

# Automatically Discovering Conceptual Neighborhoods Using Machine Learning Methods

Ling Cai<sup>1</sup> ✉ 

Center for Spatial Studies, University of California, Santa Barbara, CA, USA  
STKO Lab, Department Geography, University of California, Santa Barbara, CA, USA

Krzysztof Janowicz ✉

Universität Wien, Austria  
Center for Spatial Studies, University of California, Santa Barbara, CA, USA

Rui Zhu ✉ 

Center for Spatial Studies, University of California, Santa Barbara, CA, USA  
School of Geographical Sciences, University of Bristol, Bristol, UK

---

## Abstract

Qualitative spatio-temporal reasoning (QSTR) plays a key role in spatial cognition and artificial intelligence (AI) research. In the past, research and applications of QSTR have often taken place in the context of declarative forms of knowledge representation. For instance, conceptual neighborhoods (CN) and composition tables (CT) of relations are introduced explicitly and utilized for spatial/temporal reasoning. Orthogonal to this line of study, we focus on bottom-up machine learning (ML) approaches to investigate QSTR. More specifically, we are interested in questions of whether similarities between qualitative relations can be learned from data purely based on ML models, and, if so, how these models differ from the ones studied by traditional approaches. To achieve this, we propose a graph-based approach to examine the similarity of relations by analyzing trained ML models. Using various experiments on synthetic data, we demonstrate that the relationships discovered by ML models are well-aligned with CN structures introduced in the (theoretical) literature, for both spatial and temporal reasoning. Noticeably, even with significantly limited qualitative information for training, ML models are still able to automatically construct neighborhood structures. Moreover, patterns of asymmetric similarities between relations are disclosed using such a data-driven approach. To the best of our knowledge, our work is the first to automatically discover CNs without any domain knowledge. Our results can be applied to discovering CNs of any set of jointly exhaustive and pairwise disjoint (JEPD) relations.

**2012 ACM Subject Classification** Computing methodologies → Machine learning; Computing methodologies → Knowledge representation and reasoning; Computing methodologies → Temporal reasoning; Computing methodologies → Spatial and physical reasoning

**Keywords and phrases** Qualitative Spatial Reasoning, Qualitative Temporal Reasoning, Conceptual Neighborhood, Machine Learning, Knowledge Discovery

**Digital Object Identifier** 10.4230/LIPICs.COSIT.2022.3

**Funding** *Krzysztof Janowicz*: Funded by the National Science Foundation – OIA (2033521).

**Acknowledgements** We want to thank Yutao Zhou for discussion on experiment design/ visualization.

## 1 Introduction

Since the 90s, Qualitative Spatio-Temporal Reasoning (QSTR) has attracted attentions from researchers and practitioners in several fields, such as geographical information science, artificial intelligence and cognitive science [6, 10, 17, 13, 26, 14]. Aside from the clear connection to human representations and linguistic communication of the spatial configuration of our environment, QSTR has numerous advantages over its quantitative counterpart [16].

---

<sup>1</sup> Corresponding author



Representing qualitative information by using symbols and developing calculi to infer unknown qualitative information is the key to QSTR. Different sets of qualitative spatial relations (such as directional and topological relations) along with a system of qualitative calculi are developed [9], among which reasoning over topological relations becomes the most well-established area in QSTR.

As far as regions are concerned, the most well-known formalizations for qualitative topological relations are - the Region Connection Calculus (RCC-8) [22] and the 9-Intersection Model (9-IM) [4, 10]. Both arrive at the same conclusion that there exist eight base topological relations between regions in 2D space, although they are developed independently during the earlier 90s [2]. Those relations form the foundation for a variety of qualitative spatial reasoning techniques [8, 10, 11, 12]. Two major (and interconnected) lines of works are: (1) Composition Tables (CTs) (i.e., transitivity tables), which store possible resulting relations arising from the composition of two relations [1, 22, 24, 23]. (2) Conceptual Neighborhood Graphs (CNGs), which formalize transitions between relations. Conceptual neighbors of a relation are defined as a set of relations that can be directly transformed into/from the relation by deforming (e.g., moving and scaling) the related entities continuously (in a topological sense) [15]. In a CNG, relations are modeled as nodes and an undirected edge is established between two neighboring relations (see Figure 3f). CNGs play an essential role for reasoning with uncertain or incomplete information [14], and have been used in research of cognitive similarity assessment [19, 18] and modeling of linguistic spatial terms [7]. In addition to topological relations, composition tables, and conceptual neighborhoods have also been developed for reasoning over temporal relations [1, 15].

Those reasoning methods follow a top-down manner, which usually requires (noise-free) explicit domain knowledge. On the contrary, success in data-driven Machine Learning (ML) approaches, which are insensitive to noise and good at dealing with incomplete information as well as uncertainty, provides new opportunities to study QSTR from a bottom-up perspective. ML models rely solely on training data to discover patterns/rules that can be implicitly used for reasoning rather than explicitly injecting domain knowledge into the model. However, the question of why they succeed and whether they are able to (re)discover theories, here in the sense of rule sets or CNGs, is unexplored.

In this paper, we propose a graph-based approach to investigate similarities of qualitative relations from a bottom-up perspective. Particularly, we are interested in how the similarities derived from ML methods are related to classic theoretical studies (e.g., on conceptual neighborhoods). By conducting extensive experiments on synthetic data regarding spatial reasoning (here, *RCC-8* relations) and temporal reasoning (here, Allen's thirteen interval relations), we are able to demonstrate that ML models can automatically discover conceptual neighborhood graphs. In addition, experiment results showcase that such graphs can be easily discovered by ML methods even when limited data are available for training. Moreover, the similarities of relations are mostly asymmetric, which echos the findings in [19] from a perspective of cognitive assessment. Furthermore, patterns observed in asymmetric similarities of relations are disclosed. To the best of our knowledge, we are the first to automatically discover conceptual neighborhood graphs of qualitative relations from a bottom-up perspective by analyzing ML methods. In theory, our approach can be used to discover CNGs for any calculus with jointly exhaustive and pairwise disjoint (JEPD) relations.

The remainder of this paper is structured as follows: Section 2 introduces background about how to perform QSTR by using machine learning methods. Section 3 elaborates on the proposed graph-based approach to discover similarities among relations. Section 4 describes the generation of synthetic data, evaluation metrics, and reports experimental results. Section 5 discusses our findings and points out the direction for future studies.

## 2 Background

In this section, we introduce preliminaries of ML methods to achieve QSTR. We summarize notations and abbreviations we use in this paper in Table 1 for quick reference.

■ **Table 1** Terms and their abbreviations used in this paper.

Terms (abbrev.)	
Qualitative Spatio-temporal Reasoning (QSTR)	Conceptual Neighborhood Graphs (CNGs)
Machine Learning (ML)	Artificial Intelligence (AI)
Knowledge Graphs (KGs)	Knowledge Graph Embedding (KGE)
Composition Tables (CTs)	Jointly Exhaustive and Pairwise Disjoint (JEPD) relations
RCC-8 Relations	IR-13 Relations
disconnected (dc)	before (<)
externally connected (ec)	meets (m)
partially overlapping (po)	overlaps (o)
tangentially proper part (tpp)	during (d)
tangentially proper part inverse (tppi)	starts (s)
non-tangentially proper part (ntpp)	finishes (f)
non-tangentially proper part inverse (ntppi)	equal (=)
equal (eq)	after (>)
	met-by (mi)
	overlapped-by (oi)
	contains (di)
	started-by (si)
	finished-by (fi)

### 2.1 Qualitative Representation of Relations

In this paper, we store binary relations between entities in form of triples. A triple of the form  $\langle s, r, o \rangle$  represents an entity *subject* that has a *relation* to another entity *object*. For instance, the statement that a house is externally connected (*ec*) to a park can be represented as  $\langle house, ec, park \rangle$ . A set of such tripled is called a knowledge graph (KG). In our paper, a KG is a simple directed graph, consisting of entities being modeled as nodes and relations between them being modeled as labels of edges. Formally, it can be represented as  $G = (V, E)$ , where  $V$  and  $E$  are the set of nodes/entities and edges with relations being labels, respectively.

### 2.2 Relation Prediction Task

We will focus on a task known as relation prediction, namely inferring the relation between two entities based on other information. It is equivalent to answering the query  $\langle s, ?r, o \rangle$ . Examples include: *what is the topological relation between Los Angeles and Santa Monica?* or *what is the temporal relation between the Battle of Trafalgar and the Napoleonic Wars?*

#### 2.2.1 Symbolic Reasoning Methods

Traditionally, symbolic representations are adopted to represent entities and relations, on top of which qualitative calculi are developed to perform reasoning tasks. For instance, CTs along with path-consistency algorithms are often used to infer missing relation between entities [24]. Given that (*property A, tangential proper part (tpp), park B*) and (*park B, disconnect (dc), house C*), we are able to infer that (*property A, disconnect (dc), house C*) by checking the CT of *RCC-8*. Usually such top-down approaches (which are based on qualitative calculi) fall into the group of symbolic reasoning. Despite their great success in qualitative reasoning in the past, such approaches are faced with noticeable limitations. For instance, they are sensitive to erroneous information or noise. Moreover, they can only be applied to a limited range of reasoning tasks, do not scale well over large datasets, and cannot be easily applied in combination with numeric approaches [25].

## 2.2.2 Knowledge Graph Embedding Methods

Knowledge Graph Embedding (KGE) methods are an embedding technique in ML that has been empirically proven to be effective in reasoning in a subsymbolic way.

Generally speaking, the goal of KGE methods is to learn subsymbolic representations of entities and relations in a high-dimensional continuous vector space while preserving the connectivity between entities and relations from KGs. Typically, developing a KGE model requires the following three components.

- (I) The first is to randomly initialize subsymbolic representations for each entity/relation in a high-dimensional continuous vector space. By doing so, each entity/relation is initialized as a high-dimensional vector (a.k.a embedding or subsymbolic representation) and can be viewed as a point in such high-dimensional vector space. The vector space could be Euclidean space, Hyperbolic space, Spherical space, etc., which vary between different KGE models. The embedding of an entity  $v$ , or a relation  $r$ , can be expressed as  $\mathbf{v} \in \mathbb{U}^d$ , or  $\mathbf{r} \in \mathbb{U}^d$ , where  $\mathbb{U}$  denotes the vector space and  $d$  is its dimension.
- (II) a scoring function is required to measure the likelihood of a triple being positive (i.e., a true statement). Various KGE models specify different scoring functions. For instance, TransE [3], the first KGE model, assumes that for a triple  $\langle s, r, o \rangle$ , the relation  $r$  is a transformation operator in a vector space, which translates the subject  $s$  to the object  $o$ . Thus the embedding of an object entity  $\mathbf{o}$  should be equivalent to the resulting embedding of a subject entity  $\mathbf{s}$  being translated by the relation  $\mathbf{r}$  in the vector space. Then the distance between the embedding of the object entity and the resulting entity can be used as a scoring function:  $score(s, r, o) = \|\mathbf{s} + \mathbf{r} - \mathbf{o}\|$ . Thus, triples that are present in KGs (i.e., positive triples) will obtain a lower score while triples that are not present will gain a higher score.
- (III) an objective function is needed for training through a process of optimization. A commonly used way of constructing such an objective function is by contrasting scores obtained by positive triples with those of negative triples. Often, the objective function is built upon the task of entity prediction (namely answering queries such as  $\langle ?s, r, o \rangle$  or  $\langle s, r, ?o \rangle$ ). For each positive triple  $\langle s, r, o \rangle$ , a number of negative triples (e.g.,  $k$ ) are generated by switching the subject  $s$  and/or the object  $o$  with other randomly selected entities (e.g.,  $s_i$  or  $o_i$ ). Then an objective function  $\mathcal{L}$  can be defined to minimize scores for positive triples while maximizing scores for negative ones:

$$\mathcal{L} = -\log \sigma(\gamma - score(s, r, o)) - \frac{1}{k} \sum_{i=1}^k \log \sigma(score(s_i, r, o) - \gamma) \quad (1)$$

where  $\sigma$  is the sigmoid function and  $\gamma$  is a pre-specified hyper-parameter as a margin.  $\langle s_i, r, o \rangle$  is a negative sample of  $\langle s, r, o \rangle$ .

After a number of iterative optimization over the training data, minimizing the objective function yields embeddings (representations) for all entities and relations in the KG. The optimized KGE model then can be used in various downstream tasks, such as entity prediction relation prediction, and triple classification. A plethora of KGE models have been developed in the the past years, e.g., [3, 20] and various scoring functions have been used (refer to [27] for more details).

Here we elaborate on how to perform relation prediction (i.e., answering a query  $\langle s, ?r, o \rangle$ ) by using trained KGE methods, since they are closely related to our approach discussed in Section 3. Concretely, we enumerate all possible relations ( $r' \in \mathcal{R}$ ) to replace  $?r$  individually

and then sort these relations by  $score(s, r', o)$  in an ascending/descending order. Finally the embedding method regards the relation ranked first as the correct answer to the query. The table on the left in Figure 1 shows examples of ordered sets of relations produced by a trained KGE model regarding different testing queries.

### 3 Knowledge Graph Embedding Methods as Knowledge Miner

Similarity is one of the most commonly used measures to examine relationships of objects. For instance, domain experts introduce conceptual neighbors to indicate similar qualitative relations [15]. Likewise, in this section we introduce an approach to examine similarities between qualitative relations by analyzing trained KGE models from a bottom-up perspective. There are two steps in this approach - initial construction of a relation graph and its refinement.

The first question is how to derive similarities between any two relations in the set  $\mathcal{R}$  from a trained KGE model. Our assumption is that it would be difficult for a trained embedding model to distinguish relations that are similar in a topological sense. That is, in terms of the task of relation prediction, for a testing query (geometry A, ?r, geometry B) (whose target answer is *externally connected* (*ec*)), we hypothesize the embedding-based model may yield similar scores for (geometry A, *ec*, geometry B) and (geometry A, *partially overlap* (*po*), geometry B), because *po* and *ec* are topologically similar. Put differently, the sorted set of predicted relations reveals structural similarities among relations in the sense that similar relations are more easily confused in relation prediction (see Figure 1).

Based on this assumption, we initiate a graph in which vertices are different types of relations. For each testing query  $\langle s, ?r, o \rangle$ , a directed edge is established from the correct relation to either the relation ranked at first (top 1) or second (top 2) in the ordered list of relations. Such a choice relies on whether the relation at top 1 is the correct relation or not. When the correct is ranked at top 1, we do not introduce a loop. Instead, a directed edge starting from the correct relation to the relation at Top 2 is added. If the relation at Top 1 is not the correct, then an edge is built from the correct relation to Top 1. The resulting graph is a *directed* graph, whose edges originate from the correct relation to a relation identified as most similar to the correct by the KGE model. In a directed edge, we use terms - head and tail - to refer to the source and the target of an edge, respectively. The direction of edges reflects which candidate relation (tail) is similar to the target relation (head). Note that by such a distinction, we are able to examine the asymmetric similarities between relations.

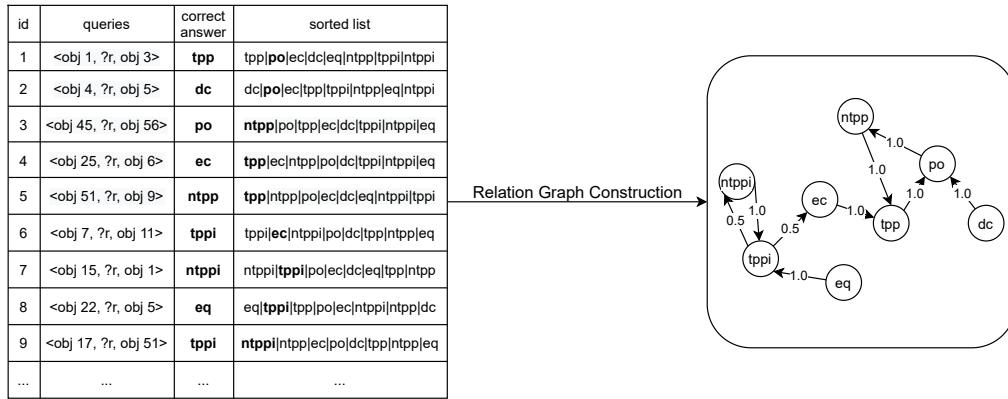
The graph constructed above only illustrates which relations are considered as similar by a KGE model, but does not quantify similarities between relations. Here, we design a weighting function to quantify these similarities. Specifically, the weight of an edge is estimated as the proportion of the number of edges from a head to a tail relation over the total number of edges originating from the head. This function can be formulated as follows:

$$weight(r_i \rightarrow r_j) = \frac{count(r_i \rightarrow r_j)}{\sum_{r' \in \mathcal{R}} count(r_i \rightarrow r')} \quad (2)$$

where  $count(r_i \rightarrow r_j)$  is the cardinality of edges originating from  $r_i$  (head) to  $r_j$  (tail) (with shortest paths). An example of the construction process is shown in Figure 1.

So far, we obtain a directed and weighted graph, which reveals the similarities between different relations; see Figure 2a. We observe that this graph is almost complete (i.e., any two relations/vertices are connected via an edge), because eventually any two relations are likely to be thought of as similar by a KGE model. However, not all these similarities are significant; for instance many edges only have marginal weights (e.g., 0.01). In order to extract significant relationships from the initial relation graph, the next step is to prune insignificant edges to get a refined graph.

### 3:6 Automatically Discover Conceptual Neighborhood



**Figure 1** Relation Graph Construction. Here nine queries are used as examples and the *sorted list* column shows relations sorted by a scoring function from a KGE method. Each relation is represented as a vertex in the graph and edges are established from the correct answer (column 3) to the relation in bold in the sorted list. Weights are calculated by using Eq. 2.

Intuitively one could enumerate different thresholds (for instance, by gradually increasing a threshold (i.e., 0.0, 0.05, 0.1, ..., 1.0)) to cut off edges whose weights are insignificant. Then one can terminate the enumeration process by manually checking whether the refined graph is aligned with our domain knowledge/cognition. However, without enough domain knowledge, it is hard to conclude which graph is meaningful and this means the proposed solution is not truly bottom-up. In order to reduce human intervention in the refinement process, we define a condition to automatically terminate the enumeration. The condition is based on the naive fact that *all relations/vertices must be preserved/connected in the graph after the refinement*, since our focus in this paper is on the relationships of all *relations*. Based on graph theory, such a fact boils down to ensuring that there is always one connected component in the graph after the refinement. Therefore, we can gradually increase thresholds by constant margins (e.g., 0.05) until the initial graph is no longer one connected component. In summary, in the process of refining relation graphs, we generate a number of candidate thresholds (within the range of (0.0, 1.0) and a step of 0.05) in an ascending order and find the maximal threshold that leads to only one connected component in the graph, which is regarded as the refined relation graph.

## 4 Experiments

In this section, we introduce the synthetic data we use to test our method, the evaluation metrics used for graph similarity measure, and present experimental results. Although theoretically our proposed approach can be applied to any set of JEPD relations to automatically discover a graph of relations, we focus on *RCC-8* and *IR-13* here.

### 4.1 Data Preparation

Since real-life datasets are usually incomplete, we generate sets of synthetic data for the purpose of demonstration. Specifically, we choose rectangles as primitive geographical entities for *RCC-8* relations and closed-intervals as primitive temporal entities for *IR-13* relations.

To generate rectangles, we first set up a main area, in which rectangles should be located. By default, the main area is set to be a  $15 \times 15$  unit square with the origin being its bottom-left corner. Then we randomly generate pairs of points within the square and each pair of points

compose the top-left corner and the bottom-right corner of a rectangle<sup>2</sup>. Finally, we compute *RCC-8* relations between any two rectangles to generate synthetic spatial relation triples. Likewise, we generate a number of closed-intervals on the x-axis within the range [0, 500]. Specifically, we randomly select two integers from the range and use the smaller one as the beginning of an interval and the bigger one as the ending of the interval. Then we compute the *IR-13* relations between any two intervals to generate synthetic temporal relation triples. We call the set of all synthetic triples as *complete synthetic data*.

However, without any prior knowledge, it is hard to decide how many rectangles/intervals should be generated within the given main area/line segment. Meanwhile, the number of rectangles/intervals generated in the same area/line may affect discovered relation graphs. Therefore, we independently generate several sets of synthetic triples for both the *RCC-8* and the *IR-13* relations with different number of rectangles/intervals (i.e., [64, 128, 256, 512, 1024]). These sets of triples have different densities of rectangles/intervals. The proportions of different relations generated with respect to different numbers of rectangles/intervals are shown in Table 2.

■ **Table 2** Relation proportions of *RCC-8* (on the left) and *IR-13* (on the right) regarding different numbers of rectangles/intervals  $N=64, 128, 256, 512$  or  $1024$ . All values are multiplied by 100.

$N$	64	128	256	512	1024
<i>dc</i>	43.5	43.4	42.7	42.8	42.3
<i>ec</i>	12.2	11.9	11.8	11.5	11.8
<i>eq</i>	1.6	0.8	0.4	0.2	0.1
<i>ntpp</i>	1.2	1.4	1.1	1.5	1.5
<i>ntppi</i>	1.2	1.4	1.1	1.5	1.5
<i>po</i>	35.6	34.8	37.4	36.4	36.7
<i>tpp</i>	2.4	3.1	2.8	3.1	3.1
<i>tppi</i>	2.4	3.1	2.8	3.1	3.1

$N$	64	128	256	512	1024
<	18.7	14.9	16.1	16.4	16.5
=	1.6	0.8	0.4	0.2	0.1
>	18.7	14.9	16.1	16.4	16.5
<i>d</i>	15.7	17.2	16.4	16.2	16.9
<i>di</i>	15.7	17.2	16.4	16.2	16.9
<i>f</i>	0.1	0.1	0.1	0.1	0.1
<i>fi</i>	0.1	0.1	0.1	0.1	0.1
<i>m</i>	0	0.1	0.1	0.1	0.1
<i>mi</i>	0	0.1	0.1	0.1	0.1
<i>o</i>	14.6	17.1	16.9	16.9	16.1
<i>oi</i>	14.6	17.1	16.9	16.9	16.1
<i>s</i>	0.1	0.2	0.1	0.1	0.1
<i>si</i>	0.1	0.2	0.1	0.1	0.1

## 4.2 Experiment Settings

We choose *HyperRotatE* [5] as the embedding model to learn subsymbolic representations of entities and relations, thanks to its ability of modeling the composition of relations (which is relevant to composition tables) and tree-like graph structures (which is useful for modeling transitive relations (e.g., *ntpp*)). This model also contains the three components mentioned in Section 2.2.2 and has a different scoring function. We use the original implementation of *HyperRotatE* to learn embeddings for entities and relations<sup>3</sup>. Hyper-parameters used for the *RCC-8* and the *IR-13* relations include learning rates: 0.05 (for the *RCC-8* relations) and 0.1 (for the *IR-13* relations), batch sizes: 1024 for both, negative samples: 64 (for the *RCC-8* relations) and 32 (for the *IR-13* relations), and dimensions: 110 (for the *RCC-8* relations)

<sup>2</sup> We ensure that each rectangle is valid. For example, if the two points in a pair align along the same axis, we will remove this pair.

<sup>3</sup> <https://github.com/HazyResearch/KGEmb>

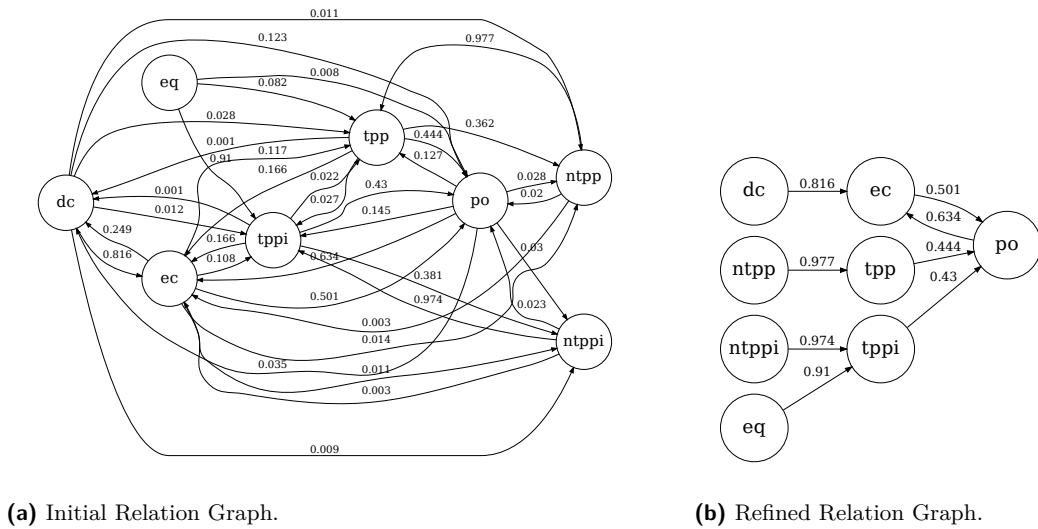
and 18 (for the *IR-13* relations). For *IR-13* relations, we use the same hyper-parameters for all synthetic data. For *RCC-8* relations, we increase the dimension of the embedding space to 200 when the number of entities is 512 or 1024<sup>4</sup>. In the experiment, we train *HyperRotatE* and then perform relation prediction over the *complete synthetic data* by default<sup>5</sup>.

### 4.3 Evaluation Metrics

In order to quantify the differences between the learned relation graph and from CNGs, we introduce three metrics to measure commonality and difference. One solution is to convert graphs to sets of edges (each edge consists of a pair of relations) and to use set operations for quantification. Three metrics can be defined: (1) *False Recall* (i.e., number of false positives): the number of edges that are in our generated graph but not in CNGs (set difference). (2) *True Recall* (i.e., number of true positives): the number of edges that are in both our generated graph and CNGs (set intersection). (3) *Failed Recall* (i.e., number of false negatives): the number of edges that are not in our generated graph but in CNGs (set difference). Clearly, a graph that is similar to CNGs should have a low *False Recall*, a high *True Recall*, and a low *Failed Recall*.

### 4.4 Experimental Results

In this section, we first show direct results from our approach introduced in Section 3. Figure 2 illustrates (a) the initial relation graph resulting from the construction steps and (b) the refined relation graph after pruning. Next we report main findings based on the refined relation graph.



■ **Figure 2** Examples of initial/refined relation graphs produced by our approach.

<sup>4</sup> When the number of entities increased to 512/1024, the model’s performance greatly deteriorated. We assume the performance is compromised due to lack of learnable parameters. Thus, we increase the dimensions to provide more learnable parameters for our models to learn.

<sup>5</sup> Note that we do not tune these hyper-parameters but choose them by empirical experiences. It is worthwhile to investigate the impact of hyper-parameters on the experiment results in the future.



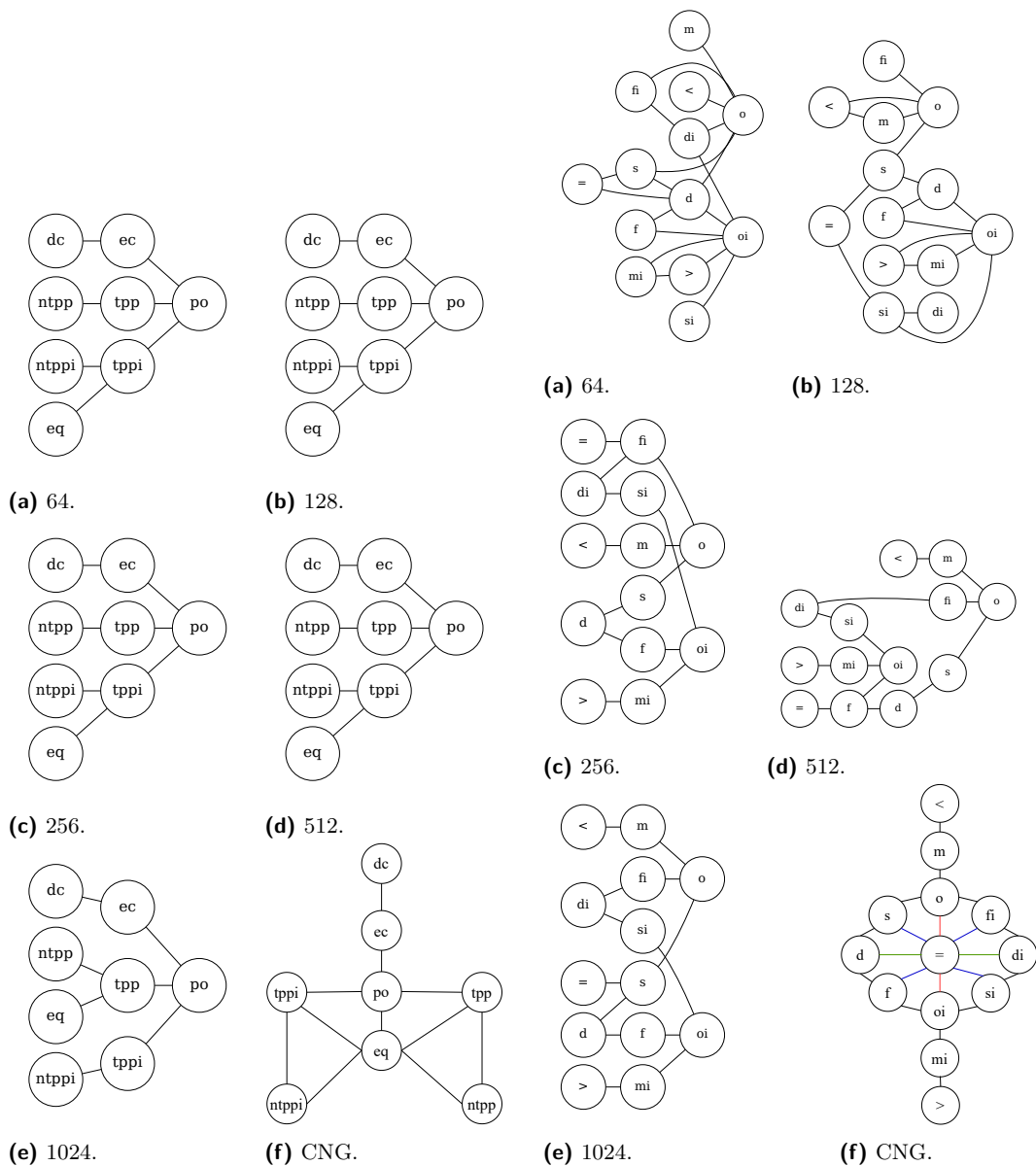
**1. Relation graphs automatically discovered by our approach are well-aligned with CNGs for both *RCC-8* and *IR-13* relations.** Figure 2b implies that our refined relation graph resembles conceptual neighborhood graphs (Figure 3f). This motivates us to examine how similar our refined relation graphs are CNGs from the literature and whether this is merely a coincidence. In order to make our refined graphs comparable with CNGs, we convert the refined graphs into undirected and unweighted relation graphs (*UU-RGs*).

Figure 3 and Figure 4 report the results for *RCC-8* and *IR-13* relations, respectively, with different number of entities being considered. Noticeably, our approach discovers stable relation graphs for both *RCC-8* and *IR-13* relations. In specific, relation graphs for *RCC-8* remain almost unchanged with an increasing number of rectangles and relation graphs for *IR-13* begin to be fixed (except for the *equal* relation) when the number of intervals is 256. This observation also aligns with the statistics shown in Table 2, in which the relation proportions become relatively stable when the number of entities reaches 256. This indicates that the KGE model is mainly affected by the proportion of relations in the synthetic data. Moreover, by comparing Figure 3a, 3b, 3c, 3d and 3e with Figure 3f (or comparing Figure 4c, 4d and 4e with Figure 4f), we can observe that the discovered relation graphs are well-aligned with the CNGs which are defined in the literature (see Figure 3f and Figure 4f), except for differences around the *equal* relation (i.e., “eq” and “=”). This observation demonstrates the ability of ML models in learning domain knowledge purely from data and the effectiveness of our approach in automatically discovering relationships of JEPD relations (*RCC-8* and *IR-13* as examples here). *This demonstrates that conceptual neighborhood graphs can be reproduced from data without any domain knowledge/inductive bias.*

As for the differences around the *equal* relation, one explanation is the lack of enough *equal* relations in our synthetic data. Because we randomly generate rectangles/intervals within a given area/segment, it is relatively rare to yield two rectangles/intervals that have the same geometry. As a result, most *equal* relations are just self-equivalent (e.g.,  $\langle s, eq, s \rangle$ ), which in fact does not provide enough useful information for the model to learn. Hence, we do not consider this a shortcoming of the model.

**2. Similarities of relations are asymmetric and certain relations are more similar. Several patterns in asymmetric similarities of relations are also disclosed.** In this experiment, we examine similarities of relations, which are quantified by weights in Eq. 2. We extract a subgraph from our initial relation graphs (see Figure 2a) that contain edges presented in the theoretical CGNs except for edges that are connected to the *equal* relation (since the *equal* relation is not well-reproduced). We set the number of entities to 1024 and run the *HyperRotatE* model for 20 times to obtain average weights/similarity scores. The extracted subgraphs for *RCC-8* and *IR-13* relations are illustrated in Figure 5.

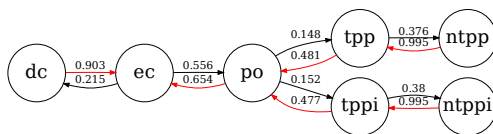
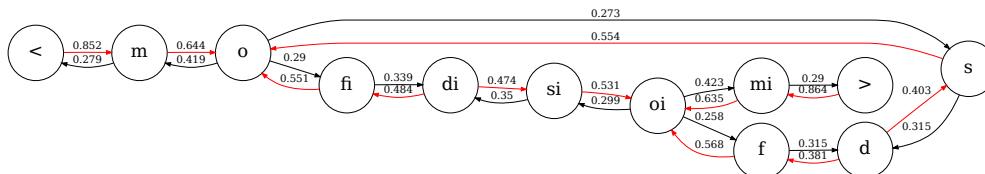
Apparently, we can observe that similarities of relations are *asymmetric*. In other words, the statement that  $a$  is similar to  $b$  differs from that  $b$  is similar to  $a$  ( $a$  and  $b$  are relations). For instance, the similarity between  $dc$  and  $ec$  is 0.903 while the inverse similarity is 0.215. Namely,  $dc$  is more similar to  $ec$  while  $ec$  is less similar to  $dc$ . In fact, Figure 5a shows  $ec$  is most similar to  $po$ , and both  $dc$  and  $po$  are most similar to  $ec$ . Meanwhile, we find that  $ec$  and  $po$  are more similar in general with higher similarities of 0.556 and 0.654. Additionally, there exist similar patterns between relations and their inverses in terms of their asymmetric similarities to other relations. For instance, Figure 5b shows  $<$  is most similar to  $m$  and  $m$  is most similar to  $o$ . In terms of their inverse relations,  $>$  is most similar to  $mi$  and  $mi$  is most similar to  $oi$ . Moreover, in Figure 5a,  $ntpp$  is most similar to  $tpp$  and  $ntppi$  is most similar to  $tppi$ . Similar patterns are shown between  $d \rightarrow f \rightarrow oi$  and  $di \rightarrow fi \rightarrow o$ , as well as between



■ **Figure 3** The relation graph of the RCC-8 relations w.r.t. different number of rectangles.

■ **Figure 4** The relation graph of the IR-13 relations w.r.t. different number of intervals.

$d \rightarrow s \rightarrow o$  and  $di \rightarrow si \rightarrow oi$ . Another interesting observation is that all neighboring relations of the *overlapping* relation (i.e.,  $po$  in *RCC-8* and  $o$  and  $oi$  in *IR-13*) are most similar to the *overlapping* relation (see the arrows that point to the *overlapping* relation). By contrast, in Figure 5b, both  $d$  and  $di$  are most similar to their neighboring relations (the red arrows around them leave out of them). Interestingly, similarity assessments in the cognitive science literature have been shown to be highly non-symmetric as well due to differences in (feature) alignment. For instance, Klippel et al. disclosed that the similarity between *RCC-8* relations vary from different scenarios (such as hurricane, cannon and geometry). In addition, Mark et al. also found that some topological relations indeed are conceptually more similar to others [21].

(a) Similarities for *RCC-8* relations.(b) Similarities for *IR-13* relations.

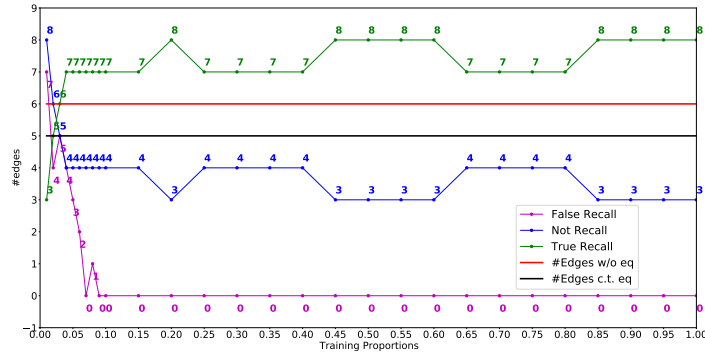
■ **Figure 5** Asymmetric similarities of relations. For two edges between two vertices, the edge with a larger weight is highlighted in red.

### 3. Even with limited training data (i.e., as low as 15% of the *complete synthetic data*), *HyperRotatE* is still capable of reproducing CNGs.

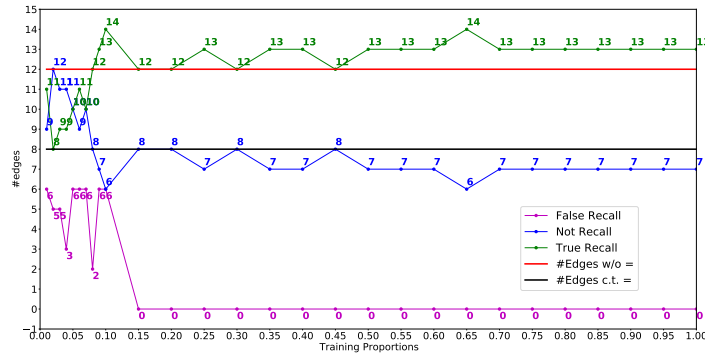
Finally, we are interested in the question of how much training data is needed for *HyperRotatE* to reproduce CNGs. In order to answer this question, we extract subsets of the *complete synthetic data* with different proportions and use the three metrics introduced in Section 4.3 to evaluate the commonality and difference between *UU-RGs* and CNGs. Experiment results with the number of entities being 256 are shown in Figure 6. Red lines and black lines are theoretical references, indicating numbers of edges that are connected to the *equal* relation and that are not in theoretical CNGs, respectively. Clearly, regarding *RCC-8* relations, when more than 10% of the *complete synthetic data* are used for training, *HyperRotatE* is able to reproduce CNGs with stable recalls. Specifically, when the proportion is larger than 10%, *False Recall* continues to be 0, *True Recall* is either 7 or 8 and *Failed Recall* is either 3 or 4. Noticeably, *True Recall* is always above the red line (i.e., 6 – the theoretical number of edges that are not connected to “=” in the CNG) and *Not Recall* is close to the black line (i.e., 5 – the theoretical number of edges that are connected to “=” in the CNG). That is, the relation graphs (except for conceptual neighbors of the *equal* relation) is well-aligned with the theoretical CNGs even when only 10% of the *complete synthetic data* are available. Similar observations are shown for *IR-13* relations; see Figure 6b; however, the same pattern is observed when the training proportion is larger than 15%. In summary, *HyperRotatE* is a robust knowledge miner, which succeeds in discovering CNGs even with limited training data.

## 5 Conclusion

In this work, we presented a graph-based approach to examine similarities among *RCC-8* and *IR-13* relations in neighborhood graphs since they are important to spatio-temporal reasoning and spatial queries. In contrast to traditional approaches that heavily rely on top-down techniques and rule sets, we address this problem in a bottom-up manner without the need of any domain knowledge. Specifically, we focus on the task of relation prediction; namely to answer the query  $\langle s, ?r, o \rangle$ . Our rationale is that it would be difficult for machine learning methods to distinguish relations that are topologically similar when predicting



(a) Quantitative comparison for *RCC-8* relations.



(b) Quantitative comparison for *IR-13* relations.

■ **Figure 6** Quantitative comparison between *UU-RGs* and *CGNs*. *UU-RGs* are reproduced w.r.t. different proportions of the *complete synthetic data* as training data. Lines in red denote the number of edges that are not connected to the *equal* relation in *CNGs* and lines in black denotes the opposite.

missing relations between two entities. Therefore, we can pull similar relations out of the relation prediction task, and then use the proposed method to construct a graph to examine the structure among relations. Our experiments on synthetic data about *RCC-8* and *IR-13* relations reveal that (1) the extracted relation graphs are well-aligned with conceptual neighborhood graphs introduced in [15] and [10] except for neighboring relations of the *equal* relation. We believe this may be caused by a lack of enough *equal* relations in generated training data, which is left for future work; that (2) similarities of relations are asymmetric, and patterns in asymmetric similarities of relations are the same as those in their inverse relations; and that (3) the presented embedding models are robust in mining qualitative spatial and temporal knowledge (i.e., *CNGs*), even with limited training data.

Theoretically, our approach could be applied to any calculus with JEPD relations [6] to automatically discover *CNGs*. We believe our research would benefit theoretical studies of *CNGs* in general and contribute to a broader field, such as geospatial artificial intelligence, by promoting a deeper understanding of what machines really learn from data in a bottom-up manner. In the future, we plan to study whether such *CNGs* will be preserved when realistic data (particularly when non-spatial information is also considered) are used at training.

---

**References**


---

- 1 James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
- 2 Roland Billen and Nico Van de Weghe. Qualitative spatial reasoning. *International Encyclopedia of Human Geography*, pages 12–18, 2009.
- 3 Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 1–9, 2013.
- 4 Lejdel Brahim, Kazar Okba, and Laurini Robert. Mathematical framework for topological relationships between ribbons and regions. *Journal of Visual Languages & Computing*, 2015.
- 5 Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of ACL*, pages 6901–6914, 2020.
- 6 Anthony G Cohn and Jochen Renz. Qualitative spatial representation and reasoning. *Foundations of Artificial Intelligence*, 3:551–596, 2008.
- 7 Matthew P Dube and Max J Egenhofer. An ordering of convex topological relations. In *GIScience*, pages 72–86. Springer, 2012.
- 8 Max J Egenhofer. Deriving the composition of binary topological relations. *Journal of Visual Languages & Computing*, 5(2):133–149, 1994.
- 9 Max J Egenhofer. The family of conceptual neighborhood graphs for region-region relations. In *GIScience*, pages 42–55. Springer, 2010.
- 10 Max J Egenhofer and Khaled K Al-Taha. Reasoning about gradual changes of topological relationships. In *TMSTRGS*, pages 196–219. Springer, 1992.
- 11 Max J Egenhofer and Robert D Franzosa. Point-set topological spatial relations. *IJGIS*, 5(2):161–174, 1991.
- 12 Max J Egenhofer and John Herring. Categorizing binary topological relations between regions, lines, and points in geographic databases. *The*, 9(94-1):76, 1990.
- 13 Andrew U Frank. Qualitative spatial reasoning about distances and directions in geographic space. *Journal of Visual Languages & Computing*, 3(4):343–371, 1992.
- 14 Christian Freksa. Qualitative spatial reasoning. In *Cognitive and linguistic aspects of geographic space*, pages 361–372. Springer, 1991.
- 15 Christian Freksa. Temporal reasoning based on semi-intervals. *AI*, 54(1-2):199–227, 1992.
- 16 Christian Freksa. Using orientation information for qualitative spatial reasoning. In *TMSTRGS*, pages 162–178. Springer, 1992.
- 17 Alexander Klippel. Spatial information theory meets spatial thinking: is topology the rosetta stone of spatio-temporal cognition? *Annals of AAG*, 102(6):1310–1328, 2012.
- 18 Alexander Klippel and Rui Li. The endpoint hypothesis: A topological-cognitive assessment of geographic scale movement patterns. In *COSIT*, pages 177–194. Springer, 2009.
- 19 Alexander Klippel, Jinlong Yang, Jan Oliver Wallgrün, Frank Dylla, and Rui Li. Assessing similarities of qualitative spatio-temporal relations. In *ICSC*, pages 242–261. Springer, 2012.
- 20 Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, volume 29, 2015.
- 21 David M Mark and Max J Egenhofer. Calibrating the meanings of spatial predicates from natural language: Line-region relations. In *Proceedings, SDH 1994*, volume 1, pages 538–553, 1994.
- 22 David A Randell, Zhan Cui, and Anthony G Cohn. A spatial logic based on regions and connection. *KR*, 92:165–176, 1992.
- 23 Jochen Renz, Debasis Mitra, et al. Qualitative direction calculi with arbitrary granularity.
- 24 Jochen Renz and Bernhard Nebel. Qualitative spatial reasoning using constraint calculi. In *Handbook of spatial logics*, pages 161–215. Springer, 2007.

### 3:14 Automatically Discover Conceptual Neighborhood

- 25 Carl Schultz, Mehul Bhatt, Jakob Suchan, and Przemysław Andrzej Wałęga. Answer set programming modulo ‘space-time’. In *IJRR*, pages 318–326. Springer, 2018.
- 26 Jan Oliver Wallgrün, Diedrich Wolter, and Kai-Florian Richter. Qualitative matching of spatial information. In *the 18th SIGSPATIAL*, pages 300–309, 2010.
- 27 Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE TKDE*, 29(12):2724–2743, 2017.