

Logic Meets Language: Hybrid Query Answering over Incomplete Text-Labelled Graphs

Yunjie He ¹, Bo Xiong ³, Daniel Hernandez ¹, Yuqicheng Zhu ^{1,2},
Yi Wang ¹, Evgeny Kharlamov ^{2,5}, Steffen Staab ^{1,4},

¹University of Stuttgart, ²Bosch Center for Artificial Intelligence,
³Iowa State University, ⁴University of Southampton ⁵University of Oslo

Correspondence: yunjie.he@ki.uni-stuttgart.de

Abstract

Text-labeled graph databases combine symbolic graph structures with rich textual descriptions, enabling expressive representations for real-world data. However, answering queries over such databases remains challenging when queries require both precise logical constraints and flexible semantic text interpretation, especially under information incompleteness. We introduce the task of hybrid query answering (hybrid QA) over *incomplete text-labeled graph databases*, which requires joint reasoning over both symbolic and textual knowledge under incompleteness. To investigate this problem, we formalize a query language that integrates logical constraints with free-form textual descriptions and construct two new benchmark datasets from biological and e-commerce domains. We propose HybridQE, a hybrid query embedding framework that aligns with our query language via embedding and combining the symbolic and textual constraints in a unified embedding space. Experiments on the proposed benchmarks show that existing methods struggle with this setting, while HybridQE provides consistent improvements across diverse hybrid query types, highlighting both the difficulty of HybridQA and the promise of hybrid query embeddings for reasoning under incompleteness.

1 Introduction

Text-labeled graph databases (He and McAuley, 2016; Jin et al., 2023; Wu et al., 2025) combine textual descriptions with graph structured data, facilitating a flexible and expressive framework for information storage (Yan et al., 2023; Kuang et al., 2023). Effective information retrieval systems over text-labeled graph databases are crucial for various downstream tasks, such as text classification (Zhao et al., 2023) and personalized recommendation (Zhu et al., 2021b).

A body of work integrates textual and relational evidence for question answering. GraftNet (Sun

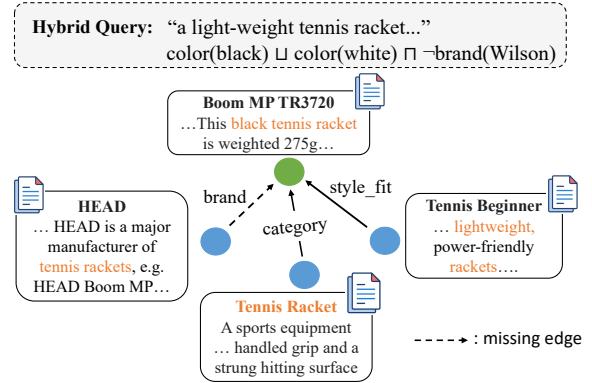


Figure 1: An illustration of a hybrid QA system over incomplete text-labeled graphs in the e-commerce setting.

et al., 2018) and PullNet (Sun et al., 2019) retrieve and reason jointly over text passages and knowledge graph triples for multi-hop natural language questions. More recently, Wu et al. (2025) introduce STaRK, a benchmark for LLM-based retrieval over semi-structured knowledge bases that pair textual entity descriptions with relational graph structure. To address this setting, Retrieval-Augmented Generation (RAG) and Graph RAG methods (Peng et al., 2025; Sun et al., 2024; Yasunaga et al., 2021) are employed to retrieve textual and relational information, respectively. When questions require evidence from both modalities simultaneously, however, methods relying on a single retrieval channel fail to capture the full evidence. Lee et al. (2025) address this with HybGRAG, a hybrid agentic retrieval framework that jointly queries text documents and graph relations.

Despite this progress, two critical issues remain unaddressed in existing work. The first is the inability to simultaneously enforce *precise logical constraints* and capture *nuanced textual semantics* within a single query. Consider the e-commerce query example: "black or white tennis racket that is not Wilson and very lightweight." It illustrates two common phenomena in real-world queries. First, semantic flexibility: the attribute "very lightweight"

Method	Logical	Semantic	Incomplet.
BM25	×	✓	×
ANCE	×	✓	×
ToG	×	✓	×
HybGRAG	×	✓	×
BetaE	✓	×	✓
GammaE	✓	×	✓
HybridQE	✓	✓	✓

Table 1: Qualitative comparison of baseline methods for different types of retrieval and query answering tasks.

conveys a nuanced, context-dependent concept that cannot be directly mapped to a discrete symbolic attribute. Second, logical ambiguity: the query allows multiple different interpretations, such as "very lightweight tennis rackets that are not Wilson" or "normal or even heavy-weighted tennis rackets that are not Wilson". Therefore, a hybrid query framework consisting of symbolic logical constraints and a textual description of the searched result is essential to bridge this gap. Under such a framework, the above query can be reformulated as, $\exists \text{color}.(\{\text{black}\} \sqcup \{\text{white}\}) \sqcap \neg \exists \text{brand}(\{\text{Wilson}\}) \sqcap \{\text{"lightweight tennis racket"}\}$, preserving both logical precision and semantic flexibility.

The second is the failure to account for *missing symbolic data*. Existing retrieval frameworks assume complete and reliable symbolic descriptions for the graph entities, whereas real-world annotations are often sparse, noisy, or outdated due to the limits of manual curation (Xu et al., 2019). As shown in Fig. 1, answering queries over text-labeled graphs frequently requires inferring the missing attributes from the item’s descriptions and relational structures. Recent work on query-embedding (QE) methods (Ren and Leskovec, 2020; Yang et al., 2022) map symbolic knowledge and queries into a vector space, but only assume symbolic queries. LLMs are employed to complete such missing information (Xu et al., 2024), however their effectiveness is limited in domains that are uncovered by the training corpus of the LLMs.

Our contributions are as follows: We formalize a hybrid query language with logical operators over both symbolic and textual modalities (§2); construct two new benchmark datasets satisfying incompleteness assumptions on text-labeled graphs (§3); propose HybridQE, a hybrid query embedding framework that jointly reasons over hybrid queries under incompleteness (§4, Table 1); and

present extensive experiments addressing four research questions on the effectiveness, robustness, and adaptability of HybridQE (§5).

2 Problem Statement

In this paper we study the hybrid query answering problem, which is an extension of the query answering problem over incomplete knowledge graphs (He et al., 2024) to include text descriptions in the data and in the queries. To formalize this problem, we next define how we extend knowledge graphs and how we extend the query language.

2.1 Text-labeled Knowledge Graphs

Knowledge graph query answering has been studied over structures $G = (V, R, E)$ where V and R are two finite disjoint sets of *nodes* and *relation names*, and $E \subseteq V \times R \times V$ is a set of labeled edges. *Text-labeled knowledge graphs* extend these structures with textual information for nodes and edges. For this extension, we assume the existence of a set T , disjoint from V and R containing all possible textual descriptions.

Definition 1 (Text-labeled knowledge graph). A text-labeled knowledge graph G is a tuple (V, R, E, T, τ) , where (V, R, E) is a knowledge graph, and $\tau : V \cup R \rightarrow T$ is a function mapping nodes and relation names to text.

2.2 Hybrid Query Language

We refer to the QA problem over text-labeled knowledge graphs as the hybrid query answering problem (HybridQA). The name *hybrid* comes from the combination of text and structure. To formally define this problem, we need first to define the query language. To this end we build on top of the $SRQI^-$ description logics formulation of query answering by He et al. (He et al., 2024).¹

Definition 2 (Hybrid queries). Hybrid queries Q over text-labeled knowledge graphs are defined by the following grammar:

$$\begin{aligned} Q &::= \top \mid \{v\} \mid \{t\} \mid \neg Q \mid Q \sqcap Q \mid \exists R.Q, \\ R &::= r \mid r^- \mid t \mid t^-, \end{aligned}$$

where $\top$ is the universal concept, t , v , and r stand for elements in sets T , V , and R , respectively.

The semantics are defined as usual for all elements, except expressions of the form $\{t\}$, called *text nominals*, which match entities whose textual

¹See Appendix A for the $SRQI^-$ syntax and semantics.

description is semantically similar to t , and expressions of the form $\exists t.Q$ and $\exists t^-.Q$, called *textual relational existentials*, where t is interpreted as a relation name. Standard node nominals $\{v\}$ match exactly the individual node $v \in V$.

2.3 Hybrid QA under Incompleteness

At inference time, the input of the HybridQA problem consists of a hybrid query Q and an incomplete text-labeled knowledge graph $G_{\text{train}} = (V, R, E_{\text{train}}, T, \tau)$. A HybridQA model produces a score for every node $v \in V$, representing how likely v is to satisfy the query Q .

Following the standard query answering setting over incomplete knowledge graphs, we assume that the observed graph G_{train} is a subgraph of a complete graph $G_{\text{test}} = (V, R, E_{\text{train}} \cup E_{\text{missing}}, T, \tau)$, where the set E_{missing} contains edges that are not available during training, and $E_{\text{missing}} \cap E_{\text{train}} = \emptyset$.

For a query Q , let $Q^{G_{\text{train}}}$ and $Q^{G_{\text{test}}}$ denote the sets of answers obtained by evaluating the query over the training graph and the complete graph, respectively. We distinguish two answer sets: the *easy answer set* $A_{\text{easy}}(Q) = Q^{G_{\text{train}}}$, containing answers directly retrievable from the observed graph, and the *hard answer set* $A_{\text{hard}}(Q) = Q^{G_{\text{complete}}} \setminus Q^{G_{\text{train}}}$, containing answers that require inference over missing edges. Our evaluation focuses on $A_{\text{hard}}(Q)$, as these answers test the model’s ability to reason under incompleteness.

Given a hybrid query Q , the goal of HybridQA is to rank entities such that nodes in $A_{\text{hard}}(Q)$ receive higher scores than non-answers.

3 HybridQA Benchmarks

We introduce two benchmarks for the HybridQA problem, ProdHybridQA and BioHybridQA, which are derived from real-world e-commerce and protein-disease data (Zhang et al., 2022c; Xu et al., 2019; Wu et al., 2025). The existing text-labeled graph QA datasets are not suitable to evaluate the HybridQA problem. Indeed, some (Wu et al., 2025) assume completeness and support only textual queries, while symbolic QA (Ren et al., 2019; Ren and Leskovec, 2020; Gregucci et al., 2025) benchmarks do not include text descriptions in the graph. Our new datasets introduce two key innovations: incomplete text-labeled knowledge graphs and hybrid queries for these graphs.

Hybrid Query Formulation. Our proposed dataset queries adhere to established query struc-

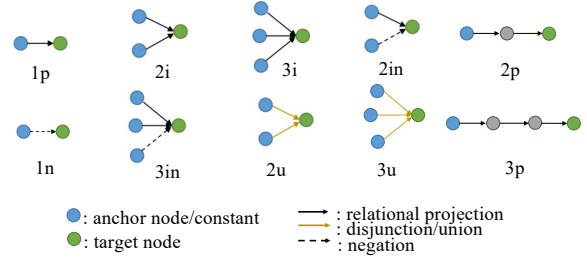


Figure 2: Query structures of our datasets queries. Anchor nodes (in cyan) and relations names are to be specified to instantiate the queries. Each structure is named using abbreviations: “p” for projection, “i” for intersection, “u” for union, and “n” for negation.

Table 2: Example queries in ProdHybridQA

Type	Hybrid Query Example
1p	“Could you find a body deodorant made in powder form that has a gentle floral fragrance?” $\sqcap$ $\exists \text{brand}.\{\text{HoneybeeGardens}\}$
2i	“Could you provide a battery that is in black color?” $\sqcap$ $\exists \text{binding}.\{\text{ConsElect}\}$ $\sqcap$ $\exists \text{brand}.\{\text{Neewer}\}$
3i	“Could you find a candle in a jar container that has a single wick?” $\sqcap$ $\exists \text{category}.\{\text{candle}\}$ $\sqcap$ $\exists \text{scnt}.\{\text{SweetPea}\}$ $\sqcap$ $\exists \text{binding}.\{\text{miscellaneous}\}$
1n	“Could you find breakfast cereal produced by the brand KIND?” $\sqcap$ $\neg \exists \text{flavor}.\{\text{oat}\}$
2in	“Could you find a candle made from a soyblend wax material?” $\sqcap$ $\exists \text{container_type}.\{\text{jar}\}$ $\sqcap$ $\neg \exists \text{manufacturer}.\{\text{MVPGroup}\}$
3in	“Could you provide a cake that comes packaged in a box?” $\sqcap$ $\neg \exists \text{manufacturer}.\{\text{SweetSams}\}$ $\sqcap$ $\exists \text{binding}.\{\text{grocery}\}$ $\sqcap$ $\exists \text{flavor}.\{\text{fruit}\}$
2u	“Could you find a blood pressure monitor that comes in a gray color?” $\sqcap$ $(\exists \text{brand}.\{\text{Beurer}\} \sqcup \exists \text{voltage_units}.\{\text{volts}\})$
3u	“Could you find an oil-based body lubricant with a ylang fragrance that has its shelf life specified in days?” $\sqcap$ $(\exists \text{brand}.\{\text{PipedreamProducts}\} \sqcup \exists \text{brand}.\{\text{FleurDeSpa}\} \sqcup \exists \text{brand}.\{\text{OPO}\})$

tures commonly used in symbolic QA tasks, i.e. 1p, 2p, 3p, 2i, 3i, 1n, 2in, 3in, 2u, and 3u.² Figure 2 illustrates the structures of these symbolic queries. We make a conjunction of each of these queries Q with a text nominal $\{t\}$ to create a hybrid query of the form $Q \sqcap \{t\}$. Table 2 shows examples queries for the ProdHybridQA dataset.

Incompleteness Conditions Our datasets introduce two complementary forms of incompleteness to ensure that hybrid query answering requires gen-

²Note that ProdHybridQA is built on a hyperedge graph, where product–attribute relationships are encoded as hyperedges rather than binary triples. Multi-hop projection queries (2p, 3p) are not defined for this dataset, since hyperedges do not compose transitively across intermediate nodes.

uine inference rather than simple lookup.

On the symbolic side, to be certain about each test query answer, we require two types of edges, edges that are at hand (i.e., in set E_{train}) and missing edges (i.e., in set E_{missing}). This controlled symbolic incompleteness is aligned with the QA setting. On the text side, real-world textual descriptions have a high variability, noise, and partial information coverage. In our generated queries, we simulated this incompleteness by generating textual descriptions from a randomly sampled subset of triples associated with the target entity. As a result, the query textual descriptions capture fragmentary information, requiring models to infer missing data from both the text descriptions and the incomplete graph structure. We refer the reader to Appendix B for full details on the query generation process and a systematic justification of how the generation procedure fulfills our dataset assumptions. Dataset statistics are provided in Table 5.

4 HybridQE for HybridQA

We next present the rationale, the query embedding design, and the training aspects of HybridQE.

4.1 Rationale

We hypothesize that a method for the HybridQA problem would benefit from encoding the symbolic and the textual parts directly, that is, without transforming one into the other. Translating symbolic queries into natural-language questions would inevitably lead to loss or distortion of the original logical meaning of operators such as conjunction, disjunction, and negation (Yih et al., 2016; Banerjee, 2023). Conversely, converting natural language questions into symbolic queries through semantic parsing also struggles to capture the nuances, often yielding incorrect or incomplete logical forms (Feng and He, 2025; Zhao et al., 2025; Shen et al., 2020).

Additionally, HybridQE is expected to support joint reasoning over hybrid queries in the presence of incompleteness. Missing information in the graph may be recoverable through either the associated textual descriptions or latent structural patterns in the database. We hypothesize that a robust hybrid QA model should therefore enable both text-driven and structure-driven completion, allowing these complementary inference pathways to interact during reasoning.

4.2 Hybrid Query Embedding

Our method, HybridQE, infers query answers by embedding both entities and hybrid queries into a shared vector space $\mathbf{E}^d$ and ranking entities by their distance to the query embedding. Entities positioned closer are predicted as more plausible answers. Text descriptions and relations are embedded to spaces $\mathbf{T}^d$ and $\mathbf{R}^d$, and then mapped to space $\mathbf{E}^d$ when representing queries and entities.³

Entity Embeddings with Query-Conditioned Similarity. We next define how HybridQE embeds entities into the vector space $\mathbf{E}^d$. Unlike prior methods (Yang et al., 2022; Ren and Leskovec, 2020), where an entity e is assigned a single static vector, HybridQE produces a query-conditioned entity embedding $e_Q \in \mathbf{E}^d$ that selectively emphasizes the portions of the entity’s text description most relevant to query Q .

Let $\xi : T \rightarrow \mathbf{T}^d$ denote a pretrained language model that maps text to vectors. For every entity $e \in V$ with text description $t_e = \tau(e)$, we obtain a token-level representation $\mathbf{E} = \xi_{\text{tok}}(t_e) \in \mathbb{R}^{n \times d}$ where n is the number of tokens in t_e and d is the hidden dimension. We also compute a pooled entity representation $\bar{e} = \text{mean-pool}(\mathbf{E}) \in \mathbb{R}^d$. For a hybrid query Q of the form $\{t\} \sqcap \phi$, we define two query-side vectors, $\mathbf{q}_t = \xi(t)$ and $\mathbf{q}_\phi = \xi(t_\phi)$, where t_ϕ is the text description resulting from verbalizing query ϕ into natural language and only serves as an auxiliary attention signal for identifying relevant tokens in entity descriptions.

To identify which tokens of the entity description are most relevant to each query component, we compute attention scores by taking the dot product of each token embedding with the corresponding query vector:

$$\alpha_t = \text{softmax}(\mathbf{E}\mathbf{q}_t) \in \mathbb{R}^n, \quad (1)$$

$$\alpha_\phi = \text{softmax}(\mathbf{E}\mathbf{q}_\phi) \in \mathbb{R}^n. \quad (2)$$

where $\mathbf{E}\mathbf{q}_t$ is the vector of dot-product scores between each of the n token embeddings and the query vector. The softmax normalizes these into a probability distribution over tokens. The resulting query-conditioned context vectors are:

$$\tilde{\mathbf{a}}_t = \mathbf{E}^\top \alpha_t \in \mathbb{R}^d \quad (3)$$

$$\tilde{\mathbf{a}}_\phi = \mathbf{E}^\top \alpha_\phi \in \mathbb{R}^d \quad (4)$$

³We use different notations for the three spaces because our method is model-agnostic. In some models the three spaces are the same, $\mathbb{R}^d$, but this is not generally true.

representing a d-dimensional summary of the entity description weighted by relevance to each query component. To modulate how much each query component should influence the final entity representation, we apply learnable gating functions:

$$\mathbf{g}_t = \sigma(\mathbf{W}_t [\bar{\mathbf{e}}; \tilde{\mathbf{a}}_t]), \quad (5)$$

$$\mathbf{g}_\phi = \sigma(\mathbf{W}_\phi [\bar{\mathbf{e}}; \tilde{\mathbf{a}}_\phi]), \quad (6)$$

where $[\cdot; \cdot]$ denotes concatenation, σ is the sigmoid activation function, $\mathbf{W}_t$ and $\mathbf{W}_\phi$ are learnable weights and the gates $\mathbf{g}_t$ and $\mathbf{g}_\phi$ dynamically adjust the contribution of each modality. The final query-specific entity embedding is produced by combining these gated contributions and projecting them into the shared embedding space via a Multi-Layer Perceptron (MLP) $\mathbf{P}_E : \mathbf{T}^d \rightarrow \mathbf{E}^d$:

$$\mathbf{e}_Q = \mathbf{P}_E(\bar{\mathbf{e}} + \mathbf{g}_t \odot \tilde{\mathbf{a}}_t + \mathbf{g}_\phi \odot \tilde{\mathbf{a}}_\phi), \quad (7)$$

where $\odot$ is the Hadamard product. This design ensures that entity embeddings are not static database artifacts but contextualized representations that reflect both what the query is asking and how the entity’s description relate to it.

Relation Embeddings. We compute the embeddings of relations by mapping the text description of each relation name $r \in R$ and its inverse r^- into the embedding space $\mathbf{R}^d$ by using their textual descriptions:

$$\mathbf{r} = \mathbf{P}_R(\xi(t_r)), \quad \mathbf{r}^- = \mathbf{P}_R(\xi(t_{r^-})), \quad (8)$$

where $\mathbf{P}_R : \mathbf{T}^d \rightarrow \mathbf{R}^d$ is a learnable projection layer, t_r and t_{r^-} are the text descriptions of relation names r and r^- .

Symbolic Query Encoder. On top of the embedding of entities and relations, we define a symbolic query encoder, $\mathbf{Emb}_\phi$. The goal is to preserve the exact logical structure of ϕ while embedding it into the shared vector space $\mathbf{E}^d$ where reasoning can be performed jointly with textual information. Unlike natural-language reformulations of symbolic queries, which risk losing their exact logical operators, our encoder preserves the logical precision by directly modeling the symbolic form using neural operators defined over the embedding space. The symbolic encoder contributes explicit logical inference capabilities inherited from neural query embedding models, which have been shown to reason over unseen patterns and missing edges

in incomplete graphs (He et al., 2024). The signatures of these neural operators are the following: Nominal : $\mathbf{E}^d \rightarrow \mathbf{E}^d$, RelT : $\mathbf{E}^d \times \mathbf{R}^d \rightarrow \mathbf{E}^d$, Intersect : $\mathbf{E}^d \times \dots \times \mathbf{E}^d \rightarrow \mathbf{E}^d$, Union : $\mathbf{E}^d \times \dots \times \mathbf{E}^d \rightarrow \mathbf{E}^d$ and Complement : $\mathbf{E}^d \rightarrow \mathbf{E}^d$. These neural operators define recursively the query embedding of a query Q as follows:

$$\mathbf{Emb}_{\{t\}} ::= \text{Nominal}(\mathbf{P}_E(\xi(t))), \quad (9)$$

$$\mathbf{Emb}_{\{e\}} ::= \text{Nominal}(e_Q),$$

$$\mathbf{Emb}_{\exists r, \phi} ::= \text{RelT}_r(\mathbf{Emb}_\phi, \mathbf{r}),$$

$$\mathbf{Emb}_{\phi_1 \sqcap \dots \sqcap \phi_n} ::= \text{Intersect}(\mathbf{Emb}_{\phi_1}, \dots, \mathbf{Emb}_{\phi_n}),$$

$$\mathbf{Emb}_{\phi_1 \sqcup \dots \sqcup \phi_n} ::= \text{Union}(\mathbf{Emb}_{\phi_1}, \dots, \mathbf{Emb}_{\phi_n}),$$

$$\mathbf{Emb}_{\neg \phi} ::= \text{Complement}(\mathbf{Emb}_\phi),$$

HybridQE is model-agnostic with respect to the exact implementation of these operators and can operate with different query embedding backbones. Instantiations for BetaE (Ren and Leskovec, 2020) and GammaE (Yang et al., 2022) are provided in Appendix C.

The layer $\mathbf{P}_E$ projects text descriptions into the same latent space as symbolic queries with the aim to supports bidirectional information flow and joint inference across modalities. Indeed, hybrid queries Q of the form $\{t\} \sqcap \phi$ that combine a textual description with a symbolic formula are particular case of the conjunction, $\mathbf{Emb}_Q = \text{Intersect}(\mathbf{Emb}_{\{t\}}, \mathbf{Emb}_\phi)$. We hypothesize that this design keeps both modalities isolated during encoding to preserve the semantic precision and flexibility in hybrid query, while the shared embedding space enables the two modalities to exchange information, allowing textual evidence to fill missing symbolic relations and symbolic constraints to filter irrelevant textual information. As a result, HybridQE preserves the exact logical structure of the symbolic query while extending neural reasoning to free-text semantics and incomplete data.

4.3 Model Training

Learning Objective. Given a hybrid query Q and a text-labeled graph G , the goal of query embedding approaches is to maximize the predictions of *positive answers* to query Q (i.e., answer entity a that are in Q^G and minimize the prediction of *negative answers* to query Q (i.e., entities b that are in $\neg Q^G$). In the representation space, the evaluation of how likely an entity is an answer to a query is computed with a distance function with signature $\text{Dist} : \mathbf{E}^d \times \mathbf{E}^d \rightarrow \mathbb{R}$. Given an entity e and a query Q , and number γ denoting a margin hyperparameter, we abbreviate

Table 3: The MRR performance of HybridQE and baseline models on ProdHybridQA and BioHybridQA benchmark datasets. The results of our methods are highlighted.

Dataset	Model	Avg	1p	2i	3i	2u	3u	1n	2in	3in	2p	3p
ProdHybridQA	BM25	9.23	9.45	12.74	11.96	13.32	11.6	6.99	4.55	3.23	-	-
	ANCE	6.49	6.98	10.32	7.88	8.09	6.74	5.03	3.87	3.07	-	-
	QA-GNN	4.49	2.34	7.49	6.23	8.14	5.92	2.35	1.34	2.13	-	-
	HybGRAG	7.73	8.95	10.02	15.28	11.63	9.84	0.84	1.49	3.85	-	-
	ToG	5.33	3.31	5.54	10.29	8.39	10.02	0.53	2.81	1.77	-	-
	PullNet	4.08	5.31	6.46	7.24	4.83	4.37	1.15	1.43	1.81	-	-
	QE+LM (modular method)	0.24	0.20	0.33	0.32	0.53	0.58	0.0001	0.002	0.005	-	-
	HybridQE (BetaE)	11.18	10.93	14.40	21.75	13.36	13.33	1.52	3.18	6.68	-	-
	HybridQE (GammaE)	14.65	15.9	20.15	28.61	17.94	15.05	4.05	4.57	8.94	-	-
BioHybridQA	BM25	6.25	8.64	9.40	10.87	5.56	3.72	5.29	5.25	4.05	6.04	3.74
	ANCE	5.48	6.75	8.28	9.01	4.84	3.83	5.35	4.24	3.74	5.58	3.25
	QA-GNN	3.51	3.39	4.24	5.18	3.23	3.57	4.93	1.85	2.86	3.27	2.56
	HybGRAG	4.55	7.96	10.29	11.92	4.84	3.87	0.51	0.83	0.52	1.45	3.35
	ToG	0.74	1.23	0.27	0.78	1.02	2.06	0.59	0.47	0.49	0.48	0.05
	PullNet	2.90	4.17	5.02	5.73	3.36	2.74	0.73	0.91	1.16	3.13	2.01
	QE+LM (modular method)	0.37	0.50	0.67	0.59	0.43	0.75	0.003	0.05	0.05	0.31	0.42
	HybridQE (BetaE)	5.76	6.02	11.48	13.75	6.08	4.85	1.62	2.96	4.68	2.96	3.19
	HybridQE (GammaE)	6.38	9.37	11.49	16.31	6.15	4.11	1.58	2.79	4.58	3.55	3.84

$\log \sigma(\gamma - \text{Dist}(\mathbf{Emb}_Q, \mathbf{Emb}_{\{b\}}))$ as Δ_Q^e . Then, given a query Q the goal of query embedding is to minimize the following loss:

$$\text{Loss}(Q) ::= \sum_{a \in Q^G} \left(\frac{1}{|N_a|} \sum_{b \in N_a} \Delta_Q^b - \Delta_Q^a \right). \quad (10)$$

where $N_a \subseteq -Q^G$ is a set of negative samples for answer a regarding query Q .

Integration of Graph Information. Symbolic query embedding methods are trained on diverse queries from a training graph (a subset of the test graph), enabling strong generalization to unseen queries during inference. This stems from query diversity, which helps learning logical connectives, and structural integration via 1p queries. However, our datasets use hybrid 1p queries, which do not fully encode complete graph structure. To further improve HybridQE, we augment the training data by incorporating the entire training graph with hybrid 1p queries with empty text query component.

5 Experiments

In this section, we study the following research questions with experiments and analyses: **RQ1:** How well does HybridQE perform on predicting answers for hybrid queries over incomplete text-labeled graph databases? **RQ2:** Does hybrid query formulation improve reasoning over single-modality queries? **RQ3:** How robust is HybridQE when integrated with different query embedding methods? **RQ4:** How does the symbolic-to-text

query ratio affect HybridQE’s performance? Experimental details are supplemented in Appendix F.

5.1 Experimental Setup

Evaluation Metrics. We evaluate HybridQE on the task of hybrid query answering under incomplete data using the ProdHybridQA and BioHybridQA datasets (see Section 3), and compare the performance against baseline methods using the retrieval metrics MRR / Hits@1 / Hits@3.

Baselines. Since no prior work directly targets hybrid query answering over incomplete text-labeled graphs, we evaluate HybridQE against a collection of adapted baselines. These include LLM-enhanced graph RAG methods (ToG (Sun et al., 2024), HybGRAG (Lee et al., 2025)), classic and neural text retrieval systems (BM25 (Robertson and Zaragoza, 2009), ANCE (Xiong et al., 2021), QA-GNN (Yasunaga et al., 2021)), QA method for semi-structured data (PullNet (Sun et al., 2019)) and a modular QE+LM approach that infers answers by combining symbolic query embedding predictions with LLM-based textual inference results. The specific introduction and implementation details on baselines are provided in Appendix F.2.

5.2 Analysis of Research Questions

Effectiveness of HybridQE (RQ1). Table 3 summarizes the MRR performance of HybridQE⁴, instantiated with different query embedding backbones, alongside adapted baselines on both benchmarks. Among the baselines, BM25 achieves the

⁴Additional metrics are reported in Appendix I

Table 4: MRR Performance of HybridQE(GammaE) on Text-Only and Symbolic-Only questions from ProdHybridQA and BioHybridQA benchmark datasets.

Dataset	1p	2i	3i	2u	3u	1n	2in	3in	2p	3p
ProdHybridQA(Hybrid)	15.9	20.15	28.61	17.94	15.05	4.05	4.57	8.94	-	-
ProdHybridQA(Text Only)	8.18	15.83	16.57	13.63	11.35	2.34	1.35	4.02	-	-
ProdHybridQA(Symbolic Only)	10.29	10.61	14.42	11.58	8.64	2.73	2.54	7.25	-	-
BioHybridQA(Hybrid)	9.37	11.49	16.31	6.15	4.11	1.58	2.79	4.58	3.55	3.84
BioHybridQA(Text Only)	5.92	7.32	9.27	5.21	2.85	0.59	0.94	0.98	0.75	1.73
BioHybridQA(Symbolic Only)	6.38	5.49	7.48	2.35	2.46	0.32	1.37	0.96	1.32	1.34

strongest overall performance by exploiting lexical overlap between textual queries and entity descriptions, but it cannot enforce logical constraints or reason over missing edges, limiting its effectiveness on structurally complex queries such as 2i and 3i. ANCE shows similar limitations despite using learned dense representations. QA-GNN’s graph propagation is severely hampered by missing edges, while PullNet’s iterative subgraph expansion terminates prematurely under incompleteness and applies symbolic constraints only as post-hoc filters, confirming that logical reasoning must be integrated during retrieval. LLM-based baselines (ToG, HybGRAG) are limited by incomplete graph structure, which disrupts multi-hop reasoning chains, and by the inevitable loss of logical precision when hybrid queries are translated into natural language. Moreover, as shown in Appendix H, these methods exhibit high sensitivity to backbone choice with no consistent winner across query types. Modular method similarly fails, achieving only 0.24 average MRR on ProdHybridQA, nearly 60 times lower than HybridQE(GammaE). Since QE+LM uses the same text encoder as HybridQE, this gap directly quantifies the value of joint hybrid reasoning over post-hoc score fusion.

HybridQE outperforms all baselines on average across both benchmarks, with HybridQE(GammaE) achieving 58.7% and 40.1% relative improvements over BM25 on ProdHybridQA and BioHybridQA respectively. Gains are most pronounced on structurally complex queries (2i, 3i), where cross-modal interaction provides the greatest benefit. Negation performance remains lower, a known limitation of neural complement operators (Ren et al., 2023), though HybridQE(GammaE) still matches or exceeds the best baseline on most negation types. We provide further case analysis and error analysis in Appendices L and K. The analysis reveals that each query family is dominated by a distinct failure mode, with negation failure accounting for up to 78.8% of errors in negation queries, partial symbolic

matching dominating intersection queries, and error compounding growing with multi-hop depth. Appendix K also identifies four concrete directions for future improvement. Beyond the transductive setting, Appendix M demonstrates robust few-shot performance (17.36 average MRR for frequency-1 entities), highlighting HybridQE’s potential for data-efficient and inductive hybrid reasoning.

Hybrid Query Formulation Impact (RQ2). To isolate the effect of hybrid query formulation, we compare the performance of HybridQE under three alternative representations of each test query: (i) **Text-Only** reformulates both the symbolic constraints and the textual condition into a single natural-language question, (ii) **Symbolic-Only** translates the entire query into a purely symbolic representation,⁵ and (iii) **Hybrid** adopts our proposed formulation, which retains symbolic and textual constraints as distinct yet jointly processed inputs. The results are summarized in Table 4. We observe that HybridQE consistently achieves superior performance when queries are expressed in hybrid form compared to single-modal variants. This trend supports our hypothesis that jointly modeling symbolic and textual constraints is effective for querying semi-structured data.

HybridQE integrated with different query embeddings (RQ3). HybridQE is built upon query embedding techniques and is designed to maintain performance across backbone models. To verify its transferability across models, we evaluate it using BetaE (Ren and Leskovec, 2020) and GammaE (Yang et al., 2022), two query embedding methods capable of modeling $SRQI^-$ description concepts. Table 3 presents the performance of our method with BetaE and GammaE on two benchmark datasets. The results demonstrate that both models effectively support HybridQE in hybrid query answering across different graphs, confirming its robustness. HybridQE (GammaE) outperforms HybridQE (BetaE), as GammaE exhibits

⁵The translation details are provided in Appendix G

superior reasoning capabilities, indicating that our method inherits and extends the reasoning ability of backbone models for hybrid query answering.

Influence of Symbolic-to-Text Query Ratio (RQ4). Hybrid queries differ in their composition, containing varying proportions of symbolic constraints and textual descriptions. A robust hybrid query answering approach should therefore adapt effectively to different levels of symbolic-textual integration. To assess this property, we analyze the performance of HybridQE under varying text-to-symbolic condition ratios. Specifically, we partition test queries into five groups based on the proportion of textual conditions: 0–20%, 20–40%, 40–60%, 60–80%, and 80–100%. 0% corresponds to purely symbolic queries, while a ratio of 100% corresponds to purely textual queries. As illustrated in Figure 3, HybridQE instantiated with GammaE exhibits stable and competitive performance across all composition ranges, demonstrating its robustness to varying degrees of symbolic and textual integration.

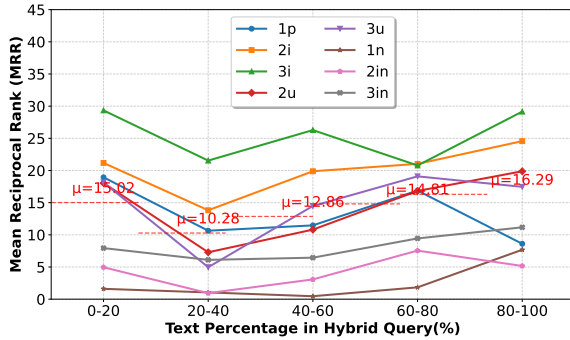


Figure 3: Performance of HybridQE (Gamma) on hybrid queries with varying text-to-symbolic ratio. μ represents the mean performance across different query types at each ratio.

6 Ablation Studies and Analysis

To assess the effectiveness of the components in HybridQE, we perform extensive ablation studies by considering several variant of HybridQE. We list HybridQE and its variants below: QCS with graph (our method): query-conditioned similarity with integration of graph information; QCS w/o graph: query-conditioned similarity only; MLP with graph: MLP network replacing query-conditioned similarity, retaining graph structure; and MLP w/o graph: MLP only. Figure 4 presents HybridQE’s performance for its variants. Firstly, the results show that incorporating graph structural

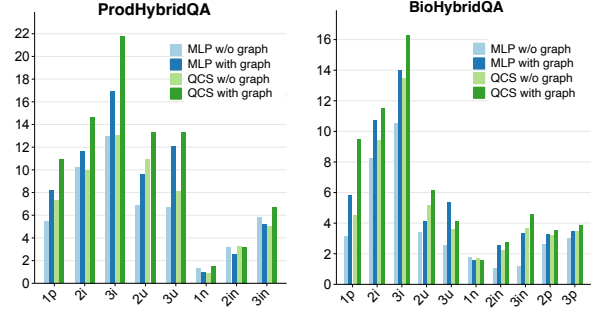


Figure 4: Ablation analysis on the performance of HybridQE (BetaE) with respect to model components.

information improves the performance, as graph-based variants consistently outperform those without. Secondly, the cross-attention mechanism further enhances performance across nearly all query types, further validating the effectiveness of our method’s components in hybrid query reasoning.

7 Related Work

Single-modality retrieval and QA. Early retrieval methods target individual modalities in isolation. Text-based QA systems (Karpukhin et al., 2020; Lee et al., 2019) answer natural language questions over unstructured passages, while query embedding (QE) methods (Ren et al., 2019; Ren and Leskovec, 2020; Yang et al., 2022; He et al., 2024, 2025) embed logical queries into a vector space for reasoning over incomplete knowledge graphs. The two families make complementary assumptions. Text retrieval treats the corpus as the only evidence and has no notion of relational structure or logical operators, whereas query embedding assumes that all query-relevant knowledge is expressible as symbolic triples and discards the textual descriptions attached to entities.

Natural language QA over knowledge graphs.

Work on answering free-form questions against a knowledge graph falls into three families. Semantic parsing translates a question into an executable logical form (Berant et al., 2013; Gu et al., 2021; Cao et al., 2022). Retrieval-based methods skip the logical form and rank answers within a question-specific subgraph (Saxena et al., 2020; He et al., 2021; Zhang et al., 2022a), and LLM-driven methods prompt the model to traverse or verbalize the graph (Luo et al., 2024; Jiang et al., 2023), with Generate-on-Graph (Xu et al., 2024) additionally letting the model supply facts the graph lacks. All three assume the query arrives as natural language, so operators such as negation and disjunction must

survive either verbalization or a parse into schema predicates, neither of which admits a constraint like *very lightweight*. Retrieval- and embedding-based variants tolerate some missing facts, but the logical structure of the question remains implicit in a learned scoring function rather than enforced.

QA over semi-structured knowledge bases.

Semi-structured knowledge bases pair relational structure with entity documents. Representation learning on such text-attributed graphs couples a language model with a graph encoder (Yasunaga et al., 2021; Zhang et al., 2022b), yielding one static representation per node that cannot emphasize the portion of a long description a particular constraint depends on. STaRK (Wu et al., 2025) benchmarks retrieval over these bases, while Graph RAG (Peng et al., 2025; Edge et al., 2024) and HybGRAG (Lee et al., 2025) combine text and relational channels through routing or fusion decisions made outside the encoder, a design our modular QE+LM baseline isolates. A parallel line answers questions over tables, text, and mixed sources (Chen et al., 2020, 2021; Zhu et al., 2021a; Oğuz et al., 2022; Ma et al., 2022; Pramanik et al., 2024) by verbalizing structured evidence into passages, which sacrifices the exact semantics of its operators. However, none of these methods support formal logical operators over free-form textual descriptions, nor account for symbolic incompleteness, the two gaps our work addresses.

8 Conclusion

We introduce hybrid query answering over incomplete text-labeled graphs, a setting where logical precision and semantic flexibility can be jointly satisfied under missing data. Our hybrid query language, two new benchmarks, and HybridQE framework together demonstrate that unified symbolic-textual reasoning is both necessary and achievable, opening a concrete path toward expressive querying over the incomplete, semi-structured knowledge bases that dominate real-world applications.

Limitations

While our work demonstrates the effectiveness of hybrid query formulation and the proposed HybridQE framework, several limitations remain. First, our main experiments focus on the transductive setting, where entities and relations observed at test time appear during training; although we

provide preliminary few-shot results, fully inductive hybrid query answering remains an open challenge. Second, HybridQE relies on the quality of pretrained text encoders to capture semantic information, which may limit performance in highly specialized domains or under noisy textual descriptions. Third, as observed in our experiments, modeling negation and complex logical operators continues to be challenging for neural query embedding methods, and further improvements in logical reasoning are needed.

Acknowledgments

The authors thank the International Max Planck Research School for Intelligent Systems for supporting Yunjie He, Yi Wang and Yuqicheng Zhu. This work was funded by the German Research Foundation (DFG) – SFB 1574 – 471687386 and partially supported by the EU project SMARTY (GA 101140087).

References

- Debayan Banerjee. 2023. [Semantic parsing for knowledge graph question answering with large language models](#). In *The Semantic Web: ESWC 2023 Satellite Events: Hersonissos, Crete, Greece, May 28 - June 1, 2023, Proceedings*, page 234–243, Berlin, Heidelberg. Springer-Verlag.
- Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. [Semantic parsing on Freebase from question-answer pairs](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544. Association for Computational Linguistics.
- Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyiu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. 2022. [KQA Pro: A dataset with explicit compositional programs for complex question answering over knowledge base](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 6101–6119. Association for Computational Linguistics.
- Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. 2021. [Open question answering over tables and text](#). In *International Conference on Learning Representations (ICLR)*.
- Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. [HybridQA: A dataset of multi-hop question answering over tabular and textual data](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036. Association for Computational Linguistics.

- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. [From local to global: A graph RAG approach to query-focused summarization](#). *CoRR*, abs/2404.16130.
- Tengfei Feng and Liang He. 2025. [RGR-KBQA: Generating logical forms for question answering using knowledge-graph-enhanced large language model](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3057–3070, Abu Dhabi, UAE. Association for Computational Linguistics.
- Cosimo Gregucci, Bo Xiong, Daniel Hernández, Lorenzo Loconte, Pasquale Minervini, Steffen Staab, and Antonio Vergari. 2025. [Is complex query answering really complex?](#) In *ICML, Proceedings of Machine Learning Research*. PMLR / OpenReview.net.
- Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. [Beyond I.I.D.: Three levels of generalization for question answering on knowledge bases](#). In *Proceedings of the Web Conference 2021*, pages 3477–3488. ACM.
- Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. [Improving multi-hop knowledge base question answering by learning intermediate supervision signals](#). In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 553–561. ACM.
- Ruining He and Julian McAuley. 2016. [Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering](#). In *Proceedings of the 25th International Conference on World Wide Web*, WWW ’16, page 507–517, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Yunjie He, Daniel Hernández, Mojtaba Nayyeri, Bo Xiong, Yuqicheng Zhu, Evgeny Kharlamov, and Steffen Staab. 2024. [Generating sroiq ontologies via knowledge graph query embedding learning](#). In *ECAI*, volume 392 of *Frontiers in Artificial Intelligence and Applications*, pages 4279–4286. IOS Press.
- Yunjie He, Bo Xiong, Daniel Hernández, Yuqicheng Zhu, Evgeny Kharlamov, and Steffen Staab. 2025. [Dage: Dag query answering via relational combinator with logical constraints](#). In *Proceedings of the ACM on Web Conference 2025*, WWW ’25, page 2514–2529, New York, NY, USA. Association for Computing Machinery.
- Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. [Struct-GPT: A general framework for large language model to reason over structured data](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251. Association for Computational Linguistics.
- Bowen Jin, Yu Zhang, Yu Meng, and Jiawei Han. 2023. [Edgeformers: Graph-empowered transformers for representation learning on textual-edge networks](#). In *ICLR*. OpenReview.net.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *EMNLP (1)*, pages 6769–6781. Association for Computational Linguistics.
- Haoyu Kuang, Jiarong Xu, Haozhe Zhang, Zuyu Zhao, Qi Zhang, Xuanjing Huang, and Zhongyu Wei. 2023. [Unleashing the power of language models in text-attributed graph](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8429–8441, Singapore. Association for Computational Linguistics.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. [Latent retrieval for weakly supervised open domain question answering](#). In *ACL (1)*, pages 6086–6096. Association for Computational Linguistics.
- Meng-Chieh Lee, Qi Zhu, Costas Mavromatis, Zhen Han, Soji Adeshina, Vassilis N. Ioannidis, Huzefa Rangwala, and Christos Faloutsos. 2025. [Hybgrag: Hybrid retrieval-augmented generation on textual and relational knowledge bases](#). In *ACL (1)*, pages 879–893. Association for Computational Linguistics.
- 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 (ICLR)*.
- Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. 2022. [Open-domain question answering via chain of reasoning over heterogeneous knowledge](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5360–5374. Association for Computational Linguistics.
- Barlas Oğuz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2022. [UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546. Association for Computational Linguistics.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. [Pytorch: An imperative style, high-performance deep learning library](#). In *NeurIPS*, pages 8024–8035.
- Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang

- Tang. 2025. [Graph retrieval-augmented generation: A survey](#). *ACM Trans. Inf. Syst.*, 44(2).
- Soumajit Pramanik, Jesujoba Alabi, Rishiraj Saha Roy, and Gerhard Weikum. 2024. [UNIQORN: Unified question answering over RDF knowledge graphs and natural language text](#). *Journal of Web Semantics*.
- Hongyu Ren, Mikhail Galkin, Michael Cochez, Zhaocheng Zhu, and Jure Leskovec. 2023. [Neural graph reasoning: Complex logical query answering meets graph databases](#). *Preprint*, arXiv:2303.14617.
- Hongyu Ren, Weihua Hu, and Jure Leskovec. 2019. [Query2box: Reasoning over knowledge graphs in vector space using box embeddings](#). In *ICLR*.
- Hongyu Ren and Jure Leskovec. 2020. [Beta embeddings for multi-hop logical reasoning in knowledge graphs](#). *NeurIPS*, 33:19716–19726.
- Stephen Robertson and Hugo Zaragoza. 2009. [The probabilistic relevance framework: Bm25 and beyond](#). *Found. Trends Inf. Retr.*, 3(4):333–389.
- Apoorv Saxena, Aditya Tripathi, and Partha Talukdar. 2020. [Improving multi-hop question answering over knowledge graphs using knowledge base embeddings](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507. Association for Computational Linguistics.
- Tao Shen, Xiubo Geng, Guodong Long, Jing Jiang, Chengqi Zhang, and Daxin Jiang. 2020. [Effective search of logical forms for weakly supervised knowledge-based question answering](#). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2227–2233. International Joint Conferences on Artificial Intelligence Organization. Main track.
- Haitian Sun, Tania Bedrax-Weiss, and William W. Cohen. 2019. [Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2380–2390. Association for Computational Linguistics.
- Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. 2018. [Open domain question answering using early fusion of knowledge bases and text](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4231–4242. Association for Computational Linguistics.
- 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 *ICLR*. OpenReview.net.
- Shirley Wu, Shiyu Zhao, Michihiro Yasunaga, Kexin Huang, Kaidi Cao, Qian Huang, Vassilis N. Ioannidis, Karthik Suresh, James Zou, and Jure Leskovec. 2025. [Stark: benchmarking llm retrieval on textual and relational knowledge bases](#). In *Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS '24*, Red Hook, NY, USA. Curran Associates Inc.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. [Approximate nearest neighbor negative contrastive learning for dense text retrieval](#). In *ICLR*. OpenReview.net.
- Huimin Xu, Wenting Wang, Xin Mao, Xinyu Jiang, and Man Lan. 2019. [Scaling up open tagging from tens to thousands: Comprehension empowered attribute value extraction from product title](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5214–5223, Florence, Italy. Association for Computational Linguistics.
- 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, Miami, Florida, USA. Association for Computational Linguistics.
- Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, Weiwei Deng, Qi Zhang, Lichao Sun, Xing Xie, and Sen Zhang Wang. 2023. [A comprehensive study on text-attributed graphs: Benchmarking and rethinking](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 17238–17264. Curran Associates, Inc.
- Dong Yang, Peijun Qing, Yang Li, Haonan Lu, and Xiaodong Lin. 2022. [Gammae: Gamma embeddings for logical queries on knowledge graphs](#). *EMNLP*.
- 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 *NAACL-HLT*, pages 535–546. Association for Computational Linguistics.
- Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. [The value of semantic parse labeling for knowledge base question answering](#). In *ACL (2)*. The Association for Computer Linguistics.
- Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022a. [Subgraph retrieval enhanced model for multi-hop knowledge base question answering](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational*

Linguistics, pages 5773–5784. Association for Computational Linguistics.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. 2022b. [GreaseLM: Graph REASONing enhanced language models for question answering](#). In *International Conference on Learning Representations (ICLR)*.

Xinyang Zhang, Chenwei Zhang, Xian Li, Xin Luna Dong, Jingbo Shang, Christos Faloutsos, and Jiawei Han. 2022c. [Oa-mine: Open-world attribute mining for e-commerce products with weak supervision](#). In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 3153–3161, New York, NY, USA. Association for Computing Machinery.

Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. 2023. [Learning on large-scale text-attributed graphs via variational inference](#). In *ICLR*. OpenReview.net.

Ruilin Zhao, Feng Zhao, and Hong Zhang. 2025. [Correcting on graph: Faithful semantic parsing over knowledge graphs with large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5364–5376, Vienna, Austria. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021a. [TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 3277–3287. Association for Computational Linguistics.

Jason Zhu, Yanling Cui, Yuming Liu, Hao Sun, Xue Li, Markus Pelger, Tianqi Yang, Liangjie Zhang, Ruofei Zhang, and Huasha Zhao. 2021b. [Textgnn: Improving text encoder via graph neural network in sponsored search](#). In *WWW*, pages 2848–2857. ACM / IW3C2.

A Syntax and Semantics of $SRQI^-$ concept

Definition 3 (Syntax of $SRQI^-$ concept descriptions). *Given a knowledge graph $G = (V, R, E)$, $SRQI^-$ concept descriptions C are generated by the grammar:*

$$C ::= \top \mid \{a\} \mid \neg C \mid C \sqcap C \mid \exists r.C \mid \exists r^-.C,$$

where $\top$ is the universal concept, $a \in V$, and $r \in R$. Nominals $\{a\}$ denote singleton sets. As abbreviations, we write $\perp = \neg\top$, $C \sqcup D = \neg(\neg C \sqcap \neg D)$, and $\forall r.C = \neg\exists r.\neg C$.

Definition 4 (Semantics of $SRQI^-$ concept). *The interpretation of C on graph G is defined recursively as:*

1. $\top^G = V$,
2. $\{a\}^G = \{a\}$,
3. $\neg C^G = V \setminus C^G$,
4. $C \sqcap D^G = C^G \cap D^G$,
5. $\exists r.C^G = \{u \mid \exists (u, r, v) \in E, v \in C^G\}$,
6. $\exists r^-.C^G = \{u \mid \exists (v, r, u) \in E, v \in C^G\}$.

For example, the concept $\exists \text{type.Coffee}$ retrieves all items of type coffee (i.e., a, b), while $\exists \text{flavor.Roast} \sqcap \exists \text{brand}.\neg \text{Starbucks}$ retrieves roast-flavored items not branded Starbucks (i.e., only a).

B Generation of Datasets

In this paper, we construct two new benchmark datasets, BioHybridQA and ProdHybridQA, for the hybrid query answering task over incomplete text-labeled graph. This section provides a detailed description of the procedure used to construct hybrid queries for our benchmark datasets. The goal of the generation process is to produce queries that (i) contain both symbolic and textual components that enforce logical constraints and capture semantic nuances respectively, and (ii) require reasoning under symbolic incompleteness.

B.1 Main Algorithm

In Algorithm 1, we present the psudo code for the generation of hybrid query answering datasets.⁶ Table 5 summarizes the statistics on the generated queries.

B.2 Fulfilling Dataset Assumptions

We now systematically justify that the generation procedure satisfies both dataset assumptions: (i). Queries contain both symbolic and textual components that enforce logical constraints and capture semantic nuances respectively. (ii) Queries require reasoning under symbolic incompleteness.

Assumption I. The symbolic component Φ , generated by GENSYMBOLICQUERY, encodes formal logical operators as a valid $SRQI^-$ concept description (He et al., 2024) that cannot be approximated by natural language without precision loss.

⁶Since code is used to generate queries for triple graphs, we adapt it on hyperedge graphs by composing nodes and hyperedge indexes into entities, and transforming hyperedges into triples of the form (hyperedge index, relation, node), where hyperedge indexes = I and nodes = N , thus $V = \{I, N\}$.

Algorithm 1: Hybrid Query Generation

Input:

- Knowledge Graph $G = (V, R, E)$, where:
- V : set of entities
- R : set of relations
- E : set of triples $(h, r, t) \in V \times R \times V$, $E = E_{test}$ where $E_{train} \subset E_{valid} \subset E_{test}$
- Ent2Text, Rel2Text
- query_structure $\in Q_{templates}$ for different query structures

Output: Hybrid query $Q = [\Phi, q]$ where Φ is symbolic query, q is textual query and query answer set A

Initialize: $A_{easy}, A_{hard} \leftarrow \emptyset, \emptyset$ // Answer set
while $A_{hard} = \emptyset$ **do**

Step 1: Entity Sampling

$e_{target} \leftarrow \text{SampleEntity}(V)$ or
 $\text{SampleEntity}(I)$ if hyperedge graph
// Randomly sample an entity as an answer entity

Step 2: Symbolic Query Generation

$\Phi \leftarrow$
 $\text{GenSymbolicQuery}(e_{target}, \text{query_structure}, G)$
// Generate a symbolic query following the query structure

Step 3: Triple Selection for Textual Query

$E_{relevant} \leftarrow \{(h, r, t) \in E_{test} \mid t = e_{target}\}$
 $E_{selected} \leftarrow$
 $\text{SelectRandomSubset}(E_{relevant}, k)$
// Select random k triples involving e_{target}

Step 4: Textual Query Generation

$q_{sym} \leftarrow \text{TriplesToQuery}(E_{selected})$
 $q \leftarrow$
 $\text{QueryToNLQuestion}(q_{sym}, \text{Ent2Text}, \text{Rel2Text})$
// Convert triples to semantically equivalent natural language question about e_{target}

Step5: Retrieve Answers for the Hybrid Query

$A_{easy} \leftarrow \{a \mid G_{train} \models q_{sym}(a) \wedge \Phi(a)\}$
 $A_{hard} \leftarrow \{a \mid G_{test} \models q_{sym}(a) \wedge \Phi(a)\} - A_{easy}$
// Derive the answer set involving with triples missing from train graph

end

Return $Q = [\Phi, q], A_{hard}$ // The hybrid query and the hard answer set

The textual component q is derived from triples *disjoint* from those used to instantiate Φ , ensuring q captures complementary semantic information rather than restating symbolic constraints. The semantic abstraction applied in `QUERYTONLQUESTION` of Step 4 employs large language models (with instructions given below) to ensure that q expresses nuanced, qualitative descriptions, such as inferring “*lightweight*” from a weight attribute value, or “*suitable for beginners*” from a style attribute that cannot be recovered by symbolic lookup or string matching.

Assumption II. Symbolic incompleteness is enforced through the construction of the hard answer set $A_{hard}(Q) = \{a \mid G_{test} \models q_{sym}(a) \wedge \Phi(a)\} \setminus A_{easy}$. By definition, every answer in $A_{hard}(Q)$ satisfies the hybrid query over the complete graph $G_{complete}$ but *not* over the observed training graph G_{train} . The **while** loop in Algorithm 1 guarantees that every generated query has a non-empty hard answer set ($A_{hard} \neq \emptyset$), ensuring that all queries in our benchmarks strictly require inference over missing information. The missing information may be recoverable via two complementary pathways that our framework is designed to exploit. The *structural pathway* leverages latent relational patterns in G_{train} to infer missing edges, the mechanism exploited by symbolic query embedding methods (Ren and Leskovec, 2020; Yang et al., 2022). The *textual pathway* leverages the entity text descriptions $\tau(v)$ to infer missing attributes from semantic evidence, for example, inferring that an entity satisfies a weight constraint from a description mentioning “lightweight design”, even when the weight triple is absent from G_{train} .

Prompt for Semantic Abstraction

Instruction. You are a natural language generation expert. Given a structured attribute description of the form [relation]: [value], rephrase it into a semantically equivalent but lexically varied natural language phrase. Apply semantic abstraction where appropriate:

- *Convert specific measurements to qualitative descriptors, e.g., weight: 275g $\rightarrow$ “a lightweight racket”*
- *Map categorical values to conceptual descriptions, e.g., style_fit: Tennis Beginner $\rightarrow$ “suitable for beginner players”*
- *Express relational attributes implicitly, e.g., brand: HEAD $\rightarrow$ “a HEAD-manufactured product”*

Do NOT alter the meaning, omit information, or introduce facts not present in the input. Output only the rephrased phrase with no explanation.

Input: [relation]: [value]

Output: [rephrased phrase]

B.3 Lexical Overlap Analysis.

A potential concern with triple-derived textual queries is that they may merely paraphrase symbolic graph attributes, reducing the task to surface-level string matching. To assess this, we measure the token-level F1 score between each textual query q and the raw attribute values of its target entity e_{target} in the graph. Specifically, token-level F1 is computed after lowercasing, punctuation removal, and stopword filtering, and is taken as the maximum F1 across all entity attributes to give the most

Table 5: Descriptive statistics for ProdHybridQA (Amazon) and BioHybridQA (Prime). ProdHybridQA is a hyperedge product graph and does not support multi-hop projection queries (2p, 3p), indicated by —.

Category	Statistic	ProdHybridQA		BioHybridQA	
		Value	Detail	Value	Detail
Graph	Product / bio entities	86,905	prod nodes	129,385	bio nodes
	Attribute entities	36,037	attr nodes	—	—
	Total entities	122,942		129,385	
	Relation types	16	attr rels	37	bio rels
	Training edges	424,476		13,064,296	
Splits	Train queries	113,682	8 structures	289,483	10 structures
	Valid queries	23,233	8 structures	58,476	10 structures
	Test queries	23,295	8 structures	58,539	10 structures
Per type		train / valid / test		train / valid / test	
	1p	13,402 / 2,748 / 2,797		26,740 / 5,441 / 5,405	
	2p	—		29,328 / 5,872 / 5,854	
	3p	—		29,526 / 5,914 / 5,906	
	2i	14,605 / 2,959 / 2,969		29,107 / 5,880 / 5,862	
	3i	14,963 / 2,997 / 2,998		29,863 / 5,975 / 5,970	
	2u	14,605 / 2,963 / 2,970		29,107 / 5,891 / 5,875	
	3u	14,963 / 2,997 / 2,999		29,863 / 5,978 / 5,982	
	1n	12,509 / 2,703 / 2,706		27,557 / 5,728 / 5,770	
	2in	13,956 / 2,894 / 2,885		28,752 / 5,835 / 5,921	
	3in	14,679 / 2,972 / 2,971		29,640 / 5,962 / 5,994	

lenient possible overlap estimate. For example, given the textual query $q = \text{“Find a lightweight tennis racket made by HEAD”}$ and entity attributes $\{\text{brand : HEAD, weight : 275g, category : Tennis Racket}\}$, the query tokens after filtering are $\{\text{lightweight, tennis, racket, made, HEAD}\}$ and the attribute tokens for brand are $\{\text{HEAD}\}$, for weight are $\{\text{275g}\}$, and for category are $\{\text{tennis, racket}\}$. The per-attribute F1 scores are therefore 0.33, 0.00, and 0.57 respectively.

Figure 5a shows the distribution of token-level F1 scores between textual query components and raw graph attribute values across both datasets. The distributions are strongly right-skewed, with the bulk of queries concentrated below $F1 = 0.4$, confirming that the majority of generated queries do not directly copy graph attribute values. The average token-level F1 is 0.34 on ProdHybridQA and 0.30 on BioHybridQA, both well below the near-verbatim threshold of 0.8. Critically, only 3.03% and 2.55% of queries in ProdHybridQA and BioHybridQA respectively exceed this threshold, indicating that near-verbatim paraphrase is rare by construction. Figure 5b further confirms that no individual query type is trivially recoverable by string matching. Together, these results confirm that our benchmarks require genuine semantic reasoning and cannot be solved by paraphrase recovery.

C Specific details of neural logical operators within HybridQE

In this section, we supplement the specific details of the operators, i.e., relational transformation, intersection operator, union operator and complement operator, of the query embedding models involved in our experiments.

C.1 BetaE

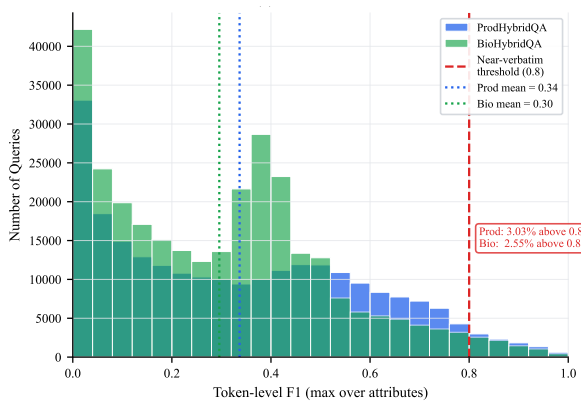
BetaE represents concepts by the Cartesian product of multiple Beta distributions: $\text{Emb}_C = [(\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n)]$ where each component is a Beta distribution $\text{Beta}(\alpha, \beta)$ controlled with two shape parameters α and β .

- **Relational Transformation** maps from one Beta embedding S to another Beta embedding S' given the relation type r . This is modeled by a transformation neural network for each relation type r using a multi-layer perceptron (MLP):

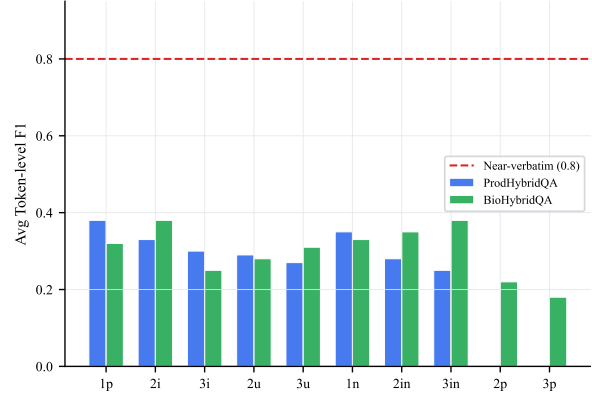
$$\text{RelT}_r(\text{Emb}_C) = \text{MLP}_r(\text{Emb}_C) \quad (11)$$

- **Intersection Operator** is modeled by taking the weighted product of the PDFs of the input Beta embeddings

$$I = \frac{1}{Z} \prod_{i=1}^n p_{\text{Emb}_C}^{w_i} \quad (12)$$



(a) Distribution of F1 scores



(b) Avg F1 per Query Type

Figure 5: Lexical Overlap Analysis between Textual Queries and Graph Attribute Values.

where $I = \text{Intersect}(\text{Emb}_{C_1}, \dots, \text{Emb}_{C_n})$ and Z is a normalization constant and $w_1, \dots, w_n$ are the weights with their sum equal to 1.

- **Complement Operator** is modeled by taking the reciprocal of the shape parameters.

$$C = \left[\left(\frac{1}{\alpha_1}, \frac{1}{\beta_1} \right), \dots, \left(\frac{1}{\alpha_n}, \frac{1}{\beta_n} \right) \right] \quad (13)$$

where $C = \text{Complement}(\text{Emb}_C)$.

- **Union Operator** BetaE does not model the specific union operator because queries involving with union can be translated into disjunctive normal form (DNF), more details can be checked in (Ren and Leskovec, 2020).
- **Distance function** Given an answer entity embedding $\mathbf{a}$ with parameters $[(\alpha_1^a, \beta_1^a), \dots, (\alpha_n^a, \beta_n^a)]$, and a query embedding $\mathbf{q}$ with parameters $[(\alpha_1^q, \beta_1^q), \dots, (\alpha_n^q, \beta_n^q)]$, we define the distance between this entity a and the query q as the sum of KL divergence between the two Beta embeddings along each dimension:

$$\text{dist}_{\text{beta}}(a; q) = \sum_{i=1}^n \text{KL}(p_{\mathbf{a},i}; p_{\mathbf{q},i}) \quad (14)$$

C.2 GammaE

GammaE models concepts by the Cartesian product of multiple Gamma distributions: $\text{Emb}_C = [(\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n)]$ where each component is a Gamma distribution $\Gamma(\alpha, \beta)$ controlled with shape parameter α and rate parameter β .

- **Relational Transformation** maps from one Gamma embedding $\mathbf{S}$ to another Gamma embedding $\mathbf{S}'$ given the relation type r . This is modeled by a transformation neural network for each relation type r using a multi-layer perceptron (MLP):

$$\text{RelT}_r(\text{Emb}_C) = \text{MLP}_r(\text{Emb}_C) \quad (15)$$

- **Intersection Operator** is modeled by taking the weighted product of the PDFs of the input Gamma embeddings

$$I = \frac{1}{Z} \prod p_{\text{Emb}_C}^{w_1} \cdots p_{\text{Emb}_C}^{w_n} \quad (16)$$

where $I = \text{Intersect}(\text{Emb}_{C_1}, \dots, \text{Emb}_{C_n})$ and Z is a normalization constant and $w_1, \dots, w_n$ are the weights with their sum equal to 1.

- **Negation Operator** is modeled by taking the reciprocal of the shape parameters α along with an elasticity parameter ϵ , such that

$$C = [\Gamma(\frac{1}{\alpha_1}, \beta_1) + \epsilon, \dots, \Gamma(\frac{1}{\alpha_n}, \beta_n) + \epsilon] \quad (17)$$

where $C = \text{Complement}(\text{Emb}_C)$.

- **Union Operator** is modeled by a Gamma mixture models, it is calculated as a weighted sum of Gamma distributions, such that

$$U = \sum_{i=1}^n \theta_i \Gamma(\alpha_i, \beta_i) \quad (18)$$

where $U = \text{Union}(\text{Emb}_{C_1}, \dots, \text{Emb}_{C_n})$ and $\theta_i = \frac{\exp(\Gamma(\alpha_i, \beta_i))}{\sum_{j=1}^n \exp(\Gamma(\alpha_j, \beta_j))}$ is the learned

Table 6: BioHybridQA Query Examples

Query Type	Hybrid Query Example
1p	[expression_present(urinary bladder), "Which gene or protein shows expression in both the endometrium and the myometrium?"]
2p	[target(ppi(PRKCG)), "Which drug targets the GABRA1 gene or protein?"]
3p	[target(associated_with(ppi(BCL2L14))), "Which biological process interacts with RPEL1 and is part of the parent-child relationship within the glyoxylate cycle?"]
2i	[interacts_with ⁻ (actin filament polymerization) $\wedge$ expression_present ⁻ (tibial nerve), "Which gene or protein has a protein-protein interaction (PPI) with HTR3A?"]
3i	[expression_present(corpus callosum) $\wedge$ interacts_with(endoplasmic reticulum membrane) $\wedge$ expression_present(esophagus), "Which gene or protein interacts with the apoptotic cleavage of cellular proteins and is associated with CADD5?"]
1n	[$\neg$ interact_with(protein binding), "Which gene or protein shows no expression in the quadriceps femoris?"]
2in	[$\neg$ expression_present(heart left ventricle) $\wedge$ type(gene/protein), "Which gene or protein is expressed in the lymph node?"]
3in	[$\neg$ synergistic_interaction(demeclocycline) $\wedge$ type(drug) $\wedge$ synergistic_interaction(quinidine), "Which drug has a synergistic interaction with magnesium glycinate?"]
2u	[indication(systemic mastocytosis) $\vee$ synergistic_interaction(aldoosterone), "Which drug is associated with the side effect of pulmonary fibrosis?"]
3u	[interacts_with(protein binding) $\vee$ expression_present(temporal lobe) $\vee$ type(gene/protein), "Which gene or protein has a protein-protein interaction with PEX5 and also interacts with the endocytic recycling process?"]

weight for each Gamma density in the Gamma mixture model and also uses the self-attention mechanism.

- **Distance function** Given an answer entity embedding $\mathbf{a}$ with parameters $[(\alpha_1^a, \beta_1^a), \dots, (\alpha_n^a, \beta_n^a)]$, and a query embedding $\mathbf{q}$ with parameters $[(\alpha_1^q, \beta_1^q), \dots, (\alpha_n^q, \beta_n^q)]$, we define the distance between this entity a and the query q as the sum of KL divergence between the two Gamma embeddings along each dimension:

$$\text{dist}_{\text{Gamma}}(a; q) = \sum_{i=1}^n \text{KL}(\Gamma_{\mathbf{a},i}; \Gamma_{\mathbf{q},i}) \quad (19)$$

D Query Types considered in Dataset ProdHybridQA

Figure 6 illustrates the query types considered in the dataset ProdHybridQA. Being different from triple graph which support multi-hop queries, hyperedge graphs do not support 2p, 3p queries.

E Examples of Queries from Dataset BioHybridQA

Table 6 presents the query examples from the dataset BioHybridQA.

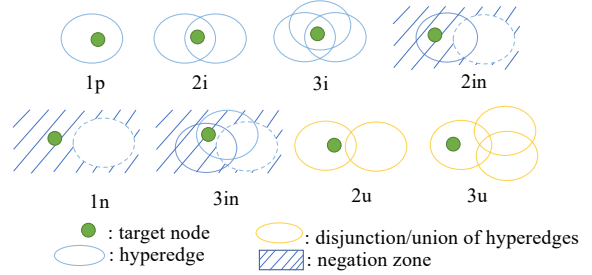


Figure 6: Query structures of the symbolic queries considered in the ProdHybridQA dataset, where anchor entities and relations are to be specified to instantiate logical queries. Each structure is named using abbreviations: "p" for projection of a hyperedge, "i" for intersection, "u" for union, and "n" for negation. For instance, "2i" indicates the conjunction of two atomic queries.

F Experiment Details

F.1 Hyperparameters and Computational Resource

All of our experiments are implemented in Pytorch (Paszke et al., 2019) framework and run on four Nvidia A100 GPU cards. Most LLM-based baselines are implemented with OpenAI API (GPT3.5-turbo for the backbone of ToG and HybGRAG) and Ollama (Deepseek-R1-Distill-Qwen-32B and Deepseek-R1-Distill-Llama-8B). For hyperparameters search, we performed a grid search of learning rates in $\{5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}\}$,

Table 7: The Hits@1 performance of HybridQE and baseline models on ProdHybridQA and BioHybridQA benchmark datasets.

Dataset	Model	1p	2i	3i	2u	3u	1n	2in	3in	2p	3p
ProdHybridQA	BM25	3.33	5.66	6.66	6.33	8.17	3	1	0.67	-	-
	ANCE	2.94	3.52	2.59	4.29	2.62	1.42	0.32	0.23	-	-
	QA-GNN	0.38	2.45	2.27	3.02	2.94	2.01	0.17	0.19	-	-
	HybGRAG	2.73	4.69	7.95	5.24	5.38	0.39	0.94	0.48	-	-
	ToG	1.38	2.19	3.59	3.75	8.03	0.14	0.97	0.54	-	-
	PullNet	2.15	2.83	3.57	1.91	1.73	0.35	0.42	0.56	-	-
	QE+LM (modular method)	0.11	0.21	0.16	0.23	0.25	0	0	0	-	-
	HybridQE (BetaE)	6.57	7.44	10.06	8.24	8.36	2.1	1.16	2.15	-	-
	HybridQE (GammaE)	9.48	12.20	16.21	10.23	13.25	3.59	1.29	2.77	-	-
BioHybridQA	BM25	1.66	3.33	5.00	1.66	2.33	1.66	1.66	1.00	3.00	0.33
	ANCE	2.23	2.11	3.14	1.38	1.72	1.32	0.95	0.94	2.87	0.84
	QA-GNN	1.07	1.94	2.39	1.02	1.94	1.1	0.78	0.78	2.05	0.95
	HybGRAG	2.31	2.92	2.58	1.79	2.94	0.29	0.18	0.17	0.98	1.57
	ToG	0.68	0.11	0.14	0.38	1.01	0.12	0.13	0.09	0.13	0.00
	PullNet	1.52	2.15	2.63	1.26	1.53	0.22	0.35	0.32	1.11	0.62
	QE+LM (modular method)	0.23	0.28	0.21	0.18	0.31	0.00	0.002	0.013	0.12	0.14
	HybridQE (BetaE)	2.68	2.35	7.67	1.93	3.73	1.01	2.18	2.52	1.45	1.22
	HybridQE (GammaE)	3.63	5.01	7.94	1.71	2.71	0.92	1.75	2.51	1.01	1.07

the batch size in $\{256, 512, 1024\}$, the negative sample sizes in $\{128, 64\}$, and the margin γ in $\{30, 40, 50, 60, 80\}$. The best hyperparameters are shown in Table 8. The code of HybridQE and the created benchmark datasets can be accessed at <https://github.com/RoyaHe/HybridQE-->.

Table 8: Hyperparameters found by grid search. d is the embedding dimension, b is the batch size, n is the negative sampling size, γ is the margin in loss, l is the learning rate, ω is the regularization parameter in the distance function.

Dataset	Model	d	b	n	γ	l
ProdHybridQA	HybridQE (BetaE)	800	512	128	50	1e-4
	HybridQE (GammaE)	800	512	128	60	1e-4
BioHybridQA	HybridQE (BetaE)	800	512	128	50	1e-4
	HybridQE (GammaE)	800	512	128	60	1e-4

F.2 Introduction and Implementation of Baselines

We introduce the technical details on the implementation of the baseline methods.

BM25, ANCE and QA-GNN We adapt the baseline methods on the hybrid query answering task by transforming the hybrid queries into natural language questions under the textualization templates in Appendix G.2 and treating the entity descriptions as the search corpus. The implementations of the baseline methods are following the open-sourced code: BM25 (<https://github.com/nhirakawa/BM25/tree/master>), ANCE (<https://github.com/microsoft/ANCE>), and QAGNN(<https://github.com/michiyaunaga/qagnn>).

ToG on HybridQA We adapt ToG on the hybrid query reasoning task through the following steps, respecting its original design:

1. **Question Transformation:** transforming the hybrid queries into natural language questions under the textualization templates in Appendix G.2
2. **Initialization:** extract topic entities from the input query, the topic entities are symbolic query entities in our implementation
3. **Iterative Exploration (Beam Search):** we iteratively perform relation exploration and entity exploration. For **relation exploration**, we retrieve candidate relations linked to the current entities from the KG and use the LLM to prune irrelevant relations, keeping the top-N most relevant ones. For **entity exploration**, we retrieve candidate entities linked by the selected relations and prune entities via the LLM, retaining the most promising ones. Finally, we **extend reasoning paths** with the selected relations and entities.
4. **Reasoning and Termination Check:** After each exploration step, the LLM evaluates whether the current paths contain enough information to answer the question.
5. **Answer Generation:** the LLM synthesizes the final answer using the explored paths or falls back to its internal knowledge if paths are insufficient.

HybGRAG on HybridQA HybGRAG is an agentic hybrid retrieval framework designed for hybrid question answering over semi-structured knowledge bases that contain both textual documents and relational graph structures. To adapt HybGRAG to our hybrid query answering setting, we reimplement its core components following the original design.

1. **Query routing and entity–relation extraction:** Given a translated natural language query, an LLM-based router first identifies the relational aspect of the query by extracting candidate topic entities and relevant relation types from a predefined schema. Based on the extracted information, the router decides whether to invoke a text retrieval module or a hybrid (graph-based) retrieval module.
2. **Retriever bank:** HybGRAG maintains a retriever bank consisting of two modules. The text retrieval module performs dense vector similarity search between the translated question and textual documents (entity descriptions). The hybrid retrieval module first extracts an ego-graph around the identified entities using the selected relations, and then ranks the associated documents using text similarity, enabling joint use of relational constraints and textual relevance.
3. **Critic module with self-reflection:** To address errors in initial routing and extraction, HybGRAG employs a critic module composed of a validator and a commenter. The validator checks whether the retrieved documents satisfy the query requirements, while the commenter provides structured corrective feedback when retrieval fails. This feedback is used to refine entity and relation extraction in subsequent iterations.
4. **Iterative refinement:** The router, retriever bank, and critic operate in an iterative loop. The process terminates when the validator accepts the retrieved results or when a maximum number of refinement steps is reached.

In our experiments, we follow this procedure to adapt HybGRAG as a baseline for hybrid query answering, while keeping its retrieval-centric and agentic design unchanged.

QE+LM on HybridQA QE+LM answers hybrid queries by independently processing symbolic and textual constraints and combining their outputs. To ensure a fair comparison with HybridQE, we use the same text encoder for the textual module. The pipeline consists of three stages:

1. **Symbolic Query Processing (QE Module):** we retrained BetaE on the training graph to embed the symbolic query into latent space and retrieve entities from the KG whose embeddings minimize distance to the query embedding, yielding answer set A_{sym} .
2. **Textual Query Processing (LM Module):** We compute embeddings for the textual query component and all entity descriptions using the same sentence-transformer encoder as HybridQE. Then, entities are ranked based on semantic similarity (e.g., cosine distance) between query and entity embeddings. Top-10 entities with highest similarity scores, yielding answer set A_{text} .
3. **Answer Fusion:** we derive the final answer set as the conjunction of A_{sym} and A_{text} , which ensures answers satisfying both structural (KG) and semantic (textual) constraints.

The key distinction between QE+LM and HybridQE is therefore architectural: QE+LM processes the two modalities in isolation and combines outputs post-hoc, while HybridQE enables bidirectional information flow between symbolic and textual components during encoding. Any performance difference thus reflects the value of joint reasoning rather than encoder quality.

PullNet on HybridQA PullNet is an iterative retrieval framework that constructs a question-specific subgraph by alternately pulling facts from a KB and text from a corpus, then reasoning over the resulting heterogeneous graph with a graph CNN to extract answers. We adapt PullNet to HybridQA as follows.

1. **Query input.** Hybrid queries are converted to natural language questions using the textualization templates in Appendix 10.
2. **Graph mapping.** Entity nodes correspond to graph entities V , fact nodes to training edges E_{train} , and text nodes to the textual descriptions $\tau(v)$ attached to each entity, replacing

PullNet’s external corpus with node-attached text descriptions.

3. **Initialisation.** The initial subgraph is seeded with the symbolic anchor entities extracted directly from ϕ , bypassing entity linking.
4. **Relation scoring.** The pull operation scores candidate relations using a weighted combination of symbolic relevance (whether the relation appears in ϕ) and textual similarity to the query’s text component t , controlled by a hyperparameter λ .
5. **Answer ranking.** Since HybridQA requires a ranked list of entities, we score all entities by their graph CNN classifier score if reached by iterative retrieval, and by text similarity to the query’s textual component otherwise. Logical constraints in ϕ are applied as post-hoc filters over the ranked list. This modular design contrasts with HybridQE’s joint encoding, and serves to isolate the contribution of integrated hybrid reasoning.

G Translation of Hybrid Queries

G.1 Transform Hybrid Queries into Symbolic Form

In this section, we describe the text-to-logic compilation step that parses the free-form text into an approximate symbolic constraint, then conjunct it with the original symbolic part in the query. Given the hybrid query $Q = \{t\} \sqcap \phi$, a LLM-based parser $\mathcal{P}_{\text{LLM}}$, and a schema linking function $\mathcal{L}$, the transformation process consists of three steps:

1. **Parsing:** We prompt a language model with task-specific instructions to translate natural language queries into conjunctive queries composed of abstract predicates and variables, which are later grounded to the database schema via similarity-based linking:

$$\hat{\phi}_t = \mathcal{P}_{\text{LLM}}(t)$$

where $\hat{\phi}_t$ represents the symbolic format of the textual query. The prompt instructions for query parsing are provided below.

2. **Grounding:** Since the predicates and entities in $\hat{\phi}_t$ may not directly correspond to schema elements in the target text-labeled graph database, we perform a grounding step

that aligns abstract symbols with concrete nodes and relations. For each entity and relation symbol in $\hat{\phi}_t$, we compute its textual similarity to candidate nodes and relation names in the database, and select the best-matching schema elements. This produces a grounded conjunctive query ϕ_t whose predicates and arguments are defined over the database schema:

$$\phi_t = \mathcal{L}(G, \hat{\phi}_t)$$

3. **Composition:** The grounded conjunctive query ϕ_t is then conjoined with the original symbolic constraints ϕ to form the final compiled symbolic query, which serves as input to downstream hybrid reasoning and retrieval:

$$Q_{\text{symbolic}} = \phi_t \sqcap \phi$$

Prompt for Text-to-Conjunctive Query Parsing

Instruction. You are an expert in semantic parsing and knowledge graph querying. Given a natural language query, translate it into a conjunctive query (CQ) that captures the logical constraints expressed in the text. Use only conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$) when necessary. Introduce variables for entities whose identity is unknown. Do not ground predicates or entities to a specific database schema; use descriptive predicate and entity names inferred from the text.

Output format. Represent the query as a set of atoms of the form $\text{predicate}(\text{subject}, \text{object})$. Use variables prefixed with $?$. Output only the conjunctive query.

G.2 Transform Hybrid Queries into Textual Form

We transform the symbolic component in hybrid queries into natural language questions using domain-specific templates that maintain the original logical operations while producing fluent questions. Table 11 lists the query structures of queries presented in our datasets. Table 10 presents the query-to-question templates for datasets ProdHybridQA and BioHybridQA. The templates systematically handle conjunctions, disjunctions, and negations while preserving the original query structure.

H Backbone Sensitivity of LLM-based Methods

A natural question is whether stronger LLM backbones improve performance on HybridQA. To investigate this, we evaluate ToG with three backbone models of varying capability: GPT-3.5-turbo,

Table 9: The Hits@3 performance of HybridQE and baseline models on ProdHybridQA and BioHybridQA benchmark datasets.

Dataset	Model	1p	2i	3i	2u	3u	1n	2in	3in	2p	3p
ProdHybridQA	BM25	8.33	14.66	11.33	13.66	10.33	5.33	4.66	3.00	-	-
	ANCE	5.29	10.85	5.33	8.22	5.05	3.94	3.92	3.14	-	-
	QA-GNN	1.42	6.27	4.29	7.54	4.25	1.38	0.92	1.32	-	-
	HybGRAG	5.94	6.49	12.85	9.59	7.43	0.71	1.13	2.73	-	-
	ToG	2.35	4.24	8.48	7.39	9.32	0.34	2.15	1.23	-	-
	PullNet	5.83	7.15	8.32	5.41	4.95	1.22	1.64	2.17	-	-
	QE+LM (modular method)	0.42	0.44	0.51	0.67	0.54	0.00	0.00	0.002	-	-
	HybridQE (BetaE)	7.04	11.35	20.01	10.59	8.12	1.45	3.01	5.83	-	-
	HybridQE (GammaE)	11.76	17.79	25.57	15.44	13.18	1.54	3.59	6.53	-	-
BioHybridQA	BM25	13.01	12.00	12.33	8.33	7.33	5.66	4.33	4.23	7.33	8.33
	ANCE	6.49	10.56	10.53	6.84	7.49	5.59	3.03	3.69	2.35	0.95
	QA-GNN	5.54	7.59	8.34	6.83	1.59	1.36	2.37	2.81	2.48	0.92
	HybGRAG	4.85	2.93	2.28	2.43	2.35	1.86	0.39	0.45	2.35	2.23
	ToG	2.48	2.92	2.27	1.47	2.04	1.19	1.34	0.15	0.21	0.01
	PullNet	4.64	5.83	6.75	3.93	3.26	0.85	1.52	1.32	3.63	2.31
	QE+LM (modular method)	1.87	1.38	1.94	1.38	1.28	0.002	0.01	0.02	0.39	0.92
	HybridQE (BetaE)	10.82	13.02	15.95	7.47	6.43	2.37	3.39	3.69	3.43	6.24
	HybridQE (GammaE)	16.67	16.61	20.72	9.81	7.42	2.71	4.83	4.28	5.65	8.72

Table 10: Query-to-Question Templates

Query Type	Template
ProdHybridQA Queries	
Base	“Please find out the products with the following attributes:”
1p/1n	+ “[relation] is/is not [entity]”
2i/2in	+ “[relation1] is [entity1], and [relation2] is/is not [entity2]”
3i/3in	+ “[relation1] is [entity1], [relation2] is [entity2], and [relation3] is/is not [entity3]”
2u	+ “[relation1] is [entity1], or [relation2] is [entity2]”
3u	+ “[relation1] is [entity1], or [relation2] is [entity2], or [relation3] is [entity3]”
BioHybridQA Queries	
Base	“Please find out items that”
1p	+ “they [relation] [entity]”
1n	+ “they do not [relation] [entity]”
2p	+ “they [relation1] something that [relation2] [entity]”
3p	+ “they [relation1] something that [relation2] something that [relation3] [entity]”
2i	+ “they [relation1] [entity1] and [relation2] [entity2]”
3i	+ “they [relation1] [entity1], [relation2] [entity2] and [relation3] [entity3]”
2in	+ “they [relation1] [entity1] but do not [relation2] [entity2]”
3in	+ “they [relation1] [entity1] and [relation2] [entity2] but do not [relation3] [entity3]”
2u	+ “they [relation1] [entity1] or [relation2] [entity2]”
3u	+ “they [relation1] [entity1] or [relation2] [entity2] or [relation3] [entity3]”

DeepSeek-R1-Distill-Qwen-32B, and DeepSeek-R1-Distill-Llama-8B. The results are reported in Figure 7. The results indicate that LLM-based methods exhibit **high sensitivity to backbone choice**. The performance varies substantially across backbone models, with no consistent winner across query types or datasets. This instability contrasts sharply with HybridQE, which maintains consistent improvements across all query types regardless of its symbolic backbone (BetaE or GammaE), as shown in RQ3.

I More Experimental Results

In this section, we provide more experimental results of HybridQE and baseline approaches in different evaluation metrics, Hits@1 and Hits@3, in

Table 7 and Table 9 respectively.

J Scalability and Runtime Analysis

In this section, we provide analysis on the scalability and runtime of HybridQE. HybridQE precomputes token-level entity representations offline, so inference does not require repeatedly running the text encoder over entity descriptions. At query time, the query-conditioned attention is implemented as batched matrix operations between query vectors and cached token embeddings of candidate entities. The full-ranking complexity is linear to the number of entities (i.e., $\mathcal{O}(|V|Ld)$), where $|V|$ is the number of entities, L is average description length, and d is embedding dimension. In Table 12, we have added runtime measurements re-

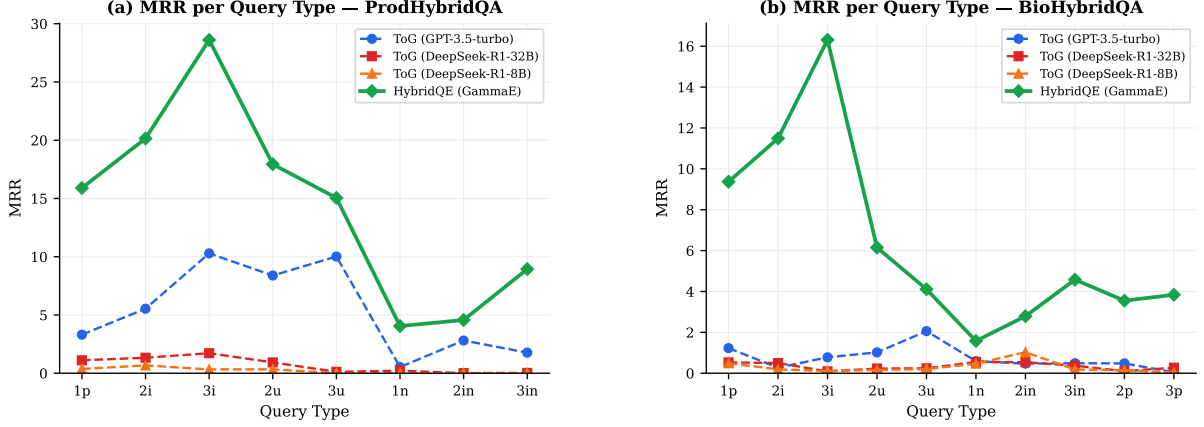


Figure 7: Backbone sensitivity analysis of ToG with three LLM backbones compared to HybridQE (GammaE).

Table 11: Query structure templates of query types in HybridQA datasets, where e_i corresponds to entity i and r_i corresponds to relation i .

Query Type	Query Structure
1p	$(e_1, (r_1))$
1n	$(e_1, (r_1, n))$
2p	$(e_1, (r_1, r_2))$
3p	$(e_1, (r_1, r_2, r_3))$
2i	$((e_1, (r_1)), (e_2, (r_2)))$
3i	$((e_1, (r_1)), (e_2, (r_2)), (e_3, (r_3)))$
2in	$((e_1, (r_1)), (e_2, (r_2, n)))$
3in	$((e_1, (r_1)), (e_2, (r_2)), (e_3, (r_3, n)))$
2u	$((e_1, (r_1)), (e_2, (r_2)), u)$
3u	$((e_1, (r_1)), (e_2, (r_2)), (e_3, (r_3)), u)$

porting training time per step, full-ranking latency per query, peak GPU memory, and token-cache storage on both datasets with both backbone models. These results show that the additional cost of query-conditioned attention is manageable at the scale of our benchmarks: full ranking over more than 120K entities can be performed within tens of milliseconds per query after token representations are cached. For larger deployment-scale graphs, HybridQE can also be used in a two-stage setting, where a lightweight retriever first retrieves a top-K candidate set and HybridQE reranks only those candidates. This preserves the model’s main benefit while reducing inference cost for larger graphs.

K Error Analysis

To better understand where HybridQE succeeds and fails, we conduct a stratified error analysis over BioHybridQA using HybridQE (GammaE). We randomly sample 30 queries from each of the 10 query types and inspect the top-10 false positive predictions per query. Each false positive is

manually labeled with its primary error source. We identify six error categories, summarized in Table 13. Table reports the percentage of top-10 false positives falling into each category. The analysis result leads to the following observations:

Firstly, negation failure (Type IV) accounts for 56.9%–78.8% of errors in negation-bearing queries, consistent with known limitations of neural complement operators in the query embedding literature. Notably, the Type IV rate decreases from 78.8% (1n) to 56.9% (3in), suggesting that the additional positive conjuncts in 2in and 3in queries provide meaningful corrective filtering through the intersection operator. To further diagnose this, we additionally sample 100 queries per negation type and categorize their top-10 false positives into finer-grained subtypes: hidden negation violations, where the violating edge is missing from the training graph (43.2% of negation false positives), and observed complement failures, where the violating edge is present yet the complement operator still fails (24.5%). The dominance of hidden negation violations indicates that negation is especially challenging under symbolic incompleteness, since the model cannot reliably negate a relation it has never observed for a given entity.

Secondly, partial symbolic match grows with query complexity. Type III errors rise from 57.8% (2i) to 69.8% (3i), with false positives typically satisfying all but one symbolic conjunct. This reflects the inherent difficulty of enforcing strict conjunction in continuous embedding spaces, where near-miss entities may receive scores close to true answers. Thirdly, union over-expansion scales with disjunct count. Type V errors increase from 58.8% (2u) to 69.4% (3u), revealing that mixture-based

Dataset	Backbone	Train time / step	Full-ranking latency / query	Peak GPU memory	Token-cache storage
ProdHybridQA	BetaE	460 ms	44 ms	13 GB	7.5GB
ProdHybridQA	GammaE	570 ms	37 ms	16 GB	7.5GB
BioHybridQA	BetaE	500 ms	80 ms	17 GB	9.3GB
BioHybridQA	GammaE	760 ms	63 ms	21 GB	9.3GB

Table 12: Runtime and scalability analysis of HybridQE on both benchmarks. Training time and full-ranking latency are measured per step and per query respectively. Token-cache storage refers to the precomputed token-level entity representations stored offline.

Error Type	Applicable Queries	Diagnostic Target
Type I: Symbolic-satisfied, text-mismatched	All	Text encoder / QCS mechanism
Type II: Text-satisfied, symbolic-violated	All	Symbolic query encoder
Type III: Partial symbolic match	All (esp. 2i, 3i, 2in, 3in)	Intersection operator tolerance
Type IV: Negation failure	1n, 2in, 3in	Neural complement operator
Type V: Union over-expansion	2u, 3u	Union operator over-generalization
Type VI: Path-composition failure	2p, 3p	Multi-hop error compounding

Table 13: Error Type Definitions and Diagnostic Targets

Query Type	MRR	Type I	Type II	Type III	Type IV	Type V	Type VI
1p	9.37	36.7	41.3	22.1	-	-	-
2i	11.49	22.5	19.6	57.8	-	-	-
3i	16.31	16.3	14.0	69.8	-	-	-
2u	6.15	12.1	10.4	18.7	-	58.8	-
3u	4.11	8.2	6.8	15.6	-	69.4	-
1n	1.58	4.7	3.7	12.8	78.8	-	-
2in	2.79	5.8	4.5	21.5	68.3	-	-
3in	4.58	7.6	6.3	29.2	56.9	-	-
2p	3.55	10.6	8.9	16.8	-	-	63.7
3p	3.84	7.9	6.5	13.1	-	-	72.5

Table 14: Error distribution (%) across query types on BioHybridQA. For each query type, we report MRR alongside the proportion of top-10 false positives attributed to each error category.

union modeling becomes less precise as the number of disjuncts grows. Fourthly, the multi-hop errors compound across projection steps. Type VI errors rise from 63.7% (2p) to 72.5% (3p), consistent with well-known error accumulation in multi-hop reasoning. Lastly, textual errors outweigh symbolic errors in most query types. Type I (text mismatch) exceeds Type II (symbolic violation) in 9 of 10 query types, suggesting that richer utilization of textual signals remains a promising and under-exploited avenue for improvement.

This analysis identifies four concrete challenges that our benchmarks help diagnose and quantify, and that future work can target: (1) sharper complement operators for negation under incompleteness, potentially through text-based negation verification where a cross-encoder checks entity descriptions for evidence of the negated relation; (2) tighter conjunction boundaries for complex intersections; (3) geometrically constrained union representations that maintain precision as the number of disjuncts

increases; and (4) intermediate-hop supervision or path-level objectives for multi-hop reasoning.

L Qualitative Case Study

To complement the quantitative results in Section 5, we present a qualitative analysis of a representative case of HybridQE. The goal is to more vividly illuminate how HybridQE reasons at intermediate steps.

Query. The 3i hybrid query

$$Q = \phi \sqcap \{t\}$$

where the symbolic and textual components are defined as:

$$\phi = \exists \text{category.}\{Baby_Bottle\} \sqcap \exists \text{material_type.}\{BPA \text{ free}\} \sqcap \text{color.}\{Pink\}$$

$t =$ “Could you help locate a bottle categorized as a baby product with a capacity measured in milliliters?”

Rank comparison. Table 15 presents an internal decomposition of HybridQE’s scoring on the hybrid query $Q = \phi \sqcap \{t\}$. For each candidate entity we report three ranks derived from a single HybridQE forward pass: the rank induced by the symbolic component score $\text{Dist}(\mathbf{Emb}_\phi, \mathbf{Emb}_e)$ alone, the rank induced by the textual component score $\text{Dist}(\mathbf{Emb}_{\{t\}}, \mathbf{Emb}_e)$ alone, and the final rank produced by the full hybrid encoder $\text{Dist}(\mathbf{Emb}_Q, \mathbf{Emb}_e)$ after the INTERSECT operator combines both components. Note that this decomposition is distinct from the query-level ablation in RQ2 (Table 3), which runs HybridQE on differently-formulated queries; here we inspect the intermediate scores produced by the two component encoders *within a single hybrid query forward pass*.

Analysis. Table 15 provides direct empirical evidence for HybridQE’s two complementary inference pathways (Section 4). The key finding concerns the correct hard-answer entity B078TGQHQQ, whose symbolic edges all belong to E_{missing} . Despite this, $\mathbf{Emb}_\phi$ assigns it a strong rank 2, reflecting the structure-driven pathway: latent relational patterns in G_{train} provide partial structural evidence even for entities with missing edges. The *textual pathway* further reinforces this recovery: B078TGQHQQ’s description explicitly contains “BPA Free” and “Pink”, the exact values of the two missing *material_type* and *color* edges in ϕ , allowing $\mathbf{Emb}_{\{t\}}$ to recover the missing symbolic information directly from text.

Additionally, the remaining entities illustrate why neither component alone suffices. B07RR2BRHD ranks higher than others under the textual component, but its weak symbolic rank 5 leads the hybrid encoder to correctly demote it to rank 2. B00DBFXG1Q achieves a strong symbolic rank 3 yet receives rank 8 from text encoder for lacking any capacity signal, and is correspondingly demoted to rank 3 by the hybrid encoder. Across all entities, the hybrid encoder consistently promotes those with mutually reinforcing evidence from both modalities and penalizes those strong on one axis but weak on the other, confirming that joint encoding enables cross-modal verification that neither modality alone can achieve.

M Few-shot Performance Analysis

Unlike backbone symbolic query embedding methods, which rely entirely on structural patterns ob-

served during training, HybridQE can leverage textual descriptions of entities to reason about those with limited or no structural evidence. Although our benchmark datasets follow a transductive setup, we examine HybridQE in few-shot scenarios to gain insights into its potential robustness on entities with sparse training evidence, a key prerequisite for inductive hybrid query answering. Figure 8 illustrates the distribution of entity mention frequencies within the training dataset.

More than 10,000 entities appear only a hand-

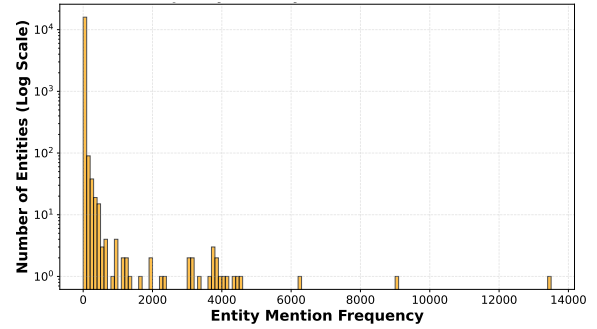


Figure 8: Distribution of entity mention frequencies within the training dataset. The x-axis denotes the frequency of entity mentions, while the y-axis corresponds to the number of entities associated with each frequency value on log scale.

ful of times, highlighting the long-tail nature of entity occurrences in real-world graphs. To understand how this frequency distribution affects model performance, we partition test queries into four groups based on the training-set frequency of the entities they reference. Table 16 presents the MRR performance of HybridQE across these frequency-stratified groups. The results reveal a non-monotonic relationship between entity frequency and performance. Rather than a simple inverse correlation, performance peaks for moderate-frequency entities (frequency 2–50, 22.40 average MRR) and declines at both extremes. This pattern reflects a tradeoff between evidence availability and entity specificity. Frequency-1 entities have minimal structural evidence, yet HybridQE still achieves 17.36 average MRR by exploiting textual descriptions through query-conditioned matching, demonstrating the value of text-driven inference when graph-based patterns are sparse. Moderate-frequency entities strike the best balance, providing enough structural and textual evidence for learning reliable hybrid representations while remaining sufficiently specific to be distinguishable. Very high-frequency entities (≥ 2000) yield the weakest

Table 15: Internal rank decomposition of HybridQE on the 3i success-case hybrid query Q . For each entity we report the rank induced by the symbolic component $\mathbf{Emb}_\phi$, the textual component $\mathbf{Emb}_{\{t\}}$, and the final hybrid encoder $\mathbf{Emb}_Q$, all derived from the same single forward pass on $Q = \phi \sqcap \{t\}$. $\downarrow$ lower rank is better.

Entity	$\mathbf{Emb}_\phi \downarrow$	$\mathbf{Emb}_{\{t\}} \downarrow$	$\mathbf{Emb}_Q \downarrow$	Description
B078TGQHQQ [†]	2	4	1	Putti Atti Silicone Baby Bottle with Handle, BPA Free, Pink
B07RR2BRHD	5	3	2	Tescoma Papu Papi Soft Bottle with Spoon, 200 ml, Pink
B00DBFXG1Q	3	8	3	BellyBottle Pregnancy Water Bottle Intake Tracker with Weekly Milestone Stickers, Pregnancy Gifts for First Time Moms, Must-Have Essentials, Pink
B07YN985ZV	4	10	4	Baby Magic Bottles (2 Pack) Milk / Orange Juice — Empties When Tilted, with 2 GosuToys Stickers
B00IM3QW7C	12	6	5	Tommee Tippee Closer to Nature Baby Bottle Decorated Pink, Breast-Like Nipple for Natural Latch, Slow Flow, BPA-Free - 0+ Months, 2 Count
B01MRKFA85	7	12	6	Baby Bottle Adapter Turns Water Bottles into Baby Bottles, Formula or Bottled Water for Babies On-The-Go by Refresh-A-Baby, Camo Green

[†] Correct hard-answer entity ($\in \mathcal{A}_{\text{hard}}$).

Table 16: The MRR performance of HybridQE on test queries grounded with varying frequency in the training graph.

Frequency	1p	2i	3i	2u	3u	1n	2in	3in	AVG
1	34.65	28.41	36.3	10.67	14.77	1.12	4.89	8.07	17.36
2-50	35.55	45.42	36.89	18.58	19.3	0.99	8.58	13.87	22.40
50-2000	12.19	19.12	17.61	12.58	9.14	0.55	0.51	8.57	10.03
≥ 2000	3.67	13.79	2.26	9.49	12.59	1.71	1.97	6.25	6.47

results (6.47 average MRR) because they tend to be highly connected hub nodes with generic descriptions, making them easily confusable under hard-answer evaluation, and because the model may overfit to their frequent patterns at the expense of discriminative features. These findings motivate two future directions: frequency-aware training strategies to mitigate hub-entity degradation, and fully inductive hybrid query answering where test-time entities are entirely unseen during training.