CylE: Cylinder Embeddings for Multi-hop Reasoning over Knowledge Graphs

Chau Duc Minh Nguyen* and **Tim French** and **Wei Liu** and **Michael Stewart**
ARC Centre for Transforming Maintenance through Data Science,
Department of Computer Science and Software Engineering
The University of Western Australia
*chau.nguyenducminh@research.uwa.edu.au
{tim.french, wei.liu, michael.stewart}@uwa.edu.au

Abstract

Recent geometric-based approaches have been shown to efficiently model complex logical queries (including the intersection operation) over Knowledge Graphs based on the natural representation of Venn diagram. Existing geometric-based models (using points, boxes embeddings), however, cannot handle the logical negation operation. Further, those using cones embeddings are limited to representing queries by two-dimensional shapes, which reduced their effectiveness in capturing entities query relations for correct answers. To overcome this challenge, we propose unbounded cylinder embeddings (namely CylE), which is a novel geometric-based model based on three-dimensional shapes. Our approach can handle a complete set of basic first-order logic operations (conjunctions, disjunctions and negations). CylE considers queries as Cartesian products of unbounded sector-cylinders and consider a set of nearest boxes corresponds to the set of answer entities. Precisely, the conjunctions can be represented via the intersections of unbounded sector-cylinders. Transforming queries to Disjunctive Normal Form can handle queries with disjunctions. The negations can be represented by considering the closure of complement for an arbitrary unbounded sector-cylinder. Empirical results show that the performance of multi-hop reasoning task using CylE significantly increases over state-of-the-art geometric-based query embedding models for queries without negation. For queries with negation operations, though the performance is on a par with the best performing geometric-based model, CylE significantly outperforms a recent distribution-based model.

1 Introduction

Multi-hop Reasoning (MHR) on Knowledge Graphs (KGs) is a primary task in answering queries over large-scale knowledge graphs. Queries can be represented using First-Order-Logic (FOL) connectives (Brachman and Levesque, 2004), involving these operations: existential quantification ($\exists$), conjunction ($\wedge$), disjunction ($\vee$) and negation ($\neg$). MHR involves learning to answer these FOL queries, which has recently received attention from several studies (Hamilton et al., 2018; Ren et al., 2020; Ren and Leskovec, 2020). A common approach is to first transform the FOL query into a computation graph (see Figure 1), where nodes represent entity constants or variables and edges map to predicates and logical operations. Representing queries in this way enables the learning process to traverse paths of KGs via the computation graph, so as to find a set of answers for a given query. However, large-scale KGs (Bollacker et al., 2008; Vrandečić and Krötzsch, 2014; Lehmann et al., 2015; Speer et al., 2017; Fellbaum, 2010; Mitchell et al., 2018) are often incomplete and noisy, which makes explicit query mechanism, such as graph traversal incapable of returning correct answers to a query.

Motivated by the challenge above, we aim to reason about incomplete KGs using MHR (Lin et al., 2018; Zhang et al., 2021a). To achieve MHR, recent studies have proposed several query embedding (QE) methods based on geometry (Hamilton et al., 2018; Ren et al., 2020; Zhang et al., 2021b) and probability distribution representations (Ren and Leskovec, 2020; Choudhary et al., 2021a). A common approach of QE in the literature is to project the FOL queries into an embedding space, allowing a model to learn the embeddings of queries and entities. Answering these queries is equivalent to finding the similarity between the embedded queries and the embedded entities. Geometric-based models using cone embeddings ConE (Zhang et al., 2021b) are shown to be superior over the others, especially the ability of handling negations. ConE represents queries as Cartesian product of sector-cones in an embedding space. ConE projects target entities as lines and queries as areas of sector-cones. Intuitively,

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 1736–1751
May 2-6, 2023 ©2023 Association for Computational Linguistics

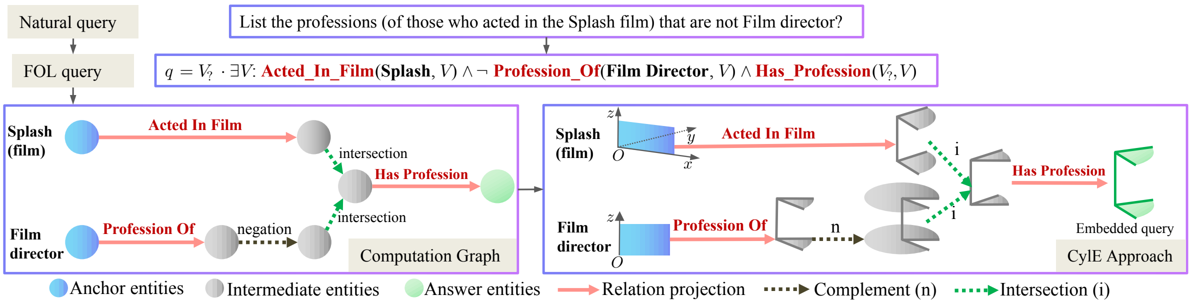

Figure 1: **(Left):** An example of a computation graph for a FOL query (rewritten from a natural query). **(Right):** The intuition of CylE is to model anchor and target entities as perpendicular boxes regarding the base of sector-cylinders. Each edge of the computation graph transforms the entity with its relation into the sector-cylinder embedding via projection, intersection or complement operations; then outputting an embedded query.

if the line is inside the area, the corresponding entity is considered as a match to the query. Nevertheless, cone embeddings are limited to the 2D space in a flat plane, to represent the query and entity embeddings. For example, one entity in a cluster of the sub-topic */movie/action* and one in another */movie/documentary* can be in the sector-cone region for those labels in the topic */movie* (see Figure 2(a)). Answering a query regarding the */movie/action* can return irrelevant entities from the */movie/documentary*. In multi-dimensional space, entities assigned to multiple labels can be achieved using multi-dimensional classification (Read et al., 2013). In the 3D space, for instance, apart from the xy-plane, semantics of entities can be additionally classified by different levels of degree in the height segment (along the z-axis).

In this paper, we propose to expand the two dimension sector-cone embeddings into the three dimension coordinate system. We represent queries by augmenting the shape of the sector-cones to be similar as unbounded sector-cylinders (shortly called *sector-cylinders*) in an embedding space, compared to closed sector-cylinders in a normal situation (see examples of unbounded sector-cylinders, their intersection or union in Figure 2 and further definitions in Section 4.1). In short, we name this approach **Cyl**inder **E**mbeddings (CylE). Answering a query is similar to finding entities that are subsets of the sector-cylinders representing queries. We investigate whether there is any improvement in answering correctly any query structures, using CylE over other approaches. Our approach can handle the conjunction as we notice that the intersection of sector-cylinders (along the z-axis) can be a sector-cylinder. Since the union of sector-cylinders is no longer a sector-cylinder, we first transform queries to Disjunctive Normal Form (DNF), which enables CylE to handle disjunction. With regard to the negation operation, we consider the closure-complement of sector-cylinders to model this operation. Our contributions are: (1) introducing the first 3D geometric-based approach to model the QE for MHR to the best of our knowledge, (2) enabling the model to handle a complete set of the basic FOL queries (existential quantification, conjunction, disjunction and negation) and (3) demonstrating that CylE significantly outperforms state-of-the-art (SOTA) geometric-based models for non-negation queries and is on par for queries with negations by empirical results.

2 Related works

Multi-hop Reasoning for logical query Studies in MHR employed approaches such as (distributions (Ren and Leskovec, 2020; Choudhary et al., 2021a; Huang et al., 2022), geometric shapes (Hamilton et al., 2018; Ren et al., 2020; Zhang et al., 2021b), fuzzy logic (Chen et al., 2022; Arakelyan et al., 2021)), others using count-min sketch (Sun et al., 2020) and neural-symbolic approach (Zhu et al., 2022), to achieve the common goal of learning representation of queries, *i.e.* query embeddings. The primary difference in these approaches is based on how queries are represented. For example, distribution-based models use Beta distributions (Ren and Leskovec, 2020) or Multivariate Gaussian distributions (Choudhary et al., 2021a). In geometric shapes, Hamilton et al. (2018) represented queries as point embeddings, Ren et al. (2020) then furthered this using box embeddings, and Bai et al. (2022) made an improvement by introducing 'particle' embeddings (a set of points using multiple vectors). Zhang et al. (2021b) have made the geometric approach more expressive using cone embeddings. Another difference is the ability to model a complete set of logical operations (Brachman and Levesque, 2004) (conjunction, disjunction and negation). Several methods (Ren and Leskovec, 2020; Zhang et al., 2021b; Bai et al., 2022; Chen et al., 2022) have achieved a coverage of all operations including the negation, compared to others (Hamilton et al., 2018; Ren et al., 2020; Choudhary et al., 2021a) without negation.

Reasoning about KGs using geometric shapes Historically, these approaches received attention since the introduction of translation-based methods (Bordes et al., 2013), rotation (Sun et al., 2019; Zhang et al., 2020) and 3D-rotation (Gao et al., 2020) for learning knowledge graph embeddings. Inspired by a Poincaré ball, other studies (Nickel and Kiela, 2017; Balažević et al., 2019) proposed hyperbolic space (non-Euclidean geometry). A common task of these studies is KGs completion. However, furthering it to MHR task poses a challenge because of the complex structures of queries (see Figure 3). Using geometric shapes for the MHR task (point Hamilton et al. (2018); Bai et al. (2022), box Ren et al. (2020), hyperbolic Choudhary et al. (2021b) and cone embeddings Zhang et al. (2021b)) have increasingly gained popularity. Cone embeddings were also mentioned in (Ganea et al., 2018), but not for the MHR task. Since existing geometric-based methods of cone embeddings rely on 2D shapes, we extend the representation learning for this geometric family to 3D shapes for the MHR task. Other studies (*e.g.* spherical text embeddings (Meng et al., 2019)) learned word embeddings for document clustering and classification tasks, but still not for the MHR task.

3 Preliminaries

3.1 Knowledge Graphs

Given a set of *vertices* (*entities*) $\mathcal{V}$ and a set of *edges* (*relations* or *predicates*) $\mathcal{E}$, we define a knowledge graph ($\mathcal{G}$) as a set of *triples*. Each triple is (v_s, e, v_o), where $(v_s, v_o \in \mathcal{V})$ and $(e \in \mathcal{E})$ is a *vertex subject*, a *vertex object* and an *edge* respectively. Assuming $(r \in \mathcal{R})$ denotes each element in a set of relation functions $(\mathcal{R})$, where (r) – associated with (e) – is a binary function $r : \mathcal{V} \times \mathcal{V} \rightarrow \{\text{True}, \text{False}\}$ that denotes an asymmetric direction of relation from (v_s) to (v_o), and vice versa. A symmetric direction of relation (non-directional relation) is $r : \mathcal{V} \times \mathcal{V} \rightarrow \{\text{True}, \text{True}\}$. Notice that there are two sets involving in edges/relations: $(\mathcal{E})$ for edge instances and $(\mathcal{R})$ for relation functions.

3.2 First-Order Logic queries

There are four basic logical operations involving in the interpretation of FOL queries[1]: conjunction ($\wedge$), disjunction ($\vee$), negation ($\neg$) and existential quantification ($\exists$). We adopt definitions and notations of BetaE (Ren and Leskovec, 2020) to assume that a FOL query consists of three folds: (1) a constant anchor entity set ($\mathcal{V}_a \subseteq \mathcal{V}$), (2) existentially quantified bound variables ($V_1, \ldots, V_k$) and (3) a target entity variable ($V_?$) to respond a certain query. A FOL query can be written in Disjunctive Normal Form (DNF) as a combination of disjunctions of conjunctive queries (c_i) in the following:

$$q[V_?] = V_? \cdot \exists V_1, \ldots, V_k : c_1 \vee c_2 \vee \ldots c_n,$$

where $c_i = e_{i1} \wedge e_{i2} \cdots \wedge e_{im}$, including at least one literal $e_{ij} = r(v_a, V)$ or $\neg r(v_a, V)$ or $r(V', V)$ or $\neg r(V', V)$, and $(v_a \in \mathcal{V}_a)$ while $V \in \{V_?, V_1, \ldots, V_k\}$, $V' \in \{V_1, \ldots, V_k\}$ and $V' \neq V$. Finding the answer entities of a query (q) is similar to searching for an answer set $[\![q]\!] \subseteq \mathcal{V}$, where $v \in [\![q]\!]$ if and only if $q[v]$ is True.

3.3 Query Decomposition

We adopt the definitions of FOL query decomposition in Zhang et al. (2021b) using a computation graph, including *vertices* and *edges* (see an example in Figure 1). Each intermediate vertex is a set of entities and each edge demonstrates relational projection or logical operations over entity sets:

- **Relation Traversal** $\rightarrow$ **Projection**: Given a set of entities $\mathcal{S} \subset \mathcal{V}$ and a relation function $r \in \mathcal{R}$, estimate adjacent entities $\cup_{v \in \mathcal{S}} \mathcal{A}(v, r)$, where $\mathcal{A}(v, r) \equiv \{v' \in \mathcal{V} : r(v, v') = \text{True}\}$.
- **Negation** $\rightarrow$ **Complement**: Given a set of entities $\mathcal{S} \subset \mathcal{V}$, estimate $\overline{\mathcal{S}}$ where $\overline{\mathcal{S}} \equiv \mathcal{V} \backslash \mathcal{S}$.
- **Conjunction** $\rightarrow$ **Intersection**: Given a number of sets of entities $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n\}$, compute the intersection $\cap_{i=1}^n \mathcal{S}_i$ of these sets.
- **Disjunction** $\rightarrow$ **Union**: Given a number of entity sets $\{\mathcal{S}_1, \ldots, \mathcal{S}_n\}$, find the union $\cup_{i=1}^n \mathcal{S}_i$.

[1]Universal quantification ($\forall$) rarely appears in the real world (Ren and Leskovec, 2020), this operation is therefore not considered.

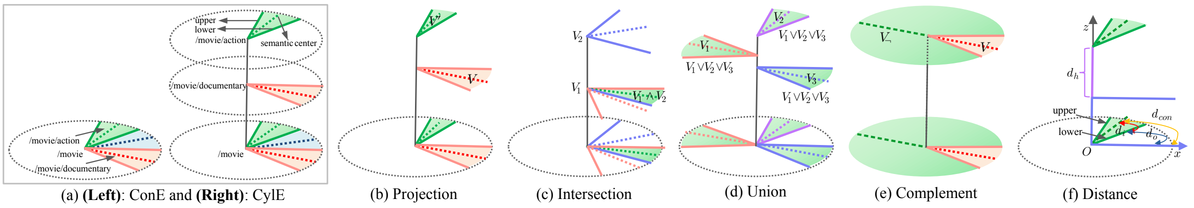

Figure 2: An overview of cone and cylinder embeddings with projection, logical operations (intersection, union, complement) and distance. The color dot line represents the semantic center axis. **(c)** Intersection between a sector-cylinder (red semantic center) and another higher sector-sector (royal blue semantic center) is the sector-cylinder having green semantic center. **(d)** Union of three sector-cylinders (having red, purple and royal blue semantic center axis) are in green regions. **(f)** Distance is from the green sector-cylinder (of an embedded query) to a purple target box (of an embedded target entity); d_i, d_o, d_h denotes the inside, outside and height distance.

4 Cylinder embeddings

We first provide a background of parameterizing cylinder (Section 4.1) and modeling cylinder embeddings for conjunctive queries (Section 4.2). Next, we show the cylinder embeddings process with logical operators (Section 4.3), then provide an optimization method (Section 4.4).

4.1 Cylinder definitions and parameterization

We define an *unbounded sector-cylinder* (without parameterizing radius) as having only two bounds for its body: the upper bound and the lower bound with an intersection as a height (see Figure 2(a)). We call the angle between the two bounds aperture, which has a range in $[0, 2\pi]$. In short, we call an unbounded sector-cylinder a *sector-cylinder* or a *cylinder* based on their apertures. Notice that a sector-cylinder becomes a cylinder when the aperture is zero. We use three variables for parameterization (the first two for the sector-cylinder's base adapt from ConE): (1) the *semantic center axis* $\theta_{ax} \in [-\pi, \pi)$ is the angle between the positive x-axis and the symmetric axis, (2) $\theta_{ap} \in [0, 2\pi]$ is the *aperture* and (3) $\theta_{he} \in (-\pi, \pi)$ is the *height*.

Note that the base of cylinders and cones share similarity in properties (semantic axis and aperture), which can be illustrated in the same plane, such as xy-plane. Thus, the base of cylinders in our study inherit some definitions and propositions from cones. These are a *cone*, a *convex cone* (Boyd et al., 2004), a *closure-complement* of a cone and a *sector-cone*. Each of these, which is defined by ConE (Zhang et al., 2021b), is a set in 2D space. Further, sector-cylinder's base has the same proposition as the sector-cone: *"always axially symmetric"* which has been proven in ConE.

Precisely, we define $(\mathbb{K})$ as a space consisting of all $(\theta_{ax}, \theta_{ap}, \theta_{he})$. An arbitrary sector-cylinder S_0 is as: $S_0 = (\theta_{ax}, \theta_{ap}, \theta_{he}) \in \mathbb{K}$. Then, a d-ary Cartesian product of sector-cylinders, called S, is a product of each sector-cylinder $S_{i=1\to d}$ (or each element of S is a d-dimensional vector in $\mathbb{K}^d$): $S = S_1 \times S_2 \cdots \times S_d$ or be rewritten as follows:

$$S := \Big((\theta^1_{ax}, \theta^1_{ap}, \theta^1_{he}), \ldots, (\theta^d_{ax}, \theta^d_{ap}, \theta^d_{he})\Big) \\ = (\boldsymbol{\theta}_{ax}, \boldsymbol{\theta}_{ap}, \boldsymbol{\theta}_{he}) \subset \mathbb{K}^d. \tag{4.1}$$

4.2 Cylinder embeddings for conjunctive queries and entities

In this section, we describe query embeddings for conjunctive queries. Note that disjunctive queries can be transformed to DNF form as a set of conjunctive queries (as mentioned in Section 3.2). We model the *embedding region* $(\mathbf{V}_q)$ for the answer set $[\![q]\!]$ of the query (q) (see Section 3.3) using a Cartesian product of sector-cylinders (as mentioned in Section 4.1) as: $\mathbf{V}_q = (\boldsymbol{\theta}_{ax}, \boldsymbol{\theta}_{ap}, \boldsymbol{\theta}_{he})$, where embedding of semantic center axis is $\boldsymbol{\theta}_{ax} \in [-\pi, \pi)^d$, embedding of aperture is $\boldsymbol{\theta}_{ap} \in [0, 2\pi]^d$ and embedding of height is $\boldsymbol{\theta}_{he} \in (-\pi, \pi)^d$; and (d) denotes the dimension of the embedding space (see Eq. (4.1)).

Next, an *arbitrary entity* $(v \in \mathcal{V})$ is represented by a Cartesian product of cylinders having zero apertures. The corresponding embedding $(\mathbf{v})$ is as: $\mathbf{v} = (\boldsymbol{\theta}_{ax}, \mathbf{0}, \boldsymbol{\theta}_{he})$. The intuition is to embed the anchor or target entity into one similar as a *perpendicular box* with regard to a base of the cylinder, Precisely, all elements of the d-dimensional vector $(\boldsymbol{\theta}_{ap})$ is equal to zero.

4.3 Cylinder embeddings with logical operators

We describe the process of modeling relational projection (*projection* module) and modeling logical operators (*intersection*, *complement*, *union* module), to embed the FOL queries in the following:

Projection module: This module aims to learn the projection operation (see Section 3.3), which outputs the adjacent entities (of an anchor entity) linking to a given relation. To this end, we maps the embedding region of an entity set ($\mathbf{V}_q$) to another ($\mathbf{V}'_q$) (see Figure 2(b)) via a mapping function (f):

$$f : \mathbb{K}^d \to \mathbb{K}^d, \mathbf{V}_q \to \mathbf{V}'_q.$$

We use a *shallow neural network* to approximately represent the mapping function $f(x)$. Overall, ($\mathbf{V}_q$) is a summation of the embeddings of entity set ($\mathbf{V}$) and a relation ($\mathbf{r}$) as: $\mathbf{V}_q = \mathbf{V} + \mathbf{r}$; where the representation of each relation ($\mathbf{r}$) is sector-cylinder embeddings. Then, a composition function $f(\mathbf{V}_q)$ has a scaling of chunking function $g(x)$ and a multilayer perceptron (**MLP**) network:

$$f(\mathbf{V}_q) = g(\mathbf{MLP}([\mathbf{V}_q])),$$

where $\mathbf{MLP} : \mathbb{R}^{3d} \to \mathbb{R}^{3d}$, and the function $g(x)$ is to split a three-dimensional vector into three d-vectors for semantic center axis, aperture and height embeddings. In addition, there is a scaling operation, adapt from Zhang et al. (2021b), involving in the function $g(x)$ to scale the semantic center axis, the aperture and the height embeddings into their normal ranges (as mentioned in Section 4.2):

$$[f(\boldsymbol{x})]_i = \begin{cases} \theta'^{i}_{ax} = \pi \tanh(\lambda_1 x_i), & \text{if } i \leq d, \\ \theta'^{i-d}_{ap} = \pi \tanh(\lambda_2 x_i) + \pi, & \text{if } i > d, \\ \theta'^{i}_{he} = 2\pi(sigmoid(\lambda_3 x_i) - 0.5), \end{cases}$$

where $[f(\boldsymbol{x})]_i$ is the i-th element of $f(\boldsymbol{x})$; $(\lambda_1, \lambda_2, \lambda_3)$ are the scaling hyper-parameters.

Intersection module: This module aims to learn the conjunction operation (see Section 3.3). Assuming a conjunction of conjunctive queries (q_i) associates with a query (q), its answer is $[\![q]\!] = \cap_{i=1}^{k} [\![q_i]\!]$. Notice that entities in the set $[\![q]\!]$ share semantic similarity with one another, as the conjunction of conjunctive queries based on sector-cylinder embeddings are conjunctive queries (see Figure 2 (c)). Supposing $\mathbf{V}_{\cap q} = (\boldsymbol{\theta}_{ax}, \boldsymbol{\theta}_{ap}, \boldsymbol{\theta}_{he})$ and $\mathbf{V}_{i,q} = (\boldsymbol{\theta}_{i,ax}, \boldsymbol{\theta}_{i,ap}, \boldsymbol{\theta}_{i,he})$ are embedding region of $[\![q]\!]$ and $[\![q_i]\!]$ respectively. To obtain the $\mathbf{V}_{\cap q}$, we then take the summation w.r.t. the number of conjunctive queries (of the Hadamard product $\odot$ between $\boldsymbol{A}_i$ and $\mathbf{V}_{i,q}$), which is shown below:

$$\mathbf{V}_{\cap q} = \textstyle\sum_i^k \boldsymbol{A}_i \odot \mathbf{V}_{i,q},$$

where $\boldsymbol{A} \in \mathbb{R}^{k \times d}$ is an attention matrix defined by:

$$\boldsymbol{A}_{i \times d} = \frac{\exp\Big(\mathbf{MLP}([\mathbf{V}_q])_i\Big)}{\sum_j^n \exp\Big(\mathbf{MLP}([\mathbf{V}_q])_j\Big)},$$

where (k) is the number of involving conjunctive queries and $[\mathbf{V}_q]_i \in \mathbb{R}^{3d}$ is a concatenation of $(\boldsymbol{\theta}_{i,ax}, \boldsymbol{\theta}_{i,ap}, \boldsymbol{\theta}_{i,he})$ for the i-th conjunctive query and $\mathbf{MLP} : \mathbb{R}^{3d} \to \mathbb{R}^{d}$. As mentioned in Ren et al. (2020), using attention mechanism is important in comparison to other approaches (*e.g.* averaging, deep sets Zaheer et al. (2017)). Note that our approach is to approximately model the conjunction operation. Further, this approach is different than that in cone embeddings which required an intermediate process (Zhang et al., 2021b): to convert the semantic center axis to points on the unit circle, then to map these points back to angles to recover the semantic center axis.

Complement module: This module aims to represent the negation operation (see Section 3.3), by finding the complementary set of $[\![q]\!]$ (or $\mathcal{V} \backslash [\![q]\!]$): $[\![\neg q]\!]$. Supposing $\mathbf{V}_q = (\boldsymbol{\theta}_{ax}, \boldsymbol{\theta}_{ap}, \boldsymbol{\theta}_{he})$ and $\mathbf{V}_{\neg q} = (\boldsymbol{\theta}'_{ax}, \boldsymbol{\theta}'_{ap}, \boldsymbol{\theta}'_{he})$ are sector-cylinder embeddings region of $[\![q]\!]$ and $[\![\neg q]\!]$ respectively. From a geometric aspect, the closure-complement is close to the set of sector-cylinders (see Figure 2(e)). Thus, the sum of apertures of ($\mathbf{V}_q$) and ($\mathbf{V}_{\neg q}$) should be close to 2π. Assuming semantic center of ($\mathbf{V}_{\neg q}$) should be opposite to those in ($\mathbf{V}_q$) while the height of ($\mathbf{V}_{\neg q}$) should be equivalent to those in ($\mathbf{V}_q$) as follows:

$$[\boldsymbol{\theta}'_{ax}]_i = \begin{cases} [\boldsymbol{\theta}_{ax}]_i - \pi, & \text{if } [\boldsymbol{\theta}_{ax}]_i \geq 0, \\ [\boldsymbol{\theta}_{ax}]_i + \pi, & \text{if } [\boldsymbol{\theta}_{ax}]_i < 0, \end{cases}$$

$$[\boldsymbol{\theta}'_{ap}]_i = 2\pi - [\boldsymbol{\theta}_{ap}]_i,$$

$$[\boldsymbol{\theta}'_{he}]_i = [\boldsymbol{\theta}_{he}]_i.$$

Note that the height variable cannot be involved in this module, as the negation should be closed without this variable. Since the negation is not closed w.r.t. entities as long as keeping the same height for ($\mathbf{V}_{\neg q}$) and ($\mathbf{V}_q$), this can be a bottleneck of negation queries under the three dimension space. This can be addressed by designing closed negation for both queries and entities; however, we leave this direction for future work.

Union module: This module aims to represent the disjunction operation (see Section 3.3). Assuming a disjunction of conjunctive queries (q_i) associates with a query (q), the aim of this operation is to represent its answer: $[\![q]\!] = \cup_{i=1}^{k} [\![q_i]\!]$. We face a challenge as the union of sector-cylinders having different height is no longer a sector-cylinder, or the union over sector-cylinders is not closed (see Figure 2(d)). As mentioned by Ren et al. (2020), there is an issue of non-scalability when modeling directly the disjunction. To address this issue, we adapt a technique of Ren et al. (2020) by transforming queries into a DNF (Davey and Priestley, 2002) (*e.g.* disjunction of conjunctive queries). The union operation in DNF is moved to the last step of converted computation graphs, which contain conjunctive queries (see Section 3.2). Thus, we can apply the other logical operations (as mentioned in above modules) to have a set of embeddings of these conjunctive queries. The answer entities are those nearest to any embeddings of these conjunctive queries (see further details of estimating the aggregated distance score in Eq. (4.2)).

4.4 Optimization method

Distance score: We define a distance score $d(\mathbf{v};\mathbf{V})$ between the embedded region entity $\mathbf{v} = (\boldsymbol{\theta}'_{ax}, \mathbf{0}, \boldsymbol{\theta}'_{he})$ and the embedded region query $\mathbf{V} = (\boldsymbol{\theta}_{ax}, \boldsymbol{\theta}_{ap}, \boldsymbol{\theta}_{he})$. Inspired by Ren et al. (2020) and Zhang et al. (2021b), we adapt two types of this distance: (1) d_{con} (for those conjunctive queries) and (2) d_{dis} (for those disjunctive queries). In terms of estimating (d_{con}), there are three terms: an outside distance (d_o), an inside distance (d_i) and a height distance (d_h) as follows:

$$d_{con}(\mathbf{v};\mathbf{V}) = d_o(\mathbf{v};\mathbf{V}) + \lambda d_i(\mathbf{v};\mathbf{V}) + d_h(\mathbf{v};\mathbf{V}),$$

where the hyper-parameter $\lambda \in (0,1)$ is to encourage the expected entity $(\mathbf{v})$ to be inside the embedded region $(\mathbf{V})$. Intuitively, $(\mathbf{v})$ is close to the upper or lower bound of $(\mathbf{V})$ when (λ) is close to zero or one. The three distances contributing to the (d_{con}) for conjunctive queries are defined by:

$$d_o = \left|\left|\min\{d_l, d_u\}\right|\right|_1, \; d_i = \left|\left|\min\{d_{ax}, d_{ap}\}\right|\right|_1,$$
$$d_h = \left|\left|\boldsymbol{\theta}'_{he} - \boldsymbol{\theta}_{he}\right|\right|_1,$$

where $(d_l = |1 - \cos(\boldsymbol{\theta}'_{ax} - \boldsymbol{\theta}_l)|)$ denotes the outside distance between the semantic center axis of the entity and the *lower bound* of the query, $(d_u = |1 - \cos(\boldsymbol{\theta}'_{ax} - \boldsymbol{\theta}_u)|)$ denotes the outside distance between the semantic center axis of the entity and the *upper bound* of the query; $(d_{ax} = |1 - \cos(\boldsymbol{\theta}'_{ax} - \boldsymbol{\theta}_{ax})|)$ denotes the inside distance between the semantic center axis of the entity and that of the query, $(d_{ap} = |1 - \cos(\boldsymbol{\theta}_{ap}/2)|)$ denotes the inside distance between the semantic center axis and either of the two bounds of the query (see Figure 2(f) for an example to estimate these distances). Further, $(\boldsymbol{\theta}_l = \boldsymbol{\theta}_{ax} - \frac{\boldsymbol{\theta}_{ap}}{2})$ is the *lower bound* and $(\boldsymbol{\theta}_u = \boldsymbol{\theta}_{ax} + \frac{\boldsymbol{\theta}_{ap}}{2})$ is the *upper bound* of the embedded query, the notation $||\cdot||_1$ is the L_1 norm. The higher value of the cosine function is, the less distance is. Notice that the maximum of this function is equivalent to one, we therefore subtract one from this, to ensure the minimum distance to be close to zero. Next, we adapt the DNF technique of Ren et al. (2020) to estimate (d_{dis}) by obtaining the minimum distance between a target entity and each conjunctive query:

$$d_{dis}(\mathbf{v};\mathbf{V}) = \min\{d_{con}(\mathbf{v};\mathbf{V}_i)\}_{i:1\rightarrow n}. \quad (4.2)$$

Training objective function: To optimize the training loss, we follow the objective function $(\mathcal{L})$ from Ren and Leskovec (2020), $\mathcal{L} = -\log\sigma(\gamma - d(\mathbf{v};\mathbf{V})) - \frac{1}{n}\sum_i^n \log\sigma(d(\mathbf{v}';\mathbf{V}) - \gamma)$ is a summation of two terms: (1) a positive loss is to minimize the distance $d(\mathbf{v};\mathbf{V})$ between a positive embedded entity $(v \in [\![q]\!])$ and an embedded query and (2) a negative sampling loss is to maximize the distance $d(\mathbf{v}';\mathbf{V})$ between negative embedded entities $(v'_{i:1\rightarrow n} \notin [\![q]\!])$ and an embedded query; where (n) is the number of negative sampling entities, $\sigma(x)$ denotes the sigmoid activation function and the hyper-parameter (γ) is a pre-fixed positive margin.

5 Experiments

5.1 Experimental designs

For benchmarking, we follow experimental designs (datasets, query structures, training and evaluation protocol) of Multi-hop Reasoning (MHR) task from (Ren and Leskovec, 2020).

Datasets: We use benchmarking datasets for the MHR task: FB15k (Bollacker et al., 2008), FB15k-237 (Toutanova and Chen, 2015) and NELL995 (Xiong et al., 2017). Using the same pre-processing steps as BetaE (Ren and Leskovec, 2020), we split each dataset into the training, validation, and test set. The aim of MHR task is to obtain *non-trivial* answers, which *cannot be discovered* by directly *traversing* the *incomplete KGs*, for each arbitrary FOL query. Please see Appendix A.1 for further details of these datasets.

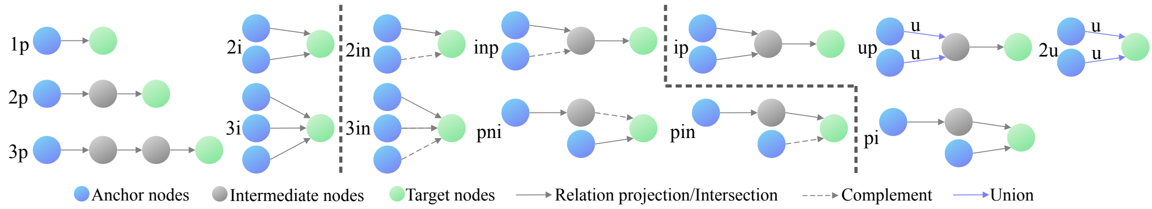

Figure 3: **(Left)** & **(Middle):** query structures are involved in the training process. **(Left)**, **(Middle)** & **(Right):** all queries are involved in the evaluation process; p is projection, i is intersection, n is negation and u is union.

Dataset	**Model**	**1p**	**2p**	**3p**	**2i**	**3i**	**ip**	**pi**	**2u**	**up**	**AVG**
FB15k	GQE	53.9	15.5	11.1	40.2	52.4	19.4	27.5	22.3	11.7	28.2
	Q2B	70.5	23.0	15.1	61.2	71.8	28.7	41.8	37.7	19.0	40.1
	BetaE	65.1	25.7	24.7	55.8	66.5	28.1	43.9	40.1	25.2	41.6
	ConE ($d = 800$)	73.3	33.8	29.2	64.4	73.7	35.7	50.9	55.7	31.4	49.8
	CylE ($d = 512$)	78.5	34.6	29.2	65.5	74.5	37.9	52.2	57.9	31.6	51.3
	CylE ($d = 800$)	**78.8**	**37.0**	**30.9**	**66.9**	**75.7**	**40.8**	**53.8**	**59.4**	**33.5**	**53.0**
FB15k-237	GQE	35.2	7.4	5.5	23.6	35.7	10.9	16.7	8.4	5.8	16.6
	Q2B	41.3	9.9	7.2	31.1	45.4	13.3	21.9	11.9	8.1	21.1
	BetaE	39.0	10.9	10.0	28.8	42.5	12.6	22.4	12.4	9.7	20.9
	ConE ($d = 800$)	41.8	12.8	11.0	32.6	47.3	14.0	25.5	14.5	10.8	23.4
	CylE ($d = 512$)	42.5	13.0	11.0	34.4	48.4	15.0	26.3	15.2	11.2	24.1
	CylE ($d = 800$)	**42.9**	**13.3**	**11.3**	**35.0**	**49.0**	**15.7**	**27.0**	**15.3**	**11.2**	**24.5**
NELL995	GQE	33.1	12.1	9.9	27.3	35.1	14.5	18.5	8.5	9.0	18.7
	Q2B	42.7	14.5	11.7	34.7	45.8	17.4	23.2	12.0	10.7	23.6
	BetaE	53.0	13.0	11.4	37.6	47.5	14.3	24.1	12.2	8.5	24.6
	ConE ($d = 800$)	53.1	16.1	13.9	40.0	50.8	17.5	26.3	15.3	11.3	27.2
	CylE ($d = 512$)	**56.5**	**17.5**	**15.6**	**41.4**	**51.2**	**19.6**	**27.2**	**15.7**	**12.3**	**28.5**
	CylE ($d = 800$)	55.7	17.5	15.1	40.7	51.1	19.1	27.1	15.4	12.2	28.2

Table 1: The average **MRR** (%) results in different query structures without negation ($\exists$, $\wedge$, $\vee$) using these datasets: FB15k, FB15k-237 and NELL995. The results of baselines (GQE, Q2B, BetaE, ConE) are taken from (Zhang et al., 2021b). Query structures with union operations ($2u/up$) are in DNF forms.

Queries: We adopt FOL query structures of (Ren and Leskovec, 2020) for the training, validation and test process. In terms of the training, there are five structures without negation ($1p/2p/3p/2i/3i$) and five structures with negation ($2in/3in/inp/pni/pin$). With regard to the evaluation process, we not only use the same query structures as those in the training process, we also use unseen structures ($ip/pi/2u/up$), which have not been involved in the training process, to evaluate the ability of generalization for the model.

Training and evaluation protocol: In terms the training process, we use Adam optimizer (Kingma and Ba, 2015). We follow the similar hyper-parameter settings of (Zhang et al., 2021b) to initialize the model, but search for the most effective combination of these hyper-parameters in the situation of cylinder embeddings (see more details in Appendix A.2). With regard to the evaluation process, we adopt the evaluation protocol of (Ren and Leskovec, 2020). There are three involving KGs: the training KG ($\mathcal{G}_{train}$ for training edges), the validation KG ($\mathcal{G}_{valid}$ for training and validation edges) and the test KG ($\mathcal{G}_{test}$ for training, validation and test edges) (see Appendix A.1). Specifically, given a test query (q) of incomplete KGs, our aim is to find *non-trivial* answers $[\![q]\!]_{\text{test}} \backslash [\![q]\!]_{\text{valid}}$ ($[\![q]\!]_{\text{valid}} \backslash [\![q]\!]_{\text{train}}$). We use the same metric Mean Reciprocal Rank (MRR) as described in (Ren et al., 2020; Ren and Leskovec, 2020; Zhang et al., 2021b) to evaluate the performance of Multi-hop Reasoning. Suppose ($\mathcal{Q}$) is a set of $[\![q]\!]_{\text{test}} \backslash [\![q]\!]_{\text{valid}}$, for each *non-trivial* answer ($v \in \mathcal{Q}$), we rank (v) against non-answer entities $\mathcal{V} \backslash [\![q]\!]_{\text{test}}$ (where v is associated with the rank r). We estimate the MRR as follows: $\text{MRR} = \frac{1}{|\mathcal{Q}|} \sum_{v \in \mathcal{Q}}^{|\mathcal{Q}|} \frac{1}{r}$. The higher MRR is, the better performance of the model is.

Baselines: There are four baseline models: GQE (Hamilton et al., 2018), Query2Box/Q2B (Ren et al., 2020), BetaE (Ren and Leskovec, 2020) and ConE (Zhang et al., 2021b). We obtain the recent results of these models from Zhang et al. (2021b), which are slightly higher than those reported by Ren and Leskovec (2020). We not only use the same embedding dimension $d = 800$ as ConE for fair comparisons, but we also run experiments using $d = 512$ for the sensitivity analysis[2].

5.2 Results

We report our main results of Multi-hop Reasoning using CylE regarding FOL queries with and without negation. Specifically, we compare the performance of CylE with these baselines: GQE, Q2B, BetaE and ConE using the same benchmarking datasets as mentioned above. We obtain the average results of five experiments for each dataset when the embedding dimension $d = 800$ (see Appendix B.1 for error bars of these results). For the sensitivity analysis with $d = 512$, we obtain results of an experiment for each dataset.

Modeling queries without negation: Table 1 demonstrates the average performance of Multi-hop Reasoning using CylE regarding existential positive first-order (EPFO) queries (a subset of FOL queries without negation), compared to baselines. Overall, CylE significantly outperforms all baselines. In comparison with the SOTA model ConE ($d = 800$), the average performance for all query structures (**AVG**) of CylE gains around 6.4%, 4.7% and 3.7% using the dataset FB15k, FB15k-237 and NELL995 respectively. More specifically, CylE achieves the highest improvement regarding ip queries, by nearly 14.3%, 12.1% and 9.1% using these datasets. In comparison with the previous model BETA, the **AVG** of CylE ($d = 800$) is also considerably higher, by around 27.4%, 17.2% and 14.6%, observed in the three datasets.

In terms of using the DNF technique, notice that the performance of answering those queries with unions ($2u$) only is significantly lower than those of queries with intersections ($2i$), by a large margin. We consistently observed this point in the three datasets. This can be due the limitation of representing union queries using the DNF technique, where it is challenging to expect an answer entity, for example, to be nearest to all conjunctive queries in the DNF form (see Eq. (4.2)). We also report the results for query structures regarding the union operation using De Morgan's (DM) law. Since there might be a problem of inconsistency with the real set union (as discussed in (Zhang et al., 2021b)), the results for union operation using DNF ($2u/up$) are higher than those using DM law (see Section 5.3 for further details).

Modeling queries with negation: Table 2 shows the average performance of Multi-hop Reasoning using CylE regarding query with negation, compared to baselines. Although ConE achieves the highest performance in average using the three datasets, the **AVG** of CylE ($d = 800$) is close to those in ConE. In comparison with the previous model BetaE, the **AVG** of CylE outperforms significantly. This increasing trend is similar to that in the ConE. Note that handling queries with negation are still challenging in all models (BetaE, ConE and CylE) since the **AVG** are significantly lower than those in queries without negation operations. This challenge may be due to a high uncertainty in the large number of answers for negation queries.

Dataset	Model	2in	3in	inp	pin	pni	AVG
FB15k	BetaE	14.3	14.7	11.5	6.5	12.4	11.8
	ConE ($d = 800$)	**17.9**	**18.7**	12.5	**9.8**	**15.1**	**14.8**
	CylE ($d = 512$)	15.6	15.9	13.3	7.5	13.6	13.2
	CylE ($d = 800$)	15.7	16.3	**13.7**	7.8	13.9	13.5
FB15k-237	BetaE	5.1	7.9	7.4	3.6	3.4	5.4
	ConE ($d = 800$)	**5.4**	**8.6**	7.8	**4.0**	**3.6**	**5.9**
	CylE ($d = 512$)	4.8	8.3	8.1	3.6	3.4	5.7
	CylE ($d = 800$)	4.9	8.3	**8.2**	3.7	3.4	5.7
NELL995	BetaE	5.1	7.8	10.0	3.1	3.5	5.9
	ConE ($d = 800$)	**5.7**	**8.1**	10.8	**3.5**	**3.9**	**6.4**
	CylE ($d = 512$)	5.6	7.5	11.2	3.4	3.7	6.3
	CylE ($d = 800$)	5.4	7.6	**11.3**	3.4	3.7	6.3

Table 2: The average **MRR** (%) in different query structures with negation using these datasets: FB15k, FB15k-237 and NELL995. The baseline results (BetaE and ConE) are taken from (Zhang et al., 2021b).

Effects of the embedding dimension: In terms of queries without negation, there is a slight difference in MRR results of CylE between the embedding dimension $d = 800$ and $d = 512$ (see Table 1). MRR results in these query structures with $d = 800$ are higher than those with $d = 512$ using these datasets FB15k and FB15k-237. In the NELL995 dataset, however, MRR results with $d = 800$ are lower than those with $d = 512$. There is a different trend in these datasets as there may be

[2]Source code is available at https://github.com/nlp-tlp/cyle

an over-fitting problem when implementing CylE using the NELL995 dataset, but not for the FB15k and FB15k-237 dataset. Notice that large number of entities increase the complexity of the model using the NELL995 dataset. Precisely, the number of entities in this dataset (63,361) is considerably larger than those in the FB15k (14,951) and FB15k-237 dataset (14,505) (see Appendix A.1). Similarly, the number of relations also contribute to the complexity of the model. Although the number of relations in the NELL995 dataset (200) is less than those in the FB15k (1,345) and FB15k-237 dataset (237), this difference is slight, compared to that in the number of entities across the three datasets. With regard to negation queries, MRR results of CylE with $d = 512$ are close to those with $d = 800$ (see Table 2). Hence, there is little effect of the embedding dimension on the performance of CylE in these query structures.

5.3 Comparisons results of disjunctive queries using DNF and De Morgan's law

Table 3 shows the comparisons of MRR results (in percentage) of query structures regarding union operations using DNF and DM transformation. We compare results of CylE with those in ConE and BETA, since these models can handle the negation operations requiring for the transformation in DM queries. Overall, the MRR in DNF query structures of the approaches using ConE and CylE are higher than those in DM query structures. There is a similar trend in ConE and CylE, since these approaches share similarity in the aperture (the boundary of shapes in sector-cone and sector-cylinder respectively). Note that we use complement operations to transform union queries into DM forms, the queries in this situation are represented in geometry (sector-cone and sector-cylinder respectively). However, not all queries are represented well in geometry as discussed in (Zhang et al., 2021b). Further, we also observe a higher margin in MRR results of CylE than those in ConE, regarding most of DNF and DM query structures, using the three datasets (FB15k, FB15k-237 and NELL995).

Dataset	Model	2u-DNF	2u-DM	up-DNF	up-DM
FB15k	BetaE	40.1	25.0	25.2	25.4
	ConE	55.7	37.7	31.4	29.8
	CylE	**59.4**	**42.4**	**33.5**	**32.0**
FB15k-237	BetaE	12.4	11.1	9.7	9.9
	ConE	14.5	13.4	10.8	9.9
	CylE	**15.3**	**13.4**	**11.2**	**10.6**
NELL995	BetaE	12.2	11.0	8.5	8.6
	ConE	15.3	**14.8**	11.3	10.8
	CylE	**15.4**	13.3	**12.2**	**11.5**

Table 3: **MRR** (%) for answering FOL disjunctive query structures using DNF and DM on these datasets: FB15k, FB15k-237 and NELL995. The results of BetaE and ConE are taken from (Zhang et al., 2021b).

6 Conclusion

We have presented a novel query embeddings (QE) model using cylinder embeddings (CylE), which can handle a complete set of arbitrary FOL queries, to perform the Multi-hop Reasoning (MHR) task. To the best of our knowledge, CylE is the first 3D geometric-based QE model for MHR. Experiments show significant performance gain over previous approaches for non-negation queries. For queries with negation operations, we face a similar challenge to previous models *i.e.* low MHR performance. This is a future direction to improve CylE on these queries. This work paves the way for opening the geometric-based QE method using three dimensional shapes.

Limitations

Although CylE can learn to achieve the Multi-hop Reasoning task or answering complex queries, several limitations in this work are taken into account. First, the modeling process of logical operators (conjunction, disjunction and negation) using geometric-based perspective is an approximate method in a learning manner which may not satisfy some logic laws. This can be addressed by using fuzzy logic under fuzzy sets representation for these logical operators Chen et al. (2022). Fuzzy logic is a learning-free manner for logical operators in FOL queries. Combining this approach with the neural models to learn these operators can be a potential approach, but we leave this extension as a direction for future work.

Another limitation is that modeling union operators in EPFO queries using the DNF technique may not find all expected answer entities. Note that modeling this operator is similar to finding nearest entities to all conjunctive queries (in the DNF form), which may not an optimal solution when the geometric embeddings of these queries locate far from one to another, as mentioned in Section 5.2, queries structure $2u$ and $2i$ in particular.

Acknowledgements

This research is supported by the Australian Research Council through the Centre for Transforming Maintenance through Data Science (grant number IC180100030), funded by the Australian Government. Wei Liu acknowledges the support from ARC Discovery Projects DP150102405. Further, the authors would like to thank School of Physics, Maths and Computing – Department of Computer Science and Software Engineering, and Kaya team – IT High Performance Computing system at The University of Western Australia (UWA) for funding facilities and resources; all the anonymous reviewers for their discussions and insightful feedback.

References

Erik Arakelyan, Daniel Daza, Pasquale Minervini, and Michael Cochez. 2021. Complex query answering with neural link predictors. In *International Conference on Learning Representations (ICLR)*.

Jiaxin Bai, Zihao Wang, Hongming Zhang, and Yangqiu Song. 2022. Query2Particles: Knowledge graph reasoning with particle embeddings. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2703–2714, Seattle, United States. Association for Computational Linguistics.

Ivana Balažević, Carl Allen, and Timothy Hospedales. 2019. *Multi-Relational Poincaré Graph Embeddings*. Curran Associates Inc., Red Hook, NY, USA.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 2787–2795, Red Hook, NY, USA. Curran Associates Inc.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Ronald Brachman and Hector Levesque. 2004. *Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Xuelu Chen, Ziniu Hu, and Yizhou Sun. 2022. Fuzzy logic based logical query answering on knowledge graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4):3939–3948.

Nurendra Choudhary, Nikhil Rao, Sumeet Katariya, Karthik Subbian, and Chandan Reddy. 2021a. Probabilistic entity representation model for reasoning over knowledge graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 23440–23451. Curran Associates, Inc.

Nurendra Choudhary, Nikhil Rao, Sumeet Katariya, Karthik Subbian, and Chandan K Reddy. 2021b. Self-supervised hyperboloid representations from logical queries over knowledge graphs. In *Proceedings of the Web Conference*, WWW '21, page 1373–1384, New York, NY, USA. Association for Computing Machinery.

Brian A Davey and Hilary A Priestley. 2002. *Introduction to lattices and order*, 2 edition. Cambridge university press.

Christiane Fellbaum. 2010. *WordNet*, pages 231–243. Springer Netherlands, Dordrecht.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 5350–5360, Red Hook, NY, USA. Curran Associates Inc.

Chang Gao, Chengjie Sun, Lili Shan, Lei Lin, and Mingjiang Wang. 2020. Rotate3d: Representing relations as rotations in three-dimensional space for knowledge graph embedding. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 385–394, New York, NY, USA. Association for Computing Machinery.

Will Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. 2018. Embedding logical queries on knowledge graphs. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Zijian Huang, Meng-Fen Chiang, and Wang-Chien Lee. 2022. Line: Logical query reasoning over hierarchical knowledge graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 615–625, New York, NY, USA. Association for Computing Machinery.

Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, page 604–613, New York, NY, USA. Association for Computing Machinery.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations,* *ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3243–3253, Brussels, Belgium. Association for Computational Linguistics.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. 2019. Spherical text embedding. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. 2018. Never-ending learning. *Communications of the ACM*, 61(5):103–115.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6341–6350, Red Hook, NY, USA. Curran Associates Inc.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

Jesse Read, Concha Bielza, and Pedro Larrañaga. 2013. Multi-dimensional classification with super-classes. *IEEE Transactions on knowledge and data engineering*, 26(7):1720–1733.

Hongyu Ren, Weihua Hu, and Jure Leskovec. 2020. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In *International Conference on Learning Representations (ICLR)*.

Hongyu Ren and Jure Leskovec. 2020. Beta embeddings for multi-hop logical reasoning in knowledge graphs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Haitian Sun, Andrew O. Arnold, Tania Bedrax-Weiss, Fernando Pereira, and William W. Cohen. 2020. Faithful embeddings for knowledge base queries. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Kristina Toutanova and Danqi Chen. 2015. 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. Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. 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, Copenhagen, Denmark. Association for Computational Linguistics.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbhakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J Smola. 2017. Deep sets. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3394–3404, Red Hook, NY, USA. Curran Associates Inc.

Yao Zhang, Hongru Liang, Adam Jatowt, Wenqiang Lei, Xin Wei, Ning Jiang, and Zhenglu Yang. 2021a. GMH: A general multi-hop reasoning model for KG completion. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3437–3446, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhanqiu Zhang, Jianyu Cai, and Jie Wang. 2020. Duality-induced regularizer for tensor factorization based knowledge graph completion. *Advances in Neural Information Processing Systems*, 33:21604–21615.

Zhanqiu Zhang, Jie Wang, Jiajun Chen, Shuiwang Ji, and Feng Wu. 2021b. Cone: Cone embeddings for multi-hop reasoning over knowledge graphs. In *Advances in Neural Information Processing Systems*, volume 34.

Zhaocheng Zhu, Mikhail Galkin, Zuobai Zhang, and Jian Tang. 2022. Neural-symbolic models for logical queries on knowledge graphs. In *International Conference on Machine Learning (ICML)*.

A Further experimental details

We provide additional details for Section 5.1 in this section, divided into parts: Datasets & query structures A.1 and training & evaluation protocol A.2.

A.1 Datasets and query structures

We train and evaluate models using the same datasets (FB15k (Bollacker et al., 2008), FB15k-237 (Toutanova and Chen, 2015) and NELL995 (Xiong et al., 2017)) as those in (Ren and Leskovec, 2020) for the task of Multi-hop Reasoning (see Table 4 for a number of entities, a number of relations, a number of edges for each dataset). These datasets has been pre-processed by (Ren and Leskovec, 2020) to generate query structures for the training/validation/test set (see Table 5 for a description of these query structures and Table 6 for a description of average number of answer entities for test queries). These datasets are available at this link [3].

Dataset	Entities	Relations	Edges			
			Training	Validation	Test	Total
FB15k	14,951	1,345	483,142	50,000	59,071	592,213
FB15k-237	14,505	237	272,115	17,526	20,438	310,079
NELL995	63,361	200	114,213	14,324	14,267	142,804

Table 4: A statistical description of number of entities, relations, training/validation/test edges, reported from (Ren and Leskovec, 2020), in three datasets: FB15k, FB15k-237 and NELL995.

A.2 Training and evaluation protocol

We compare our results with these baselines (GQE, Query2Box, BetaE and ConE), taken from (Zhang et al., 2021b). We conduct all experiments using the Pytorch framework. Our implementation is done based on the original work of BetaE (Ren and Leskovec, 2020). [4] We adopt hyper-parameters, found by (Zhang et al., 2021b): the dimension of embedding $d = 800$, $\lambda_1 = 1.0$, $\lambda_2 = 2.0$, $\lambda = 0.02$, the batch size $b = 512$ and the negative sampling size $n = 128$. We also search for these hyper-parameters for best performance in MRR: the γ in the loss function [20, 30], the learning rate $\{1e^{-4}, 5e^{-5}\}$ and the scaling weight for the height variable λ_3 $\{1.0, 2.0\}$. We use a three-layer **MLP** (for a projection module) while two-layer **MLP** (for an intersection module), using 1600 dimension for hidden layers and Swiss activation function (Ramachandran et al., 2017). We run each experiment on a single NVIDIA Tesla V100 GPU. More details of hyper-parameters are shown in Table 7. Note that we also search for hyper-parameters in terms of experiments using ReLU activation function (in MLP) for ablation study. In this situation, we follow the same found hyper-parameters in Table 7 as those in the situation using Swiss activation function (the dimension of embedding d, the batch size b, the number of negative sampling size n, the maximum number of training steps m, the scaling hyper-parameters $\lambda_{i=\{1,2,3\}}$, the controlling distance λ, the learning rate l and the γ in the loss function), except for the found $\gamma = 30$ in the loss function using the FB15k-237 dataset, during the training process.

[3] https://github.com/snap-stanford/KGReasoning

[4] https://github.com/snap-stanford/KGReasoning, licensed under the MIT License.

B Additional experimental results

B.1 Error bars

Table 8 and Table 9 show the error bars of MRR for the task of Multi-hop Reasoning using our approach with CylE, for queries without and with negation respectively using the three dataset FB15k, FB15k-237 and NELL995. More specifically, we run five experiments using different seed values in $\{0, 10, 100, 1000, 10000\}$ during the initialization process (for each dataset). We estimate the average MRR for different query structures of five experiments and obtain the standard deviation (for each dataset).

In terms of queries without negation, the standard deviation of the average MRR is small, at around 0.101, 0.039 and 0.113 using the dataset FB15k, FB15k-237 and NELL995 respectively. A similar trend is observed in queries with negation operation, since the standard deviation of the average MRR is also small using the three datasets. These error bars show a level of degree in stability of MRR (for each dataset) using different values of random seed during the initialization process.

B.2 Modeling the cardinality of answer sets using correlation coefficients

It is argued that the aperture embeddings may have a correlation with the number of the answer set $[\![q]\!]$. This correlation though is not guaranteed under different circumstances (*e.g.* entities having identical relations to one another), the learnt embeddings can have a positive relationship with the number of elements (cardinality) of $[\![q]\!]$. We follow a technique

Dataset	Training Queries		Validation Queries		Test Queries	
	1p/2p/3p/2i/3i	2in/3in/inp/pin/pni	1p	Each Other	1p	Each Other
FB15k	273,710	27,371	59,097	8,000	67,016	8,000
FB15k-237	149,689	14,968	20,101	5,000	22,812	5,000
NELL995	107,982	10,798	16,927	4,000	17,034	4,000

Table 5: A statistical description of number of training/validation/test queries in different structures, preprocessed by (Ren and Leskovec, 2020), in three datasets: FB15k, FB15k-237 and NELL995.

Dataset	1p	2p	3p	2i	3i	ip	pi	2u	up	2in	3in	inp	pin	pni
FB15k	1.7	19.6	24.4	8.0	5.2	18.3	12.5	18.9	23.8	15.9	14.6	19.8	21.6	16.9
FB15k-237	1.7	17.3	24.3	6.9	4.5	17.7	10.4	19.6	24.3	16.3	13.4	19.5	21.7	18.2
NELL995	1.6	14.9	17.5	5.7	6.0	17.4	11.9	14.9	19.0	12.9	11.1	12.9	16.0	13.0

Table 6: A statistical description of average number of answers for test queries, preprocessed by (Ren and Leskovec, 2020), in three datasets: FB15k, FB15k-237 and NELL995.

Dataset	d	b	n	m	γ	l	λ_1	λ_2	λ_3	λ
FB15k	800	512	128	450k	30	0.00005	1.0	2.0	2.0	0.02
FB15k-237	800	512	128	350k	20	0.00005	1.0	2.0	2.0	0.02
NELL995	800	512	128	350k	20	0.0001	1.0	2.0	2.0	0.02

Table 7: Found hyper-parameters for the main results in three different datasets: FB15k, FB15k-237 and NELL995. d denotes the embedding dimension, b denotes the batch size, n denotes the negative sampling size, γ denotes to control the loss function, m denotes the maximum training steps, l denotes the learning rate, $\lambda_1, \lambda_2, \lambda_3$ denote scaling hyper-parameters in the projection module (see scaling function $f(x)$ in Section 4.3) and λ is to control the distance d_{con} (see Section 4.4 in the main content).

Dataset	1p	2p	3p	2i	3i	ip	pi	2u	up	AVG
FB15k	78.8 ±0.044	37.0 ±0.156	30.9 ±0.231	66.9 ±0.093	75.7 ±0.132	40.8 ±0.257	53.8 ±0.263	59.4 ±0.210	33.5 ±0.226	53.0 ±0.101
FB15k-237	42.9 ±0.105	13.3 ±0.113	11.3 ±0.070	35.0 ±0.189	49.0 ±0.168	15.7 ±0.143	27.0 ±0.108	15.3 ±0.136	11.2 ±0.075	24.5 ±0.039
NELL995	55.7 ±0.290	17.5 ±0.112	15.1 ±0.145	40.7 ±0.223	51.1 ±0.291	19.1 ±0.147	27.1 ±0.099	15.4 ±0.154	12.2 ±0.037	28.2 ±0.113

Table 8: **MRR** (%) results of CylE with error bars for answering different FOL query structures without negation ($\exists$, $\wedge$, $\vee$) using these datasets: FB15k, FB15k-237 and NELL995.

Dataset	2in	3in	inp	pin	pni	AVG
FB15k	15.7 ±0.084	16.3 ±0.059	13.7 ±0.086	7.8 ±0.065	13.9 ±0.033	13.5 ±0.020
FB15k-237	4.9 ±0.077	8.3 ±0.093	8.2 ±0.112	3.7 ±0.058	3.4 ±0.073	5.7 ±0.042
NELL995	5.4 ±0.106	7.6 ±0.081	11.3 ±0.095	3.4 ±0.019	3.7 ±0.090	6.3 ±0.046

Table 9: **MRR** (%) results of CylE with error bars for answering negation queries using these datasets: FB15k, FB15k-237 and NELL995.

of ConE (Zhang et al., 2021b) to compute this correlation. We use two types of correlation as (Ren and Leskovec, 2020; Zhang et al., 2021b): (1) Spearman's rank-order correlation coefficient (SRCC) (to measure the monotonicity relationship or the statistical dependence between the rankings of the two variables) and (2) Pearson correlation coefficient (PCC) (to measure the linear relationship between the two variables). We do not compute SRCC and PCC regarding disjunctive queries, which is the same as (Ren and Leskovec, 2020; Zhang et al., 2021b), since we model queries with disjunctions using the DNF technique. Table 10 shows SRCC of Q2B, BetaE, ConE and CylE using the FB15k dataset. No SRCC results are available in G2B for queries with negation since this model cannot handle this operation (Ren et al., 2020). Overall, the SRCC results of CylE are significantly higher

Model	1p	2p	3p	2i	3i	ip	pi	2in	3in	inp	pin	pni
Q2B	0.30	0.22	0.26	0.33	0.27	0.14	0.30	-	-	-	-	-
BetaE	0.37	0.48	0.47	0.57	0.40	0.42	0.52	0.62	0.55	0.46	0.47	0.61
ConE	0.60	0.68	0.70	0.68	0.52	0.56	0.59	**0.84**	0.75	0.61	0.58	0.80
CylE	**0.61**	**0.84**	**0.81**	**0.74**	**0.62**	**0.79**	**0.77**	0.83	**0.78**	**0.67**	**0.66**	**0.83**

Table 10: **Spearman's rank** correlation coefficient between embeddings of learnt embeddings and a number of answers for queries using the dataset FB15k. Results of Q2B, BetaE and ConE are taken from (Zhang et al., 2021b).

Dataset	Model	1p	2p	3p	2i	3i	ip	pi	2in	3in	inp	pin	pni
FB15k-237	Q2B	0.18	0.23	0.27	0.35	0.44	0.20	0.36	-	-	-	-	-
	BetaE	0.41	0.50	0.57	0.60	0.52	0.44	0.54	0.69	0.58	0.51	0.47	0.67
	ConE	0.70	0.71	**0.74**	0.82	0.72	0.62	0.70	**0.90**	**0.83**	**0.66**	0.57	**0.88**
	CylE	**0.71**	**0.80**	0.73	**0.83**	**0.77**	**0.74**	**0.84**	0.84	0.79	0.59	**0.60**	0.83
NELL995	Q2B	0.15	0.29	0.31	0.38	0.41	0.35	0.36	-	-	-	-	-
	BetaE	0.42	0.55	0.56	0.59	0.61	0.54	0.60	0.71	0.60	0.35	0.45	0.64
	ConE	0.56	0.61	0.60	**0.79**	**0.79**	0.58	0.74	**0.90**	**0.79**	0.56	0.48	**0.85**
	CylE	**0.57**	**0.75**	**0.65**	0.73	0.72	**0.70**	**0.76**	0.85	0.76	**0.59**	**0.63**	0.81

Table 11: **Spearman's** rank correlation coefficient between embeddings of learnt embeddings and a number of answers for queries using the dataset FB15k-237 and NELL995. Rank correlation results of Q2B, BetaE and ConE are taken from (Zhang et al., 2021b).

Dataset	Model	1p	2p	3p	2i	3i	ip	pi	2in	3in	inp	pin	pni
FB15k	Q2B	0.08	0.22	0.26	0.29	0.23	0.13	0.25	-	-	-	-	-
	BetaE	0.22	0.36	0.38	0.39	0.30	0.31	0.31	0.44	0.41	0.34	0.36	0.44
	ConE	0.33	0.53	0.59	0.50	0.45	0.42	0.37	0.65	0.55	**0.50**	**0.52**	0.64
	CylE	**0.36**	**0.68**	**0.62**	**0.66**	**0.59**	**0.70**	**0.70**	**0.68**	**0.59**	0.46	0.48	**0.71**
FB15k-237	Q2B	0.02	0.19	0.26	0.37	0.49	0.20	0.34	-	-	-	-	-
	BetaE	0.23	0.37	0.45	0.36	0.31	0.33	0.32	0.46	0.41	0.39	0.36	0.48
	ConE	**0.40**	0.52	**0.61**	0.67	0.69	0.49	0.47	**0.71**	**0.66**	**0.53**	**0.47**	**0.72**
	CylE	0.36	**0.56**	0.53	**0.67**	**0.71**	**0.55**	**0.71**	0.64	0.59	0.41	0.35	0.64
NELL995	Q2B	0.07	0.21	0.31	0.36	0.29	0.34	0.24	-	-	-	-	-
	BetaE	0.24	0.40	0.43	0.40	0.39	0.40	0.40	0.52	0.51	0.26	0.35	0.46
	ConE	**0.48**	0.45	0.49	**0.72**	**0.68**	0.39	0.52	**0.74**	0.66	**0.38**	0.34	**0.69**
	CylE	0.45	**0.60**	**0.50**	0.64	0.63	**0.52**	**0.60**	0.69	**0.66**	0.35	**0.47**	0.64

Table 12: **Pearson** correlation coefficient between embeddings of learnt embeddings and a numer of answers for queries using the dataset FB15k, FB15k-237 and NELL995. Correlation results of Q2B, BetaE and ConE are taken from (Zhang et al., 2021b).

than those in ConE in most of query structures, by a large margin. These results demonstrate the expressiveness of modeling cardinality of answer sets using CylE. Note that SRCC results using CylE also significantly exceed the other previous models.

We show additional results of Spearman's Rank Correlation Coefficient (SRCC) in Table 11 using the dataset FB15k-237 and NELL995. In terms of the dataset FB15k-237, although the SRCC results in several query structures (*e.g.* $3p, 2in, 3in, inp, pni$) using the approach in CylE are lower than those in ConE, the SRCC in the rest of query structures using CylE outperform those in ConE. A slight decrease in SRCC results, from CylE to ConE, can be mostly observed in queries that are involved in negation operations. These observations are due to the fact that the height variable might not play a role in embedded queries with complement operations. Similar to the dataset FB15k-237, in the dataset NELL995, the SRCC results in some query structures (*e.g.* $2i, 3i, 2in, 3in, pni$), particularly in queries with negation operations, are also lower than those in ConE. However, most of SRCC results in queries without negation operations surpass SRCC results

Dataset	Activation	1p	2p	3p	2i	3i	ip	pi	2u	up	AVG
FB15k	ReLU	75.1	36.5	**31.0**	65.8	74.5	39.9	52.5	56.9	**33.6**	51.8
	Swiss	**78.8**	**37.0**	30.9	**66.9**	**75.7**	**40.8**	**53.8**	**59.4**	33.5	**53.0**
FB15k-237	ReLU	42.1	13.1	11.3	34.8	**49.1**	14.8	26.5	14.7	11.2	24.2
	Swiss	**42.9**	**13.3**	**11.3**	**35.0**	49.0	**15.7**	**27.0**	**15.3**	**11.2**	**24.5**
NELL995	ReLU	54.3	16.3	14.2	40.5	51.0	17.9	26.0	14.8	11.3	27.4
	Swiss	**55.7**	**17.5**	**15.1**	**40.7**	**51.1**	**19.1**	**27.1**	**15.4**	**12.2**	**28.2**

Table 13: Average **MRR** (%) results of CylE in different FOL query structures without negation ($\exists$, $\wedge$, $\vee$) using ReLU and Swiss activation for the MLP in these datasets: FB15k, FB15k-237 and NELL995.

in ConE. In comparisons with BetaE and Q2B, SRCC results in all query structures (with and without negation) using CylE are significantly higher than SRCC results in these models.

The similar trend of results are also observed in the Pearson Correlation Coefficient (PCC) between the aperture embeddings and the cardinality of answers set using the three dataset FB15k, FB15k-237 and NELL995 (see Table 12).

B.3 Further ablation study

We compare the performance of CylE using the different activation function (ReLU and Swiss) for the MLP networks (in the projection and intersection module), during the training process in this ablations study (see Table 13 and 14). Overall, the

Dataset	Activation	2in	3in	inp	pin	pni	AVG
FB15k	ReLU	15.6	16.2	13.4	7.8	13.7	13.3
	Swiss	**15.7**	**16.3**	**13.7**	**7.8**	**13.9**	**13.5**
FB15k-237	ReLU	4.7	8.1	8.1	3.7	3.2	5.6
	Swiss	**4.9**	**8.3**	**8.2**	**3.7**	**3.4**	**5.7**
NELL995	ReLU	5.1	7.5	11.2	3.3	3.5	6.1
	Swiss	**5.4**	**7.6**	**11.3**	**3.4**	**3.7**	**6.3**

Table 14: Average **MRR** (%) results of CylE for different FOL query structures with negation using ReLU and Swiss activation for MLP in these datasets: FB15k, FB15k-237 and NELL995.

average MRR results in most of query structures for the approach using Swiss activation are slightly higher than those in the approach using ReLU activation in the three datasets FB15k, FB15k-237 and NELL995. Since the Swiss activation was shown to be an efficient activation function (Ramachandran et al., 2017) for the MLP networks.

C Computational complexity

The computational complexity of CylE is similar to ConE since these models share similarity in geometric shapes. Note that the computational complexity of ConE and G2B is also similar to one another (Zhang et al., 2021b). It is arguably that CylE, ConE and G2B have similar computational complexity. Assuming a Disjunctive Normal Form (DNF) query q: that consists of conjunctive queries $q_{i:1\rightarrow n}$, where $q = q_1 \vee \cdots \vee q_n$. The computational complexity of CylE for answering q is equivalent to the computational complexity for answering the number n of conjunctive queries q_i. This answering process is involved in the estimation of a sequence of geometric sector-cylinder operations in which a constant time can be taken for each operation, then performing a range search which can be achieved using techniques according to Locality Sensitive Hashing (Indyk and Motwani, 1998).

Models	GQE	Q2B	BetaE	ConE	CylE	CylE
Emb. dimension d	800	800	800	800	800	500
Running time (s)	75.87	81.43	168.91	119.90	154.79	121.65

Table 15: Average running time (seconds) for the first 500 training steps in different approaches using the FB15k-237 dataset.

We record the average running time (seconds) for the first 500 training steps using the same dimensional embedding for all models (GQE, G2B, BetaE, ConE and CylE) and another one with lower embedding dimension for CylE, on the same single NVIDIA Tesla V100 GPU, for fair comparisons. The lower this value is, the faster training process is. Overall, the fastest model is GQE while the slowest model is BETA. The running time of CylE is slightly slower than ConE, but CylE with lower dimension ($d = 500$) is on par with ConE. Further, the running time of CylE is also faster than BetaE.

D The range values for the height

The range of values for the height variable $\boldsymbol{\theta}_{he}$ for an arbitrary embedded query $\mathbf{V}_q = (\boldsymbol{\theta}_{ax}, \boldsymbol{\theta}_{ap}, \boldsymbol{\theta}_{he})$ can be varied without any constraints. For example, this range of values can be infinite. However, there is a numerical problem in a way that values of $\boldsymbol{\theta}_{he}$

might dominate those in the semantic center $\boldsymbol{\theta}_{ax}$ and the aperture $\boldsymbol{\theta}_{ap}$. This problem can lead to a reduction in the performance of MHR in this situation. Since $\boldsymbol{\theta}_{ax} \in [-\pi, \pi)^d$ and $\boldsymbol{\theta}_{ap} \in [0, 2\pi]^d$ are in a small range of values, compared to infinite range of $\boldsymbol{\theta}_{he}$. Thus, we set a small range of values for the height variable and scale the range to $(-\pi, \pi)$, to avoid the numerical issues, making this variable have a consistent systematic range of values (based on the multiples of π) as those in the semantic center $\boldsymbol{\theta}_{ax}$ and the aperture $\boldsymbol{\theta}_{ap}$.