench. As shown in Figure 3.A, GLOW-GN consistently outperforms all baselines. Gains are most notable in domain-specific (DS) areas such as pharmaceuticals and proteins, where GLOW-GN significantly exceeds models like Qwen3-8B. These results reflect LLMs' difficulty in handling fine-grained, specialized knowledge. By integrating textual and structural signals, GLOW-GN enables stronger generalization. Even in general domains like Generic (G) and Entertainment (ET), where entities are likely seen during pretraining, GLOW-GN maintains an edge—often rivaling AskGNN without requiring additional fine-tuning.

KG Structure: We assess the effect of KG structure by comparing performance across KGs used in Amazon-Product, Arxiv, and four GLOW-Bench KGs (BioKG, CrunchBase, LinkedIMDB, YAGO4). Figure 3.B shows that GLOW-GN outperforms baselines on all domain-specific KGs and performs well on YAGO4, a generic KG. The gap narrows only on LinkedIMDB, likely due to high entity overlap with LLM training corpora.

Answer Choices (MCQ Format): We evaluate robustness under varying MCQ settings. Prior work (Zheng et al., 2024) showed LLM accuracy drops as the number of choices increases, and Figure 3.C confirms this. Still, GLOW-GN retains its edge, especially beyond two choices, underscoring the value of graph-based disambiguation under higher decision complexity.

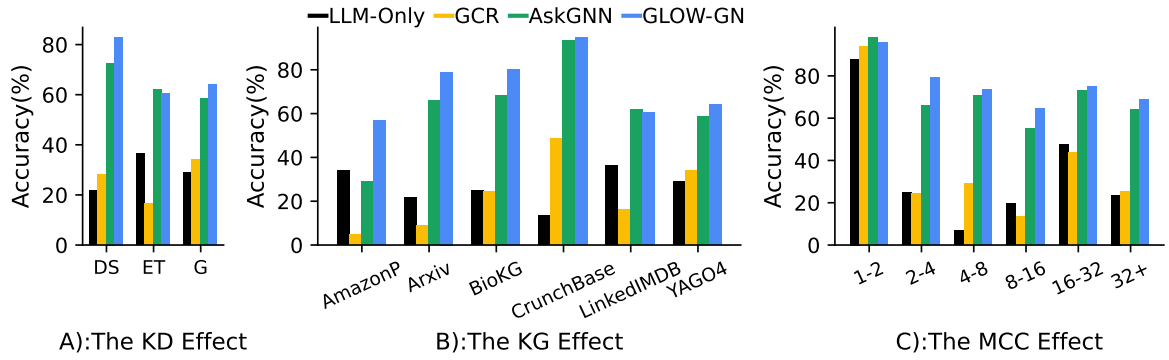

Figure 3: Effect of GLOW-Bench characteristics on the GLOW answer accuracy(%) with Qwen3-8B. The effects are grounded by A) Knowledge Domain (KD), B) Knowledge Graph (KG) , and C) Multiple Choice Count (MCC).

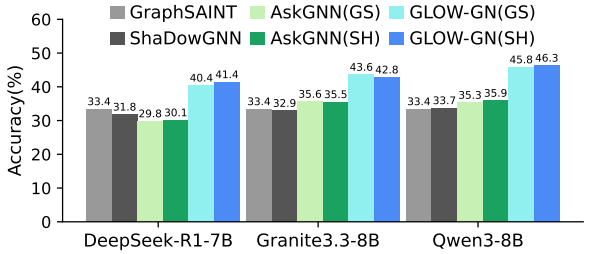

Figure 4: The GNN error propagation using GraphSAINT(GS) and ShaDowGNN(SH) models on questions with accuracy score $< 50\%$. GLOW boosts accuracy by up to 12% over weak GNNs using textual cues. AskGNN closely follows GNN performance.

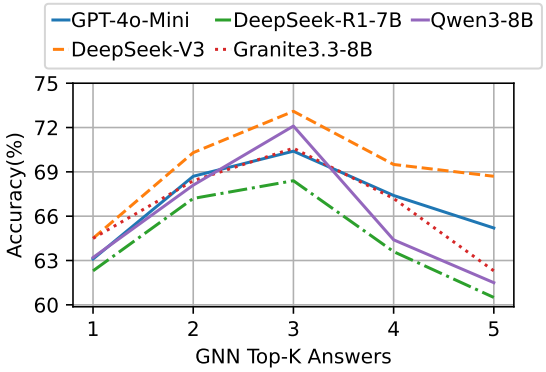

Figure 5: Impact of GNN top-K answer count on GLOW-GN's average accuracy (%) across different LLMs. GLOW-GN achieves the best performance at top-K=3, while higher values tend to mislead the LLMs.

5.4 GNN Models and Answer Selection

GNN Error Propagation Analysis: AskGNN's performance depends heavily on GNN quality. When the GNN is weak, AskGNN provides little to no gain and may even underperform the GNN itself. Figure 4 shows overall QA accuracy on OWA questions on which their corresponding GraphSAINT and ShaDowGNN models (used by AskGNN) scored below 50%. Across several LLMs, AskGNN typically matches or lags behind the GNN baseline. In contrast, GLOW-GN combines textual semantics and retrieval-based reasoning to outperform weak GNNs, with gains up to 12%. For example, on Amazon-product with Qwen3-8B, GraphSAINT scored 45%, AskGNN dropped to 29%, while GLOW-GN reached 51%. On Protein-Keyword, AskGNN achieved 35%, GraphSAINT 42%, and GLOW-GN improved to 57%.

Varying GNN Top-K Answers: Increasing top-K GNN answers confuses the LLM, lowering accuracy. Figure 5 shows GLOW-GN performs best at $K = 3$ across five LLMs, while $K = 4$ or 5 reduces performance. Large closed-weight LLMs are less sensitive to higher K values.

5.5 System Efficiency Analysis

Training Time: AskGNN's joint GNN–LLM training grows with graph size, structural complexity (e.g., node/edge types), and LLM scale, averaging 23.4 hours across tasks. GCR, fine-tuned on WebQSP, takes about 6 GPU hours for 500 epochs. In contrast, GLOW-N and GLOW-GN decouple GNN and LLM training, requiring just 1.7 hours for GNN training (Table 4). This modular design enhances scalability and avoids dependence on LLM size, easing adaptation to new domains.

Token Count: We analyze average token consumption per question across all pipelines. GLOW-N incurs the lowest token usage at 0.43K tokens per prompt, as its retrieved context ($\mathcal{RC}$) consists solely of GNN-predicted answers and candidate labels, mirroring AskGNN's structure but with reduced verbosity. GCR incurs the highest token cost due to XML-formatted triples. GoG incurs the highest token cost due to performing

Table 4: The average training time in Hours, Answer tokens count per question in (K-Tokens) and answer time per question in Seconds using Qwen3-8B LLM.

	Training Time (H)	Tokens Count (K)	Answer Time (Sec)
AskGNN	23.4	0.79	11.2
GCR	6	0.84	13.4
GoG	N/A	1.5	15.7
GLOW-G	N/A	0.78	12.6
GLOW-N	**1.7**	**0.43**	**8.2**
GLOW-GN	**1.7**	0.65	11.5

an agentic chain-of-thought rounds for missing triple generation. AskGNN averages 0.79K tokens, driven by 20 ICL examples; fewer examples significantly reduce performance. GLOW-G and GLOW-GN offer a balanced cost, with prompt length shaped by KG density and the number of neighbors connected to the question node.

Question Answering Time: In Table 4, GLOW-N is the fastest, averaging 8.2s per question due to its compact prompt. GCR is slower, as it relies on retrieved paths—often missing in OW-QA—causing hallucinations and longer processing. GoG is slowest, performing agentic chain-of-thought rounds for missing triple generation. AskGNN follows at 11.2s, hindered by reasoning over ICL examples. GLOW-GN shows moderate latency, combining GNN outputs with subgraph context. All experiments use vGPUs, not A100/H100; faster hardware would likely reduce runtime.

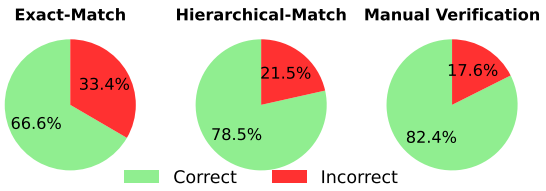

Figure 6: Human evaluation results versus Exact-match and Hierarchical-match. Hierarchical matching closely aligns with human evaluation.

Human Based Evaluation: To evaluate our LLM-as-a-Judge prompts, we manually validated the Qwen3:8B GLOW-GN answers across the GLOW-Bench question patterns, using 5 randomly selected questions per pattern. As shown in Figure 6, The Exact-match accuracy reached 66.6%, Hierarchical match 78.5%, and Manual verification is 82.4%. Notably, Hierarchical matching recovered 95.2% of manually verified correct answers, demonstrating the robustness of our prompts. Some responses were marked as non-matches hierarchically despite being valid (e.g., (CEO, Co-Founder), (20th Century Studios, Paramount Pictures), (Catalan, Spanish)), while a few cases were incorrectly flagged as non-exact matches (e.g., (American English, English)).

6 Conclusion

This paper introduces GLOW, a system for open-world QA on KGs that integrates LLMs with GNNs. GLOW uses GNN-predicted candidates and relevant subgraphs as structured context to enhance multi-hop reasoning over incomplete KGs. By combining structural and textual semantics, GLOW overcomes key limitations of closed-world and retrieval-based KGQA. It consistently achieves strong performance across question types, domains, and LLMs, even when component quality varies. On standard open-world benchmarks and our new GLOW-Bench dataset, GLOW shows significant improvements in exact and semantic accuracy. These results highlight the need for hybrid approaches tailored to open-world settings and confirm GLOW's robustness and generalizability.

7 Limitations

This work has three main limitations: First, data quality issues, such as sparse KGs with limited node and edge descriptions, impair both textual and structural semantics, reducing performance, especially for large LLMs. Second, our approach depends on high-performing GNN models for effectiveness. Third, all questions are currently framed as node classification tasks, though some may be better suited to link prediction, requiring prior evaluation and task-specific formulation.

References

Hussein Abdallah, Waleed Afandi, Panos Kalnis, and Essam Mansour. 2024. Task-oriented gnns training on large knowledge graphs for accurate and efficient modeling. In *ICDE*, pages 1833–1846. IEEE.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *CoRR*, abs/2306.04136.

DeepSeek-AI. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, and et.al. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Michael Färber, Carsten Menne, and Andreas Harth. 2018. A linked data wrapper for crunchbase. *Semantic Web*, 9(4):505–515.

Xiou Ge, Yun Cheng Wang, Bin Wang, C-C Jay Kuo, and 1 others. 2024. Knowledge graph embedding: An overview. *APSIPA Transactions on Signal and Information Processing*, 13(1).

Jiawei Gu, Xuhui Jiang, and et.al. 2024. A survey on llm-as-a-judge. *CoRR*, abs/2411.15594.

Oktie Hassanzadeh and Mariano P. Consens. 2009. Linked movie data base. In *WWW2009 Workshop on Linked Data on the Web*, volume 538 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *NeurIPS*.

Zhengyu Hu, Yichuan Li, Zhengyu Chen, Jingang Wang, Han Liu, Kyumin Lee, and Kaize Ding. 2024. Let's ask GNN: empowering large language model for graph in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 1396–1409.

Pengcheng Jiang, Lang Cao, Cao (Danica) Xiao, Parminder Bhatia, Jimeng Sun, and Jiawei Han. 2024. KG-FIT: knowledge graph fine-tuning upon open-world knowledge. In *NeurIPS*.

Jiaying Lu and Carl Yang. 2022. Open-world taxonomy and knowledge graph co-learning. In *AKBC*.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *International Conference on Learning Representations*.

Linhao Luo, Zicheng Zhao, Chen Gong, Gholamreza Haffari, and Shirui Pan. 2025. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. In *Forty-second International Conference on Machine Learning*.

Costas Mavromatis and George Karypis. 2024. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*.

OpenAI, Josh Achiam, and Steven Adler et.al. 2024. Gpt-4 technical report.

Simon Ott, Christian Meilicke, and Matthias Samwald. 2021. SAFRAN: an interpretable, rule-based link prediction method outperforming embedding models. In *AKBC*.

Andrea Rossi, Denilson Barbosa, and et.al. 2021a. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Trans. Knowl. Discov. Data*, 15(2):14:1–14:49.

Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. 2021b. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Trans. Knowl. Discov. Data*, 15(2):14:1–14:49.

George Saon, Avihu Dekel, and et.al. 2025. Granite-speech: open-source speech-aware llms with strong english asr capabilities. *Preprint*, arXiv:2505.08699.

Michael Sejr Schlichtkrull and et al. Thomas N. Kipf. 2018. Modeling relational data with graph convolutional networks. In *ESWC*, volume 10843, pages 593–607.

Dong Shu, Tianle Chen, Mingyu Jin, Chong Zhang, Mengnan Du, and Yongfeng Zhang. 2024. Knowledge graph large language model (KG-LLM) for link prediction. In *ACML*, volume 260 of *Proceedings of Machine Learning Research*, pages 143–158. PMLR.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Sijun Tan, Siyuan Zhuang, and et.al. 2025. Judgebench: A benchmark for evaluating llm-based judges. In *ICLR*. OpenReview.net.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian M. Suchanek. 2020. YAGO 4: A reasonable knowledge base. In *The Semantic Web - 17th International Conference, ESWC*, volume 12123 of *Lecture Notes in Computer Science*, pages 583–596. Springer.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In *International Semantic Web Conference*, pages 210–218. Springer.

Brian Walsh, Sameh K. Mohamed, and Vít Nováček. 2020. Biokg: A knowledge graph for relational learning on biological data. In *Proceedings of the 29th ACM Conference on Information Knowledge Management*, page 3173–3180, New York, NY, USA. Association for Computing Machinery.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021. Structure-augmented text representation learning for efficient knowledge graph completion. In *WWW*, pages 1737–1748. ACM / IW3C2.

Chengrui Wang, Qingqing Long, and et.al. 2024. Biorag: A RAG-LLM framework for biological question reasoning. *CoRR*, abs/2408.01107.

Shirley Wu, Shiyu Zhao, Michihiro Yasunaga, Kexin Huang, Kaidi Cao, Qian Huang, Vassilis N. Ioannidis, Karthik Subbian, James Zou, and Jure Leskovec. 2024. Stark: Benchmarking llm retrieval on textual and relational knowledge bases. In *Advances in Neural Information Processing Systems*, volume 37, pages 127129–127153. Curran Associates, Inc.

Yuchen Xia, Jiho Kim, and et.al. 2024. Understanding the performance and estimating the cost of LLM fine-tuning. In *IISWC*, pages 210–223. IEEE.

Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Jun Zhao, and Kang Liu. 2024. Generate-on-graph: Treat LLM as both agent and KG for incomplete knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18410–18430. Association for Computational Linguistics.

An Yang, Anfeng Li, and et.al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

Hanqing Zeng, Muhan Zhang, Yinglong Xia, and et.al. 2021. Decoupling the depth and scope of graph neural networks. In *NeurIPS*, pages 19665–19679.

Hanqing Zeng, Hongkuan Zhou, and et.al. 2020. Graphsaint: Graph sampling based inductive learning method. In *ICLR*. , GitHub Code: `https://github.com/snap-stanford/ogb/blob/master/examples/nodeproppred/mag/graph_saint.py`.

Yuyu Zhang, Hanjun Dai, Alexander J Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *AAAI*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Chujie Zheng, Hao Zhou, and et.al. 2024. Large language models are not robust multiple choice selectors. In *ICLR*. OpenReview.net.

Dongzhuoran Zhou, Yuqicheng Zhu, Yuan He, Jiaoyan Chen, Evgeny Kharlamov, and Steffen Staab. 2025. Evaluating knowledge graph based retrieval augmented generation methods under knowledge incompleteness. *Preprint*, arXiv:2504.05163.

Zhaocheng Zhu, Xinyu Yuan, Michael Galkin, Louis-Pascal A. C. Xhonneux, Ming Zhang, Maxime Gazeau, and Jian Tang. 2023. A*net: A scalable path-based reasoning approach for knowledge graphs. In *NeurIPS*.

A Appendix

A.1 GLOW Prompt Examples

User Prompt:

Predict the chemical kingdom for the drug Yohimbine from the BioKG knowledge graph.
Answer:

Question Understanding Prompt:

<system>: You are an an expert Entity-Extraction NLP system.
<user>: Given the following question, identify 1-The question main entity type, 2- the main entity, 3- the prediction label and 4- the KG name.
Question: {}
Answer:
1-Question main entity: 2-Main Entity:
3-Prediction label: 4-KG name:

Entity/Relation Linking Prompt:

<system>: You are an an expert knowledge graph entity-relation Linking NLP system.
<user>: Given the following KG Schema in the basic graph pattern CSV format, one predicate per line: node type, relation, node type.
———
KG Schema:
{}
———
1- What is the node type in the schema that corresponds to {main entity}? Return only the name.
2- Choose from the schema the BGP (node type, relation, node type) that describes the {main entity type} value {main entity}. Return only the BGP.
3- Choose from the schema the BGP (node type, relation, node type) that describe the {main entity type } {prediction label}. Return only the BGP.
———
Answer:
1-
2-
3-

Question To SPARQL Prompt:

<system>: You are an an expert text-To-SPARQL translation system.
<user>: Given the following KG Schema in the basic graph pattern CSV format, one predicate per line: node type, relation, node type.
———
KG Schema:
{}
————
Write a SPARQL query that selects the {main entity type} that satisfy the following BGPs.
1- {question node type} ,[label/name/titile] ,{question node}
2- {Other BGPs}
graph prefix: {KG Prefix}
———————
Answer: SPARQL Query
Do not return any explanation or reasoning details.

SPARQL Query Example:

```
PREFIX biokg: <http://www.biokg.com/>
SELECT ?drug as ?vt ?kingdom as ?vl
WHERE {
VALUES ?name { "Yohimbine" }
?drug biokg:NAME ?name .
?drug biokg:KINGDOM ?kingdom .}
```

Basic Prompt:

<system>: You are an expert open world question answer system.
<user>: What is the {**Prediction Label**} of the {**question entity type**} {**question entity**} from {**KG details**} knowledge graph.
- Do not return any context or analysis.
- Help: The possible list of {**Prediction Label Type**}s are: [{ **Labels List**}]
Answer

To generate an instance of this template, replace the question node type, KG, and prediction label with values from one of the queries in table 5. **An example prompt for OWA-Q #1**:

<system>: You are an expert open world question answer system.
<user>: What is the **Kingdom** of the **Drug** Yohimbine from the **BioKG , a Biomedical** knowledge graph.
- Do not return any context or analysis.
- Help: The list of **Kingdoms** are: [Organic,Non-Organic]
Answer:

GLOW-G Prompt:

<system>: You are an expert open world question answer system.
<user>: What is the **Kingdom** of the **Drug** Yohimbine from the **BioKG , a Biomedical** knowledge graph.
- Do not return any context or analysis.
- Help: The possible list of **Kingdoms** are: [**Organic,Non-Organic**]
- The **Durg** associated triples. [("Yohimbine","DDI", "DB13677"), ..]
Answer:

GLOW-N Prompt:

<system>: You are an expert open world question answer system.
<user>: What is the **Kingdom** of the **Drug** Yohimbine from the **BioKG , a Biomedical** knowledge graph.
- Do not return any context or analysis.
- Help: The possible list of **Kingdoms** are: [**Organic,Non-Organic**]
- Verify the following list of GNN Answers: [Organic,Non-Organic, ..]
Answer:

GLOW-GN Prompt:

<system>: You are an expert open world question answer system.
<user>: What is the **Kingdom** of the **Drug** Yohimbine from the **BioKG , a Biomedical** knowledge graph.
- Do not return any context or analysis.
- Help: The possible list of **Kingdoms** are : [**Organic,Non-Organic**]
- Verify the following GNN Answer: [Organic]
- The **Durg** associated triples. [("Yohimbine","DDI", "DB13677"), ..]
Answer:

A.2 LLM as-a-Judge Prompt

<system>: You are an expert LLM-as-a-Judge system.
<user>: Given the following list of predicted and true pairs of values.
-Rank the predicted value against the true value using two metrics.
1- Exact Match Rule: you compare the two strings after normalization and remove any special characters. report 1 if both values are literally and semantically equal and 0 otherwise.
2- Hierarchical/Categorical Match Rule: report 1 if the predicted value is under a subcategory or hierarchically belongs to the true value or is a synonym and 0 otherwise.
- Example:
List of pairs: [[music, art], [painter, artist],[football player, soccer player], [lawyer, judge], [lawyer, player]]
Answer: [[0,1],[0,1],[1,1],[0,1],[0,0]]
- Question:
-List of pairs: {**ListOfPairs**}
-Note: refine each pair and return Answer for exactly {**length(ListOfPairs)**} pairs without explanation.
-Finally: make sure you return only {**length(ListOfPairs)**} pair of answers.
Answer:

A.3 Full Details of Our GLOW-Bench

Table 5: Our GLOW-Bench benchmark characterizes each Open-World Question Template (OW-QT) across four KG-based features: (1) Knowledge Domain (KD): Domain-Specific (DS), Entertainment (E), or Generic (G); (2) Target Entity Type (ET); (3) Reasoning Hops (RH); and (4) Multiple Choice Count (MCC).

OW-QT	KG	KD	Target ET	RH	#Class	MCC	Label Type	Reasoning Path
1	BioKG	DS	Drug	1	2	2–4	Kingdom	Kingdom
2		DS	Drug	1	13	8–16	Superclass	Superclass
3		DS	Drug	2	29	16–32	Class	RelatedPubMed → Class
4		DS	Protein	1	18	16–32	SPECIES	SPECIES
5		DS	Protein	2	15	8–16	R.Keyword	RelatedPubMed → R.Keyword
6		DS	Protein	1	4	2–4	Family	Family
7	CrunchBase	DS	Investor	1	12	8–16	Country	Country
8		DS	Investor	1	14	8–16	InvestRegion	InvestRegion
9		DS	Investor	1	6	4–8	PositionTitle	PositionTitle
10		DS	Investor	1	11	8–16	Company	Company
11	LinkedMDB	E	Film	1	15	8–16	Language	Language
12		E	Film	1	27	16–32	Country	Country
13		E	Film	2	39	32+	Producer	Sequel → Producer
14		E	Film	2	7	4–8	Genre	Sequel → Genre
15		E	Film	1	28	16–32	Subject	Subject
16	YAGO4	G	CreativeW	2	13	8–16	ProdComp	byArtist → ProdComp
17		G	CreativeW	2	14	8–16	Publisher	isBasedOn → Publisher
18		G	CreativeW	1	25	16–32	PublishLang	Author
19		G	CreativeW	2	11	8–16	Genre	ProdComp → Genre
20		G	CreativeW	1	6	4–8	Country	Country
21		G	Person	1	3	2–4	GivenAward	GivenAward
22		G	Person	1	91	32+	Nationality	Nationality
23		G	Person	1	4	4–8	GraduateOfOrg	GraduateOfOrg
24		G	Person	1	102	32+	Occupation	Occupation
25		G	Person	1	8	4–8	SpokenLang	SpokenLang

A.4 The GNN Training/Inference Technical Details.

- For each question pattern, we train a node classification GNN using GraphSAINT with RGCN as GNN convolutional layer to support heterogeneous KG subgraphs. The initial node embeddings are iteratively aggregated with embeddings received from neighboring nodes connected to a specific relation until the embeddings of all nodes converge. The final embedding of a v_t, our main question entity, is obtained through two aggregations: an outer aggregation over each relation type and an inner aggregation over neighbouring nodes $\mathcal{RC}$ of a specific relation and defined by RGCN (Schlichtkrull and Thomas N. Kipf, 2018) as follows:

$$h_i^{(l+1)} = \sigma\Big(\sum_{r\in R}\sum_{j\in N_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\Big) \tag{2}$$

  where l is an RGCN layer, $h_j^{(l+1)}$ is the hidden embedding of node j at layer $l+1$, σ is element-wise activation function, N_i^r denotes the set of neighbour indices of node i under relation $r \in \mathcal{R}$, $c_{i,r}$ is a normalization constant that can either be learned or chosen in advance (such as $c_{i,r} = |N_i^r|$), $W_r^{(l)}$ is the weight matrix for relation r at layer l, and W_0^l is the initial weight matrix at layer l.

- To construct the training set, we exclude the GLOW-Bench question entity nodes from the training set and extract 1-hop in and out neighbor nodes subgraphs using the KGTOSA sampler (Abdallah et al., 2024). This ensures scalability to large KGs and yields task-relevant and diverse subgraphs for effective training.

- GraphSAINT Training HyperParameters: We set the input dimension (D = 128), hidden channels dimension=64, and number of layers (L = 2) of our GNN module with dropout rate 0.5 applied to each layer. We train the model with the Adam optimizer, learning rate from 5e-4, 5e-3, 1e-3, 2e-3.
- Subgraph Sampling Hyperparameters: batch-size=20000, walk-length=2, num-steps=10.
- At inference time, the node classification model M is resolved using the KG name, the question's main entity type, and the target edge. The model M predicts the top-k answer classes based on the Log-Likelihood for each class and feeds them to the LLM prompt. The trained GNN model is hosted via an inference API. At runtime, this API returns the top-K predicted answer nodes, which are then passed to the GLOW pipeline as supportive evidence for the LLM's reasoning.

A.5 The OWA-QA Detailed Results:

Table 6: The detailed accuracy (%) of GLOW compared to baseline models on existing question answering datasets from (Hu et al., 2024) and our developed benchmark GLOW-Bench. All results below use Qwen3-8B as the underlying LLM. Best results are marked in **bold**.

LLM-Model	Dataset	RH	AskGNN	GraphSAINT	LLM-Only	GCR	**GLOW-L**	**GLOW-G**	**GLOW-N**	**GLOW-GN**
Qwen3:8b	Arxiv2023	1	62	41	31	4	59	64	66	**81**
	ogbn-arxiv	1	70	59	12	14	63	71	60	**77**
	ogbn-product	1	29	45	34	5	41	44	51	**57**
	drug-superclass	1	88	84	29	3	50	77	85	**92**
	drug-kingdom	1	98	96	88	94	91	**98**	97	96
	protein-SPECIES	1	75	89	8	16	19	71	91	**92**
	protein-FAMILY	1	67	70	9	13	35	39	69	**71**
	film-country	1	55	55	40	4	40	**60**	57	55
	film-subject	1	77	81	15	7	50	30	**82**	80
	film-language	1	**87**	83	70	13	73	80	85	86
	person-nationality	1	86	89	27	38	45	89	**91**	90
	parson-graduateOfOrg	1	51	53	3	25	27	36	54	**55**
	person-occupation	1	48	46	7	12	15	38	46	**53**
	person-spokenLang	1	86	83	70	61	83	**89**	83	85
	person-givenAward	1	93	91	7	50	75	93	93	**93**
	CWork-PublishedLang	1	75	80	50	60	55	55	85	**95**
	CWork-country	1	72	51	64	43	62	75	48	**79**
	person-title	1	79	72	5	41	63	47	77	**83**
	person-Company	1	97	95	3	93	30	21	96	**97**
	person-InvestementRegion	1	98	94	7	46	95	87	96	**100**
	person-InvestementCountry	1	**98**	95	38	14	94	55	97	97
	drug-class	2	46	68	11	15	33	45	70	**73**
	protein-keyword	2	35	42	5	7	33	53	44	**57**
	film-genre	2	54	51	50	25	60	**62**	51	52
	film-producer	2	**35**	21	7	7	14	24	21	28
	CWork-ProductionCompany	2	20	22	10	29	21	**31**	25	28
	CWork-Genere	2	20	22	16	7	22	**33**	19	27
	CWork-publisher	2	31	28	33	17	58	**62**	27	32

Table 7: The detailed accuracy (%) of GLOW compared to baseline models on existing question answering datasets from (Hu et al., 2024) and our developed benchmark GLOW-Bench. All results below use GPT-4o-Mini (OpenAI et al., 2024) as the underlying LLM. Best results are marked in **bold**.

LLM-Model	Dataset	RH	AskGNN	GraphSAINT	LLM-Only	**GLOW-L**	**GLOW-G**	**GLOW-N**	**GLOW-GN**
GPT-4o-Mini	Arxiv2023	1	N/A	41	35	58	59	48	**62**
	ogbn-arxiv	1	N/A	41	35	58	59	48	**67**
	ogbn-product	1	N/A	45	N/A	N/A	N/A	N/A	N/A
	drug-superclass	1	N/A	84	8	47	57	73	**87**
	drug-kingdom	1	N/A	96	31	**100**	**100**	98	99
	protein-SPECIES	1	N/A	89	6	10	77	92	**93**
	protein-FAMILY	1	N/A	70	11	53	43	69	**72**
	film-country	1	N/A	55	47	47	**71**	57	58
	film-subject	1	N/A	**81**	15	73	38	76	75
	film-language	1	N/A	83	80	80	80	86	**88**
	person-nationality	1	N/A	89	43	56	89	**92**	87
	parson-graduateOfOrg	1	N/A	53	19	22	33	55	**57**
	person-occupation	1	N/A	46	7	23	30	46	**52**
	person-spokenLang	1	N/A	83	70	81	**91**	83	89
	person-givenAward	1	N/A	91	6	68	**96**	93	91
	CWork-PublishedLanguage	1	N/A	80	65	60	65	83	**87**
	CWork-country	1	N/A	51	62	64	77	48	**83**
	person-title	1	N/A	72	5	38	55	74	**77**
	person-company	1	N/A	95	3	6	30	96	**98**
	person-InvestementRegion	1	N/A	94	9	7	96	**100**	96
	person-InvestementCountry	1	N/A	95	41	44	**100**	97	97
	drug-class	2	N/A	68	8	27	45	**69**	67
	protein-keyword	2	N/A	42	7	28	44	40	**45**
	film-genre	2	N/A	51	68	65	**70**	50	53
	film-producer	2	N/A	21	20	17	20	21	**31**
	CWork-ProductionCompany	2	N/A	22	10	28	**42**	25	32
	CWork-Genere	2	N/A	22	27	27	**72**	19	44
	CWork-publisher	2	N/A	28	62	70	**79**	25	75

Table 8: The detailed accuracy (%) of GLOW compared to baseline models on existing question answering datasets from (Hu et al., 2024) and our developed benchmark GLOW-Bench. All results below use DeepSeek-V3 (DeepSeek-AI, 2024) as the underlying LLM. Best results are marked in **bold**.

LLM-Model	Dataset	RH	AskGNN	GraphSAINT	LLM-Only	**GLOW-L**	**GLOW-G**	**GLOW-N**	**GLOW-GN**
DeepSeek-V3	Arxiv2023	1	N/A	41	17	50	57	61	**73**
	ogbn-arxiv	1	N/A	59	58	60	62	63	**65**
	ogbn-product	1	N/A	45	N/A	N/A	N/A	N/A	N/A
	drug-superclass	1	N/A	84	3	32	48	73	**86**
	drug-kingdom	1	N/A	96	9	88	**100**	98	96
	protein-SPECIES	1	N/A	89	2	8	79	92	**93**
	protein-FAMILY	1	N/A	70	21	76	**87**	69	81
	film-country	1	N/A	55	57	65	70	57	**71**
	film-subject	1	N/A	81	23	80	38	80	**83**
	film-language	1	N/A	83	81	86	86	86	**93**
	person-nationality	1	N/A	89	72	81	87	91	**93**
	parson-graduateOfOrg	1	N/A	53	32	48	43	52	**54**
	person-occupation	1	N/A	46	19	34	34	46	**53**
	person-spokenLang	1	N/A	83	79	79	82	83	**85**
	person-givenAward	1	N/A	91	3	56	58	93	**94**
	CWork-PublishedLanguage	1	N/A	80	47	50	50	84	**90**
	CWork-country	1	N/A	51	68	75	**87**	48	79
	person-title	1	N/A	72	41	41	63	74	**77**
	person-company	1	N/A	95	3	24	33	96	**97**
	person-InvestementRegion	1	N/A	94	7	21	100	89	**100**
	person-InvestementCountry	1	N/A	95	44	58	100	97	**100**
	drug-class	2	N/A	68	7	21	39	69	**70**
	protein-keyword	2	N/A	42	3	40	**59**	49	52
	film-genre	2	N/A	51	53	60	58	50	**62**
	film-producer	2	N/A	21	10	30	**50**	14	34
	CWork-ProductionCompany	2	N/A	22	28	53	**57**	25	53
	CWork-Genere	2	N/A	22	18	22	27	16	**28**
	CWork-publisher	2	N/A	28	70	79	**87**	16	70

Table 9: The detailed accuracy (%) of GLOW compared to baseline models on existing question answering datasets from (Hu et al., 2024) and our developed benchmark GLOW-Bench. All results below use DeepSeek-R1-Distill-Qwen:7B (DeepSeek-AI, 2025) as the underlying LLM. Best results are marked in **bold**.

LLM-Model	Dataset	RH	AskGNN	GraphSAINT	LLM-Only	GCR	**GLOW-L**	**GLOW-G**	**GLOW-N**	**GLOW-GN**
	Arxiv2023	1	48	41	24	7	44	55	59	**67**
	ogbn-arxiv	1	50	59	13	15	30	30	61	**63**
	ogbn-product	1	19	45	27	12	36	38	43	**51**
	drug-superclass	1	53	84	20	5	20	21	**69**	68
	drug-kingdom	1	98	96	60	90	90	**100**	98	**100**
	protein-SPECIES	1	71	89	1	12	4	62	88	**90**
	protein-FAMILY	1	62	70	7	23	30	35	68	**69**
	film-country	1	56	55	17	34	44	52	57	**64**
	film-subject	1	78	81	8	8	11	34	76	**82**
	film-language	1	86	83	73	32	73	66	86	**88**
	person-nationality	1	85	89	22	41	30	79	**91**	89
	parson-graduateOfOrg	1	50	53	5	17	19	36	53	**54**
	person-occupation	1	47	46	3	12	15	31	46	**54**
DeepSeek-R1-Distill-Qwen:7B	person-spokenLang	1	82	83	35	56	41	79	81	**83**
	person-givenAward	1	89	91	8	23	50	70	90	**92**
	CWork-PublishedLanguage	1	71	80	50	44	50	55	77	**81**
	CWork-country	1	62	51	37	42	41	**70**	48	66
	person-title	1	73	72	5	51	44	52	80	**82**
	person-company	1	93	95	3	90	30	19	**96**	92
	person-InvestementRegion	1	92	94	3	33	64	96	**100**	**100**
	person-InvestementCountry	1	93	95	11	56	72	94	97	**97**
	drug-class	2	41	68	4	13	15	24	69	**71**
	protein-keyword	2	31	42	3	8	18	22	32	**38**
	film-genre	2	53	51	35	18	50	56	50	**57**
	film-producer	2	30	21	7	9	7	7	14	**24**
	CWork-ProductionCompany	2	18	22	10	23	17	21	25	**32**
	CWork-Genere	2	17	22	11	17	11	16	16	**27**
	CWork-publisher	2	25	28	12	23	20	**40**	16	26

Table 10: The detailed accuracy (%) of GLOW compared to baseline models on existing question answering datasets from (Hu et al., 2024) and our developed benchmark GLOW-Bench. All results below use IBM Granite-3.3-8B-instruct (Saon et al., 2025) as the underlying LLM. Best results are marked in **bold**.

LLM-Model	Dataset	RH	AskGNN	GraphSAINT	LLM-Only	GCR	**GLOW-G**	**GLOW-N**	**GLOW-GN**
	Arxiv2	1	63	41	14	14	32	63	**77**
	ogbn-arxiv	1	66	59	5	21	**31**	55	**73**
	ogbn-product	1	30	45	22	13	45	52	**54**
	drug-superclass	1	86	84	0	9	22	73	**71**
	drug-kingdom	1	97	96	8	98	95	98	**99**
	protein-SPECIES	1	75	89	2	18	45	93	**90**
	protein-FAMILY	1	67	70	0	33	36	69	**72**
	film-country	1	56	56	48	60	**66**	58	55
	film-subject	1	76	81	12	11	27	81	**85**
	film-language	1	84	83	87	46	80	87	**88**
	person-nationality	1	86	89	52	88	79	91	**92**
	parson-graduateOfOrg	1	52	53	6	14	31	56	**57**
	person-occupation	1	47	46	8	23	23	46	**50**
Granite-3.3:8B-Instruct	person-spokenLang	1	85	83	48	75	52	83	**88**
	person-givenAward	1	93	92	3	25	63	91	**94**
	CWork-PublishedLanguage	1	81	80	45	35	80	81	**94**
	CWork-country	1	72	51	58	69	**75**	59	**69**
	person-title	1	78	72	14	55	64	80	**83**
	person-company	1	97	96	3	96	42	96	**97**
	person-InvestementRegion	1	99	95	4	29	82	98	**100**
	person-InvestementCountry	1	**99**	96	39	69	78	96	97
	drug-class	2	47	68	0	18	27	70	**70**
	protein-keyword	2	35	42	2	10	36	9	**38**
	film-genre	2	54	51	35	15	52	54	**57**
	film-producer	2	**35**	21	5	5	30	23	29
	CWork-ProductionCompany	2	22	22	7	18	21	25	**32**
	CWork-Genere	2	21	22	16	22	**50**	17	33
	CWork-publisher	2	32.7	28	33	46	**58**	27	36