Spherical Embeddings for Atomic Relation Projection Reaching Complex Logical Query Answering

Anonymous Author(s)

ABSTRACT

Projecting knowledge graph queries into an embedding space using geometric models (points, boxes and spheres) can help to answer queries for large incomplete knowledge graphs. In this work, we propose a symbolic learning-free approach using fuzzy logic to address the shape-closure problem that restricted geometric-based embedding models to only a few shapes (*e.g.* ConE) for answering complex logical queries. The use of symbolic approach facilitates non-closure geometric models (*e.g.* point, box) to handle logical operators (including negation). This enabled our newly proposed spherical embeddings (SpherE) in this work to use a polar coordinate system to effectively represent hierarchical relation. Results show that the SpherE model can answer existential positive first-order logic and negation queries. We show that SpherE significantly outperforms the point and box embeddings approaches while generating semantically meaningful hierarchy-aware embeddings.

CCS CONCEPTS

• **Computing methodologies** → **Logical and relational learning**; **Artificial intelligence**; **Knowledge representation and reasoning**.

KEYWORDS

Complex logical query answering, query embeddings, knowledge graphs

ACM Reference Format:

Anonymous Author(s). 2024. Spherical Embeddings for Atomic Relation Projection Reaching Complex Logical Query Answering. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Research in *query embedding techniques* have received growing attention because of their demonstrable abilities in **C**omplex **L**ogical **Q**uery **A**nswering (CLQA) over large incomplete knowledge graphs [30, 41]. Complex queries are decomposed into atomic ones, which are connected via logical operators into a computation graph, as illustrated in the top diagram of Figure 1. With nodes and relations in the computation graph as learnable vectors, knowledge graph reasoning can be realized through parameter estimation. Downstream tasks of CLQA can be applied to question answering [39] and recommender systems [15, 16, 35].

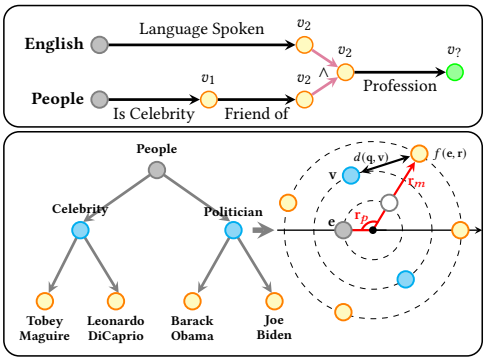


Figure 1: (Top): A complex query in a computation graph "*What is profession of someone who speaks English and is friend of a celebrity?*", has several atomic queries: "*Who speaks English?*", "*Who is a celebrity?*" and "*Who is friend of the celebrity?*", and "*What is profession of this person?*". (Bottom): Illustration of the hierarchical structure in the query using a polar coordinate system, adapted from HAKE [48].

Different mathematical interpretations of the vector space lead to several recent state-of-the-art query embedding models. Geometric interpretations are among the most popular, *e.g.* a query embedding as a point [14], a box [31] and a cone [28, 49]. A key factor for their success is the intuitive modelling of the logical operators: *conjunction*, *disjunction* and *negation* as set intersection, union and complement of the corresponding shapes, respectively.

However, such modelling has its own challenges. Taking the common shapes (*e.g.* point, box) as examples, neither point embeddings nor boxes embeddings by default can answer negation queries [32]. The complement of a point or a box is no longer a point or a box [32, 49]. The geometric-based approach limits to utilizing a few special shapes like cone embeddings [49] to resolve the issue. For example, *'the closure of complement of a cone is still a cone'* [49]. In other words, the geometric approach is challenging in representing the query using many valid geometric shapes due to the need of shape-closure. This approach suggests a necessary condition of closed complement (for answering negation queries) and closed intersection (for answering conjunctive queries) in shapes.

Probabilistic reasoning and fuzzy logic [21, 22] interpretations of the embedding vectors can address the non-geometric closure challenge while capturing the uncertain nature of such vector representations. Recent methods in probabilistic interpretations use distributions, *e.g.* beta [32]; gamma [46]; Gaussian [9, 43]), but still rely on the neural learning approaches to handle some logical operators (AND, OR). Recent works apply fuzzy logic in complex logical query answering from a query optimization perspective [1, 4], or

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

fuzzy sets viewpoint [51]. These fuzzy logic based symbolic approaches are shown to be effective in reducing the dependency on deep learning to model logical operators in complex queries, and is empirically useful in uncertainness and vagueness.

We first make a contribution in this research by proposing a unified solution to enable CLQA of geometric-based embeddings for non-closure shapes, such as point embeddings [14] and box embeddings [31]. The geometric embeddings are trained using atomic relation projection (1p) only, while a learning-free symbolic approach makes uses of fuzzy t-norms and t-conorms for logical operators to answer complex queries. Notice that point embeddings [14] and box embeddings [31] though cannot can handle negation but they can model conjunction and disjunction operators by default. This aspect can reduce the necessity of using the fuzzy symbolic approach in handling these logical operators but not negation.

This realization leads us to the next contribution of this research. We propose **Spher**ical **E**mbeddings (SpherE), which considers a central point in polar coordinate system in a similar fashion as HAKE [48], for answering atomic queries in origin or handling atomic relation projection operator only. Modelling negation, conjunction and disjunction operators using SpherE is challenging as the complement of a sphere, the intersection and union of spheres are no longer of the same spherical shape. Hence, the SpherE is the most limited geometric model compared to point and box embeddings. However, we demonstrate that the symbolic fuzzy logic realization of logical operators enables SpherE to generalize to CLQA, making SpherE achieve the full ability as ConE [49].

SpherE elegantly makes use of the symbolic approach to circumvent the non-closure problem of the spherical shape while being capable of learning hierarchy-aware vector representations. Despite the potential ability to capture hierarchical relationships using polar coordinates (as shown in the bottom diagram of Figure 1), there is little work on spherical representation on the task of CLQA.

Extensive empirical results verify that SpherE in addition to point and box embeddings can generalize to CLQA when augmenting with the symbolic approach even though these geometric models are trained on atomic queries only. This observation highlights the vital role of atomic relation projection in CLQA. Moreover, results show that using the symbolic approach over atomic projection can negate the necessary condition in shape-closure for existing geometric models in CLQA. Our results show that combining the visual intuitions of geometric embeddings with the logical intuitions of fuzzy set theory provides a promising approach towards a performant and explainable model for complex logical query answering.

2 RELATED WORKS

CLQA using geometry and probabilistic distributions. The geometric embeddings approaches include GQE [14] (vectors as points), Query2Particles [3] (particles), Query2Box [31] and NewLook [24] (hyper-boxes), HypE [10] (hyperboloids), ConE [49] (cones) with SConE [28] (simplified cones), CylE [27] (unbounded cylinders), HaLk [42] (arcs), RELG [50] (torus). The distribution-based approaches include BetaE [32] (Beta distributions) followed by its variant LinE [17], GammaE [46] (Gamma distributions), PERM [9] (Gaussian distributions), and NMP-QEM [25] and Query2GMM [43] (mixture of Gaussian distributions). A common strategy in these works is to interpret the process of logical reasoning (conjunction) in embedded queries through intersection of object or distribution shapes.

Spheres as an obvious candidate shape have been used for other tasks. For example, SEPA [13] uses hyper-sphere embeddings (in both Euclidean and Hyperbolic spaces), but for the link prediction task. Similarly, JoSE [26] uses spheres (in Euclidean space) but for the text similarity and clustering task. Besides, DGS [18] uses complex non-Euclidean geometric space to model two-view KGs: '*instance-view entities*' for cyclic structures and '*ontology-view concepts*' for hierarchical structures. DGS works on triple completion and entity typing tasks which can answer atomic queries. In this work, we use spherical embeddings for the task of answering atomic and complex query. Note that the central point of spherical embeddings in our representation is in the polar coordinate system, compared to the central point of box embeddings BoxTaxo [19] where this system is not available.

We use spherical embeddings, point and box embeddings to learn the task of atomic relation projection. Then, leveraging the same fuzzy symbolic approach over scoring values obtained from any of these geometric models can help to answer complex logical queries. Hence, the complex logical query answering relies on the basic learning of the early task of atomic relation projection. Other geometric-based embedding spaces can achieve this task, which therefore suggests to open research for future works to ease the selection of many geometric models.

Table 1: Comparison of backbone models when generalizing to CLQA using the symbolic fuzzy approach.

Backbone Approach	AQA Model	CLQA Model
GNN-QE [51]	Graph Neural Networks (R-GCNs [33], CompGCN [38], NBFNet [52])	N/A
LMPNN [40]	Knowledge graph embeddings (RESCAL [29], TransE [6], DistMult [45], ComplEx [37], ConvE [12], RotatE [34])	N/A
CQD$^{\mathcal{A}}$ [2]	Knowledge graph embeddings (ComplEx [37])	N/A
QTO [4]	Knowledge graph embeddings (TransE [6], ComplEx [37], RotatE [34])	N/A
Ours	SpherE, (GQE) [14], (Query2Box) [31],	GQE [14], Query2Box [31], BetaE [32], ConE[49]

Logic-based approach for CLQA. The logic-based approaches include these models: CQD [1], CQD$^{\mathcal{A}}$ [2], QTO [4], LMPNN [40], FIT [47], GNN-QE [51] and FuzzQE [8]. These use symbolic fuzzy logic to allow backbone models such as Knowledge Graph Embeddings KGEs (*e.g.* ComplEx [37]) or Graph Neural Networks GNNs (*e.g.* NBFNet [52]) to answer complex logical queries. Nevertheless, little work takes advantage of the symbolic approach to achieve the logical reasoning ability beyond object shapes (*e.g.* points, boxes and hyper-spheres) for negation, conjunction and disjunction.

A common thing of the backbone models in existing logic-based approaches is that these backbones (e.g., RESCAL [29], TransE [6], DistMult [45], ComplEx [37], ConvE [12], RotatE [34]) are first used for link prediction or knowledge graph completion task which can answer atomic queries. Our approach, on the other hand, use a range of backbones models as shown in Table 1, including those for answering atomic queries and those for answering complex logical queries by default. These models provide a broader view of generalizing atomic query answering (AQA) to CLQA than the existing approaches when augmenting with fuzzy logic.

3 PRELIMINARIES

3.1 Knowledge graphs

A knowledge graph ($\mathcal{G}$) is a set of triples with head, tail entity connected through relations. Let's denote the collection of entities as a set ($\mathcal{E}$), where $\mathcal{E} = \{\ldots, e_i, \ldots\}$ and $e_i \in \mathcal{E}$. The collection of relations is denoted as $\mathcal{R}$, where $\mathcal{R} = \{\ldots, r_i, \ldots\}$ and $r_i \in \mathcal{R}$. A binary relation function r (i.e. an atomic formula) defines whether there is a connection between a pair of entities e and v, where $r(e, v)$ = True if a relation r exists between e and v, and $r(e, v)$ = False otherwise. Knowledge graph queries are expected to be expressed in logical form, precisely in existential first-order logic form as defined in as follows.

3.2 Existential first-order logic EFOL queries

Following the notation of Disjunctive Normal Form (DNF) [11] in [32], an existential first-order logic (EFOL) query in DNF as a disjunction of n conjunctive queries is defined by:

$$q(v) = v \cdot \exists v_i \in \mathcal{E} : q_1 \vee \ldots \vee q_n \tag{1}$$

where the conjunctive query is $q_i = (a_i^1 \wedge \ldots \wedge a_i^m)$. An *atomic formula a* or *atom* [7] is defined by: $a_i^j = r(e, v)$ or $\neg r(e, v)$ for $r \in \mathcal{R}$, where entity e is either constant c or variable v, an answer $v_?$ is free variable, and $e, c, v_i, v_? \in \mathcal{E}$. EFOL operators are there exists ($\exists$), conjunction ($\wedge$), disjunction ($\vee$), negation ($\neg$). Figure 1 (Top) shows an EFOL query as a compound of atomic queries, connected using ($\wedge$).

3.3 Problem formulation

We hereby subscribe to a fuzzy logic interpretation of conjunction, disjunction and negation according to [21, 22].

Definition 3.1. A triangular norm (t-norm) ($\top$) is a function of two fuzzy variables, each represents a fuzzy membership value and $\top(x, y)$: $[0, 1] \times [0, 1] \longrightarrow [0, 1]$ and its dual t-conorm ($\perp$) is that function given by $\perp(x, y) = 1 - \top(1 - x, 1 - y)$ [21, 22].

Given a relation (r) between a source entity (e) and a target entity (v), we use fuzzy truth value $T^q \in [0, 1]$ as mentioned in [4] to represent atomic query $q = r(e, v)$, which suggests whether there is a link r between (e, v). The truth value $T^q(v)$ can be given with respect to a variable v, where $q = r(v', v) \wedge q'(v')$ and may be defined recursively. To generalize the atomic queries to answer complex queries, we use EFOL operators ($\exists, \neg, \wedge, \vee$) over the truth values: We follow query computation tree optimization QTO in [4] to use fuzzy logic via t-norm ($\top$) by Definition 3.1, to model these EFOL operators.

$$T^{\exists v' \cdot q}(v) = \max_{v \in \mathcal{E}} T^q(v, v'), \tag{2}$$

$$T^{\neg q}(v) = 1 - T^q(v), \tag{3}$$

$$T^{q \wedge q'}(v) = \top\big(T^q(v), T^{q'}(v)\big). \tag{4}$$

For disjunction ($\vee$), we use t-conorm based on De Morgan's law:

$$T^q(v) = \underset{1 \le i \le k}{\perp} T^{q'}(v_i) \overset{\text{DM}}{\underset{\text{law}}{:=}} \overline{\underset{1 \le i \le k}{\top} \overline{T^{q'}(v_i)}}, \tag{5}$$

where $(\overline{x})$ is negation $(\neg x)$. The problem defined in Eq. (1) can be reformulated using truth values:

$$\begin{aligned} q(v) &= \exists v \cdot \arg\max_{v \in \mathcal{E}} r(v', v) \top T^{q'}(v'), \\ &= \arg\max_{v \in \mathcal{E}} \top\big(T^{q_i}(v), \cdots, T^{q_n}(v)\big). \end{aligned} \tag{6}$$

4 METHODOLOGY

We describe an overview of this section in three stages: (1) learning atomic queries or atomic relation projection using spherical embeddings (4.1), (2) generating atomic query matrix (4.2), and (3) generalizing to complex logical query using the fuzzy symbolic approach (4.3) as shown in Figure 2. In the first stage, the task of learning to answer atomic queries can be achieved by using these geometric models (GQE, Q2B). However, these backbone models can handle complex logical queries in original setting. We propose SpherE as a basic model that can learn to answer atomic queries only rather than complex logical queries. This approach is vital to show the necessity of using the symbolic fuzzy approach to generalize to complex logical query answering in a later stage. After training a model for answering atomic queries, in the second stage, we can obtain the scoring values of all pairs of entities in the KG given a relation. Notice that the scoring value is calculated based on a distance between two entities given a relation, using a pre-trained model from the first stage (*e.g.* spherical embeddings). The scoring values are then converted to the truth values to generate the atomic query matrix of all entities for a relation. For all relations, we therefore obtain the atomic query tensor. In the third stage, we interpret fuzzy logic to generalize to complex logical queries through truth value vectors. Given an atomic query matrix and an atomic query tensor, the truth value vectors are calculated based on the product t-norm system for handling FOL logical operators ($\exists, \neg, \wedge, \vee$). We describe the details in the following:

4.1 Spherical embeddings for atomic queries

Definition 4.1. A k-sphere is a hypersphere that is a set of $(k+1)$-dimensional points, each has a constant distance (called radius) to a central point.

We represent an entity and an atomic query as the $(d-1)$-sphere using a pair of two variables $(\mathbf{c}, \tau)$ by Definition 4.1 as those in [13], where $(\mathbf{c} \in \mathbb{R}^{2d})$ is the central point and $(\tau \in \mathbb{R}^+)$ is the radius. Figure 3 shows examples of different $(d-1)$-spheres, where a 0-sphere is a point when $(\tau = 0, d = 1)$, 1-sphere is a 2D circle when $(\tau > 0, d = 2)$ and 2-sphere is a surface of 3D ball when $(\tau > 0, d = 3)$. We adapt representation learning of HAKE [48] to consider the central point $(\mathbf{c})$ in the *polar coordinate system*. Thus,

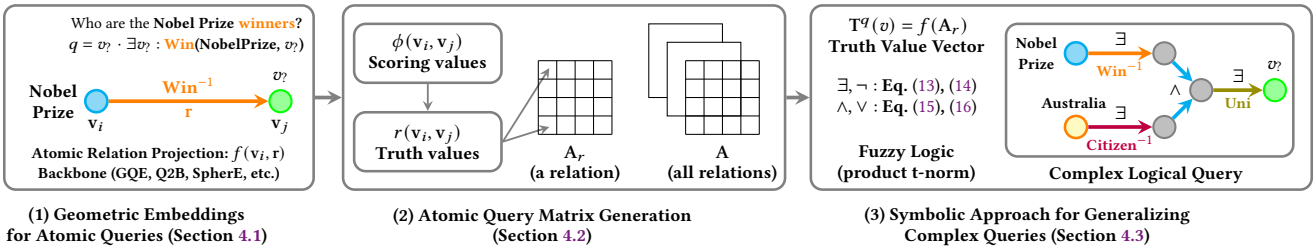


Figure 2: An overview of methodology for the use of symbolic fuzzy approach to enable geometric models to answer complex logical queries. In stage 3, a query is that "In which universities do the Nobel Prize winners who are Australian citizens graduate?" with the corresponding FOL query "$q = v_? \cdot \exists v : \text{Win}(\text{NobelPrize}, v) \wedge \text{Citizen}(\text{Australia}, v) \wedge \text{University}(v_?, v)$" [28]. This complex query has an atomic query as displayed in stage 1.

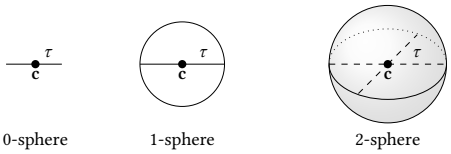


Figure 3: A point, a circle and a ball surface.

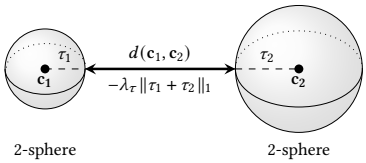


Figure 4: Illustration of distance between two 2-sphere: one for an atomic query and another for a target entity.

we use a pair of variables: radial coordinate (or modulus part) and angular coordinate (or phase part) to represent coordinate of the central point. Note that we use subscripts ($\mathbf{c}_m \in \mathbb{R}^d, \mathbf{c}_p \in \mathbb{R}^d$) to denote the modulus and phase part respectively. In general, spherical embeddings are therefore $(\mathbf{c}_m, \mathbf{c}_p, \tau) \in \mathbb{R}^{2d+1}$.

Relation projection. We model a relation projection from a head entity $\mathbf{e} = (\mathbf{e}_m, \mathbf{e}_p, \tau_e)$ to a tail entity as an atomic query $\mathbf{q} = (\mathbf{q}_m, \mathbf{q}_p, \tau_q)$ via a relation $\mathbf{r} = (\mathbf{r}_m, \mathbf{r}_p, \tau_r)$ using a *relation projection function* $f : (\mathbf{e}, \mathbf{r}) \rightarrow \mathbf{q}$:

$$\mathbf{q} = f(\mathbf{e}, \mathbf{r}). \tag{7}$$

Figure 1 (Bottom) shows an example of $f(\mathbf{e}, \mathbf{r})$ to a target entity $\mathbf{v} = (\mathbf{v}_m, \mathbf{v}_p, \tau_v)$ using spherical embeddings for only *central points* on a polar coordinate system. Radial coordinate is to distinguish entities at different hierarchical levels, angular coordinate is to distinguish entities at the similar levels. More specifically, we adapt HAKE [48] to interpret the function $f(.)$ for relation projection.

Distance functions. Given an atomic query $\mathbf{q} = (\mathbf{q}_m, \mathbf{q}_p, \tau_q)$ and a target entity $\mathbf{v} = (\mathbf{v}_m, \mathbf{v}_p, \tau_v)$, we calculate the distance $d(\mathbf{q}, \mathbf{v})$ as that in [48] as follows:

$$d(\mathbf{q}, \mathbf{v}) = \lambda_m d(\mathbf{q}_m, \mathbf{v}_m) + \lambda_p d(\mathbf{q}_p, \mathbf{v}_p), \tag{8}$$

$$d(\mathbf{q}, \mathbf{v}) \leftarrow d(\mathbf{q}, \mathbf{v}) - \lambda_\tau \|\tau_q + \tau_v\|_1, \tag{9}$$

where $d(\mathbf{q}, \mathbf{v})$ in Eq.(8) is the original distance between the two central points of corresponding hyper-spheres. ($\lambda_m, \lambda_p \in \mathbb{R}^+$) are hyper-parameters for modulus distance $d(\mathbf{q}_m, \mathbf{q}_p)$ and phase distance $d(\mathbf{q}_p, \mathbf{v}_p)$ respectively, which are as follows:

$$d(\mathbf{q}_m, \mathbf{v}_m) = \|\mathbf{q}_m - \mathbf{v}_m\|_2,$$

$$d(\mathbf{q}_p, \mathbf{v}_p) = \|\sin\left((\mathbf{q}_p - \mathbf{v}_p)/2\right)\|_1,$$

where $\|\cdot\|_{1,2}$ is the $L_{1,2}$ norm. We adjust the original distance $d(\mathbf{q}, \mathbf{v})$ by subtracting a sum of radius using a hyper parameter ($\lambda_\tau \in \mathbb{R}^+$) in Eq. (9) to control the outside and inside distance w.r.t. borders of spheres as shown in Figure 4. We use the similar loss as [49] to train the model during the optimization process:

$$\mathcal{L} = -\log \sigma\big(\gamma - d(\mathbf{q}, \mathbf{v})\big) - \frac{1}{n}\sum_{k=1}^{n} \log \sigma\big(d(\mathbf{q}, \mathbf{v}'_k) - \gamma\big),$$

where $\sigma(\cdot)$ is an activation function (*e.g.* sigmoid), ($\gamma \in \mathbb{N}^+$) is a hyper-parameter, ($\mathbf{v}'_k$) is the k-th negative answer and (n) is a number of negative sampling answers [28].

4.2 Atomic query matrix generation

Definition 4.2. Given a knowledge graph ($\mathcal{G}$), the scoring value between a pair of entities (v_i, v_j) is 1 if they are truly connected and 0 otherwise:

$$r(v_i, v_j) = \begin{cases} 1, & \text{if } (v_i, r, v_j) \in \mathcal{G}, \\ 0, & \text{otherwise } (v_i, r, v_j) \notin \mathcal{G}. \end{cases}$$

After having a pre-trained model using SpherE as described in the previous section, we can generate an *atomic query matrix* $\mathbf{A}_r \in [0, 1]^{|\mathcal{E}|\times|\mathcal{E}|}$ [1] for all entities given a relation as that in [4]. Each element of the matrix is score of atom $r(v_i, v_j)$ whether there is a relation ($r \in \mathcal{R}$) between two entities (v_i, v_j), assuming the matrix locates the i-th row and j-th column. Overall, a tensor $\mathbf{A} \in [0, 1]^{|\mathcal{R}|\times|\mathcal{E}|\times|\mathcal{E}|}$ fills scoring values for all relations. More specifically, a scoring function $\phi(\mathbf{v}_i, \mathbf{v}_j) \in \mathbb{R}$ outputs the scoring value for a pair of entities over the entity set. This function can be a distance or similarity score in geometric models (GQE, Q2B, SpherE) for the AQA or CLQA tasks, which is taken from one of

[1] This is *a.k.a.* neural adjacency or neural relation matrix.

these pre-trained models. For example, we use SpherE to obtain the scoring value as follows:

$$\phi(\mathbf{v}_i, \mathbf{v}_j) = \gamma - d(\mathbf{v}_i, \mathbf{v}_j) = \gamma - d(\mathbf{q}_i, \mathbf{v}_j), \tag{10}$$

where γ is a non-negative margin, $(\mathbf{q}, \mathbf{v})$ are spherical embeddings of an atomic query and a target entity respectively. We rewrite $d(\mathbf{v}_i, \mathbf{v}_j)$ to $d(\mathbf{q}_i, \mathbf{v}_j)$, as the atomic query has only one source entity (head or tail) (see Eq. (8) & (9) for $d(\mathbf{q}_i, \mathbf{v}_j)$) (see Appendix A.1 for further details of scoring functions in other CLQA models).

Normalization. We convert real scoring values to truth values $r(\mathbf{v}_i, \mathbf{v}_j) \in [0, 1]$ using a monotonically non-decreasing function $\sigma(x) \in [0, 1]$ such as softmax function:

$$r(\mathbf{v}_i, \mathbf{v}_j) = \sigma\big(\phi(\mathbf{v}_i, \mathbf{v}_j)\big) \cdot n_i = \frac{\exp\big(\phi(\mathbf{v}_i, \mathbf{v}_j)\big)}{\sum_j^{|\mathcal{E}|} \exp\big(\phi(\mathbf{v}_i, \mathbf{v}_j)\big)} \cdot n_i, \tag{11}$$

where (n_i) is the number of tail entities (v_j) that are reached by head entities (v_i). This is due to the fact that a head can have multiple tails [4]: $(v_i, r, v_j) \in \mathcal{G} | v \in \mathcal{E}$. To ensure the truth values are in between $[0, 1]$, we set a threshold $(t = 0.0001)$ to assign those greater than one to $(1 - t)$ by definition 4.2: $r(\mathbf{v}_i, \mathbf{v}_j) \leftarrow \min\big(r(\mathbf{v}_i, \mathbf{v}_j), 1 - t\big)$.

Non-zero values. The truth values should be equal to one by Definition 4.2 if edges exist. Besides, the relation projection process can efficiently be done by considering non-zero values. As the atomic query matrix $(\mathbf{A}_r)$ is sparse and the tensor $(\mathbf{A})$ has space complexity $|\mathcal{E}|^2 \times |\mathcal{R}|$ w.r.t. the number of entities and relations. We apply a tiny threshold (ϵ) as suggested by [4] to store and load the matrix $\mathbf{A}_r$ (or the tensor $\mathbf{A}$) efficiently in a single GPU, by setting truth values less than the threshold to zero:

$$r(\mathbf{v}_i, \mathbf{v}_j) = \begin{cases} 1, & \text{if } (v_i, r, v_j) \in \mathcal{G}, \\ 0, & \text{if } r(\mathbf{v}_i, \mathbf{v}_j) < \epsilon, \\ r(\mathbf{v}_i, \mathbf{v}_j), & \text{otherwise.} \end{cases} \tag{12}$$

4.3 Symbolic approach for generalizing complex queries

We outline a computation approach based on QTO [4] of EFOL operators in a complex logical query, which may be defined recursively, using *product* t-norm system by Definition 3.1 to interpret conjunction: $\top_{\text{prod}}(x, y) = x \cdot y$ where $x, y \in [0, 1]$. Other systems such as Gödel and Łukasiewicz [21] can be used in this situation.

Existential quantifier $(\exists)$ *and Negation* $(\neg)$. Given the atomic query matrix $\mathbf{A}_r$, we define $\mathbf{T}^q(v) \in [0, 1]^{1 \times |\mathcal{E}|}$ the embeddings of truth value $T^q(v)$ or truth value vector, which can model $(\exists)$ operator as follows:

$$\mathbf{T}^q(v) = \begin{cases} (\mathbf{A}_r)_i, & \text{if constant,} \\ \max_j\big(\mathbf{A}_r \odot \mathbf{T}^T(v')\big), & \text{if variable,} \end{cases} \tag{13}$$

where $(.)^T$ is matrix transpose operator, $\odot$ denotes element-wise operator (Hadamard product) in all sections. For negation $(\neg)$, $\mathbf{T}^q(v)$ is calculated using fuzzy negator over the atomic query matrix:

$$\mathbf{T}^q(v) = \begin{cases} (\mathbf{1} - \mathbf{A}_r)_i, & \text{if constant,} \\ \max_j\big((\mathbf{1} - \mathbf{A}_r) \odot \mathbf{T}^T(v')\big). \end{cases} \tag{14}$$

Conjunction $(\wedge)$ *and Disjunction* $(\vee)$. Assuming the intersection of truth values satisfies the *associativity* and/or *commutativity*, the computation of truth values for conjunction and disjunction are:

$$\mathbf{T}^q(v) = \prod_{1 \le i \le k} \mathbf{T}^{q'}(v_i), \tag{15}$$

$$\mathbf{T}^q(v) = \mathbf{1} - \prod_{1 \le i \le k} \Big(\mathbf{1} - \mathbf{T}^{q'}(v_i)\Big). \tag{16}$$

Solution of the EFOL query problem.

$$T^q(v^*) = \max_{v \in \mathcal{E}} \big\{\mathbf{T}^q(v)\big\}, \tag{17}$$

$$q(v^*) = \arg\max_{v \in \mathcal{E}} \big(\mathbf{T}^q(v^*)\big). \tag{18}$$

5 EXPERIMENTS

5.1 Experimental setups

Datasets. We use three benchmark datasets, pre-processed by [32], for query embeddings: FB15k [5], FB15k-237 [36] and NELL995 [44] to evaluate the atomic and complex logical query answering tasks (see Appendix B.1).

Each dataset has 14 query structures: 9 existential positive first-order logic (EPFOL) (1p, 2p, 3p, 2i, 3i, ip, pi, 2u, up) and 5 negation types (2in, 3in, inp, pin, pni) (see Appendix B.2 for the visualization of these query structures).

Baselines. We use GQE [14] and Query2Box (Q2B) [31] as baseline models. They are geometric models using translation for the relation projection. These models limit their abilities in handling EPFOL queries only (*i.e.* unable to handle negation queries). However, using symbolic approach as proposed by QTO [4] enables their abilities of answering negation queries.

Evaluation metrics. Following the evaluation protocol of Q2B [31], each query in the test/valid set has easy and hard answers. The easy answers are obtained from the training/valid graph where the hard answers are obtained from predicted links in the valid/test graph. Given the hard answers, we evaluate model performance using Mean Reciprocal Rank (MRR).

Implementation details. We train SpherE using atomic queries only (1p) and evaluate performance on path queries (1p, 2p, 3p). Then, we use QTO [4] to enable SpherE to answer complex logical queries so as to evaluate model performance on the 14 query structures. We also train GQE $(d = 800)$ and Q2B $(d = 400)$ under this setting, denoted by GQE_{1p} and Q2B_{1p}, along with the original versions of GQE and Q2B that are trained in 5 EPFOL queries (1p, 2p, 3p, 2i, 3i). In Appendix B.3, Table 8 shows the training queries using SpherE, GQE, Q2B, compared to those using BetaE, ConE and CylE.

We report main results of SpherE $(d = 256)$ with scaling radius $(\lambda_\tau = 0.02)$ in Section 5.2. We report other results of SpherE using different embedding dimensions and hyper-parameters (λ_τ) in Section 5.3. During the generation of atomic query matrices $(\mathbf{A}_r)$, we use a small threshold (ϵ) to filter all scoring values below (ϵ). We found that the threshold is to efficiently store the matrices in a single GPU as described in [4] for the FB15k and FB15k-237 datasets. In the NELL995 dataset, we apply (ϵ) with a technique to efficiently store the matrix in a GPU, save it in the storage drive, and load it

Table 2: The average MRR (%) of answering complex logical queries before and after using fuzzy logic from QTO [4]. GQE and Q2B results are taken from [32], while (GQE) and (Q2B) are trained in atomic queries (1p) only. Bold result is the best for each dataset. Indicator ↑ or ↓ is the increase or the decrease of performance after using QTO for each model.

Dataset	Model	AVG_p	AVG_n	1p	2p	3p	2i	3i	ip	pi	2u	up	2in	3in	inp	pin	pni
FB15k	GQE	28.0	-	54.6	15.3	10.8	39.7	51.4	19.1	27.6	22.1	11.6	-	-	-	-	-
	GQE + QTO	33.5↑	14.6↑	50.3 ↓	21.1↑	15.6↑	45.9↑	57.7↑	31.1↑	40.7↑	23.3↑	15.8↑	18.5↑	25.0↑	10.1↑	11.0↑	8.5↑
	GQE_{1p}	-	-	73.7	16.6	11.3	-	-	-	-	-	-	-	-	-	-	-
	GQE_{1p} + QTO	44.3↑	23.9↑	70.0↓	31.5↑	23.8↑	55.3↑	65.5↑	42.4↑	49.0↑	36.5↑	24.4↑	30.4↑	34.6↑	18.2↑	20.8↑	15.5↑
	Q2B	38.0	-	68.0	21.0	14.2	55.1	66.5	26.1	39.4	35.1	16.7	-	-	-	-	-
	Q2B + QTO	41.9↑	18.3↑	66.8↓	27.9↑	18.0↑	54.6↓	65.3↓	40.5↑	48.6↑	35.1	20.3↑	25.6↑	26.7↑	14.0↑	13.0↑	12.2↑
	$Q2B_{1p}$	-	-	78.5	13.8	7.6	-	-	-	-	-	-	-	-	-	-	
	$Q2B_{1p}$ + QTO	44.4↑	21.8↑	74.7↓	30.8↑	19.1↑	56.7↑	64.3↑	43.4↑	50.7↑	37.7↑	22.0↑	29.4↑	29.1↑	19.2↑	16.8↑	14.3↑
	SpherE	-	-	83.1	15.8	7.4	-	-	-	-	-	-	-	-	-	-	-
	SpherE + QTO	**54.8**↑	**29.9**↑	79.8↓	43.9↑	31.8↑	65.1↑	71.1↑	54.5↑	59.0↑	52.1↑	35.5↑	39.4↑	37.3↑	27.9↑	26.5↑	18.5↑
FB15k -237	GQE	16.3	-	35.0	7.2	5.3	23.3	34.6	10.7	16.5	8.2	5.7	-	-	-	-	-
	GQE + QTO	19.0↑	6.2↑	33.9↓	8.2↑	6.1↑	27.9↑	42.4↑	14.2↑	23.4↑	8.8↑	6.5↑	6.9↑	13.0↑	4.7↑	4.4↑	2.2↑
	GQE_{1p}	-	-	41.6	7.9	5.6	-	-	-	-	-	-	-	-	-	-	-
	GQE_{1p} + QTO	21.4↑	7.8↑	40.7↓	10.2↑	8.0↑	29.6↑	42.5↑	16.4↑	24.8↑	12.1↑	8.7↑	9.1↑	14.3↑	6.5↑	6.1↑	3.1↑
	Q2B	20.1	-	40.6	9.4	6.8	29.5	42.3	12.6	21.2	11.3	7.6	-	-	-	-	-
	Q2B + QTO	20.3↓	7.1↑	40.4↓	9.0↓	5.5↓	28.8↓	41.8↓	15.5↑	24.0↑	11.5↑	6.6↓	8.9↑	13.9↑	5.1↑	4.7↑	2.9↑
	$Q2B_{1p}$	-	-	42.6	6.8	4.7	-	-	-	-	-	-	-	-	-	-	-
	$Q2B_{1p}$ + QTO	21.0↑	7.2↑	41.9↓	8.8↑	6.2↑	29.5↑	43.2↑	16.0↑	24.8↑	11.6↑	6.8↑	8.8↑	13.8↑	5.6↑	5.0↑	3.0↑
	SpherE	-	-	43.3	6.9	3.6	-	-	-	-	-	-	-	-	-	-	-
	SpherE + QTO	**21.7**↑	**8.3**↑	42.0↓	10.6↑	9.1↑	29.9↑	40.8↑	16.5↑	25.0↑	12.1↑	9.2↑	9.4↑	15.2↑	7.0↑	6.4↑	3.3↑
NELL 995	GQE	18.6	-	32.8	11.9	9.6	27.5	35.2	14.4	18.4	8.5	8.8	-	-	-	-	-
	GQE + QTO	18.5↓	5.5↑	31.4↓	11.3↓	9.9↑	27.6↑	34.6↓	15.2↑	20.1↑	8.5	7.8↓	5.6↑	10.1↑	6.1↑	3.6↑	2.1↑
	GQE_{1p}	-	-	47.7	13.3	9.7	-	-	-	-	-	-	-	-	-	-	-
	GQE_{1p} + QTO	22.5↑	7.4↑	44.1↓	14.1↑	11.4↑	30.6↑	37.7↑	18.3↑	23.7↑	12.5↑	9.9↑	8.1↑	11.9↑	8.7↑	4.8↑	3.3↑
	Q2B	22.9	-	42.2	14.0	11.2	33.3	44.5	16.8	22.4	11.3	10.3	-	-	-	-	-
	Q2B + QTO	21.3↓	6.5↑	40.5↓	13.2↓	10.8↓	30.1↓	37.5↓	17.6↑	22.3↓	10.8↓	8.9↓	7.3↑	10.9↑	7.7↑	4.0↑	2.7↑
	$Q2B_{1p}$	-	-	47.7	12.6	8.8	-	-	-	-	-	-	-	-	-	-	-
	$Q2B_{1p}$ + QTO	23.3↑	7.4↑	46.2↓	14.2↑	11.2↑	32.3↑	39.6↑	19.3↑	24.6↑	12.6↑	9.8↑	8.3↑	12.1↑	8.7↑	4.8↑	3.3↑
	SpherE	-	-	60.2	15.0	9.6	-	-	-	-	-	-	-	-	-	-	-
	SpherE + QTO	**24.6**↑	**9.2**↑	53.6↓	15.3↑	13.8↑	30.8↑	37.6↑	19.5↑	24.2↑	15.1↑	11.2↑	10.0↑	13.7↑	12.1↑	6.1↑	4.1↑

in the GPU for each relation instead of all relations at once. Due to small values in atomic query matrix, we also apply negation scaling (ϵ_n) as it in QTO when generalizing atomic queries to negation queries. For further details of SpherE, please see Appendix B.4 (hyper-parameter settings) and Appendix B.5 (error bars).

5.2 Results

Table 2 shows the average (MRR %) of performance on answering negation queries (AVG_n) and answering EPFOL queries (AVG_p).

Answering negation queries (AVG_n). Without fuzzy logic, GQE and Q2B in default setting and in limited setting as that of SpherE (models trained on atomic queries 1p) cannot answer negation queries. However, using fuzzy logic unlocks the ability of handling negation queries for all geometric models in our experiments as shown in the last 5 columns in Table 2. Specifically, SpherE achieves the best performance (AVG_n) using the three datasets. Empirical results of models trained on atomic queries are to emphasize on the importance of atomic relation projection on generalizing to CLQA.

Answering EPFOL queries (AVG_p). Since the default versions of GQE and Q2B can answer the EPFOL queries, using the fuzzy symbolic approach (QTO) is not necessary in this scenario. However, Table 2 shows significant improvements in the performance of answering EPFOL queries (AVG_p), before and after implementing the symbolic approach, using the FB15k and FB15k-237 dataset (for GQE and Q2B). There are slight drops in model performance (AVG_p) using the NELL995 dataset (for GQE and Q2B), but the gap is minimal. With regard to the limited versions in GQE_{1p} and $Q2B_{1p}$ (trained on 1p queries), which are similar to SpherE, using fuzzy logic is necessary to enable the abilities of answering EPFOL in these models. Specially, SpherE (modelling the relation projection operator only) achieves the best performance of answering EPFOL queries (AVG_p) using the three datasets, compared to GQE and Q2B (modelling both the relation projection and the conjunction).

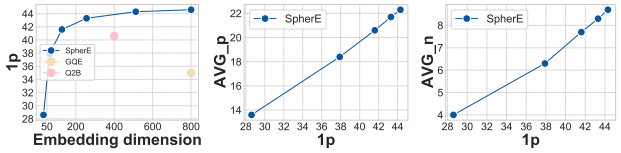
Recap. Using fuzzy logic as QTO shows a promising direction of linking atomic queries to complex logical query answering, though there is consistently a slight drop of performance in answering 1p queries of a model with and without QTO (see Table 2, column 5). This problem is due to an effect of a filtering threshold ϵ (as mentioned in Eq. (12) and Section 4.2) that neglects tiny truth values below it during the generation of atomic query matrix for the 1p.

5.3 Quantitative analysis

Table 3 shows average MRR (%) of SpherE in atomic query answering (AQA) and CLQA tasks using different embedding dimensions (d) but the same scaling radius (λ_τ).

Table 3: Performance (MRR %) of SpherE with embedding dimensions. ($*$) are for main results in Table 2.

Model/ Dataset	Dim	1p (origin)	AVG_p (QTO)	AVG_n (QTO)	#Params
GQE	$d = 800^*$	54.6	33.5	14.6	15.4M
Q2B	$d = 400^*$	68.0	41.9	18.3	8.8M
SpherE ($\lambda_\tau = .02$) FB15k	$d = 32$	46.9	21.1	6.3	1.2M
	$d = 64$	69.7	36.4	15.1	2.4M
	$d = 128$	79.0	47.8	24.5	4.9M
	$d = 256^*$	83.1	54.8	29.9	9.7M
	$d = 512$	**84.3**	**57.6**	**32.4**	**19.5M**
GQE	$d = 800^*$	35.0	19.0	6.2	13.3M
Q2B	$d = 400^*$	40.6	20.3	7.1	6.8M
SpherE ($\lambda_\tau = .02$) FB15k-237	$d = 32$	28.6	13.6	4.0	1.0M
	$d = 64$	37.9	18.4	6.3	2.0M
	$d = 128$	41.6	20.6	7.7	3.9M
	$d = 256^*$	43.3	21.7	8.3	7.8M
	$d = 512$	**44.3**	**22.3**	**8.7**	**15.6M**



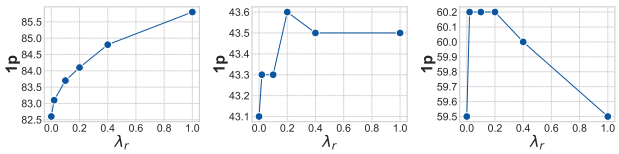
(a) (b) (c)

Figure 5: Correlation of performance (MRR %) of embedding dimension and AQA, and of AQA and CLQA using the FB15k-237 dataset.

Effect of embedding dimension on answering atomic queries. In general, there is a correlation between the size of embedding dimension and model performance (MRR %) on AQA and CLQA. Without QTO, when increasing (d) from 32 to 512, model performance on AQA (1p) significantly rises by nearly 80% (using the FB15k dataset) and by around 55% (using the FB15k-237 dataset). However, the average MRR (1p) reaches a critical point at a high embedding dimension. Figure 5a demonstrates the increasing trend, which is convergent after around ($d = 400$).

Effect of atomic queries on generalizing complex logical queries. Table 3 also shows a correlation between model performance on AQA and CLQA, particularly between the average MRR (%) in atomic queries (1p) and that in EPFOL queries (AVG_p). Figure 5b illustrates the correlation using the FB15k-237 dataset. Similarly, Figure 5c illustrates a correlation between the average MRR of 1p queries and that of negation queries (AVG_n). These correlations suggest a key role of atomic query in affecting the ability of CLQA using fuzzy logic in SpherE. Moreover, using Sphere ($d = 128$) has lower number of parameters than baselines in GQE and Q2B, but can achieve better performance of AQA and CLQA than baselines.

Effect of radius on AQA. Figure 6 shows the average MRR (%) results of SpherE in the AQA task using different scales ($\lambda_\tau \in \{0.0, 0.02, 0.1, 0.2, 0.4, 1.0\}$ contributing to the adjusted distance function in Eq. (9)) but the same embeddings dimension ($d = 256$). When this hyper-parameter is zero, the total distance of an embedded query and an embedded target answer is equal to the distance between their central points. When this hyper-parameter is one, the total distance is equal to the distance between their borders (see Figure 4 to recall a visualization this distance). In each dataset, by selecting a critical scaling hyper-parameter, we can have the best version of SpherE sufficiently performing AQA task (1p).



(a) FB15k (b) FB15k-237 (c) NELL995

Figure 6: Performance (MRR %) of SpherE using different radius scales (λ_τ) on answering atomic queries.

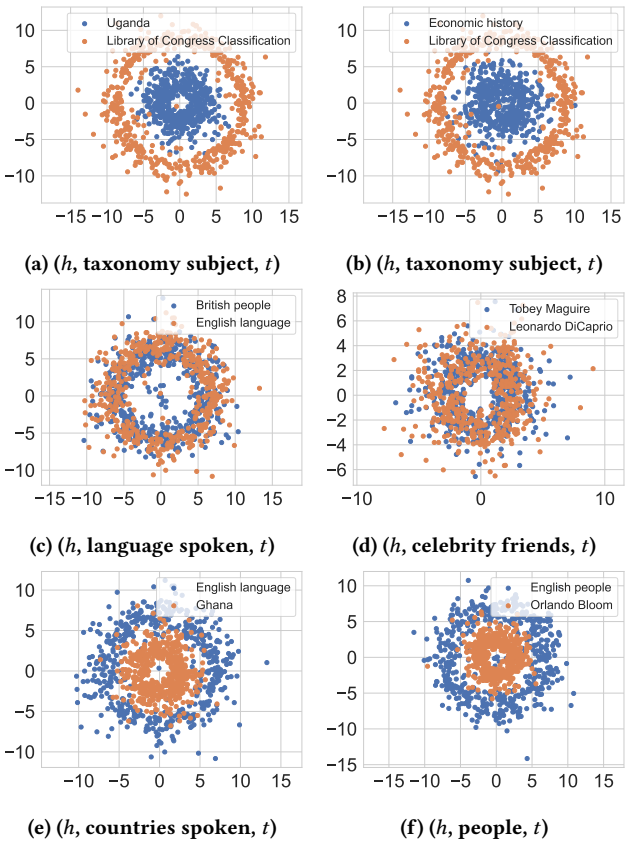
Illustration of semantic hierarchy. Figure 7 visualizes example pairs of embedded entities, each pair is (h, t) given a relation (r) based on visualization technique of HAKE [48]. The top (Figure 7a, 7b), the middle (Figure 7c, 7d), the bottom (Figure 7e, 7f) each shows respectively when the head concept is at the lower level of entity semantic hierarchies than the tail; at similar levels and when the head is at a higher level than the tail. In the top and bottom, there are clear separations between the embeddings where higher level concepts are consistently represented in the outer sides, compared to the similar level concepts in the middle. These observations show SpherE's ability of modeling semantic hierarchies as those in HAKE.

5.4 Comparison of SpherE for AQA and those models for CLQA

Table 4 shows comparisons of model performance using other models inspired by shapes (ConE [49], CylE [27], BetaE [32]) in the FB15k-237 dataset with and without symbolic approach. We reproduce results of these models, trained using complex queries, then augment with QTO to analyse whether there are improvement. Notice that these models have already achieved CLQA without QTO [4] as shown in Table 4.

Table 4: Comparison of the average MRR (%) of SpherE and CLQA models that can handle EFOL operators ($\exists, \wedge, \vee, \neg$) before and after using fuzzy logic system based on QTO [4].

Dataset	Model	Params	AVG_p	AVG_n	1p	2p	3p	2i	3i	ip	pi	2u	up	2in	3in	inp	pin	pni
FB15k-237	SpherE ($d = 512$)	15.6M	-	-	44.3	7.5	4.0	-	-	-	-	-	-	-	-	-	-	-
	SpherE + QTO		22.3↑	8.7↑	42.4↓	10.9↑	9.2↑	30.8↑	40.9↑	17.4↑	26.3↑	13.4↑	9.6↑	10.2↑	16.0↑	7.2↑	6.7↑	3.4↑
	BetaE ($d = 400$)	18.5M	20.9	5.4	39.0	10.9	10.0	28.8	42.5	12.6	22.4	12.4	9.7	5.1	7.9	7.4	3.6	3.4
	BetaE + QTO		20.2↓	8.2 ↑	38.2↓	10.1↓	8.8↓	27.6↓	36.8↓	14.9↑	24.1↑	12.5↑	8.8↓	9.0↑	15.2↑	7.3↓	6.3↑	3.1↓
	ConE ($d = 800$)	23.9M	23.4	5.9	41.8	12.8	11.0	32.6	47.3	14.0	25.5	14.5	10.8	5.4	8.6	7.8	4.0	3.6
	ConE + QTO		22.2↓	8.9↑	41.2↓	12.5↓	10.6↓	29.7↓	38.7↓	17.3↑	25.7↑	13.9↓	10.4↓	10.1↑	15.4↑	8.6↑	7.1↑	3.4↓
	CylE ($d = 800$)	39.7M	24.5	5.7	42.9	13.3	11.3	35.0	49.0	15.7	27.0	15.3	11.2	4.9	8.3	8.2	3.7	3.4
	CylE + QTO		23.8↓	9.8↑	40.9↓	13.0↓	11.0↓	33.2↓	42.4↓	19.1↑	28.2↑	15.7↑	11.1↓	10.8↑	17.9↑	9.0↑	7.6↑	3.9↑



(a) (h, taxonomy subject, t) (b) (h, taxonomy subject, t)

(c) (h, language spoken, t) (d) (h, celebrity friends, t)

(e) (h, countries spoken, t) (f) (h, people, t)

Figure 7: Visualization of entity embeddings in pairs using SpherE in the FB15k-237 dataset based on HAKE [48]. (Blue dots): Head. (Orange dots): Tail.

For negation queries, in comparison between those models with and without the symbolic approach, performance improvements are observed in all models. This trend is similar to that in SpherE (Q2B and GQE). For non-negation queries, there is a slight drop in performance using BetaE, ConE and CylE. Although model performance of CLQA using SpherE is not competitive to that in ConE and CylE, SpherE significantly outperforms BetaE and SpherE uses less parameters. The results in non-negation queries suggest that while using fuzzy logic can enable limited geometric models, such as spherical embeddings, to answer complex queries and improve model performance, the use of fuzzy logic may not be beneficial for other models that have already achieved the task of CLQA.

Table 5: The average MRR (%) of SpherE and CLQA models (BetaE [32], ConE [49], CylE [27]) in AQA.

Model	Params	1p
BetaE ($d = 400$)	18.5M	39.0
ConE ($d = 800$)	23.9M	41.8
CylE ($d = 800$)	39.7M	42.9
SpherE ($d = 512$)	15.6M	**44.3**

Table 5 shows results (extracted from Table 4) in AQA without the symbolic approach. SpherE achieves best results of AQA over other CLQA models (BetaE, ConE, CylE), but SpherE uses less parameters than that in these models. Answering atomic queries is essential as the 1p query dominates in all query structures, SpherE is at the least example to show the link from AQA to CLQA.

6 CONCLUSIONS

We propose a symbolic approach based on fuzzy logic to enable embeddings models (motivated by shapes like points, boxes and spheres) for CLQA. We introduce spherical embeddings SpherE (in polar coordinate system), and by incorporating fuzzy realisations of logical operators, we address the extra challenges brought in by the spherical shapes as compared to the more capable point and box shapes, and significantly outperforms them in the CLQA task. Results show that these models can all answer EPFOL and negation queries when augmenting with the symbolic approach. This is encouraging as these models on their own cannot handle negation queries, particularly conjunctive and disjunctive queries in the case of SpherE. Our findings suggest that fuzzy logic can help negate the necessity of closed negation and closed intersection of shapes for geometric-based models; thus highlight the key role of atomic queries in CLQA.

REFERENCES

[1] Erik Arakelyan, Daniel Daza, Pasquale Minervini, and Michael Cochez. 2021. Complex Query Answering with Neural Link Predictors. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=Mos9F9kDwkz

[2] Erik Arakelyan, Pasquale Minervini, Daniel Daza, Michael Cochez, and Isabelle Augenstein. 2023. Adapting Neural Link Predictors for Data-Efficient Complex Query Answering. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*. https://openreview.net/forum?id=1G7CBp8o7L

[3] 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*. Association for Computational Linguistics, Seattle, United States, 2703–2714. https://doi.org/10.18653/v1/2022.findings-naacl.207

[4] Yushi Bai, Xin Lv, Juanzi Li, and Lei Hou. 2023. Answering Complex Logical Queries on Knowledge Graphs via Query Computation Tree Optimization. In *Proceedings of the 40th International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 1472–1491. https://proceedings.mlr.press/v202/bai23b.html

[5] 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* (Vancouver, Canada) *(SIGMOD '08)*. Association for Computing Machinery, New York, NY, USA, 1247–1250. https://doi.org/10.1145/1376616.1376746

[6] 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* (Lake Tahoe, Nevada) *(NIPS'13)*. Curran Associates Inc., Red Hook, NY, USA, 2787–2795. https://dl.acm.org/doi/10.5555/2999792.2999923

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

[8] 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 (Jun. 2022), 3939–3948. https://doi.org/10.1609/aaai.v36i4.20310

[9] Nurendra Choudhary, Nikhil Rao, Sumeet Katariya, Karthik Subbian, and Chandan Reddy. 2021. Probabilistic Entity Representation Model for Reasoning over Knowledge Graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 23440–23451. https://proceedings.neurips.cc/paper_files/paper/2021/hash/c4d2ce3f3ebb5393a77c33c0cd95dc93-Abstract.html

[10] Nurendra Choudhary, Nikhil Rao, Sumeet Katariya, Karthik Subbian, and Chandan K Reddy. 2021. Self-supervised hyperboloid representations from logical queries over knowledge graphs. In *Proceedings of the Web Conference* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 1373–1384. https://doi.org/10.1145/3442381.3449974

[11] Brian A Davey and Hilary A Priestley. 2002. *Introduction to lattices and order*. Cambridge university press.

[12] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence* (New Orleans, Louisiana, USA) *(AAAI'18/IAAI'18/EAAI'18)*. AAAI Press, Article 221, 8 pages. https://doi.org/10.5555/3504035.3504256

[13] Cosimo Gregucci, Mojtaba Nayyeri, Daniel Hernández, and Steffen Staab. 2023. Link Prediction with Attention Applied on Multiple Knowledge Graph Embedding Models. In *Proceedings of the ACM Web Conference 2023* (Austin, TX, USA) *(WWW '23)*. Association for Computing Machinery, New York, NY, USA, 2600–2610. https://doi.org/10.1145/3543507.3583358

[14] William L. Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. 2018. Embedding Logical Queries on Knowledge Graphs. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 2030–2041. https://dl.acm.org/doi/10.5555/3326943.3327131

[15] Hengchang Hu, Wei Guo, Xu Liu, Yong Liu, Ruiming Tang, Rui Zhang, and Min-Yen Kan. 2024. User Behavior Enriched Temporal Knowledge Graphs for Sequential Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24)*. Association for Computing Machinery, New York, NY, USA, 266–275. https://doi.org/10.1145/3616855.3635762

[16] Hengchang Hu, Wei Guo, Yong Liu, and Min-Yen Kan. 2023. Adaptive Multi-Modalities Fusion in Sequential Recommendation Systems. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 843–853. https://doi.org/10.1145/3583780.3614775

[17] 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* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 615–625. https://doi.org/10.1145/3534678.3539338

[18] Roshni G. Iyer, Yunsheng Bai, Wei Wang, and Yizhou Sun. 2022. Dual-Geometric Space Embedding Model for Two-View Knowledge Graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 676–686. https://doi.org/10.1145/3534678.3539350

[19] Song Jiang, Qiyue Yao, Qifan Wang, and Yizhou Sun. 2023. A Single Vector Is Not Enough: Taxonomy Expansion via Box Embeddings. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*. Association for Computing Machinery, New York, NY, USA, 2467–2476. https://doi.org/10.1145/3543507.3583310

[20] 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*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[21] Erich Peter Klement, Radko Mesiar, and Endre Pap. 2000. *Triangular Norms*. Springer Dordrecht. https://doi.org/10.1007/978-94-015-9540-7

[22] George Klir and Bo Yuan. 1995. *Fuzzy Sets and Fuzzy Logic*. Vol. 4. Prentice Hall, New Jersey.

[23] S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22, 1 (1951), 79–86. http://www.jstor.org/stable/2236703

[24] Lihui Liu, Boxin Du, Heng Ji, ChengXiang Zhai, and Hanghang Tong. 2021. Neural-Answering Logical Queries on Knowledge Graphs. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 1087–1097. https://doi.org/10.1145/3447548.3467375

[25] Xiao Long, Liansheng Zhuang, Li Aodi, Shafei Wang, and Houqiang Li. 2022. Neural-based Mixture Probabilistic Query Embedding for Answering FOL queries on Knowledge Graphs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 3001–3013. https://aclanthology.org/2022.emnlp-main.194

[26] 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 (NIPS'19)*. Curran Associates Inc., Red Hook, NY, USA, Article 737, 10 pages. https://dl.acm.org/doi/10.5555/3454287.3455024

[27] Chau D. M. Nguyen, Tim French, Wei Liu, and Michael Stewart. 2023. CylE: Cylinder Embeddings for Multi-hop Reasoning over Knowledge Graphs. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Association for Computational Linguistics, Dubrovnik, Croatia, 1728–1743. https://doi.org/10.18653/v1/2023.eacl-main.127

[28] Chau D. M. Nguyen, Tim French, Wei Liu, and Michael Stewart. 2023. SConE: Simplified Cone Embeddings with Symbolic Operators for Complex Logical Queries. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, Toronto, Canada, 11931–11946. https://doi.org/10.18653/v1/2023.findings-acl.755

[29] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML)* (Bellevue, Washington, USA) *(ICML'11)*. Omnipress, Madison, WI, USA, 809–816. https://dl.acm.org/doi/10.5555/3104482.3104584

[30] Hongyu Ren, Mikhail Galkin, Michael Cochez, Zhaocheng Zhu, and Jure Leskovec. 2023. Neural Graph Reasoning: Complex Logical Query Answering Meets Graph Databases. *arXiv preprint arXiv:2303.14617* (2023). https://arxiv.org/abs/2303.14617

[31] 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)*. https://openreview.net/forum?id=uM4k4_nnEF

[32] 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* (Vancouver, BC, Canada) *(NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, Article 1654, 11 pages. https://dl.acm.org/doi/10.5555/3495724.3497378

[33] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web*, Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam (Eds.). Springer International Publishing, Cham, 593–607. https://doi.org/10.1007/978-3-319-93417-4_38

[34] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=HkgEQnRqYQ

[35] Zhenwei Tang, Griffin Floto, Armin Toroghi, Shichao Pei, Xiangliang Zhang, and Scott Sanner. 2023. LogicRec: Recommendation with Users' Logical Requirements. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 2129–2133. https://doi.org/10.1145/3539618.3592012

[36] 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*. Association for Computational Linguistics, Beijing, China, 57–66. https://doi.org/10.18653/v1/W15-4007

[37] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *Proceedings of The 33rd International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. 48)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.). PMLR, New York, New York, USA, 2071–2080. https://proceedings.mlr.press/v48/trouillon16.html

[38] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=BylA_C4tPr

[39] Siyuan Wang, Zhongyu Wei, Jiarong Xu, Taishan Li, and Zhihao Fan. 2023. Unifying Structure Reasoning and Language Pre-training for Complex Reasoning Tasks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2023), 1–11. https://doi.org/10.1109/TASLP.2023.3325973

[40] Zihao Wang, Yangqiu Song, Ginny Y Wong, and Simon See. 2023. Logical Message Passing Networks with One-hop Inference on Atomic Formulas. In *The Eleventh International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=SoyOsp7i_l

[41] Zihao Wang, Hang Yin, and Yangqiu Song. 2022. Logical Queries on Knowledge Graphs: Emerging Interface of Incomplete Relational Data. *Data Engineering* (2022), 3. http://sites.computer.org/debull/A22dec/A22DEC-CD.pdf#page=5

[42] Yuhan Wu, Yuanyuan Xu, Xuemin Lin, and Wenjie Zhang. 2023. A Holistic Approach for Answering Logical Queries on Knowledge Graphs. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. 2345–2357. https://doi.org/10.1109/ICDE55515.2023.00181

[43] Yuhan Wu, Yuanyuan Xu, Wenjie Zhang, Xiwei Xu, and Ying Zhang. 2024. Query2GMM: Learning Representation with Gaussian Mixture Model for Reasoning over Knowledge Graphs. In *Proceedings of the ACM on Web Conference 2024* (Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 2149–2158. https://doi.org/10.1145/3589334.3645569

[44] 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 (EMNLP)*. Association for Computational Linguistics, Copenhagen, Denmark, 564–573. https://doi.org/10.18653/v1/D17-1060

[45] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575* (2014).

[46] Dong Yang, Peijun Qing, Yang Li, Haonan Lu, and Xiaodong Lin. 2022. GammaE: Gamma Embeddings for Logical Queries on Knowledge Graphs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 745–760. https://aclanthology.org/2022.emnlp-main.47

[47] Hang Yin, Zihao Wang, and Yangqiu Song. 2024. Rethinking Complex Queries on Knowledge Graphs with Neural Link Predictors. In *The Twelfth International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=1BmveEMNbG

[48] Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020. Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 03 (Apr. 2020), 3065–3072. https://doi.org/10.1609/aaai.v34i03.5701

[49] Zhanqiu Zhang, Jie Wang, Jiajun Chen, Shuiwang Ji, and Feng Wu. 2021. ConE: Cone Embeddings for Multi-Hop Reasoning over Knowledge Graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. https://openreview.net/forum?id=Twf_XYunk5j

[50] Zhengyun Zhou, Guojia Wan, Shirui Pan, Jia Wu, Wenbin Hu, and Bo Du. 2024. Complex query answering over knowledge graphs foundation model using region embeddings on a lie group. *World Wide Web* 27, 3 (2024), 23. https://doi.org/10.1007/s11280-024-01254-7

[51] 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)*. https://proceedings.mlr.press/v162/zhu22c.html

[52] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. 2021. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 29476–29490. https://openreview.net/forum?id=DEsIX_D_vR

A METHODOLOGY

A.1 Atomic query matrix generation - additional details

Notice that the scoring function $\phi()$ using SpherE in Eq. (10) can be replaced with another one using CLQA models such as GQE [14], Q2B [31], ConE [49], BetaE [32], etc., if the atomic relation projection is learned using these models. Table 6 shows comparisons of scoring functions using distance-based of AQA models (SpherE) and CLQA models. Hyper-parameter (γ) is set to zero for simplicity.

Table 6: Scoring functions of AQA and CLQA models. KL denotes Kullback-Leibler divergence [23].

Models	Scoring function $\phi()$	Representation
SpherE	$-d(\mathbf{q}, \mathbf{v})$	$\mathbf{q}, \mathbf{v} \in \mathbb{R}^{2d+1}$
GQE [14]	$-d(\mathbf{q}, \mathbf{v})$	$\mathbf{q}, \mathbf{v} \in \mathbb{R}^{d}$
Q2B [31]	$-d_{\text{out}}(\mathbf{q}, \mathbf{v})+$ $-\lambda d_{\text{in}}(\mathbf{q}, \mathbf{v})$	$\mathbf{q} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, $\mathbf{v} = \boldsymbol{\alpha}$, $\boldsymbol{\alpha} \in \mathbb{R}^{d}, \boldsymbol{\beta} \in \mathbb{R}^{d}_{+}$
ConE [49]	$-d_{\text{out}}(\mathbf{q}, \mathbf{v})+$ $-\lambda d_{\text{in}}(\mathbf{q}, \mathbf{v})$	$\mathbf{q} = (\boldsymbol{\theta}_{ax}, \boldsymbol{\theta}_{ap})$, $\mathbf{v} = (\boldsymbol{\theta}_{ax}, \mathbf{0})$, $\boldsymbol{\theta}_{ax} \in [-\pi, \pi)^{d}$, $\boldsymbol{\theta}_{ap} \in [0, 2\pi]^{d}$
BetaE [32]	$-\text{KL}(\mathbf{v}, \mathbf{q})$	$\mathbf{q}, \mathbf{v} \in \mathbb{R}^{2d}$

B EXPERIMENTAL SETUPS

B.1 Datasets

Table 7 shows statistical description of the three datasets (FB15k [5], FB15k-237 [36] and NELL995 [44]) which are pre-processed by [32] [2]. The number of entities in the NELL995 dataset is more than four times that in the FB15k-237 dataset while both have similar number of edges. The FB15k dataset has the highest number of edges, which is more than five times that in the FB15k-237 dataset. The NELL995 has the highest complexity of atomic query matrix $|\mathcal{E}| \times |\mathcal{E}|$ and tensor $|\mathcal{E}| \times |\mathcal{E}| \times 2|\mathcal{R}|$.

B.2 Query structures available in the datasets

Figure 8 demonstrates corresponding 14 EFOL query structures in computation graphs; where source entities (constants) are in grey nodes, intermediate entities (variables) are in blue nodes, target entities (free variables) are in green nodes, relation projections ($\exists$) are black arrows, relation projections with negation ($\neg$) are dash red arrows, conjunctive connections ($\wedge$) are red arrows and disjunctive connections ($\vee$) are orange arrows.

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

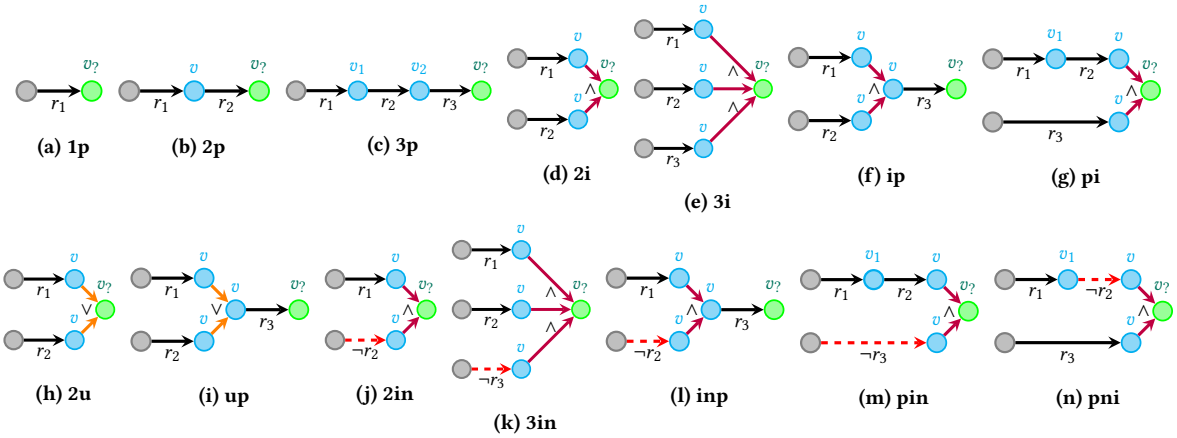


Figure 8: EFOL queries in from the BetaE [32] datasets, 'p' is projection, 'i' is intersection, 'u' is union and 'n' is negation.

Table 7: Datasets in [32]. Table is adapted from [28].

Statistics	FB15k	FB15k-237	NELL995
Train Edges	483,142	272,115	114,213
Valid Edges	50,000	17,526	14,324
Testing Edges	59,071	20,438	14,267
Total Edges	592,213	310,079	142,804
Num. Entities $\|\mathcal{E}\|$	14,951	14,505	63,361
Num. Relations $\|\mathcal{R}\|$	1,345	237	200
$\|\mathcal{E}\| \times \|\mathcal{E}\|$	$\approx 224 \times 10^6$	$\approx 210 \times 10^6$	$\approx 4{,}015 \times 10^6$
$\|\mathcal{E}\| \times \|\mathcal{E}\| \times 2\|\mathcal{R}\|$	$\approx 601 \times 10^9$	$\approx 100 \times 10^9$	$\approx 1{,}606 \times 10^9$

B.3 Comparison of query structures involving in the training process using different models

Table 8: Training query structures in different models.

Model	Training Query Structures
SpherE, GQE$_{1p}$ [14], Q2B$_{1p}$ [31]	1p
GQE [14], Q2B [31]	1p/2p/3p/2i/3i
BetaE [32], ConE [49], CylE [27]	1p/2p/3p/2i/3i/2in/3in/inp/pin/pni

B.4 Hyper-parameter settings

In terms of model implementation without QTO, we follow the implementation of BetaE [32] [3] to train and evaluate model performance of SpherE, GQE and Query2Box using either a single GPU of NVIDIA Tesla V100 or P100 under the PyTorch framework with Adam optimizer [20]. Table 9 shows a comparison of hyper-parameters settings for reproducing main results of different models (GQE, Q2B, SpherE). Hyper-parameters of GQE and Q2B are taken from [32]. In SpherE, we search for the learning rate in {0.00005, 0.0001, 0.0002, 0.0003, 0.0004}. For ablation studies, we conduct experiments using the scaling radius (λ_τ) in {0.0, 0.02, 0.1, 0.2, 0.4, 1.0} for the inside distance in Eq. (9) and using the embedding dimension in {32, 64, 128, 256, 512}.

Table 9: A summary of hyper-parameter of main results using GQE [14], Q2B [31] and SpherE. Hyper-parameter of GQE and Q2B are taken from [32].

Dataset	Model	n Batch size	b Negative sampling	d Embedding dimension	m Max steps	γ Positive margin	l Learning rate	Inside distance
FB15k	GQE	512	128	800	450k	24	0.0001	-
	Q2B			400	450k	24	0.0001	0.02
	SpherE			256	750k	36	0.0001	0.02
FB15k-237	GQE	512	128	800	450k	24	0.0001	-
	Q2B			400	450k	24	0.0001	0.02
	SpherE			256	450k	24	0.0001	0.02
NELL995	GQE	512	128	800	450k	24	0.0001	-
	Q2B			400	450k	24	0.0001	0.02
	SpherE			256	450k	24	0.0003	0.02

With regard to model implementation using fuzzy logic, we follow the implementation of QTO [4] [4] to enable pre-trained models to answer complex logical queries using a single NVIDIA Tesla V100 GPU. Table 11 shows hyper-parameters settings for reproducing main results of models (GQE, Q2B and SpherE). In SpherE, we search for the negation scaling (ϵ_n) in {1.0, 3.0, 6.0} and set the filtering threshold (ϵ) range from $[1 \times 10^{-4}, 1 \times 10^{-3}]$.

B.5 Error bars for the main results of SpherE

We conduct experiments of CLQA using SpherE five times with different random seed numbers in {0, 10, 100, 1000, 10000} for each dataset to estimate the error bars for the main results in Table 2. We estimate the error bars in two settings: (1) SpherE without QTO and (2) SpherE with QTO. Table 10 shows the average of MRR in five experiments with error bars of the two settings for each dataset. Overall, the standard deviance for each evaluation metric (MRR %)

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

[4] https://github.com/bys0318/QTO.

Table 10: Estimating error bars of MRR (in percentage) for the main results of SpherE in Table 2.

Dataset	Model	AVG_p	AVG_n	1p	2p	3p	2i	3i	ip	pi	2u	up	2in	3in	inp	pin	pni
FB15k	SpherE	-	-	83.0	15.9	7.5	-	-	-	-	-	-	-	-	-	-	-
	(std)	-	-	(0.07)	(0.08)	(0.08)	-	-	-	-	-	-	-	-	-	-	-
	SpherE + QTO	54.6	29.6	79.5	43.8	31.6	64.7	70.9	54.5	58.8	52.4	35.2	39.2	36.8	27.9	25.7	18.4
	(std)	(0.11)	(0.22)	(0.31)	(0.11)	(0.31)	(0.23)	(0.11)	(0.05)	(0.13)	(0.25)	(0.31)	(0.21)	(0.33)	(0.23)	(0.46)	(0.11)
FB15k-237	SpherE	-	-	43.2	7.0	3.7	-	-	-	-	-	-	-	-	-	-	-
	(std)	-	-	(0.05)	(0.13)	(0.13)	-	-	-	-	-	-	-	-	-	-	-
	SpherE + QTO	21.8	8.3	41.9	10.6	9.2	30.0	40.7	16.7	25.3	12.5	9.3	9.5	15.3	7.1	6.5	3.3
	(std)	(0.07)	(0.06)	(0.10)	(0.16)	(0.10)	(0.11)	(0.13)	(0.21)	(0.19)	(0.25)	(0.13)	(0.17)	(0.09)	(0.08)	(0.05)	(0.04)
NELL995	SpherE	-	-	60.2	15.0	9.7	-	-	-	-	-	-	-	-	-	-	-
	(std)	-	-	(0.04)	(0.08)	(0.11)	-	-	-	-	-	-	-	-	-	-	-
	SpherE + QTO	24.5	9.2	53.6	15.3	13.7	30.5	37.6	19.5	24.1	15.0	11.1	9.9	13.8	12.2	6.1	4.0
	(std)	(0.09)	(0.04)	(0.04)	(0.07)	(0.13)	(0.18)	(0.11)	(0.15)	(0.15)	(0.20)	(0.14)	(0.09)	(0.15)	(0.08)	(0.05)	(0.07)

Table 11: A summary of hyper-parameter using QTO [4] for reproducing main results of GQE [14], Q2B [31] and SpherE. (.) are for models trained on 1p queries only.

Dataset	Model	ϵ_n Negation scaling	ϵ Filtering threshold	d Embedding dimension	γ Positive margin	λ_τ Inside distance
FB15k	GQE	3.0	5×10^{-4}	800	24	-
	GQE_{1p}		5×10^{-4}	800	24	-
	Q2B		1×10^{-6}	400	24	0.02
	$Q2B_{1p}$		1×10^{-5}	400	24	0.02
	SpherE		1×10^{-3}	256	36	0.02
FB15k-237	GQE	1.0	1×10^{-4}	800	24	-
	GQE_{1p}		1×10^{-4}	800	24	-
	Q2B		1×10^{-9}	400	24	0.02
	$Q2B_{1p}$		1×10^{-6}	400	24	0.02
	SpherE		3×10^{-4}	256	24	0.02
NELL995	GQE	3.0	1×10^{-6}	800	24	-
	GQE_{1p}		1×10^{-5}	800	24	-
	Q2B		1×10^{-9}	400	24	0.02
	$Q2B_{1p}$		1×10^{-6}	400	24	0.02
	SpherE		1×10^{-4}	256	24	0.02

is within a small range which suggests a stable result for answering each query structure.