

# Logical Queries on Knowledge Graphs: Emerging Interface of Incomplete Relational Data

Zihao Wang<sup>†</sup>

Hang Yin<sup>‡</sup>

Yangqiu Song<sup>†</sup>

<sup>†</sup> Department of CSE, HKUST, Hong Kong, China    {zwanggc, yqsong}@cse.ust.hk

<sup>‡</sup> Tsinghua University, Beijing, China    h-yin20@mails.tsinghua.edu.cn

## Abstract

## 1 Introduction

The knowledge graph is perhaps one of the simplest representations of relational knowledge [8, 54, 38]. A knowledge graph contains a large number of triples, where each triple  $(h, r, t)$  contains a head entity  $h$ , a tail entity  $t$ , and the relation  $r$  in between. Such a simple format enables factual knowledge to be adapted to answer real-world questions [47, 59, 45, 40, 32].

Under the view of data engineering, a knowledge graph is no more than a (relational) database [1]. It is natural to expect first-order queries, a fundamental type of relational database queries that have been well-understood theoretically [30] and efficiently solved practically [29], can also be answered on knowledge graphs. However, applying the existing query-answering algorithms designed for (relational) databases can not address logical queries on knowledge graphs. The major obstacle is that large-scale knowledge graphs are inherently incomplete. Modern large-scale knowledge graphs are constructed by crowd-sourcing [54] or even by automatic information extraction pipelines [11]. Given an observed knowledge graph, the non-existence of a triple does not imply that such a triple is not part of the underlying knowledge graph. This issue is acknowledged as the open-world assumption [31]. Therefore, the actual answers to logical queries are not guaranteed by running existing query-answering algorithms on knowledge graphs.

The issue of missing knowledge finds its partial solution with learning-based models. Models that learn from the existing observed triples can predict the missing triples in the underlying knowledge graph. The research topic focusing on the prediction of missing triples with learned embeddings is known as knowledge graph representation or knowledge base completion [25, 44]. Then, the logical queries can be addressed by combining the query-answering algorithms and the learning-based models [42, 28, 16]. The participation of learning-based models brings unique strengths and weaknesses to logical query answering, especially compared to traditional query answering algorithms for relational databases [42]. For traditional databases, query-answering algorithms are discussed from the data perspective and the query perspective. The complexity of such algorithms is characterized by the data complexity (the complexity grows with the size of the database) and the query complexity (the complexity grows with the size of the query), respectively. When answering logical queries with learning-based models, it is essential to understand how learning-based models interact with knowledge graphs (model-data interaction) and logical queries (model-query interaction), besides how to design, train, and infer such models.

---

*Copyright 2004 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.*

**Bulletin of the IEEE Computer Society Technical Committee on Data Engineering**

---

In this paper, we review recent advancements in logical query answering on knowledge graphs. Our introduction begins with how to define the problems and construct datasets and then to learning-based query answering methods. Our discussion is organized from three perspectives: knowledge graphs (data) perspective, logical query perspective, and model perspective. Section 2 introduces the logical query answering tasks on knowledge graphs in the rigorous language of model theory. All existing datasets and benchmarks are discussed in Section 3 based on the rigorous definitions. Section 4 discusses the learning-based methods for knowledge graph logical query answering. Existing approaches are categorized by their interaction with the knowledge graph (data perspective) and the logical queries (query perspective). Section 5 discusses the limitation of the existing work and several possible future directions.

## 2 Preliminaries for Logical Queries

Logical queries are formally defined by  $\mathcal{L}$ -formula on an  $\mathcal{L}$ -structure, which are central concepts in model theory [36]. Specifically,  $\mathcal{L}$ -structure rigorously justifies the concept of knowledge graphs while  $\mathcal{L}$ -formula formally defines the scope of queries. Formal definitions provide a framework to investigate the data and the query aspects of the knowledge graph logical query answering tasks.

The minimal and informal description of knowledge graphs is a set of triples  $\{(h_i, r_i, t_i)\}$  that encapsulates the real-world knowledge, where  $h_i, t_i \in \mathcal{E}$  are the head and tail entities and  $r_i \in \mathcal{R}$  is the connected relation. Formally speaking, a knowledge graph is an  $\mathcal{L}_{KG}$ -structure defined by a language  $\mathcal{L}_{KG}$  for knowledge graphs. The concept of language and structure, which concerns syntax and semantics of first-order logic respectively, is defined in Definition 1 and Definition 2.

**Definition 1 (Language):** A language  $\mathcal{L}$  is a triple  $(\mathcal{E}, \mathcal{R}, \mathcal{F})$  where  $\mathcal{E}$  is the set of constant symbols,  $\mathcal{R}$  is the set of predicate symbols, and  $\mathcal{F}$  is the set of function symbols. For each  $r \in \mathcal{R}$  and  $f \in \mathcal{F}$ , associated integers  $n_r$  and  $n_f$  indicate the number of inputs for  $R$  and  $F$ , respectively.

All (classic) knowledge graphs can be described by a language  $\mathcal{L}_{KG}$  by letting (1)  $\mathcal{R}$  be the set of binary relations ( $n_r = 2$ ), (2) the constant set  $\mathcal{E}$  is finite and contains all entities, and (3)  $\mathcal{F} = \emptyset$ .  $\mathcal{E}$  and  $\mathcal{R}$  are finite sets without further classification.

**Definition 2 ( $\mathcal{L}$ -structure):** An  $\mathcal{L}$ -structure  $\mathcal{M}$  for the language  $\mathcal{L}$  is a pair  $(M, I)$ , where  $M$  is a non-empty set called the universe of  $\mathcal{M}$  and  $I$  is an interpretation function that: (i) For each  $c \in \mathcal{E}$ ,  $I(c) \in M$ ; (ii) For each  $r \in \mathcal{R}$ ,  $I(r) \subset M^{n_r}$ , where  $n_r$  is the size of relations; (iii) For each  $f \in \mathcal{F}$ , a function  $I(f) : M^{n_f} \mapsto M$ , where  $n_f$  is the number of inputs in  $f$ .

A knowledge graph  $\mathcal{KG}$  induces an  $\mathcal{L}_{KG}$ -structure by letting  $M = \mathcal{E}$  (with  $\forall e \in \mathcal{E}, I(e) = e$ ), so that each relation  $I(r)$  is a subset of  $M^2$ . The open world assumption of knowledge graphs can then be formally stated in Definition 3.

**Definition 3 (Open world assumption):** For a given language  $\mathcal{L}_{KG}$  of knowledge graph, there is an underlying  $\mathcal{L}_{KG}$ -structure  $\mathcal{KG}_u$ , and one (or more) observed  $\mathcal{L}_{KG}$ -structure(s)  $\mathcal{KG}_o$ , such that for each relation  $r \in \mathcal{R}$ , the relation set of  $\mathcal{KG}_o$  and  $\mathcal{KG}_u$  satisfies  $I_{\mathcal{KG}_o}(r) \subseteq I_{\mathcal{KG}_u}(r) \subseteq M^2$ .

Then, it is ready to define the logical queries with formal concepts, including  $\mathcal{L}$ -terms and  $\mathcal{L}$ -formulae. The current discussion focuses on first-order logic.

**Definition 4 ( $\mathcal{L}$ -term):**  $\mathcal{L}$ -terms be the smallest set of  $\mathcal{T}$  such that: (i)  $e \in \mathcal{T}$  for each constant symbol  $e \in \mathcal{E}$ ; (ii) Any variable  $v_i \in \mathcal{T}$  for  $1, 2, \dots$ ; (iii) if  $t_1, t_2, \dots, t_{n_f} \in \mathcal{T}$  and  $f \in \mathcal{F}$ , then  $f(t_1, \dots, t_{n_f}) \in \mathcal{T}$ .

Table 1: Summation of open source knowledge graphs used in the existing datasets. KGs are sorted by the number of entities.

| Knowledge graph   | # Entities | # Relations | # Total edges | Comment         | Datasets             |
|-------------------|------------|-------------|---------------|-----------------|----------------------|
| FB15k-237 [9]     | 14,505     | 237         | 310,079       | -               | [42, 43, 19, 23, 55] |
| FB15k [51]        | 14,951     | 1,345       | 592,213       | -               | [42, 43, 55]         |
| DBPedia [5]       | 34,575     | 3           | 240,942       | Hierarchical KG | [14]                 |
| WN18RR [37]       | 40,903     | 11          | 103,509       | Hierarchical KG | [24]                 |
| NELL955 [56]      | 63,361     | 200         | 142,804       | -               | [42, 43, 23, 55]     |
| DRKG [61]         | 97,238     | 107         | 5,874,271     | -               | [13]                 |
| FB400k [48, 8]    | 409,829    | 918         | 2,151,671     | -               | [41]                 |
| ogbl-wikikg2 [22] | 2,500,604  | 535         | 17,137,181    | -               | [41, 19]             |
| Freebase [8]      | 86,054,151 | 14,824      | 338,586,276   | -               | [41]                 |

**Definition 5 (Atomic  $\mathcal{L}$ -formula):**  $\phi$  is an atomic  $\mathcal{L}$ -formula if  $\phi$  is  $r(t_1, \dots, t_{n_r})$ , where  $r \in \mathcal{R}$  is a relation symbol and  $t_i$  are  $\mathcal{L}$ -terms. Given  $\mathcal{L}$ -structure  $\mathcal{M}$ , we say  $\mathcal{M} \models r(r_1, \dots, r_{n_r})$  is True if and only if the tuple  $(t_1, \dots, t_{n_r}) \in I(r)$ .

**Definition 6 ( $\mathcal{L}$ -formula):** The set of  $\mathcal{L}$ -formula is the smallest set  $\Phi$  containing all atomic formulae such that: (i) if  $\phi \in \Phi$ , then  $\neg\phi \in \Phi$ ; (ii) if  $\phi, \psi \in \Phi$ , then  $(\phi \wedge \psi) \in \Phi$  and  $(\phi \vee \psi) \in \Phi$ ; (iii) if  $\phi \in \Phi$  and  $v_i$  is any variable, then  $\exists v_i \phi \in \Phi$  and  $\forall v_i \phi \in \Phi$ .

We say a variable  $v$  is *free* if there are no associated quantifiers. Otherwise, it is *bounded*. We use  $\phi(v)$  indicates the  $\mathcal{L}$ -formula  $\phi$  contains a free variable  $v$ . An  $\mathcal{L}$ -formula without free variables is called an  $\mathcal{L}$ -sentence. Logical queries are  $\mathcal{L}$ -formulae and can be evaluated given a knowledge graph  $\mathcal{KG}$ . As studied in the database literature [53], we discuss boolean queries, set-valued queries, and aggregate queries.

A **boolean query** is an  $\mathcal{L}_{KG}$ -sentence  $s$ . Given the  $\mathcal{L}_{KG}$ -structure  $\mathcal{M}$ , its answer is the boolean value indicating whether  $\mathcal{M} \models s$  or not.

A **set-valued query** is an  $\mathcal{L}_{KG}$ -formula  $\phi$  with exactly one free variable  $v$ . Given the  $\mathcal{L}$ -model  $\mathcal{M}$ , its answer set is  $\{e : e \in \mathcal{E}, \mathcal{M} \models \phi(v = e)\}$ .

An **aggregate query** looks for the aggregation statistics of the answer set of a set-valued query, such as counting, summation, and so on.

### 3 Logical Query Answering Datasets and Evaluations

Based on the formal concepts given in Section 2, we are ready to formally justify existing datasets from the knowledge graphs (the data perspective) that are queried and the scope of logical queries (the query perspective) to be answered. Instead of going through all existing datasets, we discuss the underlying knowledge graphs and logical queries.

#### 3.1 Knowledge Graphs

Table 1 shows the knowledge graphs investigated in existing datasets. Existing evaluation includes knowledge graphs of different scales (from  $10^3$  to  $10^9$ ) and various properties. Notably, the knowledge graphs with hierarchical relation [37, 5] contain the prior and more complicated structures on entities and relations, respectively, which could be leveraged to improve the performance of query answering method. Inductive logical query answering [19] also considers an inductive KG setting [49], which is more challenging under the open-world assumption.

Table 2: Formal definitions of three typical query families. Compared to the first-order logic formula defined formally with Definition 6, three query families are defined using a subset of connectives or quantifiers (indicated by ✓).

| Query Family                            | $\wedge$ | $\vee$ | $\neg$ | $\exists$ | $\forall$ |
|---|----------|--------|--------|-----------|-----------|
| Conjunctive Query (CQ)                  | ✓        | ✗      | ✗      | ✓         | ✗         |
| Existential Positive First Order (EPFO) | ✓        | ✓      | ✗      | ✓         | ✗         |
| Existential First Order (EFO)           | ✓        | ✓      | ✓      | ✓         | ✗         |

### 3.2 Logical Queries

The scope of queries that can be answered is not rigorously justified in the existing works. Though some works claim they are addressing the first-order queries [43] or logical queries in general [33], their definitions are still a strict subset of the first-order queries defined in Definition 6. Table 2 summarizes the formal definitions for three typical query families. The scopes of three query families are sorted in increasing order, i.e.,  $CQ \subseteq EPFO \subseteq EFO$ . The universal quantifier  $\forall$  is not widely discussed because its semantics is not usually interpretable over the knowledge graphs. This is the same order for query families that learning-based query answering methods tried to solve.

However, additional assumptions are made so that queries are more suitable to be answered by learning-based methods. Common assumptions include that (1) the query is assumed to be acyclic and (2) only one free variable in the logical formula. The first assumption skips the queries that may involve hard constraint satisfaction problems and the second assumption restricts the space complexity of search within  $O(|\mathcal{E}|)$ . These assumptions are not necessary for all learning-based models. For example, BiQE [28] accepts conjunctive queries with multiple free variables, and CQD [4] answers general EPFO queries without any additional assumptions. Most existing datasets are usually constructed as companions to evaluate the proposed methods, especially query embedding methods with the operator tree representation discussed in Section 4.2.1. Such methods assume the answers with respect to the only free variable can be computed by executing learnable set operators. The execution order of set operators is decided by compiling the query formula into an operator tree. Thus, popular datasets [42, 43, 55] for EPFO and EFO queries also accept implicit assumptions: (1) the only free variable  $v$  is assumed to be the root of the set operator tree; (2) all leaf nodes must be constant entity symbols rather than existential variables.

We see that these assumptions are not related to the logic but only to make the queries easier to solve by a special class of learning-based methods. There are no serious empirical evaluations for queries without such assumptions.

### 3.3 Evaluation Metrics

The query families are categorized by the formulation of logical formulae. Moreover, for each family, it is also able to construct three types of queries. Here we list popular evaluating metrics.

- For **boolean query**, this becomes a classification problem, and thus standard classification metrics can be applied, like ROC AUC score [43, 21].
- For **set-valued query**, standard metrics include Mean Reciprocal Rank (MRR) [42], Adjusted Mean Rank Index (AMRI) [2], Average percentile rank (APR) [21], and Hits at K (H@K) [42].
- For **aggregate query**, counting is the only problem that has been considered. Spearman’s rank correlation coefficient [43] and Pearson’s correlation coefficient [35] have been proposed when the model is incapable of predicting the precise number. Otherwise, Mean Absolute Error (MAE) is used [35].

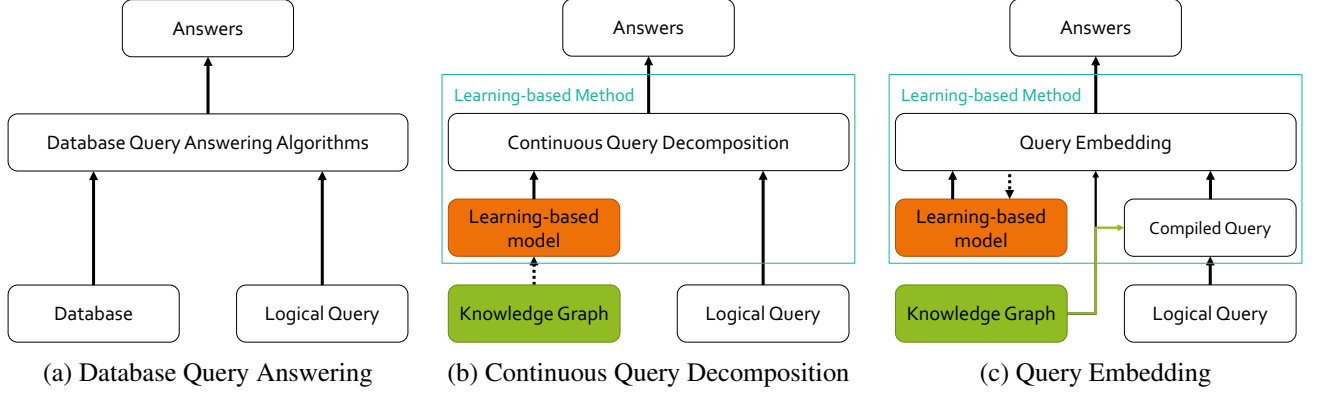


Figure 1: Paradigms for logical query answering on knowledge graphs. The solid line indicates the query answering procedures, and the dashed lines indicate the procedures to obtain the model.

Table 3: Comparison between CQD and QE methods with their training data, query answering algorithms, and solvable queries.  $\dagger$  indicates that the actual solvable queries may not be the formally defined logical query family.

| Methods                              | Training Data | Query Answering Algorithms | Solvable Queries |
|--------------------------------------|---------------|----------------------------|------------------|
| Continuous Query Decomposition (CQD) | KG triples    | Model + Optimization       | EPFO             |
| Query Embedding (QE)                 | QA samples    | Operator Tree + Model      | EFO-1 $\dagger$  |
|                                      | QA samples    | Query Graph + Model        | EPFO $\dagger$   |
|                                      | QA samples    | Sequence + Model           | EPFO $\dagger$   |

## 4 Learning-Based Methods

Traditional database query answering algorithms are executed over the raw data to find the answers, see Figure 1a. This paradigm fails in the cases of knowledge graph queries, where the underlying data is not observed. Current knowledge graph logical query answering methods address the open-world assumption with learning-based methods, see the blue boxes in Figure 1b and 1c. In this paper, we categorize the learning-based methods into two categories by how they process the queries. The first category is Continuous Query Decomposition (CQD) and the second is Query Embedding (QE). We see that learning-based models are essential for both the CQD method and the QE method. The participation of the learning-based models adds an intermediate layer between incomplete knowledge graphs and logical queries.

The key to understanding learning-based methods is not only how to define the model but also how models interact with the knowledge graph data and logical queries. The model-query and model-data interaction jointly define the scope of queries that can be answered by each method. Specifically, the model-data interaction refers to how to obtain the model from data, including both training data and algorithms, and the model-query interaction refers to how learning-based models are combined with query answering algorithms. The model-data and model-query interaction are indicated by the dashed arrows and the arrows in Figure 1b and 1c, respectively.

The rest of this section presents the details of these two types of methods. Table 3 compares these two types from three aspects, including training data (model-data interaction), query answering algorithms (model-query interaction), and queries that can be solved. The most distinct difference between CQD and QE is how they treat the model as part of the query answering algorithms. Continuous query decomposition estimates the answer embedding by solving an optimization problem with continuous objective induced from the original logical formula with logical  $t$ -norms [20]. This objective, computed by the result of the model inference, will be evaluated

multiple times during the optimization process. Therefore, the model will be inferred multiple times. This formulation is denoted as Model + Optimization in Table 3 where the term Model is placed before Optimization. In contrast, query embedding methods first compile the logical queries into various representations such as operator trees, query graphs, or sequences. Then, the answer embedding, or the embedding of logical query, is estimated by simple forward inferring the model only once. This formulation is denoted as X + Model, where the term Model is placed after X indicating one of the operator trees, query graphs, or sequences. As natural consequences of such distinct differences in query answering algorithms, CQD and QE require different types of training data to obtain the model and can solve different types of queries. Moreover, the solvable query families for QE methods differ when the queries are compiled into different representations. We note that the solvable queries are confirmed in existing works, and they can be of course extended to broader classes.

The Disjunctive Normal Form (DNF) [36] of the logical queries is of particular interest. This is because (1) all first-order queries can be converted to DNF, and (2) The answer set of DNF existential queries can be regarded as the union of the answer sets of its containing conjunctive clauses. For example, EPFO queries can be answered by compositing the solutions of conjunctive queries. Similarly, general existential logical queries can be answered once the queries with conjunction and negation can be well answered.

#### 4.1 Continuous Query Decomposition (CQD)

The model used in CQD [4] method is no more than a neural link predictor, which is the key part of knowledge graph representation [62, 25]. Specifically, CQD leverages the pretrained ComplEx embedding [52] and achieves state-of-the-art performances in EPFO queries on dataset [42]. The training of the neural link predictor is not related to the logical queries but only knowledge graphs themselves. Recent survey [25] summarizes the construction of neural link predictors and how to train neural link predictors well is discussed in [44].

CQD uses neural link predictors to solve EPFO queries with a search algorithm. It considers the EPFO query  $\phi(v)$  in the following DNF form.

$$\phi(v) = \exists x_1, \dots, x_n, (r_{11} \wedge \dots \wedge r_{1k_1}) \vee \dots \vee (r_{l1} \wedge \dots \wedge r_{lk_l}), \quad (1)$$

where  $r_{ij}$  is the atomic formula (without negation) and  $v$  is the only free variables associated to one or more  $r_{ij}$ .

Once the neural link predictor is obtained, each atomic formula  $r(h, t)$  can be evaluated with its probability  $p(h, r, t)$  of being true. Let  $p_{ij}$  be the probability of  $ij$ -th triple,  $\top$  and  $\perp$  be the  $t$ -norm and  $t$ -conorm [?], we rewrite the EPFO query in Equation (1) into the optimization problem:

$$\max_{v, x_1, \dots, x_n} (p_{11} \top \dots \top p_{1k_1}) \perp \dots \perp (p_{l1} \top \dots \top p_{lk_l}). \quad (2)$$

The objective in Problem (2) is continuous since  $p$ ,  $\top$ , and  $\perp$  is continuous. Embedding of variables  $v, x_1, \dots, x_n$  will be optimized over the continuous embedding space with gradient-based optimization or searched on the discrete set  $\mathcal{E}$  with beam-search. Therefore, complex structures of logical queries are handled by solving the optimization problem with the compositional objective function. CQD is also applicable to EPFO queries with multiple free variables in principle. However, this is not empirically justified because of the lack of corresponding datasets. The drawback of CQD is also apparent: CQD eventually estimates the embedding of variables, which remains a significant gap in answering aggregate queries.

#### 4.2 Query Embedding (QE)

In contrast to CQD, query embedding methods use explicit representations of the query structures. Direct inference of the learning-based model over specific query representation estimates the embedding of the target variable, i.e., the query embedding. Then the answers can be retrieved according to the query embedding. The general advantages and disadvantages of QE methods are also clear. On the one hand, models are inferred

Natural Language: Which cities are located on rivers that flow through Germany and France?

Logical Query:  $\phi(v) = \exists x \text{ HasRiver}(\text{Germany}, x) \wedge \text{HasRiver}(\text{France}, x) \wedge \text{PassesCity}(x, v)$

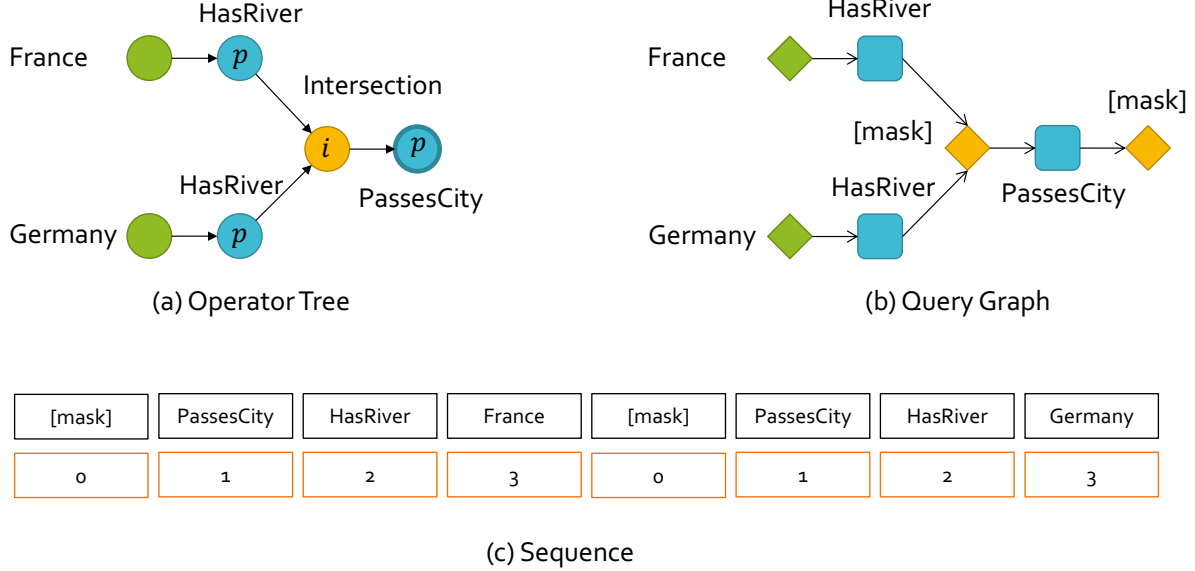


Figure 2: Three ways to represent logical query in the query embedding methods. We see the logical query is converted into three formats, including (a) operator tree [55], where each node is a set operator; (b) query graph [34], where the nodes are terms and predicates; (c) sequences [28], where a logical query is transformed as the concatenation of several multi-hop queries with a special positional encoding.

only once to obtain the answers. On the other hand, specific representation restricts the scope of solvable queries. In the current stage, there is no dominant representation of the queries. Figure 2 summarizes popular choices including operator trees [55], query graphs [34], and sequences [28]. Then we categorize different query embedding methods by their query representations and discuss each type from four perspectives: query representation, model design, training, and query answering algorithms.

#### 4.2.1 Logical Queries as Operator Trees

Operator trees materialize the procedure of searching for answers as the execution of set operators, including set projection, intersection, union, and complement (with respect to a universe). Each entity itself is regarded as the simplest singleton, a set with only one element. The set projection is converted from relations by conducting a Skolemization [36] process of a logical formula. Other set operations such as intersection, union, and difference or negation are naturally induced by logical operators. Only a subset of first-order formulae can be converted into the operator trees because such a representation of logical queries imposes additional assumptions to Definition 6, as discussed in [55]. Meanwhile, the choice of set operators is not identical. For example, the set complement can be replaced by the set difference [33], and the set union can be replaced by intersection and vice versa with the set complement, according to De Morgan’s laws. The learning-based model for operation tree representation is designed to simulate the sets in the continuous embedding spaces and the set simulations with differentiable computations (usually with neural networks). Then we discuss the spaces used to represent sets and the modeling of the set operations.

**Geometric Region Embeddings.** Sets, as a collection of discrete objects for logical query answering, are naturally related to the geometrical region of some spaces. Compared to complex variable embeddings as vectors

in CQD, embedding sets as geometric regions provides a straightforward estimation of the size of the answer set. The chosen forms of geometric regions are usually simple to be parameterized, such as the boxes used by Query2Box [42] and NewLook [33], sector cones two-dimensional spaces used by ConE [64], and hyperboloid used by HypE [14]. The set projection is usually modeled by neural networks that map the parameterized regions to new ones. Intersection and union operators in NewLook are modeled by some variants of deep set network [60] to merge multiple regions into one. Notably, the design of ConE [64] allows a closed-form set complement with sector cones, while NewLook [33] models the set difference between boxes with neural networks. How to handle negation operation using hyperboloid is not yet discussed.

**Probabilistic Distribution Embeddings.** Another thread of work models sets with probability distributions, such as Beta distribution in BetaE [43], Gaussian distribution in PERM [13], and Gamma distribution in GammaE [58]. Probabilistic distributions are more sophisticated than geometrical regions in set representation because the techniques to construct new probabilistic distributions can be applied to model set operators. For example, a mixture of probabilistic distributions can be used to construct the union operation [13, 58]. Also, altering one parameter to its reciprocal is straightforward to represent the set complement for some family of distributions [43, 58]. Meanwhile, the normalized product of the Probability Distribution Functions (PDF) of Gamma distribution is also Gamma distribution. This fact enables GammaE to simplify the design of the intersection operator. Well-defined distances and divergences between distributions are powerful tools for defining the distances between the answer sets, such as Mahalanobis distance [13] and KL-divergence [43, 58]. Following the intuition of probabilistic distributions, LinE [24] considers the histogram as the unnormalized discretization of arbitrary PDF, and Query2Particles [6] handles a set of empirical samples (particles) from an arbitrary PDF. Such designs sacrifice parts of the closed-form properties but enlarge the family of underlying distributions.

**Fuzzy Vector Embeddings.** Fuzzy vector space [27] brings supreme advantages in modeling the set operations. Set intersection, union, and negation can be precisely modeled with fuzzy logical  $t$ -norms and  $t$ -conorms [20] by embedding sets as fuzzy vectors [35, 12], i.e., vectors in  $[0, 1]^d$  where  $d$  is the dimension. The one thing left is to model the set projections with neural networks. FuzzQE [12] models the set projection by relational base decomposition [46] and LogicE [35] uses simple multi-layer perceptron. In addition, LogicE [35] models the lower and the upper bounds of the fuzzy logic truth value, which quantifies the uncertainty of each embedding to answer the aggregate query.

In contrast to various designs for embedding spaces, MLPMix [3] just embedded queries as the vectors in the Euclidean space. Such a simple embedding space does not guarantee any set properties, so it requires more advanced neural networks [50] to model set operators. Notably, MLPMix still has strong performance with simple embedding space compared to its predecessors. This might suggest that even the simplest Euclidean space is large enough. The overall performance results from the trade-off between set embeddings and set operations.

Neural symbolic methods also shed light on another way to improve performances. “Neural” indicates methods answering queries in low-dimensional embedding spaces and neural networks, while “symbolic” indicates referring to the original symbolic knowledge graphs. GNN-QE [65] estimates the probabilities of all entities given the set projection with graph neural network, NBFNet [66] over the observed knowledge graph. ENeSy [57] ensembles the neural model prediction and symbolic model prediction at each set projection, where the symbolic part is conducted by the TensorLog [15]. These methods show outstanding performances compared to the neural query embedding methods. However, scalability is always an issue since the intermediate states will unavoidably grow with the size of the underlying knowledge graph.

While various models for operator trees were proposed, the method to train the model is almost the same as the first proposed to train Query2box [42]. Let  $D(e, q)$  be the distance function between the entity  $e$  and the query embedding  $q$ , the objective function is

$$\ell_{\theta}(a, q) = -\log \sigma(\gamma - D(a, q)) - \frac{1}{k} \sum_{i=1}^k \log \sigma(D(v_i, q) - \gamma), \quad (3)$$



where  $\sigma = \frac{1}{1+e^{-x}}$  is the sigmoid function,  $\gamma$  is the margin,  $a$  is an arbitrary answer, and  $v_i$  are  $k$  negative sampled answers.

#### 4.2.2 Logical Queries as Query Graphs

Query graph presentation represents the terms and predicates as nodes and edges [16, 34]. Query graph representation does not assume the acyclicity as operator tree representation. Therefore it represents a larger set of logical queries. However, query graphs are not related to the disjunctive and negation connectives. The disjunction can be handled by converting existing queries to DNF queries (without negation), whose answer set is the combination of all conjunctive queries that can be addressed in the query graphs. How to represent the negation in the query graph has not been discussed yet.

The learning-based models for query graphs are Graph Neural Networks (GNN). MPQE [16] uses message passing networks and kgTransformer [34] uses graph transformers whose multi-layer perceptrons are upgraded by a Mixture of Experts (MoE) architecture. Besides the negative sampling objectives shown in Equation (3), kgTransformer proposed to first pretrain the model on sampled subgraphs and then finetune the model with the negative sampling loss on query answering data.

#### 4.2.3 Logical Queries as Sequences

The last type of representation is to represent the queries in sequences. BiQE [28], see Figure 2 (c), proposes to convert the query graph to multiple paths from the answer node to the anchor entity nodes. Compared to the query graphs, the sequence representation even neglects the existential variables, only the anchor entity and the relations on its path to the answer node remain. Relative positional embeddings are also used to describe the distance from each token to the answer node. This representation also focuses on conjunctive queries, which do not contain disjunction and negation. Queries with disjunctions can be handled by combining conjunctive queries in the DNF. However, it is still not known how to handle the negation operation.

Once the sequence is defined, the sequence can be put into the transformer models. The output embedding of the `[mask]` token is aggregated to obtain the answer. Unlike the query graph representation that enforces the query structures explicitly, transformers could learn implicit logical structures via the self-attention mechanism.

## 5 Future Directions

Research about knowledge graphs logical query answering, though rapidly developed in recent years, is still limited compared to the well-established study about query answering on the relational database. For example, the query types that can be answered by the model are strict fragments of first-order queries. In this section, we discuss what is left in current research compared to logical query answering on incomplete relational data, which is the ultimate goal of this line of research. Formal definitions from model theory in Section 2 picture the gap between existing works and the ambitious goal. we summarize the limitations and discuss possible future directions from the data, query, and model perspectives.

### 5.1 Towards General Relational Data

A relational language  $\mathcal{L}$  can describe general relational data. However, the language  $\mathcal{L}_{KG}$  for knowledge graphs is defined with three additional assumptions, see Section 2. By rethinking three assumptions about  $\mathcal{L}_{KG}$  and the open world assumption, we could extend the existing data model of knowledge graphs.

**KG with Relations of  $n_r > 2$ .** Knowledge graphs with  $n$ -ary relations [63] relax the first assumption that  $n_r = 2, \forall R \in \mathcal{R}$  by accepting the relations whose  $2 \leq n_r \leq n$ . Thus, each element of the knowledge graph

now is an  $n + 1$  triple list that contains  $n$  entities and one relation. The concept of  $n$ -ary relation is also related to the hyper-relation [18, 2] and the knowledge graph is extended from the directed graphs to hypergraphs.

**KG with Attributes.** Attributes expand the constant set  $\mathcal{E}$  from all entities to *entities with attribute values*, thus relaxing the second assumption. Typical attributes include entity types [5, 23], triple timestamps [26, 45], and triple probabilities [11]. Additional attributes can be described in the  $n$ -ary relations. For example, the timestamp  $t$  associated with a triple  $(s, p, o)$  can be described with 4-triples  $(s, p, o, t)$ . New attributes bring new predicates, enlarge the universe set, and enrich the query semantics. Two timestamps can be compared by “before”, “equal”, and “after”, and can be composed to a time interval, which also introduces new predicates such as “during”. The triple probabilities, as discussed in probabilistic databases [53], enable the estimation of the probabilities for boolean queries, the probability for each answer entity for set-valued queries, and the total probability for aggregate queries.

**KG with Functions.** Introducing attributes also makes it possible to define functions between attribute values. Taking the timestamp attribute as an example, new timestamps can be computed by a time movement function with a timestamp and a time interval as inputs. These features are rigorously defined in pure symbolic systems [17]. It is worth discussing how to combine them with learning-based methods.

**More Challenging Open World Assumption.** In the inductive setting, the entity set  $\mathcal{E}$  is divided into an observed set  $\mathcal{E}_o^r$  and an inempty inductive set  $\mathcal{E}_i^r$  with respect to the relation  $r \in \mathcal{R}$ , that is  $\mathcal{E}_o^r \cap \mathcal{E}_i^r = \emptyset$ ,  $\mathcal{E}_o^r \cup \mathcal{E}_i^r = \mathcal{E}$ ,  $\forall r \in \mathcal{R}$ . Under such a setting, the open world assumption in Definition 3 is rewritten as  $I_{\mathcal{KG}_o}(r) \subseteq \mathcal{E}_o^r \times \mathcal{E}_o^r \subseteq I_{\mathcal{KG}_u}(r) \subseteq \mathcal{E} \times \mathcal{E}$ . This setting makes it very hard to estimate the relations over  $\mathcal{E} \times \mathcal{E} - \mathcal{E}_o^r \times \mathcal{E}_o^r$ .

## 5.2 Towards First Order Queries and Beyond

The semantics of the logic queries lacks serious discussion. There is no standard dataset and benchmark for first-order queries with cycles, multiple free variables, or universal quantifiers even for the simplest  $L_{KG}$  language. Moreover, it is also worth discussing first-order queries over general relational data such as KGs with attributes and attribute functions. It is also worthwhile to expand the first-order logic to fixed-point or monadic second-order logic [30].

## 5.3 Towards Versatile Learning-based Models

Existing methods are incapable of addressing more and more challenging queries and complex data discussed above. It is essential to investigate the graph or sequence-based query embedding models to solve various queries. Given the combinatorial natures of the query answering problems, new opportunities could be found with neural combinatorial solvers [7]. Meanwhile, large language models, as a proven effective way to encapsulate world knowledge [39, 10], could be considered as another approach to modeling the relational data.

## 6 Conclusion

Big data and world knowledge are usually uncertain and dynamic. Logical queries over knowledge graphs are essential ways to understand the world as a simplified data model through the lens of logic and reasoning. In this paper, we structure the existing knowledge graph logical query answering tasks and methods as a three-party game between data, query, and models. With the rigorous concepts from finite model theory [36, 30], our work portrays a detailed landscape of the achievements and limitations of existing work. Our discussion shows that there are various gaps between current research to the ambitious goal in terms of data, queries, and models. We hope our paper can be a practical guide for traveling in the uncertain and dynamic data world.

## References

- [1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, Reading, Mass, 1995.
- [2] D. Alivanistos, M. Berrendorf, M. Cochez, and M. Galkin. Query Embedding on Hyper-Relational Knowledge Graphs. In *International Conference on Learning Representations*, Mar. 2022.
- [3] A. Amayuelas, S. Zhang, S. X. Rao, and C. Zhang. Neural Methods for Logical Reasoning over Knowledge Graphs. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.
- [4] E. Arakelyan, D. Daza, P. Minervini, and M. Cochez. Complex Query Answering with Neural Link Predictors. In *International Conference on Learning Representations*, 2021.
- [5] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. In K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *The Semantic Web*, Lecture Notes in Computer Science, pages 722–735, Berlin, Heidelberg, 2007. Springer.
- [6] J. Bai, Z. Wang, H. Zhang, and Y. Song. Query2Particles: Knowledge Graph Reasoning with Particle Embeddings. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2703–2714, Seattle, United States, July 2022. Association for Computational Linguistics.
- [7] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio. Neural combinatorial optimization with reinforcement learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.
- [8] K. D. Bollacker, C. Evans, P. K. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In J. T.-L. Wang, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250. ACM, 2008.
- [9] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.
- [10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [11] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr, and T. M. Mitchell. Toward an Architecture for Never-Ending Language Learning. In M. Fox and D. Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010.
- [12] X. Chen, Z. Hu, and Y. Sun. Fuzzy Logic Based Logical Query Answering on Knowledge Graphs. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 3939–3948. AAAI Press, 2022.
- [13] N. Choudhary, N. Rao, S. Katariya, K. Subbian, and C. Reddy. Probabilistic Entity Representation Model for Reasoning over Knowledge Graphs. In *Advances in Neural Information Processing Systems*, volume 34, pages 23440–23451. Curran Associates, Inc., 2021.
- [14] N. Choudhary, N. Rao, S. Katariya, K. Subbian, and C. K. Reddy. Self-Supervised Hyperboloid Representations from Logical Queries over Knowledge Graphs. In *Proceedings of the Web Conference 2021, WWW '21*, pages 1373–1384, New York, NY, USA, June 2021. Association for Computing Machinery.
- [15] W. Cohen, F. Yang, and K. R. Mazaitis. TensorLog: A Probabilistic Database Implemented Using Deep-Learning Infrastructure. *Journal of Artificial Intelligence Research*, 67:285–325, Feb. 2020.
- [16] D. Daza and M. Cochez. Message Passing Query Embedding, June 2020.
- [17] E. A. Emerson. Temporal and modal logic. In *Formal Models and Semantics*, pages 995–1072. Elsevier, 1990.

- [18] M. Galkin, P. Trivedi, G. Maheshwari, R. Usbeck, and J. Lehmann. Message Passing for Hyper-Relational Knowledge Graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7346–7359, Online, Nov. 2020. Association for Computational Linguistics.
- [19] M. Galkin, Z. Zhu, H. Ren, and J. Tang. Inductive Logical Query Answering in Knowledge Graphs. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2022 NeurIPS 2022, November 27- December 9, 2022, New Orleans*, page 25, 2022.
- [20] P. Hájek. *Metamathematics of Fuzzy Logic*, volume 4 of *Trends in Logic*. Springer Netherlands, Dordrecht, 1998.
- [21] W. L. Hamilton, P. Bajaj, M. Zitnik, D. Jurafsky, and J. Leskovec. Embedding Logical Queries on Knowledge Graphs. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2030–2041, 2018.
- [22] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Advances in Neural Information Processing Systems*, volume 33, pages 22118–22133. Curran Associates, Inc., 2020.
- [23] Z. Hu, V. Gutierrez Basulto, Z. Xiang, X. Li, R. Li, and J. Z. Pan. Type-aware Embeddings for Multi-Hop Reasoning over Knowledge Graphs. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 3078–3084, Vienna, Austria, July 2022. International Joint Conferences on Artificial Intelligence Organization.
- [24] Z. Huang, M.-F. Chiang, and W.-C. Lee. LinE: Logical Query Reasoning over Hierarchical Knowledge Graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22*, pages 615–625, New York, NY, USA, Aug. 2022. Association for Computing Machinery.
- [25] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, Feb. 2022.
- [26] Z. Jia, S. Pramanik, R. S. Roy, and G. Weikum. Complex Temporal Question Answering on Knowledge Graphs. In G. Demartini, G. Zuccon, J. S. Culpepper, Z. Huang, and H. Tong, editors, *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 792–802. ACM, 2021.
- [27] A. K. Katsaras and D. B. Liu. Fuzzy vector spaces and fuzzy topological vector spaces. *Journal of Mathematical Analysis and Applications*, 58(1):135–146, Mar. 1977.
- [28] B. Kotnis, C. Lawrence, and M. Niepert. Answering Complex Queries in Knowledge Graphs with Bidirectional Sequence Encoders. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(6):4968–4977, May 2021.
- [29] D. M. Kroenke, D. J. Auer, S. L. Vandenberg, and R. C. Yoder. *Database Processing: Fundamentals, Design, and Implementation*. Pearson, NY NY, 15th edition, 40th anniversary edition edition, 2018.
- [30] L. Libkin. *Elements of Finite Model Theory*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [31] L. Libkin and C. Sirangelo. Open and Closed World Assumptions in Data Exchange. In B. C. Grau, I. Horrocks, B. Motik, and U. Sattler, editors, *Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.
- [32] B. Y. Lin, X. Chen, J. Chen, and X. Ren. KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning. *EMNLP/IJCNLP*, 2019.
- [33] L. Liu, B. Du, H. Ji, C. Zhai, and H. Tong. Neural-Answering Logical Queries on Knowledge Graphs. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21*, pages 1087–1097, New York, NY, USA, Aug. 2021. Association for Computing Machinery.
- [34] X. Liu, S. Zhao, K. Su, Y. Cen, J. Qiu, M. Zhang, W. Wu, Y. Dong, and J. Tang. Mask and Reason: Pre-Training Knowledge Graph Transformers for Complex Logical Queries. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22*, pages 1120–1130, New York, NY, USA, Aug. 2022. Association for Computing Machinery.
- [35] F. Luus, P. Sen, P. Kapanipathi, R. Riegel, N. Makondo, T. Lebesse, and A. Gray. Logic Embeddings for Complex Query Answering. *arXiv:2103.00418 [cs]*, Feb. 2021.

- [36] D. Marker. *Model Theory: An Introduction*. Number 217 in Graduate Texts in Mathematics. Springer, New York, 2002.
- [37] G. A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, Nov. 1995.
- [38] T. Pellissier Tanon, D. Vrandečić, S. Schaffert, T. Steiner, and L. Pintscher. From Freebase to Wikidata: The Great Migration. In *Proceedings of the 25th International Conference on World Wide Web, WWW '16*, pages 1419–1428, Republic and Canton of Geneva, CHE, Apr. 2016. International World Wide Web Conferences Steering Committee.
- [39] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller. Language models as knowledge bases? 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)*, pages 2463–2473, 2019.
- [40] H. Ren, H. Dai, B. Dai, X. Chen, M. Yasunaga, H. Sun, D. Schuurmans, J. Leskovec, and D. Zhou. LEGO: Latent Execution-Guided Reasoning for Multi-Hop Question Answering on Knowledge Graphs. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8959–8970. PMLR, July 2021.
- [41] H. Ren, H. Dai, B. Dai, X. Chen, D. Zhou, J. Leskovec, and D. Schuurmans. SMORE: Knowledge Graph Completion and Multi-hop Reasoning in Massive Knowledge Graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22*, pages 1472–1482, New York, NY, USA, Aug. 2022. Association for Computing Machinery.
- [42] H. Ren, W. Hu, and J. Leskovec. Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.
- [43] H. Ren and J. Leskovec. Beta Embeddings for Multi-Hop Logical Reasoning in Knowledge Graphs. In H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, and H.-T. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual*, 2020.
- [44] D. Ruffinelli, S. Broscheit, and R. Gemulla. You CAN Teach an Old Dog New Tricks! On Training Knowledge Graph Embeddings. In *International Conference on Learning Representations*, Mar. 2020.
- [45] A. Saxena, S. Chakrabarti, and P. P. Talukdar. Question Answering Over Temporal Knowledge Graphs. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6663–6676. Association for Computational Linguistics, 2021.
- [46] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling. Modeling Relational Data with Graph Convolutional Networks. In A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, editors, *The Semantic Web, Lecture Notes in Computer Science*, pages 593–607, Cham, 2018. Springer International Publishing.
- [47] Y. Sun, Q. Shi, L. Qi, and Y. Zhang. JointLK: Joint Reasoning with Language Models and Knowledge Graphs for Commonsense Question Answering. In M. Carpuat, M.-C. de Marneffe, and I. V. M. Ruíz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 5049–5060. Association for Computational Linguistics, 2022.
- [48] A. Talmor and J. Berant. The Web as a Knowledge-Base for Answering Complex Questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
- [49] K. Teru, E. Denis, and W. L. Hamilton. Inductive Relation Prediction by Subgraph Reasoning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9448–9457. PMLR, Nov. 2020.
- [50] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021.

- [51] K. Toutanova and D. Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality*, pages 57–66, Beijing, China, July 2015. Association for Computational Linguistics.
- [52] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard. Complex Embeddings for Simple Link Prediction. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2071–2080. PMLR, June 2016.
- [53] G. Van den Broeck and D. Suciu. Query Processing on Probabilistic Data: A Survey. *Foundations and Trends® in Databases*, 7(3-4):197–341, 2017.
- [54] D. Vrandečić and M. Krötzsch. Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, Sept. 2014.
- [55] Z. Wang, H. Yin, and Y. Song. Benchmarking the Combinatorial Generalizability of Complex Query Answering on Knowledge Graphs. In J. Vanschoren and S.-K. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, Virtual*, 2021.
- [56] W. Xiong, T. Hoang, and W. Y. Wang. DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 564–573, 2017.
- [57] Z. Xu, W. Zhang, P. Ye, H. Chen, and H. Chen. Neural-Symbolic Entangled Framework for Complex Query Answering. In *Advances in Neural Information Processing Systems*, Oct. 2022.
- [58] D. Yang, P. Qing, Y. Li, H. Lu, and X. Lin. GammaE: Gamma Embeddings for Logical Queries on Knowledge Graphs, Oct. 2022.
- [59] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 535–546. Association for Computational Linguistics, 2021.
- [60] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. R. Salakhutdinov, and A. J. Smola. Deep Sets. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [61] R. Zhang, D. Hristovski, D. Schutte, A. Kastrin, M. Fiszman, and H. Kilicoglu. Drug repurposing for COVID-19 via knowledge graph completion. *Journal of Biomedical Informatics*, 115:103696, Mar. 2021.
- [62] W. Zhang, J. Chen, J. Li, Z. Xu, J. Z. Pan, and H. Chen. Knowledge Graph Reasoning with Logics and Embeddings: Survey and Perspective. *arXiv:2202.07412 [cs]*, Feb. 2022.
- [63] Y. Zhang, P. Li, H. Liang, A. Jatowt, and Z. Yang. Fact-Tree Reasoning for N-ary Question Answering over Knowledge Graphs. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 788–802. Association for Computational Linguistics, 2022.
- [64] Z. Zhang, J. Wang, J. Chen, S. Ji, and F. Wu. ConE: Cone Embeddings for Multi-Hop Reasoning over Knowledge Graphs. In *Advances in Neural Information Processing Systems*, volume 34, pages 19172–19183. Curran Associates, Inc., 2021.
- [65] Z. Zhu, M. Galkin, Z. Zhang, and J. Tang. Neural-Symbolic Models for Logical Queries on Knowledge Graphs, May 2022.
- [66] Z. Zhu, Z. Zhang, L.-P. Xhonneux, and J. Tang. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. In *Advances in Neural Information Processing Systems*, volume 34, pages 29476–29490. Curran Associates, Inc., 2021.