Combining Inductive and Deductive Reasoning for Query Answering over Incomplete Knowledge Graphs

Medina Andresel*
AIT Austrian Institute of Technology
Vienna, Austria
medina.andresel@ait.ac.at

Trung-Kien Tran
Bosch Center for Artificial Intelligence
Renningen, Germany
trungkien.tran@de.bosch.com

Csaba Domokos
Bosch Center for Artificial Intelligence
Renningen, Germany
csaba.domokos@de.bosch.com

Pasquale Minervini
University of Edinburgh
Edinburgh, United Kingdom
p.minervini@ed.ac.uk

Daria Stepanova
Bosch Center for Artificial Intelligence
Renningen, Germany
daria.stepanova@de.bosch.com

ABSTRACT

Current methods for embedding-based query answering over incomplete Knowledge Graphs (KGs) only focus on inductive reasoning, i.e., predicting answers by learning patterns from the data, and lack the complementary ability to do deductive reasoning, which requires the application of domain knowledge to infer further information. To address this shortcoming, we investigate the problem of incorporating ontologies into embedding-based query answering models by defining the task of embedding-based ontology-mediated query answering. We propose various integration strategies into prominent representatives of embedding models that involve (1) different ontology-driven data augmentation techniques and (2) adaptation of the loss function to enforce the ontology axioms. We design novel benchmarks for the considered task based on the LUBM and the NELL KGs and evaluate our methods on them. The achieved improvements in the setting that requires both inductive and deductive reasoning are from 20% to 55% in HITS@3.

CCS CONCEPTS

• **Computing methodologies** → *Description logics*; *Machine learning approaches*; • **Computer systems organization** → *Neural networks.*

KEYWORDS

Knowledge Graphs, Ontologies, Embeddings, Query Answering, Neuro-Symbolic AI

*The work has been done during the PhD sabbatical of Medina Andresel at Bosch Center for Artificial Intelligence

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614816

ACM Reference Format:
Medina Andresel, Trung-Kien Tran, Csaba Domokos, Pasquale Minervini, and Daria Stepanova. 2023. Combining Inductive and Deductive Reasoning for Query Answering over Incomplete Knowledge Graphs. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614816

1 INTRODUCTION

Knowledge Graphs (KGs) have recently received much attention due to their relevance in various applications, such as natural question answering or web search. Prominent KGs include NELL [8], YAGO [31], and Wikidata [12]. A KG describes facts about entities by interconnecting them via relations, e.g., *hasAlumnus*(*mit*, *bob*) in Figure 1 states that Bob is an MIT alumnus.

A crucial task in leveraging information from knowledge graphs is that of answering logical queries such as *Who works for Amazon and has a degree from MIT?*, which can be formally written as $q(X) \leftarrow \mathit{degreeFrom}(X, \mathit{mit}) \land \mathit{worksFor}(X, \mathit{amazon})$. Answering such queries is very challenging when KGs are incomplete, which is often the case due to their (semi-) automatic construction, and obtaining complete answers typically requires further domain knowledge, i.e., the application of deductive reasoning. For instance, *mary* is a missing but desired answer of q that can be obtained by combining the fact *managerAt*(*mary*, *amazon*), predicted using machine learning models, and the axiom stating that *managerAt* implies *worksFor* in the ontology $\mathcal{O}$ of Figure 1. Therefore, such a task requires both inductive and deductive reasoning.

Recently, *Knowledge Graph Embedding* (KGE) techniques [33] that can be used to predict missing facts have been proposed to answer logical queries over incomplete KGs [20, 30, 36, 37, 39]. At the same time, in the Knowledge Representation and Reasoning area answering queries over incomplete data has also received a lot of attention and one of the most successful approaches for this task is to exploit ontologies when querying KGs, referred to as *Ontology-Mediated Query Answering* [OMQA, 38].

While promising, existing embedding-based methods do not take ontologies, which formalize domain knowledge, into account. Since large portions of expert knowledge can be conveniently encoded using ontologies, the benefits of coupling ontology reasoning and

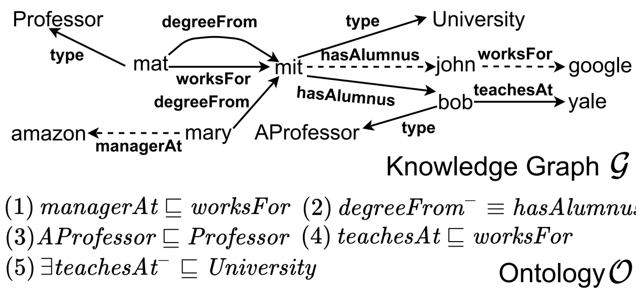

Figure 1: An exemplary ontology O and a KG $\mathcal{G}$. Solid edges illustrate existing facts in $\mathcal{G}$, and dashed ones indicate missing facts that could be predicted using KG embeddings.

embedding methods for KG completion are evident and have been acknowledged in several works, e.g. [18, 28]. However, to the best of our knowledge, coupling inductive and deductive reasoning to answer queries over incomplete KGs has not been considered yet.

Answering queries over the KG augmented with triples resulting from the naive process of interchangeably using embedding methods and ontology reasoning, comes with a big scalability challenge [27] and commonly known error accumulation issues. In practice, we need to restrict ourselves to computing merely small subsets of likely fact predictions required for answering a given query; thus more sophisticated proposals are needed. Hence, we investigate three open questions: (1) How to adapt existing OMQA techniques to the setting of KGEs? (2) How do different data augmentation strategies impact the accuracy of existing embedding models for the OMQA task? (3) Does enforcing ontology axioms in the embedding space via loss function help to improve inductive and deductive reasoning performance?

We answer (1)-(3) by the following contributions:

- We formally define the novel task of *Embedding-Based Ontology-Mediated Query Answering* (E-OMQA), analyze and systematically compare various extensions of embedding-based query answering models to incorporate ontologies.
- We propose ontology-driven strategies for sampling queries to train embedding models for query answering, as well as a loss function modification to enforce the ontology axioms within the embedding space, and demonstrate the effectiveness of these proposals on widely-used representatives of query-based and atom-based models.
- As no previous benchmarks exist for E-OMQA, we design new ones using LUBM and NELL, i.e., well-known benchmarks for OMQA and embedding models, respectively.
- Extensive evaluation shows that enforcing the ontology via the loss function, in general, improves the deductive power regardless of how the training data is sampled, while ontology-driven sampling strategy has a further significant positive impact on performance. We obtain overall improvements, ranging from 20% to 55% in HITS@3, in the settings that require both inductive and deductive reasoning.

Table 1: DL syntax and semantics defined using FO interpretations $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ with a non-empty domain $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$. C and D denote concepts in *DL-Lite$_{\mathcal{R}}$*.

DL Syntax	Semantics
e	$e^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
A (resp. p)	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ (resp. $p^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$)
$\exists p$	$(\exists p)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \exists d', (d, d') \in p^{\mathcal{I}}\}$
p^{-}	$(p^{-})^{\mathcal{I}} = \{(d', d) \mid (d, d') \in p^{\mathcal{I}}\}$.
$C \sqsubseteq D$ (resp. $p \sqsubseteq s$)	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (resp. $p^{\mathcal{I}} \subseteq s^{\mathcal{I}}$)
$A(c)$ (resp. $p(c, c')$)	$c \in A^{\mathcal{I}}$ (resp. $(c, c') \in p^{\mathcal{I}}$)

2 PRELIMINARIES

Knowledge Graphs and Ontologies. We assume a signature consisting of countable pairwise disjoint sets $\mathbf{E}, \mathbf{C}$, and $\mathbf{R}$ of entities (constants), concepts (types), and roles (binary relations), respectively. A knowledge graph $\mathcal{G}$ (*a.k.a.* ABox) is a set of triples, such as $(mit, type, University)$ and $(bob, worksFor, mit)$, where $mit, bob \in \mathbf{E}$, $worksFor, type \in \mathbf{R}$, and $University \in \mathbf{C}$. These triples can also be represented as $University(mit)$[1] and $worksFor(bob, mit)$. An ontology O (*a.k.a.* TBox) is a set of axioms in Description Logics [4] over the signature $\Sigma = \langle \mathbf{E}, \mathbf{C}, \mathbf{R} \rangle$. We focus on ontologies in *DL-Lite$_{\mathcal{R}}$* DL fragment [7] that have the following syntax:

$$A \sqsubseteq A' \quad A \sqsubseteq \exists p \quad \exists p \sqsubseteq A \quad \exists p^{-} \sqsubseteq A \quad p \sqsubseteq s \quad p^{-} \sqsubseteq s,$$

where $A, A' \in \mathbf{C}$ are concepts and $p, s \in \mathbf{R}$ are roles and p^{-} denotes the inverse relation of p. The KG and its ontology from Figure 1 are in *DL-Lite$_{\mathcal{R}}$*. The *DL-Lite$_{\mathcal{R}}$* syntax and semantics are summarized in Table 1. Given a KG $\mathcal{G}$ and an ontology O, an interpretation $\mathcal{I}$ *is a model of* $\mathcal{G}$ *w.r.t* O if $\mathcal{I}$ satisfies each fact in $\mathcal{G}$ and each axiom in O. For *DL-Lite$_{\mathcal{R}}$*, a *canonical model* exists that can be homomorphically mapped into any other model, obtained from the *deductive closure* $O^{\infty}(\mathcal{G})$, which extends $\mathcal{G}$ with triples derived from existing triples in $\mathcal{G}$ by exhaustively applying axioms in O [7].

Ontology-Mediated Query Answering. A *query atom* is an expression of the form $p(T_1, T_2)$, where $p \in \mathbf{R}$, and each $T_i \in \mathbf{V} \cup \mathbf{E}$ is called a *term*, with $\mathbf{V}$ disjoint with $\mathbf{E}, \mathbf{C}$, and $\mathbf{R}$ being a set of variables. A *monadic conjunctive query* (CQ) $q(X)$ is a First-Order (FO) formula of the form $q(X) \leftarrow \exists \vec{Y}.p_1(\vec{T_1}) \wedge \cdots \wedge p_n(\vec{T_n})$ where each $p_i(\vec{T_i})$ is a query atom, and $vars(q) = \{X\} \cup \vec{Y}$ denotes the set of variables appearing in q, with $X \notin \vec{Y}$ being the *answer variable*. In this work, we focus on *monadic Existential Positive FO* (EPFO) queries, i.e., unions of monadic CQs [10]. For a query $q(X)$ and a KG $\mathcal{G}$, a constant a is an answer of q if a mapping $\pi : var(q) \mapsto \mathbf{E}$ exists, s.t. $q\pi \in \mathcal{G}$ and $\pi(X) = a$; $q[\mathcal{G}]$ are the answers of q on $\mathcal{G}$.

Ontology-Mediated Query Answering (OMQA) concerns answering queries by accounting for both the KG and the accompanying ontology. Since the model constructed from $O^{\infty}(\mathcal{G})$ can be homomorphically mapped to every other model, the deductive closure can be used to evaluate queries [7].

Definition 2.1. Given a KG $\mathcal{G}$ and an ontology O, an entity a from $\mathcal{G}$ is a *certain answer* of $q(X)$ over $(\mathcal{G}, O)$ if a is an answer to $q(X)$ over $O^{\infty}(\mathcal{G})$. We use $q[\mathcal{G}, O]$ to denote the set of *certain answers of* q *over* $(\mathcal{G}, O)$.

[1] Unary facts can also be modeled using the binary *type* relation.

Let q and q' be two monadic queries over $(\mathcal{G}, \mathcal{O})$, then q is *contained* in q' w.r.t. $\mathcal{O}$ if $q[\mathcal{G}, \mathcal{O}] \subseteq q'[\mathcal{G}, \mathcal{O}]$; we call q a *specialization* of q' (written as $q' \overset{s}{\rightsquigarrow} q$), and q' a *generalization* of q (written as $q \overset{g}{\rightsquigarrow} q'$). Query generalizations and specializations can be obtained by exploiting ontology axioms; such process (and result) is referred to as *query rewriting*.

Example 2.2. Consider the KG $\mathcal{G}$ in Figure 1 and the query $q(X) \leftarrow \text{type}(X, \text{Professor}) \wedge \text{degreeFrom}(X, \text{mit})$. Since mat $\in q[\mathcal{G}]$, it is a certain answer. Moreover, according to $\mathcal{O}$,AProfessor is a sub-type of Professor and degreeFrom is inverse of hasAlumnus, thus bob is also a certain answer. For $q'(X) \leftarrow \text{type}(X, \text{AProfessor}) \wedge \text{degreeFrom}(X, \text{mit})$ it holds that $q \overset{s}{\rightsquigarrow} q'$ as mat $\notin q'[\mathcal{G}, \mathcal{O}]$.

Embedding-Based Approximate Query Answering. Since, in reality, KGs might be missing facts, existing query answering techniques designed for complete data might not compute all answers. In such settings, one assumes that the given KG $\mathcal{G}$ is a subset of a complete but unobservable KG $\mathcal{G}^i$, and one aims at estimating the likely answers to q over $\mathcal{G}^i$. E.g., $\mathcal{G}^i$ for the graph $\mathcal{G}$ given in Figure 1 includes the links denoted by dashed edges. In practice, to evaluate the accuracy of a considered method, $\mathcal{G}^i$ is typically fixed at the beginning, and $\mathcal{G}$ is created by removing facts from $\mathcal{G}^i$.

The set of all answers to a given query q comprises those that can be obtained by directly querying the given KG $\mathcal{G}$ only and those that require predicting missing KG facts. Thus, one typically distinguishes easy and hard answers as follows:

Definition 2.3. Given a KG $\mathcal{G}$, a subgraph of a complete but unobservable KG $\mathcal{G}^i$ and a query $q(X)$, a is an *easy answer* to q if $a \in q[\mathcal{G}]$, and it is a *hard answer* to q if $a \in q[\mathcal{G}^i] \backslash q[\mathcal{G}]$.

Recently, embedding-based methods have been proposed for *approximate answering of existential positive FO queries* over incomplete KGs [9, 25, 30, 36, 37]. Broadly, such methods can be divided into two categories: *query-based* [9, 25, 30, 36, 37] and *atom-based* [2]. Generally, any neural QA model relies on an embedding function which maps entities and relations into a d-dimensional embedding space. It then computes a score of each entity c for being an answer to a given query q over $\mathcal{G}^i$ via a scoring function $\phi_q(\mathbf{c}) : \mathbb{R}^d \mapsto [0, 1]$, where $\mathbf{c}$ denotes the embedding vector of c.[2] Using these scoring functions, the final *embedding QA function* $\mathcal{E}_\mathcal{G}$ takes as input a query and returns its approximate answers over the knowledge graph $\mathcal{G}^i$, i.e., answers that have the scoring above some predefined threshold. We say that $\mathcal{E}_\mathcal{G}$ is *reliable w.r.t.* $\mathcal{G}^i$ whenever for each query q, c is an approximate answer to q iff c is an answer to q over $\mathcal{G}^i$. Clearly, the challenge of identifying hard answers is still valid also for embedding QA models.

3 EMBEDDING-BASED OMQA

Existing methods for embedding-based query answering compute approximate answers to queries over an unobservable KG $\mathcal{G}^i$ by performing inductive reasoning. However, they are not capable of simultaneously applying deductive reasoning, and thus cannot account for ontologies with which KGs are often accompanied.

To address this shortcoming, we propose ways to combine inductive and deductive reasoning for approximate query answering over incomplete KGs. For that, we first formalize the task of *Embedding-based Ontology-Mediated Query Answering (E-OMQA)* in which both types of reasoning are exploited. The goal of this task is to approximate certain answers to OMQs over $\mathcal{G}^i$.

Definition 3.1 (E-OMQA). Let $\mathcal{G}$ be a KG, which is a subgraph of a complete but not observable KG $\mathcal{G}^i$, let $\mathcal{O}$ be an ontology and q a query. *Embedding-based ontology-mediated query answering* is concerned with constructing an embedding function $\mathcal{E}_{\mathcal{G},\mathcal{O}}$ that is reliable w.r.t. $\mathcal{O}^\infty(\mathcal{G}^i)$.

Note that, $q[\mathcal{G}^i, \mathcal{O}]$ subsumes both $q[\mathcal{G}^i]$, the answers requiring inductive reasoning, and $q[\mathcal{G}, \mathcal{O}]$, the answers computed via deductive reasoning only. Analogously as for embedding-based query answering, for E-OMQA, we distinguish between easy certain answers and hard certain answers as follows.

Definition 3.2. Given a KG $\mathcal{G}$, a subgraph of a complete but unobservable KG $\mathcal{G}^i$, an ontology $\mathcal{O}$ and a query $q(X)$, a is an *easy certain answer* to q if $a \in q[\mathcal{G}, \mathcal{O}]$, and it is a *hard certain answer* to q if $a \in q[\mathcal{G}^i, \mathcal{O}] \backslash q[\mathcal{G}, \mathcal{O}]$.

Next, we discuss several embedding-based methods for ontology-mediated query answering under incompleteness.

Query Rewriting over Pre-trained Models. In the traditional OMQA setting, each query q can be evaluated by first rewriting q into a set of FO-queries $Q_\mathcal{O}$ and then evaluating each query in $Q_\mathcal{O}$ over $\mathcal{G}$ alone. For E-OMQA, this amounts to constructing an embedding QA function $\mathcal{E}_\mathcal{G}$ for $\mathcal{G}$ alone and using it to compute the answers to all queries in $Q_\mathcal{O}$. For *DL-Lite*$_\mathcal{R}$, such FO-rewriting is obtained by extensively applying ontology axioms in a specializing fashion, which results in the so-called *perfect reformulation* [7].

Ontology-Aware Models. An alternative to query rewriting is to develop an embedding query answering function that accounts for axioms in $\mathcal{O}$. To the best of our knowledge, there are no KGE models that directly address the problem of E-OMQA. Thus, we suggest the following: (1) Train existing embedding models for QA on the data derived from $\mathcal{O}^\infty(\mathcal{G})$ instead of $\mathcal{G}$; (2) Develop an *ontology-aware* embedding model that will be trained on $\mathcal{G}$ but will have special terms in the training objective structurally enforcing $\mathcal{O}$. (3) Combine (1) and (2), i.e., train ontology-aware embedding models on the data derived from $\mathcal{O}^\infty(\mathcal{G})$.

While the proposed approaches can be realized on top of any embedding model for logical query answering, in this work, we verify their effectiveness on a query-based model *Query2Box* [36] and an atom-based model *CQD* [2]. In Section 3.1, we present several effective ontology-driven training methods for realizing (1). As for (2), building on *Query2Box*, in Section 3.2 we develop its ontology-aware version. Moreover, we build an ontology-aware extension of *CQD*, on top of the neural link predictor using *adversarial sets regularization* (ASR) [32] to enforce the ontology axioms. We chose this approach, since it is general and allows us to incorporate rules into any off-the-shelf neural link predictor. In our experiments, we use ComplEx-N3 [29] as it requires minimal modification to CQD and outperforms other neural link predictors (see [29]). Finally, we evaluate the effectiveness of our ontology-driven strategies from Section 3.1 on the extended models described in Section 3.2 and

[2] Bold small letters denote vector representations.

Table 2: Rules to specialize and generalize an atom β from $q(X) \leftarrow \alpha \wedge \beta$, where $A, B \in \mathbf{C}$, $p, r, s \in \mathbf{R}$ and $T, T_1, T_2 \in vars(q) \cup \mathbf{E}$. The operators $\overset{s}{\rightsquigarrow}$ and $\overset{g}{\rightsquigarrow}$ are used for constructing specializations and generalizations respectively of a given query.

(R1) If $A \sqsubseteq B \in \mathcal{O}$	then:	$\alpha \wedge \text{type}(T,B) \overset{s}{\rightsquigarrow} \alpha \wedge \text{type}(T,A)$	$\alpha \wedge \text{type}(T,A) \overset{g}{\rightsquigarrow} \alpha \wedge \text{type}(T,B)$
(R2) If $\exists p \sqsubseteq A \in \mathcal{O}$	then:	$\alpha \wedge \text{type}(T_1,A) \overset{s}{\rightsquigarrow} \alpha \wedge p(T_1,T_2)$	$\alpha \wedge p(T_1,T_2) \overset{g}{\rightsquigarrow} \alpha \wedge \text{type}(T_1,A)$
(R3) If $A \sqsubseteq \exists p \in \mathcal{O}$	then:	$\alpha \wedge p(T_1,T_2) \overset{s}{\rightsquigarrow} \alpha \wedge \text{type}(T,A)$	$\alpha \wedge \text{type}(T_1,A) \overset{g}{\rightsquigarrow} \alpha \wedge p(T_1,T_2)$
(R4) If $\exists p^- \sqsubseteq A \in \mathcal{O}$	then:	$\alpha \wedge A(T) \overset{s}{\rightsquigarrow} \alpha \wedge p(T_2,T_1)$	$\alpha \wedge p(T_2,T_1) \overset{g}{\rightsquigarrow} \alpha \wedge \text{type}(T_1,A)$
(R5) If $A \sqsubseteq \exists p \in \mathcal{O}$	then:	$\alpha \wedge p(T_1,T_2) \overset{s}{\rightsquigarrow} \alpha \wedge \text{type}(T,A)$	$\alpha \wedge \text{type}(T_1,A) \overset{g}{\rightsquigarrow} \alpha \wedge p(T_1,T_2)$
(R6) If $p \sqsubseteq s \in \mathcal{O}$	then:	$\alpha \wedge s(T_1,T_2) \overset{s}{\rightsquigarrow} \alpha \wedge p(T_1,T_2)$	$\alpha \wedge p(T_1,T_2) \overset{g}{\rightsquigarrow} \alpha \wedge s(T_1,T_2)$
(R7) If $s^- \sqsubseteq p \in \mathcal{O}$	then:	$\alpha \wedge p(T_1,T_2) \overset{s}{\rightsquigarrow} \alpha \wedge s(T_2,T_1)$	$\alpha \wedge s(T_1,T_2) \overset{g}{\rightsquigarrow} \alpha \wedge p(T_2,T_1)$
(R8) If $\theta{:}vars(q) \mapsto vars(q) \cup \mathbf{E}$	then:	$\alpha \wedge p(T_1,T_2) \wedge p(T_1',T_2') \in q \quad \overset{s}{\rightsquigarrow}$	$\alpha\theta \wedge p(T_1,T_2)\theta$
$s.t.\ \theta(T_i) = \theta(T_i')$		$\alpha \wedge p(T_1,T_2)$ s.t. T_1 or $T_2 \in \mathbf{E} \quad \overset{g}{\rightsquigarrow}$	$\alpha \wedge p(Z,T_2)$ or $\alpha \wedge p(T_1,Z)$

verify the feasibility of the classical query-rewriting approach in the knowledge graph embedding setting.

3.1 Ontology-Driven Data Sampling

Let $Q_{\mathcal{G}}$ be the set of all possible queries that can be formed using the signature Σ. During the training process, existing embedding models are trained on a set of sampled queries of certain shapes and their answers over the KG $\mathcal{G}$.

3.1.1 Random Query Sampling. The existing sampling procedure from the literature [19, 36] arbitrarily chooses entities and relations in the graph to construct queries of various shapes. Query2Box is trained on complex queries involving multiple atoms, while CQD is trained only on atomic queries, as it relies on a neural link predictor. For verifying how well the model generalizes, the test set includes queries whose shapes have not been encountered during training.

Naturally, this procedure is not guaranteed to capture ontological knowledge that comes with the knowledge graph, and using it over $\mathcal{O}^{\infty}(\mathcal{G})$ could generate a bias towards concepts and roles that are very general. Moreover, using all possible queries from $Q_{\mathcal{G}}$ with their certain answers might be infeasible in practice. In the following, we discuss various options for guiding the sampling of queries to train ontology-aware knowledge graph embedding models for query answering.

3.1.2 Incorporating Query Rewritings and Certain Answers. The first natural attempt to incorporate ontologies is to consider certain answers, which for *DL-Lite*$_{\mathcal{R}}$ can be done efficiently. An example of this training case is to randomly sample query $q(Y) \leftarrow \exists X.\text{hasAlumnus}(\text{mit}, X) \wedge \text{worksFor}(X,Y)$ and, given $(\mathcal{G}, \mathcal{O})$ in Fig. 1, use it along with all its certain answers: mit, yale during training. To incorporate the ontology, we can randomly sample queries over the KG, using the standard procedure, and then add their generalizations and specializations obtained using the rules in Tab. 2. To rewrite a query we select an atom and apply an ontology axiom. For example, the first rule **(R1)** applies a concept inclusion axiom, while **(R6)** applies a role inclusion.

The specializations of a query q (i.e. *Spec*(q)), incorporate more specific information regarding the answers of q, while the generalizations of q (i.e. *Gen*(q)) incorporate additional related entities.

Example 3.3. Take the queries $q_1(X) \leftarrow \exists Y.\text{type}(X, \text{University})$ and $q_2(X) \leftarrow \exists Z.\text{teachesAt}(Z,X)$. Using **R2** in Table 2 and (5) in Figure 1 we get $q_1 \overset{s}{\rightsquigarrow} q_2$, i.e., q_2 is a specialization of q_1.

In general, there are exponentially many rewritings, thus we fix a rewriting depth, up to which the respective training queries are generated, via a dedicated parameter.

3.1.3 Strategic Ontology-Based Sampling. While adding generalizations and specializations of randomly selected queries should partially reflect the background knowledge, many relevant axioms can be overlooked if they are not explicitly captured in the data. To overcome this, we consider the set of target query shapes as directed acyclic graphs (DAGs) of the form (N, E), where N is a set of nodes and $E \subseteq N \times N$ is a set of directed edges. The set of training queries is then obtained by applying a labeling function that assigns symbols in Σ to nodes and edges based on the ontology.

Definition 3.4 (Query Shape). A *query shape* S is a tuple (N, E, n) such that (N, E) is a DAG and $n \in N$ is the *distinguished node* of S (i.e., the node for the answer variable). For a given set of relations and constants in Σ, *a labeling function* $f : N \cup E \mapsto \Sigma \cup \mathbf{V}$ maps each node to either a variable or an entity and each edge to a relation symbol in the KG signature Σ.

We rely on the ontology when labeling query shapes to create semantically meaningful queries. Let $\sqsubseteq^*$ be the reflexive and transitive closure of $\sqsubseteq$. Then, for a given relation p we have:

- $inv(p) = \{p' \mid p \sqsubseteq p'^- \in \mathcal{O}\}$, $dom(p) = \{A \mid \exists p' \sqsubseteq A' \in \mathcal{O}$ s.t. $p \sqsubseteq^* p', A \sqsubseteq^* A'$ or $A' \sqsubseteq^* A\}$,
- $range(p) = \{A \mid \exists p'^- \sqsubseteq A' \in \mathcal{O}$ s.t. $p \sqsubseteq^* p', A \sqsubseteq^* A'$ or $A' \sqsubseteq^* A\}$,
- $follows(p) = \{p' \mid range(p) \cap dom(p') \neq \emptyset\}$,
- $inter_r(p) = \{p' \mid range(p) \cap range(p') \neq \emptyset$ or $p_1 \in inv(p), p_2 \in inv(p')$ and $dom(p_1) \cap dom(p_2) \neq \emptyset\}$,
- $inter_d(p) = \{p' \mid dom(p) \cap dom(p') \neq \emptyset$ or $p_1 \in inv(p), p_2 \in inv(p')$ and $range(p_1) \cap range(p_2) \neq \emptyset\}$.

Intuitively, for a given relation p, the set $inv(p)$ contains all inverse relations of p, $dom(p)$ contains all domain types for p, $range(p)$ all range types for p, $follows(p)$ stores all relations p' which can follow p, and $inter_r(p)$, $inter_d(p)$ contain resp. all relations p' which can intersect with p on range and domain positions. Then, for each shape we label nodes and edges to create queries

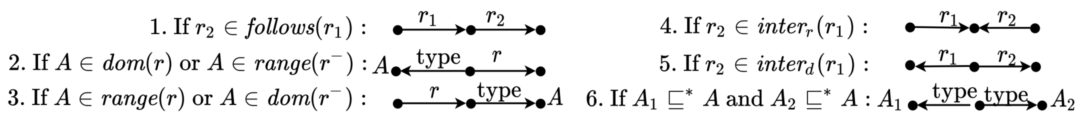

Figure 2: Ontology-driven rules to label query shapes; r^- denotes any of $\mathit{inv}(r)$.

that are valid w.r.t. $\mathcal{O}$ as shown in Figure 2. Note that this query sampling process uses only the ontology, i.e., it is data independent.

3.2 Ontology-Aware Query2Box

Query2Box [36] is a prominent *query-based* embedding models, in which entities and queries are embedded as *points* and *boxes*, resp., in a d-dimensional vector space. A d*-dimensional embedding* is a function φ that maps $c \in \mathbf{E} \cup \mathbf{C}$ to $\mathbf{c} \in \mathbb{R}^d$ and a query q to $\mathbf{q}=(\mathbf{cen}_q, \mathbf{off}_q) \in \mathbb{R}^d \times \mathbb{R}^d_{\geq 0}$, which is used to define a *query box* as

$$\mathrm{box}_q = \{\mathbf{v} \in \mathbb{R}^d \mid \mathbf{cen}_q - \mathbf{off}_q \preceq \mathbf{v} \preceq \mathbf{cen}_q + \mathbf{off}_q\},$$

where $\preceq$ is the element-wise inequality, $\mathbf{cen}_q$ is the center of the box, and $\mathbf{off}_q$ is the positive offset of the box, modeling its size. The score for an entity c being an answer to q is computed based on the distance from $\mathbf{c}$ to box_q. The Query2Box model relies on the following geometric operators.

3.2.1 Projection. Let $S \subseteq \mathbf{E} \cup \mathbf{C}$ be a set of entities, and $r \in \mathbf{R}$ a relation. Intuitively, the *projection* operator performs graph traversal, e.g. given an entity e, the projection operator for the relation r provides the box corresponding to the set $\{e' \in \mathbf{E} \cup \mathbf{C} \mid r(e, e') \in \mathcal{G}\}$. Given the embedding $\mathbf{r} = (\mathbf{cen}_r, \mathbf{off}_r) \in \mathbb{R}^d \times \mathbb{R}^d_{\geq 0}$ for the relation r, we model the projection of a box $\mathbf{v} = (\mathbf{cen}_v, \mathbf{off}_v)$ by applying element-wise summation $\mathbf{v} + \mathbf{r} = (\mathbf{cen}_v + \mathbf{cen}_r, \mathbf{off}_v + \mathbf{off}_r)$. This relational translation [5] operation corresponds to the translation and enlargement of the box $\mathbf{v}$.

3.2.2 Intersection. Given a set of entity sets $\{S_1, \ldots, S_n\}$, each of which is represented by a box in Query2Box, the *intersection* operator computes their intersection. The intersection $\mathbf{w} = (\mathbf{cen}_w, \mathbf{off}_w)$ of a set of boxes $\{(\mathbf{cen}_{v_1}, \mathbf{off}_{v_1}), \ldots, (\mathbf{cen}_{v_n}, \mathbf{off}_{v_n})\}$ for $\{S_1, \ldots, S_n\}$ is modeled by applying the following operations:

$$\mathbf{cen}_w = \sum_{i=1}^{n} \Phi\big(\mathrm{NN}(\mathbf{cen}_{v_1}), \ldots, \mathrm{NN}(\mathbf{cen}_{v_n})\big)_i \odot \mathbf{cen}_{v_i},$$
$$\mathbf{off}_w = \min(\mathbf{off}_{v_1}, \ldots, \mathbf{off}_{v_n}) \odot \sigma\big(\Psi(\mathbf{off}_{v_1}, \ldots, \mathbf{off}_{v_n})\big),$$

where $\odot$ and min denote the element-wise multiplication and minimum, respectively. $\mathrm{NN}\colon \mathbb{R}^d \to \mathbb{R}^d$ is a 2-layer feed-forward neural network having the same dimensionality for the hidden layers as for the input layer. Φ and σ stand for the softmax and sigmoid functions, resp., applied in a dimension-wise manner. Ψ is a permutation invariant function composed of a 2-layer feed-forward network followed by element-wise mean operation and a linear transformation. The center $\mathbf{cen}_w$ is calculated as the weighted mean of the box centers $\mathbf{cen}_{v_1}, \ldots, \mathbf{cen}_{v_n}$. This geometric intersection provides a smaller box that lies inside a given set of boxes – for more details, we refer the reader to [36].

The goal of the Query2Box model is to learn the embedding of queries, such that the *distance* between the box corresponding to the query and its answers is minimized, while the *distance* to this box from other randomly sampled non-answers is maximized.

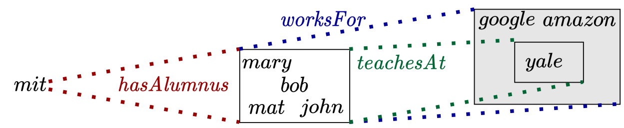

Figure 3: Illustration of our extension of Query2Box. The KG nodes and relations are embedded as points and projection operators, resp. The axiom teachesAt $\sqsubseteq$ worksFor **is captured by the inclusion of the respective boxes for queries.**

In what follows, we present our proposal for integrating ontological axioms into the Query2Box model. Similarly to [36], we define the distance between $\mathbf{q} \in \mathbb{R}^d \times \mathbb{R}^d_{\geq 0}$ and $\mathbf{v} \in \mathbb{R}^d$ as $d(\mathbf{q}, \mathbf{v}) = \|\mathbf{cen}_q - \mathbf{v}\|_1$, namely the L_1 distance from the entity $\mathbf{v}$ to the center of the box. Using the sigmoid function we transform the distance into the $(0, 1)$ interval, that is, $p(\mathbf{v} \mid \mathbf{q}) = \sigma\big(-(d(\mathbf{q}, \mathbf{v}) - \gamma)\big)$, where $\gamma > 0$ is a margin, which denotes the probability of $v \in q[\mathcal{G}^i, \mathcal{O}]$.

Note that for every ontological axiom its both left- and right-hand side can be turned into queries. When embedding those queries as boxes, axioms can be naturally enforced if in the vector space the inclusion of the boxes corresponding to the respective queries is ensured. For a query q, let $\mathit{Gen}(q) = \{q_1 \ldots q_n\}$ be the set of all generalizations of q based on $\mathcal{O}$. Given a train query q and $v \in q[\mathcal{G}, \mathcal{O}]$, we aim at maximizing $\prod_{i=1}^{n} p(\mathbf{v} \mid \mathbf{q}_i)^{\beta_i}$, where $\beta_i \geq 0$ is a weighting parameter for all $i = 1, \ldots, n$. This is achieved by minimizing the negative log-likelihood: $-\log\Big(\prod_{i=1}^{n} p(\mathbf{v} \mid \mathbf{q}_i)^{\beta_i}\Big) = -\sum_{i=1}^{n} \beta_i \log\big(p(\mathbf{v} \mid \mathbf{q}_i)\big)$. By exploiting that $\sigma(x) = 1 - \sigma(-x)$, for any $\mathbf{v}'_j \notin q[\mathcal{G}, \mathcal{O}]$, we have $p(\mathbf{v}' \mid \mathbf{q}) = 1 - p(\mathbf{v} \mid \mathbf{q}_i) = \sigma(d(\mathbf{q}, \mathbf{v}) - \gamma)$.

Our goal is to enforce that if $q' \in \mathit{Gen}(q)$ then the box of q' contains the box of q. In order for that to hold, we need to ensure that, if a is an answer to q then the distance not only between a and q should be minimized, but also between a and all generalizations of q. The following training objective reflects our goal:

$$L = -\sum_{i=1}^{n} \beta_i \log \sigma\big(\gamma - d(\mathbf{v}, \mathbf{q}_i)\big) - \sum_{j=1}^{k} \frac{1}{k} \log \sigma(d(\mathbf{v}'_j; \mathbf{q}) - \gamma),$$

where $\mathbf{v}'_j \notin q[\mathcal{G}, \mathcal{O}]$ is a random entity for all $j = 1, \ldots, k$ obtained via negative sampling. In our experiments, we use $\beta_i = |\mathit{Gen}(q)|^{-1} = 1/n$.

Example 3.5. In Figure 3, the entities and relations are embedded into the vector space as points and projection operators, resp. The embedding of $q(Y) \leftarrow \exists X.\mathrm{hasAlumnus}(\mathrm{mit}, X) \wedge \mathrm{worksFor}(X, Y)$ is represented by the larger gray box, obtained by applying the projection hasAlumnus to the embedding of entity mit followed by the projection on worksFor. To enforce teachesAt $\sqsubseteq$ worksFor we ensure that the box of $q'(Y) \leftarrow \exists X.\text{hasAlumnus}(\text{mit}, X) \wedge \text{teachesAt}(X, Y)$, is contained in the box corresponding to q.

Conceptually, our training data sampling techniques and the loss function modifications are flexible in terms of the DL, in which the ontology is encoded. The only restriction is the existence of efficient query rewriting algorithms.

3.3 Ontology-aware CQD

A prominent atom-based query-answering method is CQD [2], which relies on neural link predictors for answering atomic subqueries, and then aggregates the resulting scores via t-norms.

We now describe how we inject the ontology axioms into the neural link predictor employed by CQD [2]. For that we rely on the FO translation of the DL axioms. Following [32], for each rule the goal is to identify the entity embeddings which maximize an *inconsistency loss*, i.e., the entities for which the scoring of the head is much lower compared to the scoring of the body. For example, given the rule Γ: $\forall X, Y$ teachesAt$(X, Y) \rightarrow$ type$(Y, \text{University})$, the goal is to map the variables to d-dimensional embeddings, i.e. $\phi : var(\Gamma) \mapsto \mathbb{R}^d$, s.t. $[score_{\text{teachesAt}}(\phi(X), \phi(Y)) - score_{\text{type}}(\phi(Y), \textbf{University})]_+$ is maximal, where $[x]_+ = max([x], 0)$ with $[x]$ being the integral part of x, and $score_r$ being the scoring function for the relation r determining whether there is an r-edge between any two given entities. Mapping ϕ determines a so-called *adversarial input set*, which is used as an adaptive regulariser for the neural link predictor. The inconsistency loss is then incorporated into the final loss function of the ontology-aware model which tries to minimize the maximal inconsistency loss while learning to predict the target graph over the given sets of correct triples. In experiments we rely on the existing implementation of the adversarial sets regularisation method in ComplEx-N3, which is the default neural link predictor for CQD.

4 EVALUATION

We evaluate our training strategies on popular models: Query2Box (*Q2B*) and *CQD*, as well as our ontology-aware adaptations *O2B* and CQD^{ASR}. Specifically, we test their ability to perform inductive reasoning, deductive reasoning, and their combination.

4.1 Benchmarks for E-OMQA

Since the task of embedding-based ontology mediated query answering has not been considered in the literature before, no benchmarks for it existed prior to our work. Thus, we have created two novel benchmarks based on LUBM [17] and NELL [8] KGs, available online[3]. LUBM has a rich ontology including domain and range axioms as well as concept and role inclusions, while the NELL KG is accompanied with a more simple ontology containing only (inverse) role inclusions. Following common practice, each input KG is completed w.r.t. inverse edges. In Table 3 we present the number of ontology axioms of various types as well as the number of (materialized) triples, entities and relations in these datasets.

4.1.1 Query and Answers Sampling. We use the same type of queries (corresponding to directed acyclic graphs with entities as the source nodes, also known as *anchors*) as in prior work [36] (see Figure 4). We assume that each input KG is complete (i.e. $\mathcal{G}^i$) and then partition it into $\mathcal{G}_{valid}$ for validation and $\mathcal{G}_{train}$ for training by discarding 10% of edges at each step; this yields $\mathcal{G}_{train} \subsetneq \mathcal{G}_{valid} \subsetneq \mathcal{G}$. We then create several training sets of queries according to our ontology-aware data sampling strategies from Section 3.1:

- *plain*: the training queries are randomly sampled from $\mathcal{G}_{train}$, and we take their plain answers, i.e. over $\mathcal{G}_{train}$.
- *gen*: queries in *plain* augmented with their ontology-based generalizations[4]; answers are certain, i.e., over $\mathcal{O}^\infty(\mathcal{G}_{train})$.
- *spec*: queries from *gen* augmented with specializations;
- *onto*: queries from Section 3.1, with randomly chosen percentage of valid entities as anchors; all answers are certain.

Specializations and generalizations non-compliant with the shapes from Figure 4 are discarded. Note that for NELL, the *plain* data is exactly the one from [37]. We observe that the number of 1p queries obtained for *gen* and *spe* settings are identical. This is probably because the set of 1p queries in *plain* covers all edges in the training KG. For the LUBM dataset, we have created the training and testing sets from scratch, and the 1p queries in *plain* do not contain the entire training KG. The *onto* set of queries leverages the proposed ontology-driven technique, given that the ontology covers all relations and concepts in the KG and describes how they interact, i.e., the ontology axioms support all the constructed queries, and we chose 50 % of valid entities as anchors. As there are too many queries to chose from, due to the large number of relations, we had to select a smaller number of valid entities as anchors, namely 20-30%. This explains the smaller number of 1p queries. Moreover, the NELL ontology does not contain interesting axioms that can be leveraged by ontology-driven query sampling technique, thus to obtain *onto* we had to rely on the patterns from the data alone.

We generate three different test sets for verifying the ability of the query answering model to perform inductive reasoning, deductive reasoning and their combination. More formally,

- **Inductive case (I)**. Is the model able to predict missing answers to queries over the complete, but not observable KG $\mathcal{G}^i$? (accounts for the standard test case)
- **Deductive case (D)**. Is the model able to predict answers that can be inferred from the known triples in $\mathcal{G}_{train}$ using ontology?
- **Inductive + Deductive case (I+D)**. Is the model able to predict missing answers inferred from the complete but not observable KG $\mathcal{G}^i$ using axioms in $\mathcal{O}$?

For test case **I**, respectively **I+D**, test queries are randomly sampled over $\mathcal{G}$, respectively $\mathcal{O}^\infty(\mathcal{G})$, while for **D** they are randomly sampled over $\mathcal{O}^\infty(\mathcal{G}_{train})$ s.t. they cannot be trivially answered over $\mathcal{G}_{train}$, and unseen during training. In each test case the validation queries are generated similarly but over $\mathcal{G}_{valid}$.

The size of each training/testing set, and the number of queries per shape for each of the considered cases are presented in Table 4.

For each test and validation query, we measure the accuracy based on *hard (certain) answers*, i.e., those that cannot be trivially answered over $\mathcal{G}_{train}$ (or $\mathcal{G}_{valid}$ for test queries) and require prediction of missing edges and/or application of ontology axioms (see Definition 2.3 and 3.2).

[3] https://github.com/medinaandresel/eomqa

[4] This is similar to random sampling over $\mathcal{O}^\infty(\mathcal{G}_{train})$ but unlike the deductive closure, our procedure is guaranteed to terminate. We used the rewriting depth of up to 10.

Table 3: The total number of axioms $|\mathcal{O}|$ and of each type, the size of the input KG $|\mathcal{G}|$, the number of entities $|\mathrm{E}|$, the number of relations $|\mathrm{R}|$, and the number of materialized triples $|\mathcal{O}^{\infty}(\mathcal{G})|$.

Dataset	Ontology $\mathcal{O}$						KG $\mathcal{G}$			
	$\lvert\mathcal{O}\rvert$	$A \sqsubseteq A'$	$p \sqsubseteq s$	$p^{-} \sqsubseteq s$	$\exists p \sqsubseteq A$	$\exists p^{-} \sqsubseteq A$	$\lvert\mathcal{G}\rvert$	$\lvert\mathrm{E}\rvert$	$\lvert\mathrm{R}\rvert$	$\lvert\mathcal{O}^{\infty}(\mathcal{G})\rvert$
LUBM	68	13	5	28	11	11	284K	55684	28	565K
NELL	307	–	92	215	–	–	285K	63361	400	497K

Table 4: Number of test and train queries of each shape in each of the settings.

Dataset	Train/Test	Query Shape								
		1p	2p	3p	2i	3i	ip	pi	2u	up
LUBM	*Plain*	110000	110000	110000	110000	110000	–	–	–	–
	Gen	117124	136731	150653	181234	208710	–	–	–	–
	Spe	117780	154851	173678	271532	230085	–	–	–	–
	Onto	116893	166159	333406	212718	491707	–	–	–	–
	I	8000	8000	8000	8000	8000	8000	8000	8000	8000
	D	1241	4701	6472	3829	4746	7393	7557	4986	7122
	I+D	8000	8000	8000	8000	8000	8000	8000	7986	8000
NELL	*Plain*	107982	107982	107982	107982	107982	–	–	–	–
	Gen	174310	408842	864268	398412	930787	–	–	–	–
	Spe	174310	419664	906609	401954	936537	–	–	–	–
	Onto	114614	542923	864268	629144	930787	–	–	–	–
	I	15688	3910	3918	3828	3786	3932	3895	3940	3966
	D	346	4461	4294	4842	5996	7295	5862	5646	6894
	I+D	8000	8000	8000	8000	8000	8000	8000	7990	8000

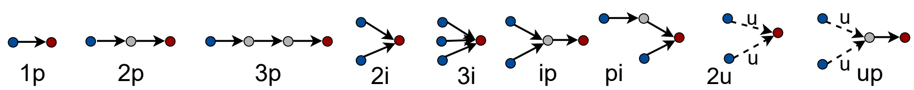

Figure 4: Query shapes considered in our experiments, where blue nodes correspond to anchor entities and red ones to answer variables; p stands for projection, i for intersection and u for union. The first five shapes are used in training.

4.2 Models and Evaluation Procedure

We consider *Q2B*, *O2B*, *CQD* and CQD^{ASR} trained in each described setting: i.e., M_x, where $M \in \{Q2B, O2B, CQD, CQD^{ASR}\}$ and $x \in$ {plain, gen, spec, onto}; $Q2B_{plain}$ and CQD_{plain} are taken as baselines. *Q2B* and *O2B* are trained on five query shapes that require projection and intersection, while *CQD* and CQD^{ASR} are trained on atomic queries. We have configured both *Q2B* and *O2B* systems as follows: The size of the embedding dimension was set to 400, and the models were trained for 15×10^4 steps using Adam optimizer with an initial learning rate of 10^{-4} and the batch size of 512. The rest of the parameters were set in the same way as in [36].

For CQD, we used the code from [2] with ComplEx-N3 [29] employed as the base model. The embedding size was set to 1000, and the regularisation weight was selected based on the validation set by searching in $\{10^{-3}, 5 \times 10^{-3}, \ldots, 10^{-1}\}$. For LUBM, the regularization weight was set to 0.1 in the *gen*, *spe*, and *onto* settings, and to 0.01 in the *plain* setting. For NELL, the regularization weight was set to 0.005 in the *plain* setting, to 0.001 in the *gen* and *spe* settings, and to 0.05 in the *onto* setting. For CQD^{ASR} we have additionally used a regularisation weight of the following values: 10^{-2}, 10^{-3} and 10^{-4}. The batch size of the adversarial examples was set to 32.

We evaluated the models periodically and report the test results of the models with the best performance on the validation dataset. The performance of each trained model is measured using standard metric HITS@K for K=3 (HITS@3), indicating the frequency that the correct answer is ranked among the top-3 results.

4.3 Evaluation Results

4.3.1 Inductive+Deductive case. First, we present the detailed results for the most challenging setting (**I+D**) for LUBM and NELL in Table 5 and Figure 5.

Based on the average accuracy of the models across all query shapes reported in Table 5, the improvements of the proposed ontology-aware adaptations of *Q2B* and *CQD* are evident. For LUBM *O2B* trained using the *onto* strategy improves the *Q2B* baseline by almost 50%, while in case of *CQD*, 54% enhancement is achieved. For NELL similar behaviour is observed with the improvement of almost 20% for *Q2B*, and 25% for *CQD*. Next, we discuss the impact of each of the proposed techniques for the E-OMQA task.

The first observation is that incorporating the ontology in the training data is crucial as both *Q2B* and *CQD* trained in settings *gen* and *onto* yield significant improvements over the baselines. Additional incorporation of specializations (setting *spec*) does not seem

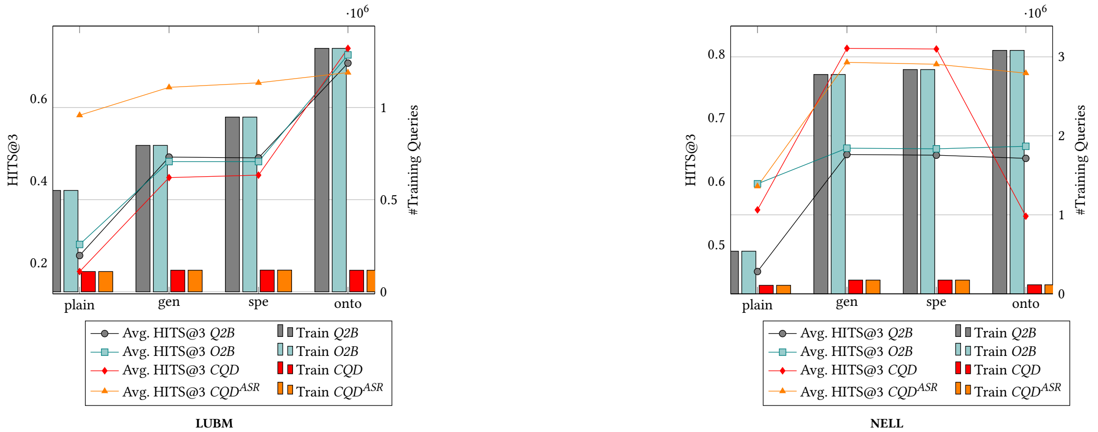

Figure 5: Performance of *Q2B*,*O2B*,*CQD*, and *CQD*^{ASR} on I+D and size of the training set for each setting *plain*, *gen*, *spe*, *onto*. The number of training queries is scaled by multiplying with 10^6.

Table 5: HITS@3 scores in the inductive and deductive setting (I+D) for each query shape (the higher the better)

Dataset	LUBM										NELL									
Model	**Avg.**	**1p**	**2p**	**3p**	**2i**	**3i**	**ip**	**pi**	**2u**	**up**	**Avg.**	**1p**	**2p**	**3p**	**2i**	**3i**	**ip**	**pi**	**2u**	**up**
$Q2B_{plain}$	0.218	0.173	0.101	0.107	0.433	0.546	0.167	0.200	0.133	0.100	0.458	0.516	0.343	0.286	0.747	0.81	0.404	0.447	0.325	0.241
$O2B_{plain}$	0.245	0.235	0.109	0.095	0.488	0.584	0.176	0.218	0.2	0.103	**0.596**	**0.79**	0.409	**0.359**	**0.904**	**0.936**	0.479	**0.521**	**0.666**	0.303
CQD_{plain}	0.179	0.109	0.058	0.104	0.384	0.502	0.130	0.187	0.092	0.046	0.555	0.664	0.383	0.304	0.853	0.903	0.471	0.512	0.599	0.306
CQD^{ASR}_{plain}	**0.56**	**0.682**	**0.589**	**0.393**	**0.659**	**0.664**	**0.547**	**0.488**	**0.509**	**0.509**	0.592	0.716	**0.518**	0.337	0.807	0.831	**0.547**	0.513	0.614	**0.445**
$Q2B_{gen}$	0.458	0.592	0.267	0.129	**0.789**	**0.870**	0.360	0.282	0.552	0.279	0.642	0.858	0.485	0.397	0.928	0.95	0.538	0.539	0.768	0.312
$O2B_{gen}$	0.447	0.577	0.257	0.114	0.777	0.859	0.359	0.27	0.546	0.264	0.652	0.859	0.494	0.420	0.928	0.953	0.552	0.559	0.77	0.329
CQD_{gen}	0.408	0.539	0.214	0.098	0.710	0.791	0.304	0.302	0.513	0.208	**0.809**	**0.903**	**0.775**	**0.473**	**0.957**	**0.969**	**0.821**	**0.757**	**0.886**	**0.743**
CQD^{ASR}_{gen}	**0.628**	**0.733**	**0.640**	**0.413**	0.717	0.720	**0.598**	**0.599**	**0.653**	**0.582**	0.787	0.9	0.771	0.467	0.919	0.924	0.793	0.723	0.846	0.741
$Q2B_{onto}$	0.687	0.762	0.617	0.447	0.868	0.915	0.693	0.555	0.732	0.600	0.636	0.858	0.472	0.398	0.927	0.948	0.529	0.524	0.747	0.317
$O2B_{onto}$	0.707	**0.771**	0.629	0.476	**0.878**	**0.927**	0.694	0.619	**0.752**	**0.618**	0.655	0.862	0.498	0.423	**0.933**	**0.953**	0.557	0.555	0.773	0.340
CQD_{onto}	**0.723**	0.752	**0.681**	**0.481**	0.870	0.924	**0.735**	**0.728**	0.738	0.604	0.545	0.667	0.368	0.293	0.848	0.904	0.453	0.506	0.595	0.275
CQD^{ASR}_{onto}	0.664	0.753	**0.681**	0.421	0.744	0.755	0.643	0.666	0.704	0.615	**0.77**	**0.9**	**0.741**	**0.456**	0.922	0.923	**0.77**	**0.701**	**0.851**	**0.672**

to have a major impact though (see Figure 5). On LUBM, for all models, the advantage of the ontology-driven query sampling (i.e., *onto* setting) is significant compared to *gen* setting. Remarkably, for LUBM CQD_{onto}, resp. CQD^{ASR}_{onto} trained on less data than CQD_{gen}, resp. CQD^{ASR}_{gen} results in higher accuracy. This shows that random query sampling is not adequate for E-OMQA. The ontology for NELL is not expressive enough, thus, when generating training queries in the *onto* setting (see Table 3 for statistics) we proceeded in a bottom-up fashion as follows: We randomly labeled query shapes which produce answers, and constructed their generalizations as before; thus the settings *gen* and *onto* are similar, but *onto* has significantly less atomic queries, which explains why CQD_{gen} outperforms CQD_{onto} on NELL.

On average ontology-aware models (i.e., *O2B* and CQD^{ASR}) significantly outperform their baselines (i.e., *Q2B* and *CQD*, resp.) for the majority of training data sampling strategies. This trend is more prominent for atom-based models on the complex LUBM ontology, and for query-based ones on NELL, which is less expressive.

In Table 6 we present the detailed results for the query rewriting over pre-trained embeddings. In order to evaluate this procedure, for each hard answer a we take the best (i.e., minimum) ranking among all rankings generated by all queries in the rewriting of each test query. In other words, we take the minimal distance between the embedding of a and all rewritings of q. Note that, for measuring the performance we use the pre-trained models $Q2B_{plain}$, and CQD_{plain} obtained after 450K training steps. Due to the reliance on particular query shapes of the respective models, the complete rewriting for each query is not guaranteed. In Table 6, we present the results for this method compared to the *plain* setting. Minor improvements of only at most 10% are observed.

4.3.2 Inductive case. Next, we present the average HITS@3 metric for the inductive **I** test case (see Figure 6). For **I** ontology-injection methods do not yield any improvement, which is expected, since ontologies cannot handle missing edges and facts in a KG that are not inferred from the data using ontological reasoning.

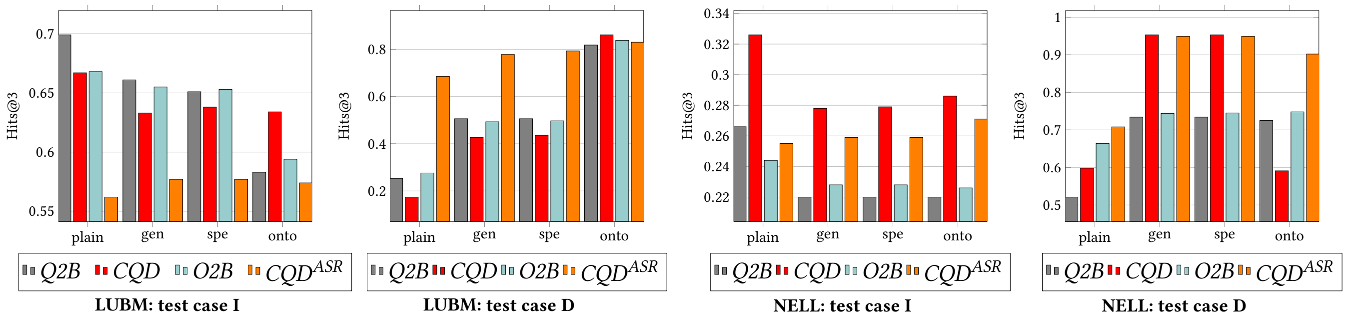

Figure 6: Comparison of *Q2B*,*O2B*, *CQD* and CQD^{ASR} in each training setting for test cases (I) and (D)

Table 6: Avg. HITS@3 for QA of shapes 1p, 2p, 3p, 2i, 3i using rewriting on top of pre-trained *plain* model vs the *plain* model.

Models	Test Case D		Test Case I+D	
	LUBM	NELL	LUBM	NELL
$Q2B_{plain}$	0.189	0.617	0.193	0.539
$Q2B^{rew}_{plain}$	0.248	0.683	0.261	0.639
Gain	+0.059	+0.066	+0.068	+ 0.1
CQD_{plain}	0.225	0.656	0.231	0.621
CQD^{rew}_{plain}	0.228	0.743	0.249	0.708
Gain	+0.003	+0.087	+0.018	+ 0.087

4.3.3 Deductive case. For the test case **D**, when the ontology is simple (e.g., NELL), CQD and Q2B are able to more or less learn to apply the ontology rules when they are explicitly injected in the training set. Moreover, the results on NELL in the *plain* setting show that rule enforcement is also competitive for deductive reasoning. For query-based models the best performance is achieved by combining ontology-driven sampling with rule enforcement, while for atom-based models, the inclusion of generalizations seems to be already sufficient.

For expressive ontologies, such as LUBM, the ontology-driven query sampling is crucial for optimal performance. For query-based models the best results are achieved when the ontology-driven query sampling is combined with rule enforcement, while for atom-based models rule enforcement does not seem to be necessary.

5 RELATED WORK

The task of answering queries that involve multiple atoms using embedding techniques has recently received a lot of attention (see [35] for overview). The existing proposals can be divided into *query-based* (e.g., [9, 25, 30, 36, 37, 39, 44, 45]) and *atom-based* (e.g., [2, 3]). The works [14] and [6] study the relation between the problem of conjunctive QA in the embedding space and over probabilistic databases. Our work is different from the above proposals in that along with the data we also rely on ontologies.

Integration of ontologies into KG embeddings has been recently actively investigated, for instance, in [1, 13, 16, 21, 23, 24, 26, 32, 41] (see also [43]), but these works typically focus on the task of link prediction rather than query answering. Recently, a type-aware model (called TEMP) for query answering over incomplete KGs has been proposed [22]. While TEMP allows for the exploitation of the type information, to the best of our knowledge it cannot handle more complex ontological axioms, which are the focus of our work.

The capability of embeddings to model hierarchical data has been explored in several works, e.g., [18, 34]. Another relevant direction is concerned with the construction of models for $\mathcal{EL}$ ontologies in the embedding space [28]. While the above works are related, they do not touch upon the problem of OMQA, studied in this work.

The problem of ontology-mediated query answering has been considered in the area of knowledge representation and reasoning (see e.g. [38] for an overview), but available methods, e.g.[11, 15], only focus on logic-based deductive reasoning, but do not aim at predicting missing links in knowledge graphs using machine learning approaches.

6 CONCLUSION

We have presented methods for Embedding-based Ontology Mediated Query Answering (E-OMQA) that operate in the embedding space to enable simultaneous inductive and deductive reasoning over the incomplete data. Experiments show that embedding-based methods for query answering applied naively or combined with query rewriting techniques are not effective. At the same time, our ontology-aware extensions of the popular models for embedding-based QA and the proposed ontology-driven training strategies yield promising results on the novel benchmarks that we introduce for the considered task.

For future work we plan to study the effectiveness of our methods for embedding-based ontology mediated query answering for other more complex query forms [37, 40, 42], e.g., queries with negation, as well as evaluate the proposed approach for the cases when the ontology is more expressive.

ACKNOWLEDGMENTS

Pasquale was partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreement no. 875160, ELIAI (The Edinburgh Laboratory for Integrated Artificial Intelligence) EPSRC (grant no. EP/W002876/1), an industry grant from Cisco, and a donation from Accenture LLP; and is grateful to NVIDIA GPU donations. This work was partially funded by the European project SMARTEDGE (grant number 101092908).

REFERENCES

[1] Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. BoxE: A Box Embedding Model for KB Completion. In *NeurIPS*.

[2] Erik Arakelyan, Daniel Daza, Pasquale Minervini, and Michael Cochez. 2021. Complex Query Answering with Neural Link Predictors. In *ICLR*.

[3] Erik Arakelyan, Pasquale Minervini, and Isabelle Augenstein. 2023. Adapting Neural Link Predictors for Complex Query Answering. *CoRR* abs/2301.12313 (2023).

[4] Franz Baader, Ian Horrocks, and Ulrike Sattler. 2009. Description Logics. In *Handbook on Ontologies*. 21–43.

[5] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*. 2787–2795.

[6] Stefan Borgwardt, İsmail İlkan Ceylan, and Thomas Lukasiewicz. 2019. Ontology-Mediated QA over Log-Linear Probabilistic Data. In *AAAI*. 2711–2718.

[7] Diego Calvanese, Giuseppe De Giacomo, *et al* Rosati. 2007. Tractable Reasoning and Efficient Query Answering in Description Logics: The *DL-Lite* Family. *J. Aut. R.* 39, 3 (2007), 385–429.

[8] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an Architecture for Never-Ending Language Learning. In *AAAI*.

[9] Nurendra Choudhary, Nikhil Rao, and Sumeet Katariya. 2021. Self-Supervised Hyperboloid Represent. from Logical Queries over KGs. In *WWW*. 1373–1384.

[10] Nilesh N. Dalvi and Dan Suciu. 2007. Efficient query evaluation on probabilistic databases. *VLDB J.* 16, 4 (2007), 523–544.

[11] Thomas Eiter, Magdalena Ortiz, Mantas Simkus, Trung-Kien Tran, and Guohui Xiao. 2012. Query Rewriting for Horn-SHIQ Plus Rules. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*, Jörg Hoffmann and Bart Selman (Eds.). AAAI Press.

[12] Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandecic. 2014. Introducing Wikidata to the Linked Data Web. In *ISWC*.

[13] Bahare Fatemi, Siamak Ravanbakhsh, and David Poole. 2019. Improved KG Embedding Using Background Taxonomic Info. In *IAAI*. 3526–3533.

[14] Tal Friedman and Guy Van den Broeck. 2020. Symbolic Querying of Vector Spaces. In *UAI*. 1268–1277.

[15] Birte Glimm, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. 2011. Conjunctive Query Answering for the Description Logic SHIQ. *CoRR* abs/1111.0049 (2011).

[16] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2016. Jointly Embedding Knowledge Graphs and Logical Rules. In *EMNLP*. 192–202.

[17] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. 2005. LUBM: A benchmark for OWL knowledge base systems. *J. Web Semant.* 3, 2-3 (2005), 158–182.

[18] Víctor Gutiérrez-Basulto and Steven Schockaert. 2018. From Knowledge Graph Embedding to Ontology Embedding?. In *KR*. 379–388.

[19] William L. Hamilton, Payal Bajaj, and Marinka Zitnik *et al.* 2018. Embedding Logical Queries on Knowledge Graphs. In *NeurIPS*. 2030–2041.

[20] William L. Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. 2018. Embedding Logical Queries on Knowledge Graphs. In *NeurIPS 2018*. 2030–2041.

[21] Junheng Hao, Muhao Chen, and *et al* Wenchao Yu. 2019. Universal Representation Learning of KBs by Jointly Embedding Instances and Ontological Concepts. In *SIGKDD*. 1709–1719.

[22] Zhiwei Hu, Víctor Gutiérrez-Basulto, Zhiliang Xiang, Xiaoli Li, Ru Li, and Jeff Z. Pan. 2022. Type-aware Embeddings for Multi-Hop Reasoning over Knowledge Graphs. In *IJCAI 2022*. 3078–3084.

[23] Nitisha Jain, Trung-Kien Tran, Mohamed H. Gad-Elrab, and Daria Stepanova. 2021. Improving Knowledge Graph Embeddings with Ontological Reasoning. In *ISWC 2021*.

[24] Seyed Mehran Kazemi and David Poole. 2018. SimplE Embedding for Link Prediction in Knowledge Graphs. In *Neurips*. 4289–4300.

[25] Bhushan Kotnis, Carolin Lawrence, and Mathias Niepert. 2021. Answering Complex Queries in KGs with Bidirectional Sequence Encoders. In *AAAI 2021*. 4968–4977.

[26] Denis Krompaß, Stephan Baier, and Volker Tresp. 2015. Type-Constrained Representation Learning in Knowledge Graphs. In *ISWC (1)*, Vol. 9366. 640–655.

[27] Denis Krompaß, Maximilian Nickel, and Volker Tresp. 2014. Querying Factorized Probabilistic Triple Databases. In *ISWC*.

[28] Maxat Kulmanov, Wang Liu-Wei, Yuan Yan, and Robert Hoehndorf. 2019. EL Embeddings: Geometric construction of models for the Description Logic EL ++. *CoRR* abs/1902.10499 (2019).

[29] Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical Tensor Decomposition for KB Completion. In *ICML*, Vol. 80. 2869–2878.

[30] Lihui Liu, Boxin Du, Heng Ji, ChengXiang Zhai, and Hanghang Tong. 2021. Neural-Answering Logical Queries on Knowledge Graphs. In *KDD*. 1087–1097.

[31] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *CIDR*.

[32] Pasquale Minervini, Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2017. Adversarial Sets for Regularising Neural Link Predictors. In *UAI*.

[33] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* 104, 1 (2016), 11–33.

[34] Dhruvesh Patel, Shib Sankar Dasgupta, Michael Boratko, Xiang Li, Luke Vilnis, and Andrew McCallum. 2020. Representing Joint Hierarchies with Box Embeddings. In *AKBC*.

[35] Hongyu Ren, Mikhail Galkin, and Michael Cochez *et al.* 2023. Neural Graph Reasoning: Complex Logical Query Answering Meets Graph Databases. *CoRR* abs/2303.14617 (2023).

[36] Hongyu Ren, Weihua Hu, and Jure Leskovec. 2020. Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. In *ICLR*.

[37] Hongyu Ren and Jure Leskovec. 2020. Beta Embeddings for Multi-Hop Logical Reasoning in Knowledge Graphs. In *NeurIPS*.

[38] Thomas Schneider and Mantas Simkus. 2020. Ontologies and Data Management: A Brief Survey. *KI* 34, 3 (2020), 329–353.

[39] Haitian Sun, Andrew O. Arnold, and Tania Bedrax-Weiss *et al.* 2020. Faithful Embeddings for Knowledge Base Queries. In *NeurIPS*.

[40] Zihao Wang, Hang Yin, and Yangqiu Song. 2021. Benchmarking the Combinatorial Generalizability of Complex QA on KGs. In *NeurIPS Datasets and Benchmarks*.

[41] Bo Xiong, Nico Potyka, Trung-Kien Tran, Mojtaba Nayyeri, and Steffen Staab. 2022. Faithful Embeddings for $\mathcal{EL}^{++}$ KBs. In *ISWC*, Vol. 13489. 22–38.

[42] Hang Yin, Zihao Wang, and Yangqiu Song. 2023. On Existential First Order Queries Inference on Knowledge Graphs. *CoRR* abs/2304.07063 (2023).

[43] Wen Zhang, Jiaoyan Chen, Juan Li, Zezhong Xu, Jeff Z. Pan, and Huajun Chen. 2022. Knowledge Graph Reasoning with Logics and Embeddings: Survey and Perspective. *CoRR* abs/2202.07412 (2022).

[44] Zhanqiu Zhang, Jie Wang, Jiajun Chen, Shuiwang Ji, and Feng Wu. 2021. ConE: Cone Embeddings for Multi-Hop Reasoning over Knowledge Graphs. In *NeurIPS 2021*. 19172–19183.

[45] Zhaocheng Zhu, Mikhail Galkin, Zuobai Zhang, and Jian Tang. 2022. Neural-Symbolic Models for Logical Queries on KGs. In *ICML*, Vol. 162. 27454–27478.