GEOMETRIC RELATIONAL EMBEDDINGS

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart zur Erlangung der Würde eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

BO XIONG

aus Nanchang, China

Hauptberichter:	Prof. Dr. Steffen Staab
Mitberichter:	Prof. Dr. Mathias Niepert
	Prof. Dr. Isabel Valera
Tag der mündlichen Prüfung:	24.07.2024

Institut für Künstliche Intelligenz (KI) der Universität Stuttgart

2024

ABSTRACT

In classical AI, symbolic knowledge is typically represented as relational data within a graph-structured framework, a.k.a., relational knowledge bases (KBs). Relational KBs suffer from incompleteness and numerous efforts have been dedicated to KB completion. One prevalent approach involves mapping relational data into continuous representations within a low-dimensional vector space, referred to as relational representation learning. This facilitates the preservation of relational structures, allowing for effective inference of missing knowledge from the embedding space. Nevertheless, existing methods employ pure-vector embeddings and map each relational object, such as entities, concepts, or relations, as a simple point in a vector space (typically Euclidean $\mathbb{R}$). While these pure-vector embeddings are simple and adept at capturing object similarities, they fall short in capturing various discrete and symbolic properties inherent in relational data.

This thesis surpasses conventional vector embeddings by embracing geometric embeddings to more effectively capture the relational structures and underlying discrete semantics of relational data. Geometric embeddings map data objects as geometric elements, such as points in hyperbolic space with constant negative curvature or convex regions (e.g., boxes, disks) in Euclidean vector space, offering superior modeling of discrete properties present in relational data. Specifically, this dissertation introduces various geometric relational embedding models capable of capturing: 1) complex structured patterns like hierarchies and cycles in networks and knowledge graphs; 2) intricate relational/logical patterns in knowledge graphs; 3) logical structures in ontologies and logical constraints applicable for constraining machine learning model outputs; and 4) high-order complex relationships between entities and relations.

Our results obtained from benchmark and real-world datasets demonstrate the efficacy of geometric relational embeddings in adeptly capturing these discrete, symbolic, and structured properties inherent in relational data, which leads to performance improvements over various relational reasoning tasks.

ZUSAMMENFASSUNG

In der klassischen Künstlichen Intelligenz wird symbolisches Wissen in der Regel als relationale Daten in einem graphenstrukturierten Rahmen repräsentiert, auch als relationale Wissensdatenbanken (KBs) bekannt. Relationale KBs leiden unter Unvollständigkeit und Rauschen, und zahlreiche Bemühungen wurden der Vervollständigung von KBs gewidmet. Ein verbreiteter Ansatz besteht darin, relationale Daten in kontinuierliche Repräsentationen in einem niederdimensionalen Vektorraum abzubilden, bekannt als relationales Repräsentationslernen. Dies erleichtert die Bewahrung relationaler Strukturen und ermöglicht eine direkte Inferenz von fehlendem Wissen aus dem Einbettungsraum.

Dennoch verwenden bestehende Methoden reine Vektor-Einbettungen und ordnen jedes relationale Objekt, wie Entitäten, Konzepte oder Relationen, als einfachen Punkt in einem Vektorraum (typischerweise euklidisch $\mathbb{R}$) zu. Obwohl diese reinen Vektor-Einbettungen einfach sind und Objektähnlichkeiten gut erfassen können, sind sie weniger geeignet, um verschiedene diskrete und symbolische Eigenschaften, die in KBs vorhanden sind, zu erfassen.

Diese Dissertation übertrifft herkömmliche Vektoreinbettungen, indem sie geometrische Einbettungen annimmt, um die relationalen Strukturen und die zugrunde liegende diskrete Semantik relationaler Daten effektiver zu erfassen. Geometrische Einbettungen ordnen Datenobjekte als geometrische Elemente zu, wie Punkte im hyperbolischen Raum mit konstanter negativer Krümmung oder konvexe Bereiche (z. B. Boxen, Scheiben) im euklidischen Vektorraum, und bieten damit eine überlegene Modellierung diskreter Eigenschaften, die in relationalen Daten vorhanden sind.

Insbesondere führt diese Dissertation verschiedene Modelle geometrischer relationer Einbettungen ein, die in der Lage sind, folgendes zu erfassen: 1) komplexe strukturierte Muster wie Hierarchien und Zyklen in Netzwerken und Wissensgraphen; 2) komplexe relationale/logische Muster in Wissensgraphen; 3) logische Strukturen in Ontologien und logische Einschränkungen, die zur Einschränkung von Ausgaben von maschinellem Lernen verwendet werden können; und 4) komplexe Beziehungen höherer Ordnung zwischen Entitäten und Relationen.

Unsere Ergebnisse, die aus Benchmark- und realen Datensätzen gewonnen wurden, zeigen die Wirksamkeit geometrischer relationer Einbettungen bei der geschickten Erfassung dieser diskreten, symbolischen und strukturierten Eigenschaften, die in relationalen Daten vorhanden sind.

ACKNOWLEDGMENTS

It would not be possible without the collaboration with my co-authors, fellow colleagues, supervisors, and the support from my family.

Foremost, I would like to thank Steffen Staab, my primary supervisor, who consistently sets the highest standards for "what is good research". In the early stages of my research, Prof. Staab displayed exceptional patience and kindness, especially when my English-speaking and scientific presentation skills were less proficient. His insightful feedback on my research and critical thinking has been invaluable, and he has provided hands-on guidance in paper writing, teaching me how to conduct impactful and meaningful research.

Meanwhile, I am equally grateful to Mathias Niepert and Isabel Valera, who served as members of my Thesis Advisory Committee (TAC). Their constructive comments and discussions during my TAC meetings greatly shaped the quality of the research.

I would like to thank some fellow co-authors whose guidance greatly influenced my research: Michael Cochez for discussions on learning with hyperbolic geometry; Shirui Pan for discussions on graph neural networks in various geometric spaces; Nico Potyka, a former postdoc in my research group, for discussions on Description Logic and ontology embeddings.

I also extend my heartfelt thanks to many collaborators who have significantly contributed to the development of my work. I acknowledge the valuable contributions of Mojtaba Nayyeri, Shichao Zhu, Trung-Kien Tran, Carl Yang, Daniel Daza, Zifeng Ding, Chengjin Xu, Chuan Zhou, Jiaying Lu, Ming Jing, Linhao Luo, Yuqicheng Zhu, Yunjie He, Zihao Wang, Özge Erten, Cosimo Gregucci, Daniel Hernandez, and many others. Their collaboration and insights have enriched my academic pursuits.

My Ph.D. project was mainly supported by two parties: 1) KnowGraphs (Knowledge Graphs at Scale), in which I worked as the ESR 5 (Early Stage Researcher) for three years; 2) the International Max Planck Research School for Intelligent Systems (IMPRS-IS), from which my Ph.D. thesis advisory commitment (TAC) team was formed. I would like to express my sincere gratitude to KnowGraphs and IMPRS-IS for providing essential support and instruments in facilitating my research journey.

In addition, my research journey was enriched by two secondments. I am deeply thankful to Michel Dumontier of Maastricht University in the Netherlands, who generously supported and supervised my exploration of ontology embeddings for Cell ontology. I am equally appreciative of Roberto Navigli at Babelscape in Rome, who served as an excellent host during my secondment. My time in Rome was not

only academically enriching but also personally fulfilling, as I had the opportunity to make new friends and cherish the experiences that contributed to my growth.

Last but not least, I would like to thank my family. I would not be able to finish my Ph.D. without their truly love.

CONTENTS

1

INTRODUCTION

1.1 BACKGROUND, MOTIVATION, AND CHALLENGES

Representation learning plays a pivotal role in modern machine learning, providing the capability to acquire compact, continuous, and lower-dimensional representations for a variety of real-world data, including images [76], words [155], and documents [81]. These distributional representations allow for the discrimination of relevant distinctions while disregarding irrelevant variations among these objects. They serve as valuable inputs for various machine learning tasks, such as image recognition [76], text categorization [155], and disease diagnosis [81].

The predominant approach in many current works maps objects into a low-dimensional vector space, typically represented in the Euclidean space $\mathbb{R}^d$. We refer to this representation as a plain vector embedding. The rationale behind using plain vector embeddings is their ability to preserve 'similarities' between objects through pairwise distances or inner products in the vector space. For example, images from the same categories or words occurring in similar linguistic contexts are mapped to vectors that are 'near' in the embedding space.

Unlike the approaches prevalent in modern machine learning, classical AI, such as knowledge representations and reasoning [95] and statistical relational learning [132], relies on symbolic knowledge. Symbolic knowledge is often represented as structured and relational data that delineate semantic relationships among entities and/or concepts. Typically, this representation takes the form of a set of factual statements, each encapsulating a fact that describes a semantic relationship involving two or more entities and/or classes. Additionally, with the aid of complex mathematical constructs, symbolic knowledge can also be described as a set of logical statements, each describing a logical relationship among various entities and/or concepts. These factual and logical statements together form a symbolic knowledge base, storing relational knowledge over a specific domain. Such symbolic knowledge plays a crucial role in various applications, including biomedical [41, 134] and intelligent systems [148].

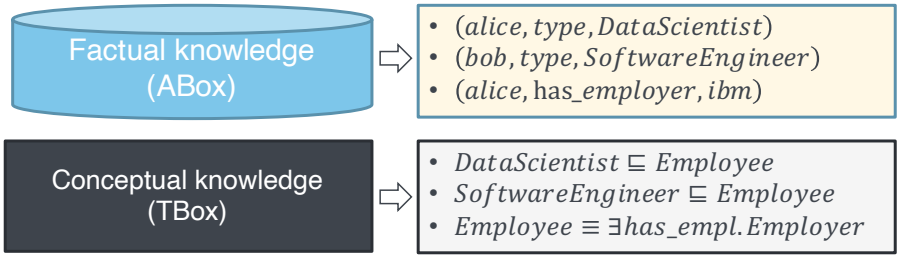


Figure 1.1: A schematic illustration of a symbolic knowledge base. It consists of a ABox describing the factual knowledge over entities and a TBox describing the conceptual knowledge over concepts.

The relational knowledge expressed in a symbolic knowledge base can be categorized into two categories as shown in Fig 1.1:

- **Factual knowledge** describes relationships among different entities. This is typically described in a knowledge graph, where factual knowledge is represented by a set of factual statements in the form of triples (h, r, t), with h, t being the entities and r being the label of a binary relation between them. Beyond binary relations, a knowledge graph can be extended to represent high-order or multi-fold relations among multiple entities, such as the co-authorship relationships in a scientific network.

- **Conceptual knowledge** describes relationships among different concepts. This is typically described in concept ontologies, where concepts are organized in a concept hierarchy with hierarchical and conceptual relationships such as "is_a" or "has_part". By applying complex mathematical constructs such as intersection, existential, and universal quantifications, ontologies can be extended to also describe logical relationships among concepts.

Following the conventions of the semantic web community, we call the part of the factual knowledge an ABox (assertional knowledge) and the part of conceptual knowledge a TBox (terminological knowledge).

Relational representation learning acts as a bridge, transforming discrete and symbolic relational data into continuous and low-dimensional representations. This approach connects classical knowledge representation with modern vector-based machine learning. The benefits of using vector representations for relational data are twofold: 1) Vector representations capture not only the relational structure suitable for classical reasoning but also the similarity and analogical structure between entities/concepts, facilitating analogical and similarity-based reasoning. 2) The learned vector representations are robust to incomplete and noisy data, making them more suitable for reasoning tasks in real-world scenarios. In the real world, many relational datasets,

such as Freebase [16] and DBpedia [6], are curated through human efforts. Despite the already substantial volume of this data, it remains incomplete and noisy.

Vector representations have proven beneficial for various instances of relational data, spanning graphs, knowledge or multi-relational graphs, and ontologies:

- **Graph embeddings** map nodes in graphs into vectors while preserving the graph structure [85]. Typical approaches include random walk-based approaches [51] and graph neural networks [30]. These learned embeddings can be used to predict missing node types (i.e.,m node classification), missing edges between nodes (i.e., link prediction), and graph-level properties (i.e., graph classifications).

- **Knowledge or multi-relational graph embeddings** map both entities and relations into vector space while preserving their relational structures in the embedding space [83]. This is achieved by modeling relations between entities as functions (i.e., functional methods) [19] or by modeling the plausibility of a fact as a three-way interaction (i.e., semantic matching methods) [122]. Knowledge graph embeddings can effectively infer missing relational facts, even in incomplete knowledge graphs, which is a process known as knowledge graph completion.

- **Ontology embeddings** map concepts in ontologies into vectors, while preserving the ontological structure [69]. Unlike knowledge graph embeddings that focus on factual knowledge, ontology embeddings consider ontological knowledge. Encoding the logical structure in a standard embedding space is challenging but crucial for ontology embeddings.

Challenges. These learned representations enable effective inference of missing knowledge directly from the embedding space. However, most relational representation learning approaches consider plain Euclidean vectors as the embeddings, which is similar to the embeddings of image and text data, but they may fall short in capturing crucial properties of relational data that are not easily modeled in a plain, low-dimensional vector space. These properties are typically structural, discrete, and symbolic, while plain vector embeddings are designed to capture similarity only. Fig 1.2 shows some examples of these discrete properties. We elaborate these properties as follow:

- **Structural patterns.** Relational data exhibit highly complex structural patterns such as hierarchies and cycles. Typical examples

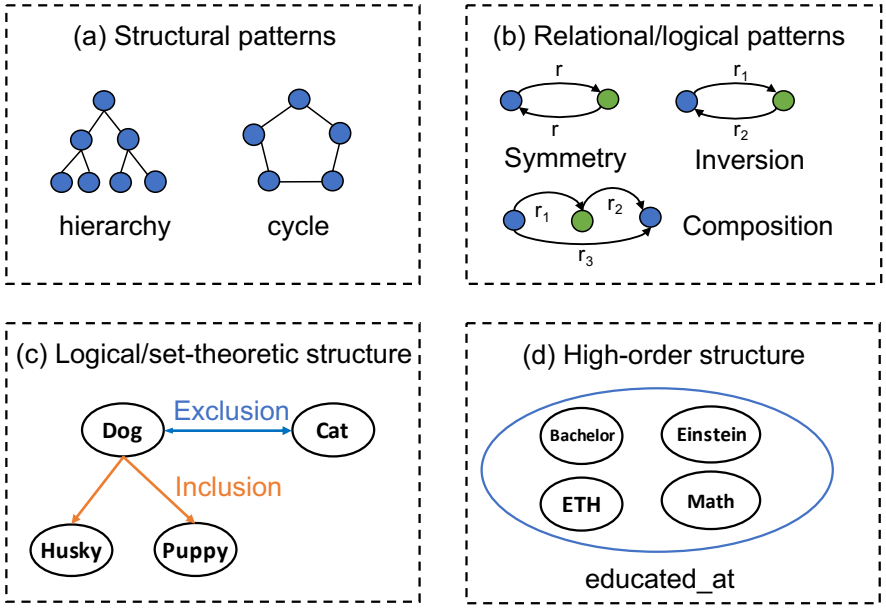


Figure 1.2: A schematic illustration of the discrete properties in relational data. (a) Structural patterns include hierarchical and cyclic structures in graphs; (b) Relational patterns are logical rules/implication over relations; (c) A logical/set-theoretical structure described by logical or set operators (inclusion and exclusion). (d) A high-order structure described by multi-fold relations among multiple entities.

include WordNet [111] describing the word sense hierarchy and Gene ontology [5] describing the hierarchy of gene functions. The plain Euclidean-based vector embeddings are designed for flat data but struggle with modeling data with complex structural patterns. For example, the number of nodes in a hierarchy grows exponentially while the volume of Euclidean space only grows linearly w.r.t the radius.

- **Relational/logical patterns.** Multi-relational data or knowledge graphs, describe relational facts between entities. These relations exhibit many relational or logical patterns such as symmetry (e.g., *has_friend*), inversion (*is_director_of* and *is_directed_by*) and implication (e.g., *mother_of* $\rightarrow$ *parent_of*). Modeling these relational or logical patterns is of great importance to the embeddings as it not only imropves expressiveness of knowledge graph embedding models, but more importantly, facilitates guaranteed generalization capability, i.e., once the patterns are learned, facts that adhere these patterns can be inferred.

- **Logical/set-theoretic structure.** Relational data may have been defined by applying set-theoretic operators such as set inclusion and set exclusion or logical operators such as logical intersection

and negation. This is especially useful when describing conceptual/schematic knowledge over concepts. In this sense, conceptual knowledge is described as logical statements expressed in the form of Description Logic (DL) languages. Embedding the logical structure expressed in the logical statements in DL is challenging as the logical structures in the statements are supposed to preserved in the embedding space, which is beyond the similarity that is preservation in plain vector embeddings. Furthermore, the logical structure in relational data can also be used to describe relational constraints over the output of machine learning models. Embedding these constraints is important for many multi-label machine learning models as it guarantees predictive coherence of predictions.

- **High-order structure.** Relational data may describe high-order relational facts, where each fact can describe a complex multi-fold relationship between multiple entities and/or relations. For example, in hyper-relational or n-ary relational knowledge graphs, each triple fact is contextualized by a set of qualifiers with each qualifier being an relation-entity pair. Another example is the nested relational knowledge graphs, in which a fact can describe a relationship over other facts, a.k.a., facts over facts. Most knowledge graph embedding approaches are designed for triple-based knowledge graphs and fail to capture high-order knowledge. Modeling these high-order knowledge is challenging as logical properties may exist not only in the triple level but also in the high-order structure level. For example, in a query of hyper-relational facts, attaching qualifiers to a fact may only narrow down the answer set but never enlarge it, a.k.a., qualifier monotonicity. In nested relational knowledge graphs, nested relational facts may express some logical rules.

Going beyond plain vector embeddings, *geometric relational embeddings* replace the plain vector representations with more advanced geometric objects, such as convex regions [92, 135], probabilistic density functions [136, 166], geometric elements of non-Euclidean manifolds [26], and their combinations [152]. Different from plain vector embeddings, geometric relational embeddings provide a rich geometric inductive bias for modeling various discrete properties of relational data while being still able to capture similarity. For example, embedding ontological concepts as convex regions allows for modeling not only similarity of concepts but also set-based and logical operators over concepts, such as set inclusion, set intersection [187] and logical negation [202]. This is very useful for concept embeddings in ontologies. On the other hand, representing data on non-Euclidean Riemannian manifolds allows for capturing complex structural patterns, such as representing hierarchies in hyperbolic space [26] and cycles in spherical space.

Geometric relational embeddings have been successfully applied in many relational reasoning tasks including but not limited to knowledge graph (KG) completion [1], ontology/hierarchy reasoning [162], hierarchical multi-label classification [130], and logical query answering [136]. However, different downstream applications require varying capabilities from underlying embeddings and, hence, appropriate choice requires a sufficiently precise understanding of embeddings' characteristics and task requirements.

1.2 RESEARCH QUESTIONS

We investigate the following research questions (**RQs**):

RQ 1: How to faithfully modeling complex graph structural patterns such as hierarchies and cycles in graph data and what are the suitable geometric inductive biases for these structural patterns? how to develop the corresponding graph neural network components that are suitable for modeling these structural patterns (Chapter 3.1).

RQ 2: For multi-relational or knowledge graphs that simultaneously exhibit both complex graph structural patterns (e.g., hierarchies and cycles) and complex relational patterns (e.g., symmetry, anti-symmetry, inversion, and composition), how to faithfully modeling both of these patterns in a single embedding space? (Chapter 3.2).

RQ 3: For ontological data where facts are expressed as logical statements/axioms with Description Logic languages, how to faithfully represent these logical statements/axioms while preserving the underlying logical structure with embeddings? what are the geometric inductive biase for representing concepts? (Chapter 4).

RQ 4: How to represent relational constraints for machine learning models such that the model can produce outputs that are logically coherent to the relational constraints? (Chapter 5).

RQ 5: How to embed relational data with high-order relational structure such as hyper-relational knowledge graphs and nested relational knowledge graphs, in a way that the underlying logical properties such as logical patterns inherent in the relational data can be still modeled? (Chapter 6).

1.3 THESIS CONTRIBUTIONS AND OUTLINE

To address these limitations, this dissertation goes beyond vector embeddings and proposes various geometric embeddings that faithfully model various discreate properties of different types of relational data. Geopmetric embeddings, instead of mapping relational objects as plain vectors in Euclidean space, encode relational objects as geometric elements (e.g., balls, boxes, and convex cones) or as elements in non-Euclidean manifolds (e.g., hyperbolic or spherical space). The primal advantage of geometric embeddings is the faithful encoding of discreate properties of relational data. In this thesis, we addressed several encoding issues of vector embeddings over relational data with geometric embeddings (Cf. Fig. 1.3). Our contributions and the remaining content of the thesis are summarised as below:

In Chapter 2, we introduce some necessary preliminaries, foundational concepts, and related works that are relevant to this dissertation.

In Chapter 3, we introduce pseudo-Riemannian manifold embeddings. This chapter makes the following two contributions.

- We present a principled framework, pseudo-Riemannian GCN, which generalizes GCNs into pseudo-Riemannian manifolds with indefinite metrics, providing more flexible inductive biases to accommodate complex graphs with mixed topologies. We also defined neural network operations in pseudo-Riemannian manifolds with novel geodesic tools, to stimulate the applications of pseudo-Riemannian geometry in geometric deep learning. Extensive evaluations on three standard tasks demonstrate that our model outperforms baselines that operate in Riemannian manifolds.

- We proposes UltraE, an ultrahyperbolic KG embedding method in a pseudo-Riemannian manifold that interleaves hyperbolic and spherical geometries, allowing for simultaneously modeling multiple hierarchical and non-hierarchical structures in KGs. We derive a relational embedding by exploiting the pseudo-orthogonal transformation, which is decomposed into various geometric operators including circular rotations/reflections and hyperbolic rotations, allowing for inferring complex relational patterns in KGs. On three standard KG datasets, UltraE outperforms many previous Euclidean and non-

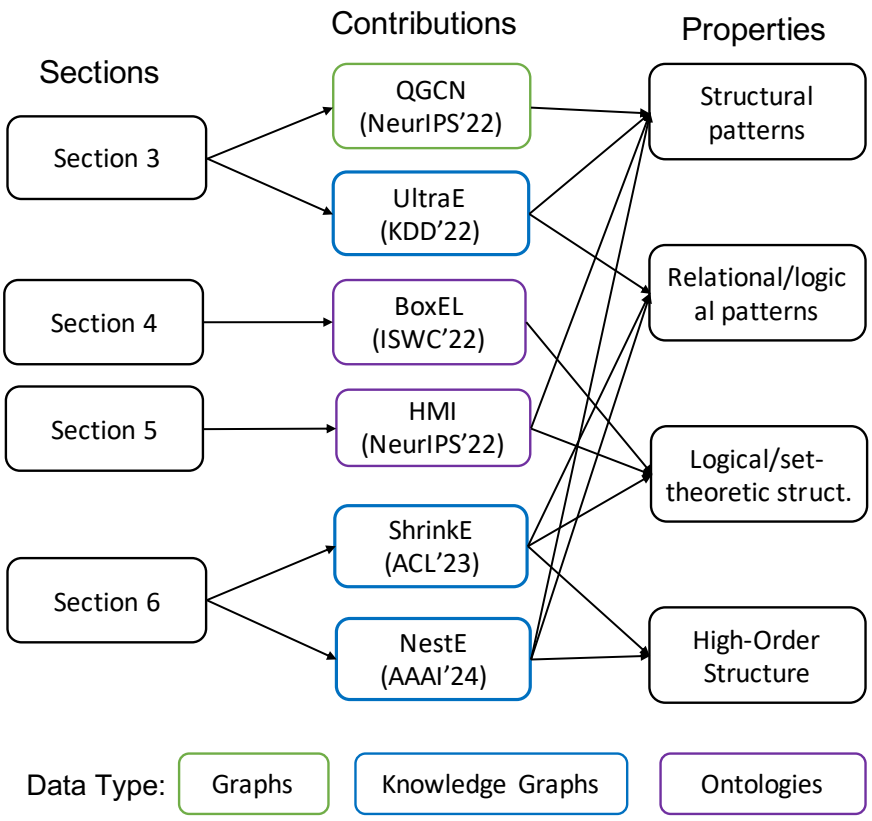


Figure 1.3: An overview of the proposed methodologies, the corresponding properties and the relational data types.

Euclidean counterparts, especially in low-dimensional setting.

In Chapter 4, we focus on ontology embeddings and make the following contribution.

- We propose BoxEL, a geometric knowledge base embedding method that explicitly models the logical structure expressed by the theories of $\mathcal{EL}^{++}$. Different from the standard KGEs that simply ignore the analytical guarantees, BoxEL provides *soundness* guarantee for the underlying logical structure by incorporating background knowledge into machine learning tasks, offering a more reliable and logic-preserved fashion for knowledge base reasoning. The empirical results further demonstrate that BoxEL outperforms previous KGEs and $\mathcal{EL}^{++}$ embedding approaches on subsumption reasoning over three ontologies and predicting protein-protein interactions in a real-world biomedical knowledge base.

In Chapter 5, we exploit ontology embeddings on improving machine learning models and make the following contribution.

- We focus on a structured multi-label prediction task whose output is supposed to respect the implication and exclusion constraints. We show that such a problem can be formulated in a hyperbolic Poincaré ball space whose linear decision boundaries (Poincaré hyperplanes) can be interpreted as convex regions. The implication and exclusion constraints are geometrically interpreted as insideness and disjointedness, respectively. Experiments on 12 datasets show significant improvements in mean average precision and lower constraint violations, even with an order of magnitude fewer dimensions than baselines.

In Chapter 6, we introduce two geometric embeddings for high-order relational knowledge graphs.

- We present *ShrinkE*, a geometric hyper-relational KG embedding method aiming to explicitly model these patterns. ShrinkE models the primal triple as a spatial-functional transformation from the head into a relation-specific box. Each qualifier "shrinks" the box to narrow down the possible answer set and, thus, realizes qualifier monotonicity. The spatial relationships between the qualifier boxes allow for modeling core inference patterns of qualifiers such as implication and mutual exclusion. Experimental results demonstrate ShrinkE's superiority on three benchmarks of hyper-relational KGs.

- We propose FactE, a family of hypercomplex embeddings capable of embedding both atomic and nested factual knowledge. This framework effectively captures essential logical patterns that emerge from nested facts. Empirical evaluation demonstrates the substantial performance enhancements achieved by FactE compared to existing baseline methods. Additionally, our generalized hypercomplex embedding framework unifies previous algebraic (e.g., quaternionic) and geometric (e.g., hyperbolic) embedding methods, offering versatility in embedding diverse relation types.

In Chapter 7, we conclude the whole dissertation, summarize the limitation, and foresee some future works.

1.3.1 *Publications*

This dissertation contains material in papers that were published already in conference proceedings as listed below. Unless explicitly indicated, these papers were significantly contributed by the author in terms of idea generation, experimentation, paper writing and so on.

Published research papers that directly contribute to this dissertation:

- **B. Xiong,** M. Nayyeri, L. Luo, Z. Wang, S. Pan, S. Staab. *NestE: Modeling Nested-Relational Structure for Knowledge Graph Reasoning.* The 38th Annual Conference on Artificial Intelligence. **AAAI 2024**. [186]

- **B. Xiong,** M. Nayyeri, S. Pan, S. Staab. *Shrinking Embeddings for Hyper-Relational Knowledge Graphs.*. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. **ACL 2023**.[184]

- **B. Xiong,** M. Cochez, M. Nayyeri, S. Staab. *Hyperbolic Embedding Inference for Structured Multilabel Prediction.* Advances in Neural Information Processing Systems. **NeurIPS 2022**. [182]

- **B. Xiong*****,** S. Zhu*, N. Potyka, S. Pan, C. Zhou, S. Staab. *Pseudo-Riemannian Graph Convolutional Networks*. Advances in Neural Information Processing Systems. **NeurIPS 2022**.[190]

- **B. Xiong,** N. Potyka, T. Tran, M. Nayyeri, S. Staab. *Faithful Embeddings for EL++ Knowledge Bases*. International Semantic Web Conference. **ISWC 2022**. [187]

- **B. Xiong**, S. Zhu, M. Nayyeri, C. Xu, S. Pan, C. Zhou, S. Staab. *Ultrahyperbolic Knowledge Graph Embeddings*. In Proceedings of The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. **SIGKDD 2022**. [189]

Unpublished survey/position paper that contributes to the foundation chapter of the dissertation:

- **B. Xiong,** M. Nayyeri, M. Jin, Y. He, M. Cochez, S. Pan, S. Staab. *Geometric Relational Embeddings: A Survey*. Technique Report, 2023. [185]

Published tutorial proposals related to this dissertation:

- **Bo Xiong**, Mojtaba Nayyeri, Daniel Daza, Michael Cochez. *Reasoning beyond Triples: Recent Advances in Knowledge Graph Embeddings.* Tutorial at The 32nd ACM International Conference on Information and Knowledge Management. **CIKM 2023**.

- Min Zhou, Menglin Yang, **Bo Xiong**, Hui Xiong, Irwin King. *Hyperbolic Graph Neural Networks: A Tutorial on Methods and Applications.* Tutorial at The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. **SIGKDD 2023**.

Papers that were co-authored by the author and were relevant but not included to this dissertation:

- Y. Tan, H. Lv, Z. Zhou, W. Guo, **B. Xiong**, W. Liu, C. Chen, S. Wang, and C. Yang. *Logical Relation Modeling and Mining in Hyperbolic Space for Recommendation.* In Proceedings of The 40th IEEE International Conference on Data Engineering. **ICDE 2024**.[154]

- M. Nayyeri, **B. Xiong,** M.Mohammadi, M. Mahfuja Akter, M. Mohtashim Alam, J. Lehmann, S. Staab. *Knowledge Graph Embeddings using Neural Ito Process: From Multiple Walks to Stochastic Trajectories..* In Findings of the 60th Annual Meeting of the Association for Computational Linguistics. **Findings of ACL 2023**.[118]

- J. Lu, J. Shen, **B. Xiong**, W. Ma, S. Staab, and C. Yang. *HiPrompt: Few-Shot Biomedical Knowledge Fusion via Hierarchy-Oriented Prompting*. In Proceedings of The 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. **SIGIR 2023**. [105]

- C. Xu, F. Su, **B. Xiong**, J. Lehmann. *Time-aware Entity Alignment using Temporal Relational Attention*. In Proceedings of the ACM Web Conference. **WWW 2022**.[191]

- Y. Zhu, N. Potyka, **B. Xiong**, K. Tran, M. Nayyeri, S. Staab and E. Kharlamov. International Semantic Web Conference. Poster & Demo track of the International Semantic Web Conference.

- Y. He, M. Nayyeri, **B. Xiong**, Y. Zhu, E. Kharlamov and S. Staab. Can Pattern Learning Enhance Complex Logical Query Answering? Poster & Demo track opf the International Semantic Web Conference.

- L. Luo, J. Ju, **B. Xiong**, Y, Li, G. Haffari, and S. Pan. *ChatRule: Mining Logical Rules with Large Language Models for Knowledge Graph Reasoning.* arXiv preprint arXiv:2309.01538, 2023.

All code and datasets related to the dissertation are available at DaRUS (the data repository of the University of Stuttgart).

- Xiong, Bo; Potyka, Nico; Tran, Trung-Kien; Nayyeri, Mojtaba; Staab, Steffen, 2024, "Code for Faithful Embeddings for EL++ Knowledge Bases", https://doi.org/10.18419/darus-3989, DaRUS, V1.

- Xiong, Bo; Nayyeri, Mojtaba; Pan, Shirui; Staab, Steffen, 2024, "Code for Shrinking Embeddings for Hyper-relational Knowledge Graphs", https://doi.org/10.18419/darus-3979, DaRUS, V1.

- Xiong, Bo; Nayyeri, Mojtaba; Cochez, Michael; Staab, Steffen, 2024, "Code for Hyperbolic Embedding Inference for Structured Multi-Label Prediction", https://doi.org/10.18419/darus-3988, DaRUS, V1.

- Xiong, Bo; Nayyeri, Mojtaba; Luo, Linhao; Wang, Zihao; Pan, Shirui; Staab, Steffen, 2024, "Replication Data for NestE: Modeling Nested Relational Structures for Knowledge Graph Reasoning (AAAI'24)", https://doi.org/10.18419/darus-3978, DaRUS, V1.

- Xiong, Bo; Zhu, Shichao; Nayyeri, Mojtaba; Xu, Chengjin; Pan, Shirui; Staab, Steffen, 2024, "Code for Ultrahyperbolic Knowledge Graph Embeddings", https://doi.org/10.18419/darus-4342, DaRUS, V1.

- Xiong, Bo; Zhu, Shichao; Potyka, Nico; Pan, Shirui; Zhou, Chuan; Staab, Steffen, 2024, "Code for Pseudo-Riemannian Graph Convolutional Networks", https://doi.org/10.18419/darus-4340, DaRUS, V1.

2

FOUNDATIONS

In this chapter, we introduce necessary preliminaries, foundational concepts, and related works that are relevant to this dissertation.

2.1 RELATIONAL DATA

In this dissertation, our primary focus is *relational data* that describes diverse relationships between entities and/or concepts in a graph-structured format. We choose to model relational data as a graph because it offers greater flexibility for integrating new sources of data [75]. This is in contrast to the standard relational data model where a schema must be pre-defined and adhered to at each step. Graph-structured data models have found extensive use in organizing various real-world relational data, such as information networks, knowledge graphs, and biomedical ontologies.

2.1.1 *Graph-Structured Data Models*

There are several graph-structured data models, such as directed edge-labeled graphs, heterogeneous graphs, and property graphs, which we introduce as follows.

Directed edge-labeled graphs, also called multi-relational graphs [123], are one of the graph-structured data models. A directed edge-labeled graph is defined by a set of nodes representing entities or concepts, like *New York* and *USA*, and a set of directed labeled edges connecting these nodes, with each labeled edge representing a relationship between these connected nodes, such as (*New York*, *CityOf*, *USA*). Formally, a directed edge-labeled graph is defined as:

Definition 1 (Directed edge-labeled graph [75])**.** *A directed edge-labeled graph is a tuple $G = (V, E, L)$, where $V \subseteq \boldsymbol{Con}$ is a set of nodes, $L \subseteq \boldsymbol{Con}$ is a set of edge labels, and $E \subseteq V \times L \times V$ is a set of edges, where **Con** denotes a countably infinite set of constants.*

Note that this definition is very flexible, as we do not assume that V and L are disjoint. In principle, a node can also serve as an edge label, and nodes and edge labels can be present without any associated edge. Moreover, although the edge is directional, bidirectional edges can be simply represented with two edges with inverse directions.

One limitation of this definition is that it does not distinguish between nodes and the type of nodes but rather expresses the type as a relation, e.g., (*New York*, *Type*, *City*).

Heterogeneous graphs or heterogeneous information networks [78] represent relational data as a set of nodes and a set of edges, with each node and edge associated with a type or label. A heterogeneous graph is formally defined as follows.

Definition 2 (Heterogeneous graph [75])**.** *A heterogeneous graph is a tuple $G = (V, E, L, l)$, where $V \subseteq \boldsymbol{Con}$ is a set of nodes, $L \subseteq \boldsymbol{Con}$ is a set of edge/node labels, $E \subseteq V \times L \times V$ is a set of edges, and $l : V \to L$ maps each node to a label, where $\boldsymbol{Con}$ denotes a countably infinite set of constants.*

In contrast to a directed edge-labeled graph, a heterogeneous graph encodes the type of a node as part of the node itself, rather than modeling types of nodes with a *Type* relation. Hence, one of the main advantages of a heterogeneous graph is that it allows for explicit distinction between nodes and the types of nodes. This is particularly useful when the type of each node is unique. However, a heterogeneous graph cannot express multiple types for a single node (i.e., many-to-one mapping).

Property graphs constitute a graph-structured data model that provides additional flexibility when modeling complex relations. Notably, a property graph allows for annotating more intricate details to each edge, such as the degree and major obtained by a person from a university. This is represented by a set of property–value pairs associated with edges. Unlike both directed edge-labeled graphs and heterogeneous graphs, annotating edges with additional properties is not straightforward. A property graph is defined as follows.

Definition 3 (Property graph [75])**.** *A property graph is a tuple $G = (V, E, L, P, U, e, l, p)$, where $V \subseteq \boldsymbol{Con}$ is a set of node ids, $E \subseteq \boldsymbol{Con}$ is a set of edge ids, $L \subseteq \boldsymbol{Con}$ is a set of labels, $P \subseteq \boldsymbol{Con}$ is a set of properties, $U \subseteq \boldsymbol{Con}$ is a set of values, $e : E \to V \times V$ maps an edge id to a pair of node ids, $l : V \cup E \to 2^L$ maps a node or edge id to a set of labels, and $p : V \cup E \to 2^{P \times U}$ maps a node or edge id to a set of property-value pairs.*

In contrast to directed edge-labeled graphs and heterogeneous graphs, a property graph allows a node or edge to have several values for a given property. Property graphs can be converted to/from directed edge-labeled graphs, and this process is called reification.

In summary, each of these three models has its advantages and disadvantages. Directed edge-labeled graphs offer a simple model, while property graphs provide a more flexible choice. The selection of a model typically depends on practical factors such as available implementations for different models, etc. For a detailed discussion, we recommend readers refer to [75].

2.1.2 *Graphs, Knowledge Graphs, and Ontologies*

We now introduce some popular instances of relational data that have been considered in the machine learning community, including (homogeneous) graphs, knowledge graphs, and ontologies.

Homogeneous graphs can be viewed as a special case of directed edge-labeled graphs or heterogeneous graphs in which there is only one type of edges and only one type of nodes. Specifically, a homogeneous graph is defined as follows.

Definition 4 (Homogeneous graph)**.** *A homogeneous graph is a tuple $G = (V, E)$, where V is a set of nodes, and $E \subseteq V \times V$ is a set of edges, with each edge connecting two nodes.*

A homogeneous graph is an *undirected* graph if all edges are bidirectional, meaning that if $(V_i, V_j) \in E$ holds, then $(V_j, V_i) \in E$ also holds. For example, in co-author networks, the co-authorship relationship is an *undirected* edge. Otherwise, the graph is called a *directed* graph. For example, in citation networks, the citation relationship is a directed edge.

Note that a homogeneous graph is the minimal model of graph-structured data. To allow for multiple node types, homogeneous graphs can be extended to single-relational graphs, defined as,

Definition 5 (Single-relational graph)**.** *A single-relational graph is a tuple $G = (V, E, L, l)$, where $V \subseteq \boldsymbol{Con}$ is a set of nodes, $L \subseteq \boldsymbol{Con}$ is a set of node labels, $E \subseteq V \times V$ is a set of edges, and $l : V \to L$ maps each node to a label, where $\boldsymbol{Con}$ denotes a countably infinite set of constants.*

Node classification. Single-relational graphs are useful for applications that involve only one type of edges but multiple types of nodes, such as node classification. Given a single-relational graph $G = (V, E, L, l)$ and the known labeling of nodes, *node classification* aims to classify the missing labeling of nodes based on the node features and the graph's structure.

$$f_{nc} : V \to L \tag{2.1}$$

For example, in a social network, node classification could involve predicting the communities of users based on their connections and shared content.

Link prediction is a task where the objective is to predict the likelihood of the existence of an edge (link) between two nodes in a graph or the specific edge label. Namely,

$$f_{lp} : V \times 0, 1 \quad \text{or} \quad f_{lp} : V \times V \to L \tag{2.2}$$

Link prediction is often used to predict missing or future connections in a graph. For instance, in a citation network, link prediction could involve predicting potential future collaborations between authors based on their previous co-authorships.

Graph classification involves assigning a label or category to an entire graph. The goal is to learn a model that can distinguish between different types or classes of graphs. Given a graph and a set of graph labels L_G, graph classification is defined as

$$f_{gc} : G \to L_G \tag{2.3}$$

For example, in chemical informatics, graph classification might be used to predict whether a molecular graph represents a toxic or non-toxic compound based on its structure.

Graph reconstruction aims to reconstruct a graph by preserving the pair-wise distance of nodes. One straightforward method is to minimize the graph distortion [9, 54] given by,

$$\frac{1}{|V|^2} \sum_{u,v} \left(\left(\frac{d\left(\mathbf{u}, \mathbf{v}\right)}{d_G(u, v)} \right)^2 - 1 \right)^2, \tag{2.4}$$

where $d(\mathbf{u}, \mathbf{v})$ is the distance function in the embedding space and $d_G(u, v)$ is the graph distance (the length of shortest path) between node u and v, $|V|$ is the number of nodes in graph. The objective function is to preserve all pairwise graph distances. Motivated by the fact that most of graphs are partially observable, we can also minimize an alternative loss function [97, 120] that preserves local graph distance, given by,

$$\mathcal{L}(\Theta) = \sum_{(u,v)\in\mathcal{D}} \log \frac{e^{-d(\mathbf{u},\mathbf{v})}}{\sum_{\mathbf{v}'\in\mathcal{E}(u)} \exp^{-d(\mathbf{u},\mathbf{v}')}}, \tag{2.5}$$

where $\mathcal{D}$ is the connected relations in the graph, $\mathcal{E}(u) = \{v|(u, v) \notin D \cup u\}$ is the set of negative examples for node u, $d(\mathbf{u}, \mathbf{v})$ is the distance function in the embedding space. For evaluation, we apply the mean average precision (mAP) to evaluate the graph reconstruction task. mAP is a local metric that measures the average proportion of the nearest points of a node which are actually its neighbors in the original graph. The mAP is defined as Eq. (2.6).

$$\text{mAP}(f) = \frac{1}{|V|} \sum_{u\in V} \frac{1}{\deg(u)} \sum_{i=1}^{|\mathcal{N}_u|} \frac{|\mathcal{N}_u \cap R_{u,v_i}|}{|R_{u,v_i}|}, \tag{2.6}$$

where f is the embedding function, $|V|$ is the number of nodes in graph, $\deg(u)$ is the degree of node u, $\mathcal{N}_u$ is the one-hop neighborhoods in the graph, $\mathbb{R}_{u,v}$ denotes whether two nodes u and v are connected.

Knowledge graphs can be viewed as an instance of directed edge-labeled graphs or multi-relational graphs in which nodes represent entities while edges represent relations between those entities. Like directed edge-labeled graphs, in the classical form of knowledge graphs, there is also no explicit distinction between entities and concepts.

Definition 6 (Knowledge graph)**.** *A knowledge graph is a tuple $G = (V, R, E)$, where V is a set of entities, R is a set of relations, and $E \subseteq V \times R \times V$ is a set of triples each describing a relationship between two entities.*

A knowledge graph serves as a structured representation of knowledge in a graph-based format. In this formalization, the components of the knowledge graph are:

- **Entities (V):** Entities represent the fundamental building blocks or objects within the knowledge graph. These could be people, places, events, or any distinguishable item of interest.

- **Relations (R):** Relations define the connections or associations between entities in the knowledge graph. These relationships capture the nuanced connections that exist in the real world, expressing how entities are related to one another.

- **Triples (E):** The set of triples $E \subseteq V \times R \times V$ describe relationships between entities. Each triple consists of a subject entity, a relation, and an object entity, collectively forming a statement about the knowledge contained in the graph.

Knowledge graphs provide a structured and semantically rich representation of information, allowing for the modeling of complex relationships and dependencies. It serves as a powerful tool for organizing, linking, and querying data in a way that reflects the inherent connections present in the real world.

Hyper-relational knowledge graph. In hyper-relational knowledge graph, each fact is a hyper-relational fact represented in the form of a primal triple coupled with a set of qualifiers. Namely,

Definition 7 (Hyper-relational knowledge graph)**.** *A hyper-relational knowledge graph is a tuple $G = (V, R, E)$, where V is a set of entities, R is a set of relations, and E is a set of hyper-relational fact.*

Definition 8 (Hyper-relational fact)**.** *Let $\mathcal{E}$ and $\mathcal{R}$ denote the sets of entities and relations, respectively. A hyper-relational fact $\mathcal{F}$ is a tuple $(\mathcal{T}, \mathcal{Q})$, where $\mathcal{T} = (h, r, t)$, $h, t \in \mathcal{E}, r \in \mathcal{R}$ is a primal triple and $\mathcal{Q} = \{(k_i : v_i)\}_{i=1}^{m}$ $k_i \in \mathcal{R}, v_i \in \mathcal{E}$ is a set of qualifiers. We call the number of involved entities in $\mathcal{F}$, i.e., $(m + 2)$, the arity of the fact.*

A hyper-relational fact reduces to a triple/binary fact when $m = 0$. When $m > 0$, each qualifier can be viewed as an auxiliary description that contextualizes or specializes the semantics of the primal triple. In typical open-world settings, facts with the same primal triple might have different numbers of qualifiers. To characterize this property, we introduce the concepts of partial fact and qualifier monotonicity in hyper-relational knowledge graphs.

Definition 9 (Partial fact [64])**.** *Given two facts $\mathcal{F}_1 = (\mathcal{T}, \mathcal{Q}_1)$ and $\mathcal{F}_2 = (\mathcal{T}, \mathcal{Q}_2)$ that share the same primal triple. We call $\mathcal{F}_1$ a partial fact of $\mathcal{F}_2$ iff $\mathcal{Q}_1 \subseteq \mathcal{Q}_2$.*

In this work, we follow the monotonicity assumption by restricting the model to respect the monotonicity property.[1] For this purpose, we consider the monotonicity of query and inference.

Definition 10 (Qualifier monotonicity)**.** *Let* $\mathrm{QA}(\cdot)$ *denote a query answering model taking a query and a knowledge graph as input and outputting the set of answer entities. Given any pair of queries* $q_1 = ((h, r, x?), \mathcal{Q}_1)$ *and* $q_2 = ((h, r, x?), \mathcal{Q}_2)$ *that share the same primal triple and* $\mathcal{Q}_1 \subseteq \mathcal{Q}_2$*, qualifier monotonicity is given iff,*

$$\mathrm{QA}(q_2; \mathrm{KG}) \subseteq \mathrm{QA}(q_1; \mathrm{KG}). \tag{2.7}$$

Qualifier monotonicity implies that attaching any qualifiers to a query does not enlarge the answer set of the possible tail entities, and inversely, removing the qualifiers from a query can only return more possible tail entities. This implies that if a fact is true, then all its partial facts must also be true (a.k.a. weakening of inference rule), i.e.,

$$(\mathcal{T}, \mathcal{Q}_1) \wedge (\mathcal{Q}_2 \subseteq \mathcal{Q}_1) \rightarrow (\mathcal{T}, \mathcal{Q}_2)\,. \tag{2.8}$$

Nested relational knowledge graphs. Given a knowledge graph G, which can be viewed an atomic factual knowledge graph, and each of the triple $(h, r, t) \in \mathcal{T}$ is referred to as an atomic triple. The nested triple and nested relational knowledge graph are defined as follows.

Definition 11 (Nested triple)**.** *Given an atomic factual knowledge graph* $G = (\mathcal{V}, \mathcal{R}, \mathcal{T})$*, a set of nested triples is defined by* $\widehat{\mathcal{T}} = \{\langle T_i, \widehat{r}, T_j\rangle : T_i, T_j \in \mathcal{T}, \widehat{r} \in \widehat{\mathcal{R}}\}$*, where* $\mathcal{T}$ *is the set of atomic triples and* $\widehat{\mathcal{R}}$ *is the set of nested relation names.*

Definition 12 (Nested relational knowledge graph)**.** *Given a knowledge graph* $G = (\mathcal{V}, \mathcal{R}, \mathcal{T})$*, a set of nested relation names* $\widehat{\mathcal{R}}$*, and a set of nested triples* $\widehat{\mathcal{T}}$ *defined on* G *and* $\widehat{\mathcal{R}}$*, a nested relational knowledge graph is defined as* $\widehat{G} = (\mathcal{V}, \mathcal{R}, \mathcal{T}, \widehat{\mathcal{R}}, \widehat{\mathcal{T}})$*.*

We can now define triple prediction and conditional link prediction [38] as follows.

Definition 13 (Triple prediction)**.** *Given a nested relational knowledge graph* $\widehat{G} = (\mathcal{V}, \mathcal{R}, \mathcal{T}, \widehat{\mathcal{R}}, \widehat{\mathcal{T}})$*, the triple prediction problem involves answer-*

1 Some kinds of qualifiers may represent semantically opaque contexts. For instance, ((*Crimea*, *belongs_to*, *Russia*), {(*said_by*, *Putin*)}) does not imply the primary triple and should therefore be excluded.

ing a query $\langle T_i, \widehat{r}, ?t\rangle$ *or* $\langle h?, \widehat{r}, T_j\rangle$ *with* $T_i, T_j \in \mathcal{T}$ *and* $\widehat{r} \in \widehat{\mathcal{R}}$*, where the variable* $?h$ *or* $?t$ *needs to be bounded to an atomic triple within* $\widehat{G}$*.*

Definition 14 (Conditional link prediction)**.** *Given a nested relational knowledge graph* $\widehat{G} = (\mathcal{V}, \mathcal{R}, \mathcal{T}, \widehat{\mathcal{R}}, \widehat{\mathcal{T}})$*, let* $T_i = (h_i, r_i, t_i)$ *and* $T_j = (h_j, r_j, t_j)$*. The conditional link prediction problem involves queries* $\langle T_i, \widehat{r}, (h_j, r_j, ?)\rangle$*,* $\langle T_i, \widehat{r}, (?, r_j, t_j)\rangle$*,* $\langle (h_i, r_i, ?), \widehat{r}, T_j\rangle$*, or* $\langle (?, r_i, t_i), \widehat{r}, T_j\rangle$*, where the variables need to be bound to entities within* $\widehat{G}$*.*

Description Logics. Description Logics (DLs) are a family of formal knowledge representation languages used to represent and reason about the semantics of information in a structured and logical manner. They are a subset of first-order logic and are designed to express and reason about concepts, relationships, and individuals in a domain. DLs provide a formal foundation for knowledge representation and are widely used in the field of artificial intelligence, particularly in areas such as ontology engineering and knowledge-based systems.

1. **Concepts:** Represent abstract classes or sets of individuals. Concepts are used to classify and categorize entities in a domain.

2. **Roles (Properties):** Describe relationships between individuals or between individuals and concepts. Roles can be used to represent attributes, roles, or associations between entities.

3. **Individuals:** Represent specific instances or objects in the domain. Individuals are members of concepts and can be related to each other through roles.

4. **Axioms:** Specify constraints and relationships between concepts and roles, providing the logical foundation for reasoning in the knowledge base.

Definition 15 (DL knowledge base)**.** *DL knowledge base* K *is defined as a tuple* $(\mathrm{A}, \mathrm{T}, \mathrm{R})$*, where* A *is the A-Box: a set of assertional axioms;* T *is the T-Box: a set of class (aka concept/terminological) axioms; and* R *is the R-Box: a set of relation (aka property/role) axioms.*

A Description Logic knowledge base is a structured collection of information using Description Logics. It typically consists of:

1. **TBox (Terminological Box):** Contains the terminology or concept hierarchy of the domain. It defines the relationships between concepts and includes axioms that express constraints on the concepts.

Table 2.1: Syntax and semantic of $\mathcal{EL}^{++}$ (role inclusions and concrete domains are omitted).

	Name	Syntax	Semantics
Constructors	Top concept	$\top$	$\Delta^{\mathcal{I}}$
	Bottom concept	$\bot$	$\emptyset$
	Nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
	Conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
	Existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}$ $(x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
ABox	Concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
	Role assertion	$r(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$
TBox	Concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

2. **ABox (Assertional Box):** Contains assertions about individuals in the domain. It specifies which individuals belong to which concepts and how they are related through roles.

3. **RBox (Role Box):** Contains information about roles and their properties, including relationships between roles.

The combination of the TBox, ABox, and RBox provides a comprehensive representation of the knowledge about a domain in a formal and logical manner. Description Logics and their associated knowledge bases are commonly used in various applications, including semantic web technologies, ontology development, knowledge-based systems, and the representation of domain-specific knowledge in AI systems.

Description logic $\mathcal{EL}^{++}$. The DL $\mathcal{EL}^{++}$ underlies multiple biomedical KBs like GALEN [134] and the Gene Ontology [41]. Formally, the syntax of $\mathcal{EL}^{++}$ is built up from a set N_I of *individual names*, N_C of *concept names* and N_R of *role names* (also called *relations*) using the constructors shown in Table 4.2, where N_I, N_C and N_R are pairwise disjoint. Strictly speaking, $\mathcal{EL}^{++}$ also allows for concrete domains, but we do not make use of them here.

The semantics of $\mathcal{EL}^{++}$ is defined by *interpretations* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where the domain $\Delta^{\mathcal{I}}$ is a non-empty set and $\cdot^{\mathcal{I}}$ is a mapping that associates every individual with an element in $\Delta^{\mathcal{I}}$, every concept name with a subset of $\Delta^{\mathcal{I}}$, and every relation name with a relation over $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. An *interpretation* is satisfied if it satisfies the corresponding semantic conditions. The syntax and the corresponding semantics (i.e., interpretation of concept expressions) of $\mathcal{EL}^{++}$ are summarized in Table 4.2.

An $\mathcal{EL}^{++}$ KB $(\mathcal{A}, \mathcal{T})$ consists of an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$. The *ABox* is a set of *concept assertions* $(C(a))$ and *role assertions* $(r(a,b))$, where C

is a concept, r is a relation, and a, b are individuals. The TBox is a set of *concept inclusions* of the form $C \sqsubseteq D$. Intuitively, the ABox contains instance-level information (e.g. $\mathsf{Person}(\mathsf{John})$), $\mathsf{isFatherOf}(\mathsf{John}, \mathsf{Peter})$), while the TBox contains information about concepts (e.g. $\mathsf{Parent} \sqsubseteq \mathsf{Person}$). Every $\mathcal{EL}^{++}$ KB can be transformed such that every TBox statement has the form $C_1 \sqsubseteq D$, $C_1 \sqcap C_2 \sqsubseteq D$, $C_1 \sqsubseteq \exists r.C_2$, $\exists r.C_1 \sqsubseteq D$, where C_1, C_2, D can be the top concept, concept names or nominals and D can also be the bottom concept [7]. The normalized KB can be computed in linear time by introducing new concept names for complex concept expressions and is a conservative extension of the original KB, i.e., every model of the normalized KB is a model of the original KB and every model of the original KB can be extended to be a model of the normalized KB [7].

2.2 RELATIONAL REPRESENTATION LEARNING

2.2.1 *Graph Representation Learning*

Given a graph $G = (V, E)$, the goal of graph representation learning is to learn a node mapping function $f : V \rightarrow \mathbb{R}^d$, which project each node into low dimensional vectors in space $\mathbb{R}^d$, where $d \ll |V|$, while preserving the graph structure and node attributes. The learned representation $\mathbf{H}$ can be applied to downstream tasks such as node classification, link prediction, and graph reconstruction.

2.2.2 *Graph Neural Networks.*

Graph neural networks (GNNs) are a class of neural networks designed to operate on graph-structured data. GNNs are particularly effective for tasks where relationships or dependencies between data points can be naturally represented as a graph. A more specific type of GNN is known as Graph convolutional networks (GCNs). Given a graph G, a GNN processes node features X and adjacency information A to learn a mapping $f : \mathbb{R}^{|V| \times d} \times \mathbb{R}^{|V| \times |V|} \rightarrow \mathbb{R}^{|V| \times o}$, where d is the input feature dimension, o is the output dimension, and $|V|$ is the number of nodes.

The GNN processes information in an iterative manner through layers. At each layer, the hidden representations $H^{(l+1)}$ are computed as a function of the previous layer's representations $H^{(l)}$:

$$H^{(l+1)} = \sigma\left(A \cdot H^{(l)} \cdot W^{(l)} + b^{(l)}\right)$$

where $W^{(l)}$ and $b^{(l)}$ are learnable parameters for layer l, $\cdot$ denotes matrix multiplication, and σ is a non-linear activation function.

GNNs can be applied to a variety of tasks, such as node classification, link prediction, and graph classification, making them versatile for learning from graph-structured data.

2.2.3 *Knowledge Graph Embeddings.*

A knowledge graph embedding is a function $f : V \cup R \to \mathbb{R}^d$ that projects entities and relations into a continuous vector space of dimension d. Formally:

$$f(e_i) \in \mathbb{R}^d, \quad \forall e_i \in V$$
$$f(r_j) \in \mathbb{R}^d, \quad \forall r_j \in R$$

The scoring function $s : V \times R \times V \to \mathbb{R}$ evaluates the compatibility of a triple (h, r, t) in the knowledge graph. A common scoring function, exemplified by TransE, is defined as:

$$s(h, r, t) = \text{dist}(\mathbf{f}(h) + \mathbf{f}(r), \mathbf{f}(t))$$

where $\mathbf{f}(h)$, $\mathbf{f}(r)$, and $\mathbf{f}(t)$ are the embeddings of the head entity, relation, and tail entity, respectively. The function $\text{dist}(\cdot, \cdot)$ measures the dissimilarity between two vectors, often represented by the Euclidean distance:

$$\text{dist}(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_2$$

The objective of knowledge graph embeddings is to learn vectors that minimize the scores for true triples $(h, r, t) \in E$ and simultaneously maximize the scores for negative samples, enhancing the model's ability to capture and predict complex relationships within the knowledge graph.

Two common evaluation metrics for knowledge graph embeddings are Mean Reciprocal Rank (MRR) and Hit@k. Mean Reciprocal Rank (MRR is a metric used to evaluate the ability of a model to rank the correct entity (or relationship) higher in the list of candidates. It is particularly relevant for tasks like link prediction. MRR is calculated as the average of the reciprocal ranks of the correct entities:

$$\text{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\text{rank}_i}$$

Where: N is the total number of test instances. rank_i is the rank of the correct entity for the i-th test instance. The higher the MRR, the better the model is at correctly ranking the true entities higher in the list.

Hit@k is another metric that measures the proportion of test instances where the correct entity (or relationship) appears in the top k ranked candidates. This metric is also used for tasks like link prediction. The formula for Hit@k is given by:

$$\text{Hit@k} = \frac{1}{N}\sum_{i=1}^{N}\text{hit}_i$$

Where N is the total number of test instances. hit_i is a binary indicator function, which equals 1 if the correct entity is among the top k ranked candidates for the i_{th} test instance, and 0 otherwise. Hit@k provides insights into the model's ability to retrieve the correct entities within the top k positions.

2.3 GEOMETRIC EMBEDDINGS.

Vector embeddings map objects to a low-dimensional vector space. Vector embeddings have been developed to learn representations of objects that allow for distinguishing relevant differences and ignoring irrelevant variations between objects. When vector embeddings are used to embed structural/relational data, they fail to represent key properties of relational data that cannot be easily modeled in a plain, low-dimensional vector space. For example, relational data may have been defined by applying set operators such as set inclusion and exclusion [187], logical operations such as negation [136], or they may exhibit relational patterns like the symmetry of relations [1] and structural patterns (e.g., trees and cycles) [26, 189].

Going beyond plain vector embeddings, *geometric relational embeddings* replace the vector representations with more advanced geometric objects, such as convex regions [92, 135, 187], density functions [136, 166], elements of hyperbolic manifolds [26], and their combinations [152]. Geometric relational embeddings provide a rich geometric inductive bias for modeling relational/structured data. For example, embedding objects as convex regions allows for modeling not only similarity but also set-based and logical operators such as set inclusion, set intersection [187] and logical negation [202] while representing data on non-Euclidean manifolds allows for capturing complex structural patterns, such as representing hierarchies in hyperbolic space [26]. We group them into three lines of works.

2.3.1 *Distribution-based Embeddings*

Probability distributions provide a rich geometry of the latent space. Their density can be interpreted as soft regions and it allows us to model uncertainty, asymmetry, set inclusion/exclusion, entailment, and so on.

Gaussian embeddings. Word2Gauss [163] maps words to multi-dimensional Gaussian distributions over a latent embedding space such that the linguistic properties of the words are captured by the relationships between the distributions. A Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is parameterized by a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$ (usually a diagonal matrix for the sake of computing efficiency). The model can be optimized by an energy function $-E\left(\mathcal{N}_i, \mathcal{N}_j\right)$ that is equivalent to the KL-divergence $D_{\mathrm{KL}}\left(\mathcal{N}_j \| \mathcal{N}_i\right)$ defined as

$$D_{\mathrm{KL}}\left(\mathcal{N}_j \| \mathcal{N}_i\right) = \int_{x \in \mathbb{R}^n} \mathcal{N}\left(x; \mu_i, \Sigma_i\right) \log \frac{\mathcal{N}\left(x; \mu_j, \Sigma_j\right)}{\mathcal{N}\left(x; \mu_i, \Sigma_i\right)} dx. \tag{2.9}$$

KG2E [73] extends this idea to knowledge graph embedding by mapping entities and relations as Gaussians. Given a fact (h, r, t), the scoring function is defined as $f(h, r, t) = \frac{1}{2}\left(\mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}_h, \mathcal{N}_r\right) + \mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}_r, \mathcal{N}_t\right)\right)$. The covariances of entity and relation embeddings allow us to model uncertainties in knowledge graphs. While modeling the scores of triples as KL-divergence allows us to capture asymmetry. TransG [181] generalizes KG2E to a Gaussian mixture distribution to deal with multiple relation semantics revealed by the entity pairs. For example, the relation HasPart has at least two latent semantics: composition-related as (*Table*, *HasPart*, *Leg*) and location-related as (*Atlantics*, *HasPart*, *NewYorkBay*).

2.3.2 *Region-based Embeddings*

Mapping data as convex regions is inspired by the Venn diagram. Region-based embeddings nicely model the set theory that can be used to capture uncertainty [33], logical rules [1], transitive closure [162], logical operations [135], etc. Several convex regions have been explored including balls, boxes, and cones.

Ball embeddings associate each object w with an n-dimensional ball $\mathbb{B}_w(\mathbf{c}_w, r_w)$, where $\mathbf{c}_w$ and r_w are the central point and its radius of the ball, respectively. A ball is defined as the set of vectors whose Euclidean distance to $\mathbf{c}_w$ is less than r_w: $\mathbb{B}(\mathbf{c}_w, r_w) = \{\mathbf{p} | \|\mathbf{c}_w - \mathbf{p}\| < r_w\}$. In ElEm [92], each concept in $\mathcal{EL}^{++}$ ontologies is represented as an open n-ball, and subsumption relations between concepts are modeled as ball containment. This explicit modeling of subsumption structure leads to significant improvements in predicting human protein-protein interactions. [50] represents categories with n-balls while considering tree-structured category information. By embedding subordinate relations between categories as ball containment, it shows promising results on NLP tasks compared to conventional word embeddings.

Box embeddings represent objects with d dimensional rectangles, i.e., a Cartesian product of d closed intervals denoted by $\prod_{i=1}^{d} \left[x_i^{\mathrm{m}}, x_i^{\mathrm{M}}\right]$, where $x_i^{\mathrm{m}} < x_i^{\mathrm{M}}$ and x_i^{m} and x_i^{M} are lower-left and top-right coordinates of boxes [162]. Box embeddings capture anticorrelation and disjoint concepts better than order embeddings. Joint Box [131] improves box embeddings' ability to express multiple hierarchical relations (e.g., hypernymy and meronymy) and proposes joint embedding of these relations in the same subspace to enhance entity characterization, resulting in improved performance. BoxE [1] proposes embedding entities and relations as points and boxes in knowledge bases, improving expressivity by modeling rich logic hierarchies in higher-arity relations. BEUrRE [33] is similar to BoxE but it differs in embedding entities and relations as boxes and affine transformations, respectively, which enables better modeling marginal and joint entity probabilities. Query2Box [135] shares BoxE's idea, embedding queries as boxes, with answer entities inside, to support a range of querying operations, including disjunctions, in large-scale knowledge graphs. For multi-label classification, MBM [130] proposes a multi-label box model that marries neural networks with [162], where labels are embedded as boxes so that label-label relations can be easily modeled.

Cone embedding was first proposed in Order embedding (OE) [159] to represent a partial ordered set (poset). However, this method is restricted to axis-parallel cones and it only captures positive correlations as any two axis-parallel cones intersect at infinite. Recent cone

embeddings formulate cones with additional angle parameters. As one of the main advantages, angular cone embedding has been used to model the negation operator since the angular cone is closed under negation. For example, negation can be modeled as the polarity of a cone as used in ALC ontology [126] or the complement of a cone as used in ConE for logical queries [202].

2.3.3 *Manifold/Differential Geometry.*

A smooth manifold $(\mathcal{M}, g)$ is defined as a smooth manifold equipped with a metric g, which induces a scalar product for each point $\mathbf{x} \in \mathcal{M}$ on the *tangent* space $\mathcal{T}_{\mathbf{x}}\mathcal{M}$. For each point $\mathbf{x}$, the metric tensor is defined as $g_{\mathbf{x}} : \mathcal{T}_{\mathbf{x}}\mathcal{M} \times \mathcal{T}_{\mathbf{x}}\mathcal{M} \to \mathbb{R}$, which varies smoothly with the point $\mathbf{x}$ and induces geometric notions such as geodesic distances, angles and curvatures.

Riemannian manifold. A Riemannian Manifold $(\mathcal{M}, g)$ is a manifold $\mathcal{M}$ equiped with a Riemannian metric g. The Riemannian metric is positive definite that means $\forall \boldsymbol{\xi} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}, g_{\mathbf{x}}(\boldsymbol{\xi}, \boldsymbol{\xi}) > 0$ iff $\boldsymbol{\xi} \neq \mathbf{0}$. We denote the *curvature* of a manifold as K and different notions of *curvature* quantify the ways in which a surface is locally curved around a point. There are three different types of constant curvature Riemannian manifold $\mathcal{M}$ with respect to the sign of curvature: hyperboloid $\mathbb{H}_K$ (negative curvature), hypersphere $\mathbb{S}_K$ (positive curvature) and Euclidean space $\mathbb{R}$ (zero curvature).

$$\mathcal{M} = \begin{cases} \mathbb{S}_K^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \langle \mathbf{x}, \mathbf{x} \rangle_2 = 1/K\}, & \text{if } K > 0 \\ \mathbb{E}^n = \mathbb{R}^n, & \text{if } K = 0 \\ \mathbb{H}_K^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = 1/K\}, & \text{if } K < 0 \end{cases} \tag{2.10}$$

where $\langle \cdot, \cdot \rangle_2$ is the standard Euclidean inner product, and $\langle \cdot, \cdot \rangle_{\mathcal{L}}$ is the Lorentz inner product. For $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^{n+1}$, the Lorentz inner product $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}}$ is defined as follows.

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} = -x_1 y_1 + \sum_{i=2}^{n+1} x_i y_i, \tag{2.11}$$

Product manifolds. Given a sequence of smooth manifolds $M_1, M_2, ..., M_k$, the product manifold is defined as the Cartesian product $M = M_1 \times M_2 \times ... \times M_k$, with the metric tensor $g(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{k} g_i(u_i, v_i)$. Correspondingly, the points $\mathbf{x} \in \mathcal{M}$ can be represented as their coordinates $\mathbf{x} = (x_1, ..., x_k), x_i \in \mathcal{M}_i$.

Geodesics. A smooth curve γ of minimal length between two points $\mathbf{x}$ and $\mathbf{y}$ is called a geodesic, which can be seen as the generalization of a straight-line in Euclidean space. Formally, the geodesic is defined as $\gamma_{\mathbf{x}\to\boldsymbol{\xi}}(\tau) : I \to \mathcal{M}$ from an interval $I = [0, 1]$ of the reals to the metric space $\mathcal{M}$, which maps a real value $\tau \in I$ to a point on the manifold $\mathcal{M}$ with initial velocity $\boldsymbol{\xi} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$. By the means of geodesic, the exponential map can be defined as $\exp_{\mathbf{x}}(\boldsymbol{\xi}) = \gamma_{\mathbf{x}\to\boldsymbol{\xi}}(1)$.

Exponential and logarithmic map in Riemannian manifold. The connections between manifolds and *tangent* space are established by the differentiable exponential map and logarithmic map. The exponential map $\exp_{\mathbf{x}}$ at $\mathbf{x}$ gives a way to project back a vector $\mathbf{v} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$ to a point $\exp_{\mathbf{x}}(\mathbf{v}) \in \mathcal{M}$ on the manifold. And the logarithmic map $\log_{\mathbf{x}} : \mathcal{M} \to \mathcal{T}_{\mathbf{x}}\mathcal{M}$ is defined as the inverse of the exponential map. Common realizations of Riemannian manifolds are hypersphere $\mathbb{S}_K$, the Euclidean space $\mathbb{R}$, and the hyperboloid $\mathbb{H}_K$. And we summarize *expmap* and *logmap* operations in these three spaces compactly in Table 2.2.

Table 2.2: Summary of *expmap* and *logmap* in $\mathbb{R}$, $\mathbb{S}_K$ and $\mathbb{H}_K$.

Operations	
expmap in $\mathbb{R}$	$\exp_{\mathbf{x}}(\mathbf{v}) = \mathbf{x} + \mathbf{v}$
logmap in $\mathbb{R}$	$\log_{\mathbf{x}}(\mathbf{y}) = \mathbf{y} - \mathbf{x}$
expmap in $\mathbb{S}_K$	$\exp^K_{\mathbf{x}}(\mathbf{v}) = \cos\left(\sqrt{\lvert K\rvert}\lVert\mathbf{v}\rVert_2\right) + \sin\left(\sqrt{\lvert K\rvert}\lVert\mathbf{v}\rVert_2\right)\frac{\mathbf{v}}{\sqrt{\lvert K\rvert}\lVert\mathbf{v}\rVert_2}$
logmap in $\mathbb{S}_K$	$\log^K_{\mathbf{x}}(\mathbf{y}) = \frac{\cos^{-1}(K\langle\mathbf{x},\mathbf{y}\rangle_2)}{\sqrt{1-K^2\langle\mathbf{x},\mathbf{y}\rangle_2^2}}\left(\mathbf{y} - K\langle\mathbf{x},\mathbf{y}\rangle_2\mathbf{x}\right)$
expmap in $\mathbb{H}_K$	$\exp^K_{\mathbf{x}}(\mathbf{v}) = \cosh\left(\sqrt{\lvert K\rvert}\lVert\mathbf{v}\rVert_{\mathcal{L}}\right)\mathbf{x} + \sinh\left(\sqrt{\lvert K\rvert}\lVert\mathbf{v}\rVert_{\mathcal{L}}\right)\frac{\mathbf{v}}{\sqrt{\lvert K\rvert}\lVert\mathbf{v}\rVert_{\mathcal{L}}}$
logmap in $\mathbb{H}_K$	$\log^K_{\mathbf{x}}(\mathbf{y}) = \frac{\cosh^{-1}(K\langle\mathbf{x},\mathbf{y}\rangle_{\mathcal{L}})}{\sqrt{K^2\langle\mathbf{x},\mathbf{y}\rangle_{\mathcal{L}}^2-1}}\left(\mathbf{y} - K\langle\mathbf{x},\mathbf{y}\rangle_{\mathcal{L}}\mathbf{x}\right)$

2.3.4 *Riemannian manifold embeddings*

Hyperbolic Embeddings The Poincaré ball $\left(\mathbb{D}^n, g^{\mathbb{D}}\right)$ is one of the models of hyperbolic geometry that is very suitable for representing hierarchies due to its exponentially growing volume [3]. The Poincaré ball is defined as an open n-ball $\mathbb{D}^n = \{\mathbf{x} \in \mathbb{R}^n : \lVert\mathbf{x}\rVert < 1\}$ equipped with a Riemannian metric $g^{\mathbb{D}}_{\mathbf{x}} = \lambda^2_{\mathbf{x}} g^E$, where $\lambda_{\mathbf{x}} = \frac{2}{1-\lVert\mathbf{x}\rVert^2}$, $g^E = \mathbf{I}_n$ is the Euclidean metric tensor, $\lambda_{\mathbf{x}}$ is the *conformal factor*, and $\lVert\cdot\rVert^2$ denotes the L^2 norm in Euclidean space. The distance between two points $\mathbf{x}, \mathbf{y} \in \mathbb{D}^n$ can be defined by

$$d_{\mathbb{D}}(\mathbf{x}, \mathbf{y}) = \cosh^{-1}\left(1 + 2\frac{\lVert\mathbf{x} - \mathbf{y}\rVert^2}{\left(1 - \lVert\mathbf{x}\rVert^2\right)\left(1 - \lVert\mathbf{y}\rVert^2\right)}\right) \tag{2.12}$$

The Poincaré ball has been used for modeling hierarchical structures [175]. MuRP [11] models multi-relational data by transforming entity embeddings by Möbius matrix-vector multiplication and addition. To capture both hierarchy and logical patterns e.g., symmetry and composition, AttH [26] models relation transformation by rotation/reflection and also embeds entities on the Poincaré ball. FieldH and FieldP [119] embed entities on trajectories that lie on hyperbolic manifolds namely Hyperboloid and Poincaré ball to capture heterogeneous patterns formed by a single relation (e.g., a combination of loop and path) besides hierarchy and logical patterns. [128] embeds nodes on the extended Poincaré Ball and polar coordinate system to address numerical issues when dealing with the embedding of neighboring nodes on the boundary of the ball on previous models. HyboNet [32] proposes a fully hyperbolic method that uses Lorentz transformations to overcome the incapability of previous hyperbolic methods of fully exploiting the advantages of hyperbolic space.

Spherical embeddings A spherical manifold $\mathcal{M} = \{x \in \mathbb{R}^d | \ \|x\| = 1\}$ has been used as embedding space in several works to model loops in graphs. MuRS [171] models relations as linear transformations on the tangent space of spherical manifold, together with spherical distance in score formulation. The spherical distance between the transformed head and tail is used to formulate the score function. A more sophisticated approach is FiledP [119] in which entity spaces are trajectories on a sphere based on relations to model more complex patterns compared to MuRS. 5*E [117] embeds entities on flows on the product space of a complex projective line which is called the Riemann sphere. The projective transformation then covers translating, rotation, homothety, reflection, and inversion which are powerful for modeling various structures and patterns such as combination of loop and loop, loop and path, and two connected path structures.

Mixed manifold embeddings Relational data exhibit heterogeneous structures and patterns where each class of patterns and structures can be modeled efficiently by a particular manifold. Therefore, combining several manifolds for embedding is beneficial and addressed in the literature. MuRMP [171] extends MuRP [11] to a product space of spherical, Euclidean, and hyperbolic manifolds. GIE [23] improves the previous mixed models by computing the geometric interaction on tangent space of Euclidean, spherical and hyperbolic manifolds via an attention mechanism to emphasize on the most relevant geometry and then projects back the obtained vector to the hyperbolic manifold for score calculation. DGS [79] targets the same problem of heterogeneity of structures and utilizes the hyperbolic space, spherical space, and intersecting space in a unified framework for learning embeddings of different portions of two-view knowledge graphs (ontology and in-

stance levels) in different geometric space. DyERNIE [71] is a temporal knowledge graph embedding on product manifolds that models the evolution of entities over time on tangent space of product manifolds. While the reviewed works provide separate spherical, Euclidean, and hyperbolic manifolds and act on Riemannian manifolds, In this model, relations are modeled by the pseudo orthogonal transformation which is decomposed into cosine-sine forms covering hyperbolic/circular rotation and reflection. This model can capture heterogeneous structures and logical patterns.

2.3.5 *Pseudo-Riemannian Manifolds*

A pseudo-Riemannian manifold [125] $(\mathcal{M}, g)$ is a smooth manifold $\mathcal{M}$ equipped with a nondegenerate and indefinite metric tensor g. Nondegeneracy means that for a given $\boldsymbol{\xi} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$, for any $\boldsymbol{\zeta} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$ we have $g_{\mathbf{x}}(\boldsymbol{\xi}, \boldsymbol{\zeta}) = 0$, then $\boldsymbol{\xi} = \mathbf{0}$. The metric tensor g induces a scalar product on the *tangent* space $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ for each point $\mathbf{x} \in \mathcal{M}$ such that $g_{\mathbf{x}} : \mathcal{T}_{\mathbf{x}}\mathcal{M} \times \mathcal{T}_{\mathbf{x}}\mathcal{M} \to \mathbb{R}$, where the *tangent* space $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ can be seen as the first order local approximation of $\mathcal{M}$ around point $\mathbf{x}$. The elements of $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ are called tangent vectors. *Indefinity* means that the metric tensor could be of arbitrary signs. A principal special case is the Riemannian geometry, where the metric tensor is positive definite (i.e. $\forall \boldsymbol{\xi} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}, g_{\mathbf{x}}(\boldsymbol{\xi}, \boldsymbol{\xi}) > 0$ iff $\boldsymbol{\xi} \neq \mathbf{0}$).

2.3.5.1 *Pseudo-hyperboloid*

By analogy with hyperboloid and sphere in Euclidean space. Pseudo-hyperboloids are defined as the submanifolds in the ambient pseudo-Euclidean space $\mathbb{R}^{s,t+1}$ with the dimensionality of $d = s + t + 1$ that uses the scalar product as $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^{s,t+1}, \langle \mathbf{x}, \mathbf{y} \rangle_t = -\sum_{i=0}^{t} x_i y_i + \sum_{j=t+1}^{s+t} x_j y_j$. The scalar product induces a norm $\|\mathbf{x}\|_t^2 = \langle \mathbf{x}, \mathbf{x} \rangle_t$ that can be used to define a pseudo-hyperboloid $\mathcal{Q}_\beta^{s,t} = \{\mathbf{x} = (x_0, x_1, \cdots, x_{s+t})^\top \in \mathbb{R}^{s,t+1} : \|\mathbf{x}\|_t^2 = \beta\}$, where β is a nonzero real number parameter of curvature. $\mathcal{Q}_\beta^{s,t}$ is called *pseudo-sphere* when $\beta > 0$ and *pseudo-hyperboloid* when $\beta < 0$. Since $\mathcal{Q}_\beta^{s,t}$ is interchangeable with $\mathcal{Q}_{-\beta}^{t+1,s-1}$, we consider the pseudo-hyperbololid here. Following the terminology of special relativity, a point in $\mathcal{Q}_\beta^{s,t}$ can be interpreted as an event [150], where the first $t + 1$ dimensions are time dimensions and the last s dimensions are space dimensions. Hyperbolic $\mathbb{H}$ and spherical $\mathbb{S}$ manifolds can be defined as the special cases of pseudo-hyperboloids by setting all time dimensions except one to be zero and setting all space dimensions to be zero, respectively, i.e. $\mathbb{H}_\beta = \mathcal{Q}_\beta^{s,1}, \mathbb{S}_{-\beta} = \mathcal{Q}_\beta^{0,t}$.

2.3.5.2 *Geodesic tools of pseudo-hyperboloid*

Geodesic. A generalization of a *straight-line* in the Euclidean space to a manifold is called the *geodesic* [54, 178]. Formally, a geodesic γ is defined as a constant speed curve $\gamma : \tau \mapsto \gamma(\tau) \in \mathcal{M}, \tau \in [0,1]$ joining two points $\mathbf{x}, \mathbf{y} \in \mathcal{M}$ that minimizes the length, where the length of a curve is given by $L(\gamma) = \int_0^1 \sqrt{\left\| \frac{d}{dt}\gamma(\tau) \right\|_{\gamma(\tau)}} dt$. The geodesic holds that $\gamma^* = \arg\min_\gamma L(\gamma)$, such that $\gamma(0) = \boldsymbol{x}, \gamma(1) = \boldsymbol{y}$, and $\left\| \frac{d}{d\tau}\gamma(\tau) \right\|_{\gamma(\tau)} = 1$.

By the means of the geodesic, the distance between $\mathbf{x}, \mathbf{y} \in \mathcal{Q}_\beta^{s,t}$ is defined as the arc length of geodesic $\gamma(\tau)$.

Exponential and logarithmic maps. The connections between manifolds and *tangent* space are established by the differentiable exponential map and logarithmic map. The exponential map at $\mathbf{x}$ is defined as $\exp_{\mathbf{x}}(\boldsymbol{\xi}) = \gamma(1)$, which gives a way to project a vector $\boldsymbol{\xi} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$ to a point $\exp_{\mathbf{x}}(\boldsymbol{\xi}) \in \mathcal{M}$ on the manifold. The logarithmic map $\log_{\mathbf{x}} : \mathcal{M} \to \mathcal{T}_{\mathbf{x}}\mathcal{M}$ is defined as the inverse of the exponential map (i.e. $\log_{\mathbf{x}} = \exp_{\mathbf{x}}^{-1}$). Note that since $\mathcal{Q}_\beta^{s,t}$ is a geodesically complete manifold, the domain of the exponential map $\mathcal{D}_x$ is hence defined on the entire tangent space, i.e. $\mathcal{D}_x = \mathcal{T}_{\mathbf{x}}\mathcal{Q}_\beta^{s,t}$. However, as we will explain later, the logarithmic map $\log_{\mathbf{x}}(\mathbf{y})$ is only defined when there exists a a length-minimizing geodesic between $\mathbf{x}, \mathbf{y} \in \mathcal{Q}_\beta^{s,t}$.

Parallel transport. Given the geodesic $\gamma(\tau)$ on $\mathcal{Q}_\beta^{s,t}$ passing through $\mathbf{x} \in \mathcal{Q}_\beta^{s,t}$ with the tangent direction $\boldsymbol{\xi} \in \mathcal{T}_{\mathbf{x}}\mathcal{Q}_\beta^{s,t}$, the parallel transport of $\boldsymbol{\zeta} \in \mathcal{T}_{\mathbf{x}}\mathcal{Q}_\beta^{s,t}$ is defined as Eq. (2.13), where $\boldsymbol{\xi} = \log_{\mathbf{x}}(\mathbf{y})$.

$$P^{\beta}_{\mathbf{x}\to\mathbf{y}}(\boldsymbol{\zeta}) = \begin{cases} \frac{\langle \boldsymbol{\zeta}, \boldsymbol{\xi} \rangle}{\|\boldsymbol{\xi}\|} \left[\mathbf{x} \sinh(\tau\|\boldsymbol{\xi}\|) + \frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|} \cosh(\tau\|\boldsymbol{\xi}\|) \right] + \left(\boldsymbol{\zeta} - \frac{\langle \boldsymbol{\zeta}, \boldsymbol{\xi} \rangle}{\|\boldsymbol{\xi}\|^2} \boldsymbol{\xi} \right), & \text{if } \langle \boldsymbol{\xi}, \boldsymbol{\xi} \rangle_t > 0 \\ \frac{\langle \boldsymbol{\zeta}, \boldsymbol{\xi} \rangle}{\|\boldsymbol{\xi}\|} \left[\mathbf{x} \sin(\tau\|\boldsymbol{\xi}\|) - \frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|} \cos(\tau\|\boldsymbol{\xi}\|) \right] + \left(\boldsymbol{\zeta} + \frac{\langle \boldsymbol{\zeta}, \boldsymbol{\xi} \rangle}{\|\boldsymbol{\xi}\|^2} \boldsymbol{\xi} \right), & \text{if } \langle \boldsymbol{\xi}, \boldsymbol{\xi} \rangle_t < 0 \\ \langle \boldsymbol{\zeta}, \boldsymbol{\xi} \rangle \left(\tau \mathbf{x} + \frac{1}{2}\tau^2 \boldsymbol{\xi} \right) + \boldsymbol{\zeta}, & \textit{otherwise} \end{cases} \tag{2.13}$$

Distance. By the means of the geodesic, the distance between $\mathbf{x}, \mathbf{y} \in \mathcal{Q}_\beta^{s,t}$ is defined as the arc length of geodesic $\gamma(\tau)$, which can be formulated by using *logmap*, given by Eq. (2.14).

$$\mathrm{D}_\gamma(\mathbf{x}, \mathbf{y}) = \sqrt{\left| \|\log_{\mathbf{x}}(\mathbf{y})\|_t^2 \right|}. \tag{2.14}$$

Geodesical connectedness. A pseudo-Riemannian manifold $\mathcal{M}$ is *connected* iff any two points of $\mathcal{M}$ can be joined by a piecewise (broken) geodesic with each piece being a smooth geodesic. The manifold is

geodesically connected (or *g-connected*) iff any two points can be smoothly connected by a geodesic, where the two points are called *g-connected*, otherwise called *g-disconnected*. Different from Riemannian manifolds in which the geodesical completeness implies the g-connectedness (Hopf–Rinow theorem [146]), pseudo-hyperboloid is a geodesically complete but not *g-connected* manifold where there exist points that cannot be smoothly connected by a geodesic [57]. Formally, in the pseudo-hyperboloid, two points $\mathbf{x}, \mathbf{y} \in \mathcal{Q}_\beta^{s,t}$ are *g-connected* iff $\langle \mathbf{x}, \mathbf{y} \rangle_t < |\beta|$. The set of *g-connected* points of $\mathbf{x} \in \mathcal{Q}_\beta^{s,t}$ is denoted as its *normal neighborhood* $\mathcal{U}_\mathbf{x} = \left\{ \mathbf{y} \in \mathcal{Q}_\beta^{s,t} : \langle \mathbf{x}, \mathbf{y} \rangle_t < |\beta| \right\}$. For *g-disconnected* points $\mathbf{x}, \mathbf{y} \in \mathcal{Q}_\beta^{s,t}$, there does not exist a tangent vector $\boldsymbol{\xi}$ such that $\mathbf{y} = \exp_\mathbf{x}^\beta(\boldsymbol{\xi})$, which implies that its inverse $\log_\mathbf{x}^\beta(\cdot)$ is only defined in the *normal neighborhood* of $\mathbf{x}$. In a nutshell, the geodesic tools for the *g-disconnected* cases are not well-defined, making it impossible to define corresponding vector operations.

3

GEOMETRIC EMBEDDING OF STRUCTURAL AND RELATIONAL PATTERNS

Real-world relational data exhibit various graph structural patterns such as hierarchies and cycles. Recent works find that non-Euclidean Riemannian manifolds provide specific inductive biases for embedding hierarchical or spherical data. However, they cannot align well with data of mixed graph topologies. To address this, we consider a larger class of pseudo-Riemannian manifolds that generalize hyperboloid and sphere. We develop novel geodesic tools that allow for extending neural network operations into geodesically disconnected pseudo-Riemannian manifolds. As a consequence, we derive a pseudo-Riemannian graph convolutional networks (GCNs) that model data in pseudo-Riemannian manifolds of constant nonzero curvature. Our method provides a geometric inductive bias that is sufficiently flexible to model mixed topologies like hierarchical graphs with cycles. Based on these geodesic tools, we further generalize knowledge graph embeddings into such space by simultaneously considering mixed structural patterns and relational patterns. To capture various relational patterns, we formulate each relation as a pseudo-orthogonal transformation that can be decomposed into a circular rotation and a hyperbolic rotation on the pseudo-Riemannian manifold.

In this chapter, we first introduce some backgrounds on pseudo-Riemannian manifolds. Next, we generalize GCNs into a pseudo-Riemannian manifold, namely pseudo-hyperboloid. Finally, we extend knowledge graph embeddings into such space by further considering relational patterns.

3.1 PSEUDO-RIEMANNIAN GRAPH CONVOLUTIONAL NETWORKS

In this section, we first introduce the background and motivation. Next, we describe how to tackle the *g-disconnectedness* in pseudo-Riemannian manifolds. Then we present the pseudo-Riemannian GCNs based on the proposed geodesic tools.

3.1.1 *Background and Motivation*

Learning from graph-structured data is a pivotal task in machine learning, for which graph convolutional networks (GCNs) [20, 85, 158, 180] have emerged as powerful graph representation learning techniques. GCNs exploit both features and structural properties in graphs, which makes them well-suited for a wide range of applications. For this purpose, graphs are usually embedded in Riemannian manifolds equipped with a positive definite metric. Euclidean geometry is a special case of Riemannian manifolds of constant zero curvature that can be understood intuitively and has well-defined operations. However, the representation power of Euclidean space is limited [150], especially when embedding complex graphs exhibiting hierarchical structures [15]. Non-Euclidean Riemannian manifolds of constant curvatures provide an alternative to accommodate specific graph topologies. For example, hyperbolic manifold of constant negative curvature has exponentially growing volume and is well suited to represent hierarchical structures such as tree-like graphs [62, 89, 183, 194, 195]. Similarly, spherical manifold of constant positive curvature is suitable for embedding spherical data in various fields [44, 110, 179] including graphs with cycles. Some recent works [42, 54, 101, 201, 203] have extended GCNs to such non-Euclidean manifolds and have shown substantial improvements.

The topologies in real-world graphs [15], however, usually exhibit highly heterogeneous topological structures, which are best represented by different geometrical curvatures. A globally homogeneous geometry lacks the flexibility for modeling complex graphs [63]. Instead of using a single manifold, product manifolds [63, 143] combining multiple Riemannian manifolds have shown advantages when embedding graphs of mixed topologies. However, the curvature distribution of product manifolds is the same at each point, which limits the capability of embedding topologically heterogeneous graphs. Furthermore, Riemannian manifolds are equipped with a positive definite metric disallowing for the faithful representation of the negative eigenspectrum of input similarities [96].

Going beyond Riemannian manifolds, pseudo-Riemannian manifolds equipped with indefinite metrics constitute a larger class of geometries, pseudo-Riemannian manifolds of constant nonzero curvature do not only generalize the hyperbolic and spherical manifolds, but also contain their submanifolds (Cf. Fig. 3.1), thus providing inductive biases specific to these geometries. Pseudo-Riemannian geometry with constant zero curvature (i.e. Lorentzian spacetime) was applied to manifold learning for preserving local information of non-metric data [150] and embedding directed acyclic graph [40]. To model complex

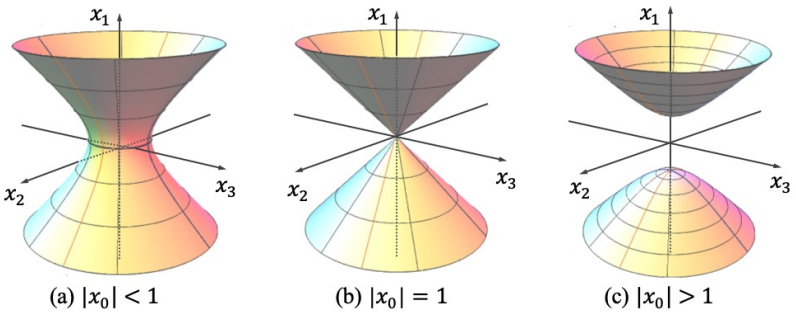


Figure 3.1: The different submanifolds of a four-dimensional pseudo-hyperboloid of curvature -1 with two time dimensions. By fixing one time dimension x_0, the induced submanifolds include (a) an one-sheet hyperboloid, (b) the double cone, and (c) a two-sheet hyperboloid.

graphs containing both hierarchies and cycles, pseudo-Riemannian manifolds with constant nonzero curvature have recently been applied into graph embeddings using non-parametric learning [97, 142], but the representation power of these works is not on par with the Riemannian counterparts yet, mostly because of the absence of geodesic tools to extend neural network operations into pseudo-Riemannian geometry.

In this work, we take the first step to extend GCNs into pseudo-Riemannian manifolds foregoing the requirement to have a positive definite metric. Exploiting pseudo-Riemannian geometry for GCNs is non-trivial because of the *geodesical disconnectedness* in pseudo-Riemannian geometry. There exist broken points that cannot be smoothly connected by a geodesic, leaving necessary geodesic tools undefined. To deal with it, we develop novel geodesic tools that empower manipulating representations in geodesically disconnected pseudo-Riemannian manifolds. We make it by finding diffeomorphic manifolds that provide alternative geodesic operations that smoothly avoid broken cases. Subsequently, we generalize GCNs to learn representations of complex graphs in pseudo-Riemannian geometry by defining corresponding operations such as *linear transformation* and *tangential aggregation*. Different from previous works, the initial features of GCN could be fully defined in the Euclidean space. Thanks to the diffeomorphic operation that is bijective and differentiable, the standard gradient descent algorithm can be exploited to perform optimization.

To summarize, our main contributions are as follows: 1) We present neural network operations in pseudo-Riemannian manifolds with novel geodesic tools, to stimulate the applications of pseudo-Riemannian geometry in geometric deep learning. 2) We present a principled

framework, pseudo-Riemannian GCN, which generalizes GCNs into pseudo-Riemannian manifolds with indefinite metrics, providing more flexible inductive biases to accommodate complex graphs with mixed topologies. 3) Extensive evaluations on three standard tasks demonstrate that our model outperforms baselines that operate in Riemannian manifolds.

3.1.2 *Methodology*

3.1.2.1 *Diffeomorphic geodesic tools*

One standard way to tackle the *g-disconnectedness* in differential geometry is to introduce diffeomorphic manifolds in which the operations are well-defined. A diffeomorphic manifold can be derived from a diffeomorphism, defined as follows.

Definition 16 (Diffeomorphism [125])**.** *Given two manifolds $\mathcal{M}$ and $\mathcal{M}'$, a smooth map $\psi : \mathcal{M} \to \mathcal{M}'$ is called a diffeomorphism if ψ is bijective and its inverse ψ^{-1} is smooth as well. If a diffeomorphism between $\mathcal{M}$ and $\mathcal{M}'$ exists, we call them diffeomorphic and write $\mathcal{M} \simeq \mathcal{M}'$.*

For pseudo-Riemannian manifolds, the following diffeomorphism [97] decomposes pseudo-hyperboloid into the product manifolds of an unit sphere and the Euclidean space.

Theorem 1 (Theorem 4.1 in [97])**.** *For any point $\mathbf{x} \in \mathcal{Q}_\beta^{s,t}$, there exists a diffeomorphism $\psi : \mathcal{Q}_\beta^{s,t} \to \mathbb{S}_1^t \times \mathbb{R}^s$ that maps $\mathbf{x}$ into the product manifolds of an unit sphere and the Euclidean space.*

The diffeomorphism is given in the Appendix A.1.1. In light of this, we introduce a new diffeomorphism that maps $\mathbf{x}$ to the product manifolds of sphere with curvature $-1/\beta$ and the Euclidean space.

Theorem 2. *For any point $\mathbf{x} \in \mathcal{Q}_\beta^{s,t}$, there exists a diffeomorphism $\psi : \mathcal{Q}_\beta^{s,t} \to \mathbb{S}_{-\beta}^t \times \mathbb{R}^s$ that maps $\mathbf{x}$ into the product manifolds of a sphere and the Euclidean space (proof in the Appendix A.1.2).*

Compared with Theorem 1, this diffeomorphism preserves the curvatures in the diffeomorphic components, making it satisfy some geometric properties, e.g. the mapped point $\mathbf{x}_t \in \mathbb{S}_{-\beta}^t$ still lies on

the surface of the pseudo-hyperboloid, making moving the tangent vectors from the pseudo-hyperboloid to the diffeomorphic manifold easy as we explained later. We call ψ as the spherical projection.

Exponential and logarithmic maps. Since pseudo-hyperboloid $\mathcal{Q}_\beta^{s,t}$ is *g-disconnected*, we propose to transfer the *logmap* and *expmap* into the diffeomorphic manifold $\psi : \mathcal{Q}_\beta^{s,t} \to \mathbb{S}_{-\beta}^t \times \mathbb{R}^s$, since the product manifold $\mathbb{S}_{-\beta}^t \times \mathbb{R}^s$ is *g-connected*. To map tangent vectors between tangent space of $\mathcal{Q}_\beta^{s,t}$ and $\mathbb{S}_{-\beta}^t \times \mathbb{R}^s$, we exploit *pushforward* that induce a linear approximation of smooth maps on tangent spaces.

Definition 17 (Pushforward)**.** *Suppose that $\psi : \mathcal{M} \to \mathcal{M}'$ is a smooth map, then the differential of ψ: $d\psi$ at point $\mathbf{x}$ is a linear map from the tangent space of $\mathcal{M}$ at $\mathbf{x}$ to the tangent space of $\mathcal{M}'$ at $\psi(\mathbf{x})$. Namely, $d\psi : \mathcal{T}_x\mathcal{M} \to \mathcal{T}_{\psi(x)}\mathcal{M}'$.*

Intuitively, *pushforward* can be used to *push* tangent vectors on $\mathcal{T}_x\mathcal{Q}_\beta^{s,t}$ *forward* to tangent vectors on $\mathcal{T}_{\psi(x)}\mathbb{S}_{-\beta}^t \times \mathbb{R}^s$. Based on this, the new *logmap* and its inverse *expmap* can be defined by Eq. (3.1).

$$\widehat{\log}_{\mathcal{Q}_\beta^{s,t}}(\mathbf{x}) = \psi^{-1}(\log_{\mathbb{S}_{-\beta}^t \times \mathbb{R}^s}(\psi(\mathbf{x}))), \widehat{\exp}_{\mathcal{Q}_\beta^{s,t}}(\boldsymbol{\xi}) = \psi^{-1}(\exp_{\mathbb{S}_{-\beta}^t \times \mathbb{R}^s}(\psi(\boldsymbol{\xi}))), \tag{3.1}$$

where $\psi(\cdot)$ is the spherical projection and $\psi^{-1}(\cdot)$ is the inverse. The mapping of tangent vectors is achieved by *pushforward* operations. The operations $\log_{\mathbb{S}_{-\beta}^t \times \mathbb{R}^s}(\cdot)$ and $\exp_{\mathbb{S}_{-\beta}^t \times \mathbb{R}^s}(\cdot)$ in the product manifolds can be defined as the concatenation of corresponding operations in different components.

$$\log_{\mathbb{S}_{-\beta}^t \times \mathbb{R}^s}(\mathbf{x}') = \log_{\mathbb{S}_{-\beta}^t}(\mathbf{x}'_t) \,\|\, \log_{\mathbb{R}^s}(\mathbf{x}'_s), \exp_{\mathbb{S}_{-\beta}^t \times \mathbb{R}^s}(\boldsymbol{\xi}) = \exp_{\mathbb{S}_{-\beta}^t}(\boldsymbol{\xi}_t) \,\|\, \exp_{\mathbb{R}^s}(\boldsymbol{\xi}_s), \tag{3.2}$$

where $||$ denotes the concatenation, $\mathbf{x}' = \psi_{\mathbb{S}}(\mathbf{x})$ consists of spherical features $\mathbf{x}'_t \in \mathbb{S}_{-\beta}^t$ and Euclidean features $\mathbf{x}'_s \in \mathbb{R}^s$. $\boldsymbol{\xi}$ is the tangent vector induced by $\mathbf{x}$ on $\mathcal{Q}_\beta^{s,t}$.

We choose points where space dimension $\mathbf{s} = \mathbf{0}$ as the reference points due to the following property.

Theorem 3. *For any reference point $\mathbf{x} = \begin{pmatrix} \mathbf{t} \\ \mathbf{s} \end{pmatrix} \in \mathcal{Q}_\beta^{s,t}$ with space dimension $\mathbf{s} = \mathbf{0}$, the induced tangent space of $\mathcal{Q}_\beta^{s,t}$ is equal to the tangent space of its diffeomorphic manifold $\mathbb{S}_{-\beta}^t \times \mathbb{R}^s$, namely, $\mathcal{T}_{\mathbf{x}}(\mathbb{S}_{-\beta}^t \times \mathbb{R}^s) = \mathcal{T}_{\mathbf{x}}\mathcal{Q}_\beta^{s,t}$. (proof in the Appendix A.1.3).*

The intuition of proof is that if space dimension $\mathbf{s} = \mathbf{0}$, the pushforward (differential) function just influences time dimension, for which

the mapping is just an identity function (see Appendix A.1.2). In this way, although we transfer *logmap* and *expmap* to the diffeomorphic manifold $\mathbb{S}^t_{-\beta} \times \mathbb{R}^s$, the diffeomorphic operations $\widehat{\log}_{\mathbf{x}}(\cdot)$ and $\widehat{\exp}_{\mathbf{x}}(\cdot)$ are still bijective functions from the pseudo-hyperboloid to the tangent space of the manifold itself. Hence, our final operations are actually still defined in the tangent space of the pseudo-hyperboloid. Note that such property only holds when our Theorem 5 is applied and the special reference points with space dimension $\mathbf{s} = \mathbf{0}$ are chosen.

By leveraging the new *logmap* and *expmap*, we further formulate the diffeomorphic version of tangential operations as follows.

Tangential operations. For function $f : \mathbb{R}^d \to \mathbb{R}^{d'}$, the pseudo-hyperboloid version $f^{\otimes} : \mathcal{Q}^{s,t}_{\beta} \to \mathcal{Q}^{s',t'}_{\beta}$ with $s + t = d$ and $s' + t' = d'$ can be defined by the means of $\widehat{\log}^{\beta}_{\mathbf{x}}(\cdot)$ and $\widehat{\exp}^{\beta}_{\mathbf{x}}(\cdot)$ as Eq. (3.3).

$$f^{\otimes}(\cdot) := \widehat{\exp}^{\beta}_{\mathbf{x}}\left(f\left(\widehat{\log}^{\beta}_{\mathbf{x}}(\cdot)\right)\right), \tag{3.3}$$

where $\mathbf{x}$ is the reference point. Note that this function is a morphism (i.e. $(f \circ g)^{\otimes} = f^{\otimes} \circ g^{\otimes}$) and direction preserving (i.e. $f^{\otimes}(\cdot)/\|f^{\otimes}(\cdot)\| = f(\cdot)/\|f(\cdot)\|$) [54], making it a natural way to define pseudo-hyperboloid version of vector operations such like scalar multiplication, matrix-vector multiplication, tangential aggregation and point-wise non-linearity and so on.

Parallel transport. Parallel transport is the generalization of Euclidean translation into manifolds. Formally, for any two points $\mathbf{x}$ and $\mathbf{y}$ connected by a geodesic, parallel transport $P^{\beta}_{\mathbf{x}\to\mathbf{y}}(\boldsymbol{\xi}) : \mathcal{T}_{\mathbf{x}}\mathcal{M} \to \mathcal{T}_{\mathbf{y}}\mathcal{M}$ is an isomorphism between two tangent spaces by moving one tangent vector $\boldsymbol{\zeta} \in \mathcal{T}_x\mathcal{M}$ with tangent direction $\boldsymbol{\xi} \in \mathcal{T}_x\mathcal{M}$ to another tangent space $\mathcal{T}_{\mathbf{y}}\mathcal{M}$. The parallel transport in pseudo-hyperboloid can be defined as the combination of Riemannian parallel transport [56]. However, the parallel transport has not been defined when there does not exist a geodesic between $\mathbf{x}$ and $\mathbf{y}$. i.e., the tangent vector $\boldsymbol{\zeta}$ induced by $\mathbf{x}$ can not be transported to the tangent space of points outside of the normal neighborhood $\mathcal{U}_{\mathbf{x}}$. Intuitively, the normal neighborhoods satisfy the following property.

Theorem 4. *For any point $\mathbf{x} \in \mathcal{Q}^{s,t}_{\beta}$, the union of the normal neighborhood of $\mathbf{x}$ and the normal neighborhood of its antipodal point $-\mathbf{x}$ cover the entire manifold. Namely, $\mathcal{U}_{\mathbf{x}} \cup \mathcal{U}_{-\mathbf{x}} = \mathcal{Q}^{s,t}_{\beta}$ (proof in the Appendix A.1.4).*

This theorem ensures that if a point $\mathbf{y} \notin \mathcal{U}_{\mathbf{x}}$, its antipodal point $-\mathbf{y} \in \mathcal{U}_{\mathbf{x}}$. Besides, $\mathcal{T}_{\mathbf{y}}\mathcal{M}$ is parallel to $\mathcal{T}_{-\mathbf{y}}\mathcal{M}$. Hence, $P^{\beta}_{\mathbf{x}\to\mathbf{y}}$ can be alternatively defined as $P^{\beta}_{\mathbf{x}\to-\mathbf{y}}$ for broken points. This result is crucial

to define the pseudo-hyperbolic addition, such as bias translation, detailed in section 3.1.2.2.

Broken geodesic distance. By the means of geodesic, the induced distance between $\mathbf{x}$ and $\mathbf{y}$ in pseudo-hyperboloid is defined as the arc length of geodesic $\gamma(\tau)$, given by $\mathrm{d}_{\mathrm{fl}}(\mathbf{x},\mathbf{y}) = \sqrt{\|\log_{\mathbf{x}}(\mathbf{y})\|_t^2}$. For broken cases in which $\log_{\mathbf{x}}(\mathbf{y})$ is not defined, one approach is to use approximation like [97]. Different from that, we define following closed-form distance, given by Eq. (3.18).

$$\mathrm{D}_{\gamma}(\mathbf{x},\mathbf{y}) = \begin{cases} \mathrm{d}_{\gamma}(\mathbf{x},\mathbf{y}), & \text{if } \langle \mathbf{x},\mathbf{y}\rangle_t < |\beta| \\ \pi\sqrt{|\beta|} + \mathrm{d}_{\gamma}(\mathbf{x},-\mathbf{y}), & \text{if } \langle \mathbf{x},\mathbf{y}\rangle_t \geq |\beta| \end{cases} \tag{3.4}$$

The intuition is that when $\mathbf{x},\mathbf{y} \in \mathcal{Q}_{\beta}^{s,t}$ are *g-disconnected*, we consider the distance as $\mathrm{d}_{\mathrm{fl}}(\mathbf{x},\mathbf{y}) = \mathrm{d}_{\mathrm{fl}}(\mathbf{x},-\mathbf{x}) + \mathrm{d}_{\mathrm{fl}}(-\mathbf{x},\mathbf{y})$ or $\mathrm{d}_{\mathrm{fl}}(\mathbf{x},\mathbf{y}) = \mathrm{d}_{\mathrm{fl}}(\mathbf{x},-\mathbf{y}) + \mathrm{d}_{\mathrm{fl}}(-\mathbf{y},\mathbf{y})$. Since $\mathrm{d}_{\mathrm{fl}}(\mathbf{x},-\mathbf{x}) = \mathrm{d}_{\mathrm{fl}}(-\mathbf{y},\mathbf{y}) = \pi\sqrt{|\beta|}$ is a constant and $\mathrm{d}_{\gamma}(-\mathbf{x},\mathbf{y}) = \mathrm{d}_{\mathrm{fl}}(\mathbf{x},-\mathbf{y})$, the distance between broken points can be calculated as $\mathrm{d}_{\mathrm{fl}}(\mathbf{x},\mathbf{y}) = \pi\sqrt{|\beta|} + \mathrm{d}_{\mathrm{fl}}(\mathbf{x},-\mathbf{y})$.

To clarify theoretical contributions, our Theorem 5 is nessasary for our Theorem 3 while Theorem 3 is nessasary for transforming the GCN operations directly into the tangent space of the pseudo-hyperboloid. Besides, we are the first to formulate the diffeomorphic *expmap*, *logmap* and tangential operations of pseudo-hyperboloid to avoid broken cases. The theoretical properties of parallel transport and geodesic distance are discussed in the literature [56, 97]. However, we re-formulate parallel transport with Theorem 4 to avoid broken issues and propose a new distance measure using the broken geodesic (Eq.3.18), which is different from the approximated distance in [97].

3.1.2.2 *Model architecture*

GCNs can be interpreted as performing neighborhood aggregation after a linear transformation on node features of each layer. We present pseudo-Riemannian GCNs ($\mathcal{Q}$-GCN) by deriving corresponding operations with the developed geodesic tools in the $\mathcal{Q}_{\beta}^{s,t}$.

Feature initialization. We first map the features from Euclidean space to pseudo-hyperboloid, considering that the input features of nodes usually live in Euclidean space. Following the feature transformation from Euclidean space to pseudo-hyperboloid in [97], we initialize the node features by performing a differentiable mapping $\varphi : \mathbb{R}_{*}^{t+1} \times \mathbb{R}^{s} \to \mathcal{Q}_{\beta}^{s,t}$ that can be implemented by a double projection [97] based on Theorem 5, i.e. $\varphi = \psi^{-1} \circ \psi$. The intuition is that we first

map the Euclidean features into diffeomorphic manifolds $\mathbb{S}^t_{-\beta} \times \mathbb{R}^s$ via $\psi(\cdot)$, and then map them into the pseudo-hyperboloid $\mathcal{Q}^{s,t}_{\beta}$ via $\psi^{-1}(\cdot)$, where the mapping functions are given by Eq. (3.5).

$$\psi(\mathbf{x}) = \begin{pmatrix} \sqrt{|\beta|}\frac{\mathbf{t}}{\|\mathbf{t}\|} \\ \mathbf{s} \end{pmatrix} \quad , \quad \psi^{-1}(\mathbf{z}) = \begin{pmatrix} \frac{\sqrt{|\beta|+\|\mathbf{v}\|^2}}{\sqrt{|\beta|}}\mathbf{u} \\ \mathbf{v} \end{pmatrix}, \tag{3.5}$$

where $\mathbf{x} = (\mathbf{t}, \mathbf{s})^\top \in \mathcal{Q}^{s,t}_{\beta}$ with $\mathbf{t} \in \mathbb{R}^t_*$ and $\mathbf{s} \in \mathbb{R}^s$. $\mathbf{z} = (\mathbf{u}, \mathbf{v})^\top \in \mathbb{S}^t_{-\beta} \times \mathbb{R}^s$ with $\mathbf{u} \in \mathbb{S}^t_{-\beta}$ and $\mathbf{v} \in \mathbb{R}^s$.

Tangential aggregation. The linear combination of neighborhood features is lifted to the tangent space, which is an intrinsic operation in differential manifolds [27, 101]. Specifically, $\mathcal{Q}$-GCN aggregates neighbours' embeddings in the tangent space of the reference point $\mathbf{o}$ before passing through a tangential activation function, and then projects the updated representation back to the manifold. Formally, at each layer ℓ, the updated features of each node i are defined as Eq. (3.6).

$$\mathbf{h}_i^{\ell+1} = \widehat{\exp}_{\mathbf{o}}^{\beta_{\ell+1}}\left(\sigma\left(\sum_{j\in\mathcal{N}(i)\cup\{i\}} \widehat{\log}_{\mathbf{o}}^{\beta_\ell}\left(W^\ell \otimes^{\beta_\ell} \mathbf{h}_j^\ell \oplus^{\beta_\ell} \mathbf{b}^\ell\right)\right)\right), \tag{3.6}$$

where $\sigma(\cdot)$ is the activation function, β_ℓ and $\beta_{\ell+1}$ are two layer-wise curvatures, $\mathcal{N}(i)$ denotes the one-hop neighborhoods of node i, and the $\otimes, \oplus$ denote two basic operations, i.e. tangential transformation and bias translation, respectively.

Tangential transformation. We perform Euclidean transformations on the tangent space by leveraging the *expmap* and *logmap* in Eq. (3.1). Specifically, we first project the hidden feature into the tangent space of *south pole* $\mathbf{o} = [|\beta|, 0, , ..., 0]$ using *logmap* and then perform Euclidean matrix multiplication. Afterwards, the transformed features are mapped back to the manifold using *expmap*. Formally, at each layer ℓ, the tangential transformation is given by $W^\ell \otimes^\beta \mathbf{h}^\ell := \widehat{\exp}_{\mathbf{o}}^{\beta}(W^\ell \widehat{\log}_{\mathbf{o}}^{\beta}(\mathbf{h}^\ell))$, where $\otimes^\beta$ denotes the pseudo-hyperboloid tangential multiplication, and $W^\ell \in \mathbb{R}^{d'\times d}$ denotes the layer-wise learnable matrix in Euclidean space.

Bias translation. It is noteworthy that simply stacking multiple layers of the tangential transformation would collapse the composition [27, 54], i.e. $\exp_{\mathbf{o}}^{\beta} ...(W^1 \log_{\mathbf{o}}^{\beta}(\exp_{\mathbf{o}}^{\beta}(W^0 \log_{\mathbf{o}}^{\beta}(x)))) = \exp_{\mathbf{o}}^{\beta}(W^0 \times W^1 \times ... \times \log_{\mathbf{o}}^{\beta}(x))$, which means that these multiplications can simply be performed in Euclidean space except the first *logmap* and last *expmap*. To avoid model collapsing , we perform bias translation after the tangential transformation. By the means of pseudo-hyperboloid *parallel*

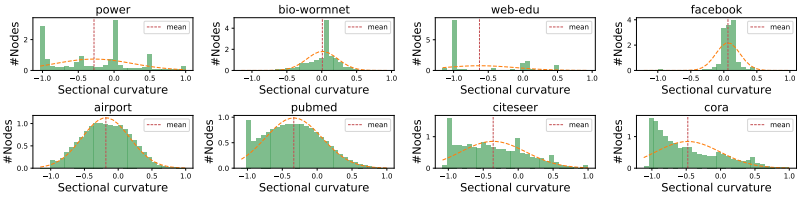


Figure 3.2: The histograms of sectional curvature for all used datasets.

transport, the bias translation can be performed by parallel transporting a tangent vector $\mathbf{b}^\ell \in \mathcal{T}_{\mathbf{o}}\mathcal{Q}_\beta^{s,t}$ to the tangent space of the point of interest. Finally, the transported tangent vector is mapped back to the manifold with *expmap*. Considering that $\widehat{\exp}(\cdot)$ is only defined at point $\mathbf{x} \in \mathcal{Q}_\beta^{s,t}$ with the space dimension $\mathbf{x}_s = \mathbf{0}$, we perform the original $\exp_{\mathcal{Q}_\beta^{s,t}}(\cdot)$ at the point of interest. The bias translation is formally given by:

$$\widetilde{\mathbf{h}}^\ell \oplus^\beta \mathbf{b}^\ell := \begin{cases} \exp^\beta_{\widetilde{\mathbf{h}}^\ell}\left(P^\beta_{\mathbf{o}\to\widetilde{\mathbf{h}}^\ell}\left(\mathbf{b}^\ell\right)\right), & \text{if } \langle \mathbf{o}, \widetilde{\mathbf{h}}^\ell\rangle_t < |\beta| \\ -\exp^\beta_{-\widetilde{\mathbf{h}}^\ell}\left(P^\beta_{\mathbf{o}\to-\widetilde{\mathbf{h}}^\ell}\left(\mathbf{b}^\ell\right)\right), & \text{if } \langle \mathbf{o}, \widetilde{\mathbf{h}}^\ell\rangle_t \geq |\beta| \end{cases} \tag{3.7}$$

where $\widetilde{\mathbf{h}}^\ell = W^\ell \otimes^\beta \mathbf{h}^\ell$, $\oplus^\beta$ denotes the pseudo-hyperboloid addition. For the broken cases where $\langle \mathbf{o}, \widetilde{\mathbf{h}}^\ell\rangle_t \geq |\beta|$, the parallel transport $P^\beta_{\mathbf{o}\to\widetilde{\mathbf{h}}^\ell}$ is not defined. In this case, we parallel transport $\mathbf{b}^\ell$ to the tangent space of the antipodal point $-\widetilde{\mathbf{h}}^\ell$, and then perform $\exp^\beta_{-\widetilde{\mathbf{h}}^\ell}$ to map it back to the manifold. Note that the case $\langle \mathbf{o}, \widetilde{\mathbf{h}}^\ell\rangle_t = |\beta|$ occurs if and only if $\mathbf{h} = -\mathbf{o}$, in which case $P^\beta_{\mathbf{o}\to-\widetilde{\mathbf{h}}^\ell}\left(\mathbf{b}^\ell\right) = P^\beta_{\mathbf{o}\to-\mathbf{o}}\left(\mathbf{b}^\ell\right) = -\mathbf{b}$.

3.1.3 *Experiments*

We evaluate the effectiveness of $\mathcal{Q}$-GCN on graph reconstruction, node classification and link prediction. Firstly, we study the geometric properties of the used datasets including the graph sectional curvature [63] and the δ-hyperbolicity [62]. Fig. 3.2 shows the histograms of sectional curvature and the mean sectional curvature for all datasets. It can be seen that all datasets have both positive and negative sectional curvatures, showcasing that all graphs contain mixed graph topologies. To further analyze the degree of the hierarchy, we apply δ-hyperbolicity to identify the tree-likeness, as shown in Table 3.1. We conjecture that the datasets with positive graph sectional curvature or larger δ-hyperbolicity should be suitable for pseudo-hyperboloid with a smaller time dimension, while datasets with negative graph sectional curvature or smaller δ-hyperbolicity should be aligned well with pseudo-hyperboloid with a larger time dimension.

Table 3.1: The δ-hyperbolicity distribution of all used datasets.

Datasets	0	0.5	1.0	1.5	2.0	2.5
Power	0.4025	0.1722	0.1436	0.0773	0.0639	0.0439
Bio-Worm	0.5635	0.3949	0.0410	0	0	0
Web-Edu	0.9532	0.0468	0	0	0	0
Facebook	0.8209	0.1569	0.0221	0	0	0
Airport	0.6376	0.3563	0.0061	0	0	0
Pubmed	0.4239	0.4549	0.1094	0.0112	0.0006	0
CiteSeer	0.3659	0.3538	0.1699	0.0678	0.0288	0
Cora	0.4474	0.4073	0.1248	0.0189	0.0016	0.0102

3.1.3.1 *Graph reconstruction*

Datasets and baselines. We benchmark graph reconstruction on four real-world graphs including 1) Web-Edu [59]: a web network consisting of the *.edu* domain; 2) Power [174]: a power grid distribution network with backbone structure; 3) Bio-Worm [36]: a worms gene network; 4) Facebook [108]: a dense social network from Facebook. We compare our method with Euclidean GCN [85], hyperbolic GCN (HGCN) [27], spherical GCN, and product manifold GCNs (κ-GCN) [9] with three signatures (i.e. $\mathbb{H}^5 \times \mathbb{H}^5$, $\mathbb{H}^5 \times \mathbb{S}^5$ and $\mathbb{S}^5 \times \mathbb{S}^5$). Besides, five variants of our model are implemented with different time dimension in $[1, 3, 5, 7, 10]$ for comparison.

Experimental settings. We use one-hot embeddings as initial node features following [9, 63, 97]. To avoid the time dimensions being $\mathbf{0}$, we uniformly perturb each dimension with a small random value in the interval $[-\epsilon, \epsilon]$, where $\epsilon = 0.02$ in practice. In addition, the same 10-dimensional embedding and 2 hidden layers are used for all baselines to ensure a fair comparison. The learning rate is set to 0.01, the learning rate of curvature is set to 0.0001. $\mathcal{Q}$-GCN is implemented with the Adam optimizer. We repeat the experiments 10 times via different random seeds influencing weight initialization and data batching.

Table 3.2: The graph reconstruction results in mAP (%), top three results are highlighted. Standard deviations are relatively small (in range $[0, 1.2 \times 10^{-2}]$) and are omitted.

Model	Web-Edu	Power	Bio-Worm	Facebook
Curvature	-0.6	-0.3	0.0	0.1
GCN ($\mathbb{E}^{10}$)	83.66	86.61	90.19	81.73
HGCN ($\mathbb{H}^{10}$)	88.33	93.80	93.12	83.40
GCN ($\mathbb{S}^{10}$)	82.72	92.73	88.98	81.04
κ-GCN ($\mathbb{H}^5 \times \mathbb{H}^5$)	89.21	94.40	94.00	84.94
κ-GCN ($\mathbb{S}^5 \times \mathbb{S}^5$)	86.70	94.58	90.36	84.56
κ-GCN ($\mathbb{H}^5 \times \mathbb{S}^5$)	87.96	**95.82**	94.74	87.73
$\mathcal{Q}$-GCN ($\mathcal{Q}^{9,1}$)	87.03	94.35	92.83	81.60
$\mathcal{Q}$-GCN ($\mathcal{Q}^{7,3}$)	**99.67**	**100.00**	**97.23**	**87.74**
$\mathcal{Q}$-GCN ($\mathcal{Q}^{5,5}$)	**98.49**	**100.00**	**95.75**	**87.03**
$\mathcal{Q}$-GCN ($\mathcal{Q}^{3,7}$)	**97.31**	95.08	**90.14**	**91.75**
$\mathcal{Q}$-GCN ($\mathcal{Q}^{0,10}$)	82.57	94.20	88.67	83.81

Results. Table 3.2 shows the mean average precision (mAP) [63] results of graph reconstruction on four datasets. It shows that $\mathcal{Q}$-GCN

achieves the best performance across all benchmarks compared with both Riemannian space and product manifolds. We observe that by setting proper signatures, the product spaces perform better than a single geometry. It is consistent with our statement that the expression power of a single view geometry is limited. Specifically, all the top three results are achieved by $\mathcal{Q}$-GCN, with one exception on Power where $\mathbb{H}^5 \times \mathbb{S}^5$ achieved the third-best performance. More precisely, for datasets that have smaller graph sectional curvature like Web-Edu, Power and Bio-Worm, $\mathcal{Q}^{7,3}$ perform the best, while $\mathcal{Q}^{3,7}$ perform the best on Facebook with positive sectional curvature. We conjecture that the number of time dimensions controls the geometry of the pseudo-hyperboloid. We find that the graphs with more hierarchical structures are inclined to be embedded with fewer time dimensions. By analyzing the sectional curvature in Fig. 3.2, we find that this makes sense as the mean sectional curvature of Power, Bio-Wormnet and Web-Edu are negative while it is negative for Facebook. Such results give us an intuition to determine the best time dimension based on the geometric properties of graphs.

3.1.3.2 *Node classification and link prediction*

Datasets and baselines. We consider four benchmark datasets: Airport, Pubmed, Citeseer and Cora, where Airport is airline networks, Pubmed, Citeseer and Cora are three citation networks. We observe that the graph sectional curvatures of the four datasets are consistently negative without significant differences in Fig. 3.2, hence we report the additional δ-hyperbolicity for comparison in Table 3.3. GCN [85], GAT [158], SAGE [70] and SGC [180] are used as Euclidean GCN counterparts. For non-Euclidean GCN baselines, we compare HGCN [27] and κ-GCN [9] with its three variants as explained before. For $\mathcal{Q}$-GCN, we empirically set the time dimension as $[1, 2, 3, 14, 15, 16]$ as six variants since these settings best reflect the geometric properties of hyperbolic and spherical space, respectively.

Experimental settings. For node classification, we use the same dataset split as [196] for citation datasets, where 20 nodes per class are used for training, and 500 nodes are used for validation and 1000 nodes are used for testing. For Airport, we split the dataset into 70/15/15. For link prediction, the edges are split into 85/5/10 percent for training, validation and testing for all datasets. To ensure a fair comparison, we set the same 16-dimension hidden embedding, 0.01 initial learning rate and 0.0001 learning rate for curvature. The optimal regularization with weight decay, dropout rate, the number of layers and activation functions are obtained by grid search for

Table 3.3: ROC AUC (%) for Link Prediction (LP) and F1 score for Node Classification (NC).

Dataset	Airport		Pubmed		CiteSeer		Cora	
δ-hyperbolicity	1.0		3.5		4.5		11.0	
Method	LP	NC	LP	NC	LP	NC	LP	NC
GCN	89.24±0.21	81.54±0.60	91.31±1.68	79.30±0.60	85.48±1.75	72.27±0.64	88.52±0.85	81.90±0.41
GAT	90.35±0.30	81.55±0.53	87.45±0.00	78.30±0.00	87.24±0.00	71.10±0.00	85.73±0.01	83.05±0.08
SAGE	89.86±0.52	82.79±0.17	90.70±0.07	77.30±0.09	90.71±0.20	69.20±0.10	87.52±0.22	74.90±0.07
SGC	89.80±0.34	80.69±0.23	90.54±0.07	78.60±0.30	89.61±0.23	71.60±0.03	89.42±0.11	81.60±0.43
HGCN ($\mathbb{H}^{16}$)	**96.03±0.26**	**90.57±0.36**	**96.08±0.21**	**80.50±1.23**	**96.31±0.41**	68.90±0.63	91.62±0.33	79.90±0.18
κ-GCN ($\mathbb{H}^{16}$)	**96.35±0.62**	**87.92±1.33**	96.60±0.32	77.96±0.36	95.34±0.16	**73.25±0.51**	94.04±0.34	79.80±0.50
κ-GCN ($\mathbb{S}^{16}$)	90.38±0.32	81.94±0.58	94.84±0.13	78.80±0.49	95.79±0.24	72.13±0.51	93.20±0.48	81.08±1.45
κ-GCN ($\mathbb{H}^8 \times \mathbb{S}^8$)	93.10±0.49	81.93±0.45	94.89±0.19	79.20±0.65	93.44±0.31	73.05±0.59	92.22±0.48	79.30±0.81
$\mathcal{Q}$-GCN ($\mathcal{Q}^{15,1}$)	**96.30±0.22**	**89.72±0.52**	95.42±0.22	**80.50±0.26**	94.76±1.49	72.67±0.76	93.14±0.30	80.57±0.20
$\mathcal{Q}$-GCN ($\mathcal{Q}^{14,2}$)	94.37±0.44	84.40±0.35	**96.86±0.37**	**81.34±1.54**	94.78±0.17	**73.43±0.58**	93.41±0.57	81.62±0.21
$\mathcal{Q}$-GCN ($\mathcal{Q}^{13,3}$)	92.53±0.17	82.38±1.53	**96.20±0.34**	**80.94±0.45**	94.54±0.16	**74.13±1.41**	93.56±0.18	79.91±0.42
$\mathcal{Q}$-GCN ($\mathcal{Q}^{2,14}$)	90.03±0.12	81.14±1.32	94.30±1.09	78.40±0.39	94.80±0.08	72.72±0.47	**94.17±0.38**	**83.10±0.35**
$\mathcal{Q}$-GCN ($\mathcal{Q}^{1,15}$)	89.07±0.58	81.24±0.34	94.66±0.18	78.11±1.38	**97.01±0.30**	**73.19±1.58**	**94.81±0.27**	**83.72±0.43**
$\mathcal{Q}$-GCN ($\mathcal{Q}^{0,16}$)	89.01±0.61	80.91±0.65	94.49±0.28	77.90±0.80	**96.21±0.38**	72.54±0.27	**95.16±1.25**	82.51±0.32

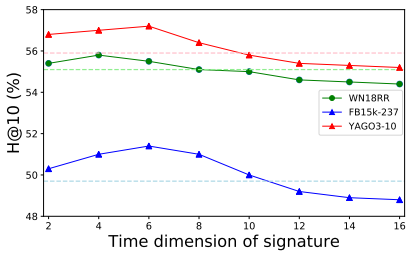


Figure 3.3: The mAP of graph reconstruction with varying number of time dimensions.

each method. We report the mean accuracy over 10 random seeds influencing weight initialization and batching sequence.

Results. Table 3.3 shows the averaged ROC AUC for link prediction, and F1 score for node classification. As we can see from the δ-hyperbolicity, Airport and Pubmed are more hierarchical than CiteSeer and Cora. For Airport and Pubmed with dominating hierarchical properties (lower δ), $\mathcal{Q}$-GCNs with fewer time dimensions achieve the results on par with hyperbolic space based methods such as HGCN [27], κ-GCN ($\mathbb{H}^{16}$). While for CiteSeer and Cora with less tree-like properties (higher δ), $\mathcal{Q}$-GCNs achieve the state-of-the-art results, showcasing the flexibility of our model to embed complex graphs with different curvatures. More specifically, $\mathcal{Q}$-GCN with more time dimensions consistently performs best on Cora. While for CiteSeer, albeit $\mathcal{Q}$-GCN achieves the best results, the corresponding best variants are not consistent on both tasks.

3.1.3.3 *Parameter sensitivity and analysis*

Time dimension. We study the influence of time dimension for graph reconstruction by setting varying number of time dimensions under

the condition of $s + t = 10$. Fig. 3.3 shows that the *time dimension* t acts as a knob for controlling the geometric properties of $\mathcal{Q}_\beta^{s,t}$. The best performance are achieved by neither hyperboloid ($t = 1$) nor sphere cases ($t = 10$), showcasing the advantages of $\mathcal{Q}_\beta^{s,t}$ on representing graphs of mixed topologies. It shows that on Web-Edu, Power and Bio-Worm with smaller mean sectional curvature, after the optimal value is reached at a lower t, the performance decrease as t increases. While on Facebook with larger (positive) mean sectional curvature, as t rises, the effect gradually increases until it reaches a peak at a higher t. It is consistent with our hypothesis that graphs with more hierarchical structure are inclined to be embedded in $\mathcal{Q}_\beta^{s,t}$ with smaller t, while cyclical data is aligned well with larger t. The results give us an intuition to determine the best time dimension based on the geometric properties of graphs.

$\mathcal{Q}$-NN VS $\mathcal{Q}$-GCN. We also introduce $\mathcal{Q}$-NN, a generalization of MLP into pseudo-Riemannian manifold, defined as multiple layers of $f(\mathbf{x}) = \sigma^{\otimes}(W \otimes^\beta \mathbf{x} \oplus^\beta \mathbf{b})$, where $\sigma^{\otimes}$ is the tangential activation. Table 3.4 shows that $\mathcal{Q}$-NN with appropriate time dimension outperforms MLP and HNN on node classification, showcasing the expression power of pseudo-hyperboloid. Furthermore, compared with the results of $\mathcal{Q}$-GCN in Table 3.3, $\mathcal{Q}$-GCN performs better than $\mathcal{Q}$-NN, suggesting that the benefits of the neighborhood aggregation equipped with the proposed GCN operations.

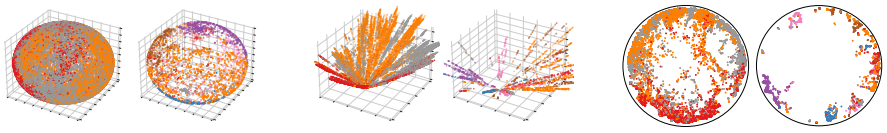
Table 3.4: The graph reconstruction results in mAP (%), top three results are highlighted. Standard deviations are relatively small (in range $[0, 1.2 \times 10^{-2}]$) and are omitted.

Method	Pubmed	CiteSeer	Cora
MLP	72.30±0.30	60.22±0.42	55.80±0.08
HNN	74.60±0.40	59.92±0.87	59.60±0.09
$\mathcal{Q}$-NN ($\mathcal{Q}^{15,1}$)	74.31±0.33	59.33±0.35	60.38±0.56
$\mathcal{Q}$-NN ($\mathcal{Q}^{14,2}$)	**76.26±0.31**	**64.33±0.35**	62.77±0.30
$\mathcal{Q}$-NN ($\mathcal{Q}^{13,3}$)	75.85±0.79	63.65±0.57	59.04±0.45
$\mathcal{Q}$-NN ($\mathcal{Q}^{2,14}$)	74.44±0.68	60.48±0.29	63.85±0.22
$\mathcal{Q}$-NN ($\mathcal{Q}^{1,15}$)	73.44±0.28	60.33±0.40	**64.85±0.24**
$\mathcal{Q}$-NN ($\mathcal{Q}^{0,16}$)	73.31±0.17	61.05±0.22	63.96±0.41

Computation efficiency. We compare the running time of $\mathcal{Q}$-GCN, GCN and Prod-GCN per epoch. Table 3.5 shows that $\mathcal{Q}$-GCN achieves higher efficiency than Prod-GCN ($\mathbb{H}^5 \times \mathbb{S}^5$). This is mainly owing to that the component $\mathbb{R}$ in our diffeomorphic manifold ($\mathbb{S} \times \mathbb{R}$) runs faster than non-Euclidean components in $\mathbb{H}^5 \times \mathbb{S}^5$. The additional running time mainly comes from the mapping operations and the

Table 3.5: The running time (seconds) of graph reconstruction on Web-Edu and Facebook.

Manifolds	Web-Edu	Facebook
GCN ($\mathbb{E}^{10}$)	2284	5456
Prod-GCN ($\mathbb{H}^5 \times \mathbb{S}^5$)	4336	10338
$\mathcal{Q}$-GCN ($\mathcal{Q}^{9,1}$)	2769	6981
$\mathcal{Q}$-GCN ($\mathcal{Q}^{7,3}$)	3363	6303
$\mathcal{Q}$-GCN ($\mathcal{Q}^{5,5}$)	3620	7142
$\mathcal{Q}$-GCN ($\mathcal{Q}^{3,7}$)	3685	7512
$\mathcal{Q}$-GCN ($\mathcal{Q}^{1,9}$)	3532	7980
$\mathcal{Q}$-GCN ($\mathcal{Q}^{0,10}$)	2778	7037



(a) Spherical projection (b) Hyperbolic projection (3D) (c) Hyperbolic projection (disk)

Figure 3.4: Visualization of the learned embeddings for link prediction on Pubmed (left) and Cora (right), where the colors denote the class of nodes. We apply (a) spherical projection, (b) hyperbolic projection (3D) and (c) hyperbolic projection (Poincaré disk) on the learned embeddings of $\mathcal{Q}$-GCN to visualize various views of the learned embeddings.

projection from the time dimensions to S. The running time grows when increasing the number of time dimensions. Overall, albeit slower than Euclidean GCN, the running time of all variants of $\mathcal{Q}$-GCN is smaller than the twice of time in Euclidean GCN, which is within the acceptable limits.

Visualization. To visualize the embeddings learned by $\mathcal{Q}$-GCN, we use UMAP tool [1] to project the learned embeddings for Link Prediction on Pubmed and Cora into low-dimensional spherical and hyperbolic spaces. The projections include spherical projection into 3D sphere, hyperbolic projection into 3D plane, and 2D Poincaré disk. As shown in Fig. 3.4 (a,b), for Pubmed with more hyperbolic structures, the class separability is more significant in hyperbolic projection than that is in spherical projection. While the corresponding result is opposite for less tree-like Cora. Furthermore, Fig. 3.4 (c) provides a more clear insight of the hierarchy. It shows that there are more hub nodes near the origin of Poincaré disk in Pubmed than in Cora, showcasing the dominating tree-likeness of Pubmed.

3.1.4 *Conclusion*

In this paper, we generalize GCNs to pseudo-Riemannian manifolds of constant nonzero curvature with elegant theories of diffeomorphic geometry tools. The proposed $\mathcal{Q}$-GCN have the flexibility to fit complex graphs with mixed curvatures and have shown promising results on graph reconstruction, node classification and link prediction. One limitation might be the choice of time dimension, we provide some insights to decide the best time dimension but this could still be improved, which we left for our future work. The developed geodesic tools are application-agnostic and could be extended to more deep

1 https://umap-learn.readthedocs.io/en/latest/index.html

learning methods. We foresee our work would shed light on the direction of non-Euclidean geometric deep learning.

3.2 PSEUDO-RIEMANNIAN KNOWLEDGE GRAPH EMBEDDINGS

Recent knowledge graph (KG) embeddings have been advanced by hyperbolic geometry due to its superior capability for representing hierarchies. The topological structures of real-world KGs, however, are rather heterogeneous, i.e., a KG is composed of multiple distinct hierarchies and non-hierarchical graph structures. Therefore, a homogeneous (either Euclidean or hyperbolic) geometry is not sufficient for fairly representing such heterogeneous structures. To capture the topological heterogeneity of KGs, we present an ultrahyperbolic KG embedding (UltraE) in an ultrahyperbolic (or pseudo-Riemannian) manifold that seamlessly interleaves hyperbolic and spherical manifolds. In particular, we model each relation as a pseudo-orthogonal transformation that preserves the pseudo-Riemannian bilinear form. The pseudo-orthogonal transformation is decomposed into various operators (i.e., circular rotations, reflections and hyperbolic rotations), allowing for simultaneously modeling heterogeneous structures as well as complex relational patterns. Experimental results on three standard KGs show that UltraE outperforms previous Euclidean, hyperbolic, and mixed-curvature KG embedding approaches.

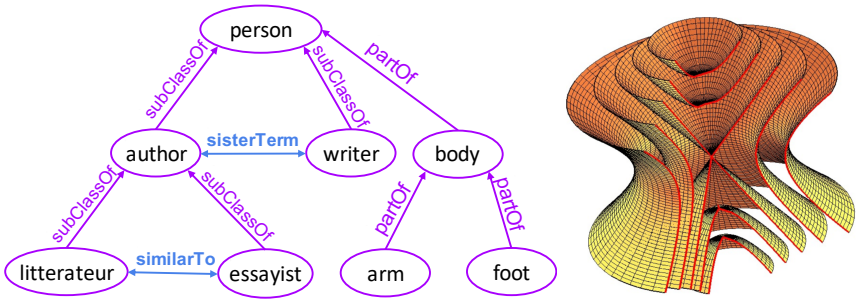
3.2.1 *Background and Motivation*

Knowledge graph (KG) embeddings, which map entities and relations into a low-dimensional space, have emerged as an effective way for a wide range of KG-based applications [21, 24, 77]. In the last decade, various KG embedding methods have been proposed. Prominent examples include the *additive* (or *translational*) family [19, 99, 173] and the *multiplicative* (or *bilinear*) family [100, 123, 193]. Most of these approaches, however, are built on the Euclidean geometry that suffers from inherent limitations when dealing with hierarchical KGs such as WordNet [111]. Recent studies [27, 121] show that hyperbolic geometries (e.g., the Poincaré ball or Lorentz model) are more suitable for embedding hierarchical data because of their exponentially growing volumes. Such *tree-like* geometric space has been exploited in developing various hyperbolic KG embedding models such as MuRP [12], RotH [26] and HyboNet [31], boosting the performance of link prediction on KGs with rich hierarchical structures and remarkably reducing the dimensionality.

Although hierarchies are the most dominant structures, the real-world KGs usually exhibit heterogeneous topological structures, e.g., a KG consists of multiple hierarchical and non-hierarchical relations. Typically, different hierarchical relations (e.g., *subClassOf* and *partOf*) form distinct hierarchies, while various non-hierarchical relations (e.g., *similarTo* and *sisterTerm*) capture the corresponding interactions between the entities at the same hierarchy level [10]. Fig.3.5(a) shows an example of KG consisting of a heterogeneous graph structure. However, current hyperbolic KG embedding methods such as MuRP [12] and HyboNet [31] can only model a globally homogeneous hierarchy. RotH [26] implicitly considers the topological "heterogeneity" of KGs and alleviates this issue by learning relation-specific curvatures that distinguish the topological characteristics of different relations. However, this does not entirely solve the problem, because hyperbolic geometry inherently *mismatches* non-hierarchical data (e.g., data with cyclic structure) [63].

To deal with data with heterogeneous topologies, a recent work [172] learns KG embeddings in a product manifold and shows some improvements on KG completion. However, such product manifold is still a homogeneous space in which all data points have the same degree of heterogeneity (i.e., hierarchy and cyclicity), while KGs require relation-specific geometric mappings, e.g., relation *partOf* should be more "hierarchical" than relation *similarTo*. Different from previous works, we consider an ultrahyperbolic manifold that seamlessly interleaves the hyperbolic and spherical manifolds. Fig.3.5 (b) shows an example of ultrahyperbolic manifold that contains multiple distinct geometries. Ultrahyperbolic manifold has demonstrated impressive capability on embedding graphs with heterogeneous topologies such as hierarchical graphs with cycles [97, 142]. However, such powerful representation space has not yet been exploited for embedding KGs with heterogeneous topologies.

In this paper, we propose ultrahyperbolic KG embeddings (UltraE), the first KG embedding method that simultaneously embeds multiple distinct hierarchical relations and non-hierarchical relations in a single but heterogeneous geometric space. The intuition behind the idea is that there exist multiple kinds of local geometries that could describe their corresponding relations. For example, as shown in Fig. 3.6(a), two points in the same *circular conic section* are described by spherical geometry, while two points in the same half of a *hyperbolic conic section* can be described by hyperbolic geometry. In particular, we model entities as points in the ultrahyperbolic manifold and model relations as pseudo-orthogonal transformations, i.e., isometries in the ultrahyperbolic manifold. We exploit the theorem of hyperbolic Cosine-Sine decomposition [149] to decompose the pseudo-orthogonal

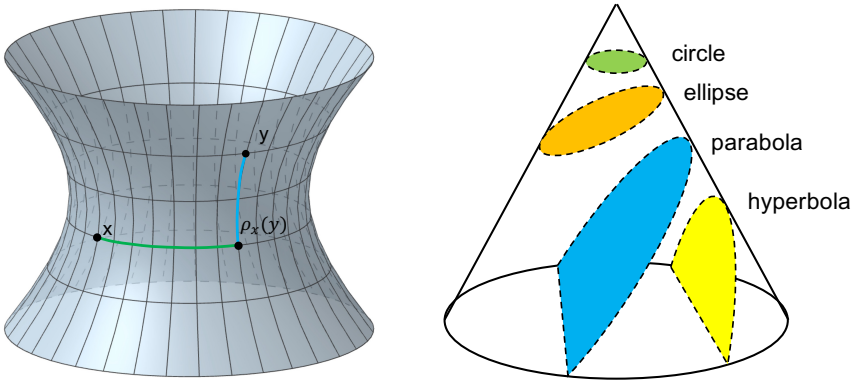


(a) (b)

Figure 3.5: (a) A KG contains multiple distinct hierarchies (e.g., *subClassOf* and *partOf*) and non-hierarchies (e.g., *similarTo* and *sisterTerm*). (b) An ultrahyperbolic manifold generalizing hyperbolic and spherical manifolds (figure from [97]).

matrices into various geometric operations including circular rotations/reflections and hyperbolic rotations. Circular rotations/reflections allow for modeling relational patterns (e.g., composition), while hyperbolic rotations allow for modeling hierarchical graph structures. As Fig. 3.6(b) shows, a combination of circular rotations/reflections and hyperbolic rotations induces various geometries including circular, elliptic, parabolic and hyperbolic geometries. These geometric operations are parameterized by Givens rotation/reflection [26] and trigonometric functions, such that the number of relation parameters grows linearly w.r.t embedding dimensionality. The entity embeddings are parametrized in Euclidean space and projected to the ultrahyperbolic manifold with differentiable and bijective mappings, allowing for stable optimization via standard Euclidean based gradient descent algorithms.

Contributions. Our key contributions include: 1) We propose a novel KG embedding method, dubbed UltraE, that models entities in an ultrahyperbolic manifold seamlessly covering various geometries including hyperbolic, spherical and their combinations. UltraE enables modeling multiple hierarchical and non-hierarchical structures in a single but heterogeneous space; 2) We propose to decompose the relational transformation into various operators and parameterize them via Givens rotations/reflections such that the number of parameters is linear to the dimensionality. The decomposed operators allow for modeling multiple relational patterns including inversion, composition, symmetry, and anti-symmetry; 3) We propose a novel Manhattan-like distance in the ultrahyperbolic manifold, to retain the identity of indiscernibles while without suffering from the broken geodesic issues. 4) We show the theoretical connection of UltraE with some existing approaches. Particularly, by exploiting the theorem of Lorentz transformation, we identify the connections between multiple hyperbolic KG embedding methods, including MuRP, RotH/RefH



(a) (b)

Figure 3.6: (a) An illustration of a spherical (*green*) geometry in the circular conic section and a hyperbolic (*blue*) geometry in the hyperbolic conic section. The Manhattan-like distance of two points is defined by summing up the *energy* moving from one point to another point with a circular rotation and a hyperbolic rotation. $\rho_{\mathbf{x}}(\mathbf{y})$ is a projection of $\mathbf{y}$ on an circular conic section crossing $\mathbf{x}$, such that $\rho_{\mathbf{x}}(\mathbf{y})$ and $\mathbf{x}$ are connected by a circular rotation while $\rho_{\mathbf{x}}(\mathbf{y})$ and $\mathbf{y}$ are connected by a hyperbolic rotation. (b) An illustration of geometries covered by the circular and hyperbolic rotation, including circular, elliptic, parabolic, and hyperbolic geometries.

and HyboNet. 5) We conduct extensive experiments on three standard benchmarks, and the experimental results show that UltraE outperforms previous Euclidean, hyperbolic and mixed-curvature (product manifold) baselines on KG completion tasks.

3.2.2 *Methodology*

Let $\mathcal{E}$ and $\mathcal{R}$ denote the set of entities and relations. A KG $\mathcal{K}$ consists of a set of triples $(h, r, t) \in \mathcal{K}$ where $h, t \in \mathcal{E}, r \in \mathcal{R}$ denote the head, the tail and their relation, respectively. The objective is to associate each entity with an embedding $\mathbf{e} \in \mathbb{U}^{p,q}$ in the ultrahyperbolic manifold, as well as a relation-specific transformation $f_r : \mathbb{U}^{p,q} \to \mathbb{U}^{p,q}$ that transforms one entity to another one in the ultrahyperbolic manifold.

3.2.2.1 *Relation as Pseudo-Orthogonal Matrix*

We propose to model relations as pseudo-orthogonal (or J-orthogonal) transformations [74], a generalization of *orthogonal transformation* in

pseudo-Riemannian geometry. Formally, a real, square matrix $\mathbf{Q} \in \mathbb{R}^{d\times d}$ is called J-orthogonal if

$$\mathbf{Q}^T\mathbf{J}\mathbf{Q} = \mathbf{J}, \tag{3.8}$$

where $\mathbf{J} = \begin{bmatrix} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_q \end{bmatrix}$, $p + q = d$ and $\mathbf{I}_p$, $\mathbf{I}_q$ are identity matrices. $\mathbf{J}$ is called a signature matrix of signature (p, q). Such J-orthogonal matrices form a multiplicative group called pseudo-orthogonal group $O(p, q)$. Conceptually, a matrix $\mathbf{Q} \in O(p, q)$ is an *isometry* (distance-preserving transformation) in the ultrahyperbolic manifold that preserves the bilinear form (i.e., $\forall \mathbf{x} \in \mathbb{U}^{p,q}, \mathbf{Q}\mathbf{x} \in \mathbb{U}^{p,q}$). Therefore, the matrix acts as a linear transformation in the ultrahyperbolic manifold.

There are two challenges to model relations as J-orthogonal transformations: 1) J-orthogonal matrix requires $\mathcal{O}(d^2)$ parameters. 2) Directly optimizing the J-orthogonal matrices results in constrained optimization, which is practically challenging within the standard gradient based framework.

Hyperbolic Cosine-Sine Decomposition. To solve these issues, we seek to decompose the J-orthogonal matrix by exploiting the Hyperbolic Cosine-Sine (CS) Decomposition.

Proposition 1 (Hyperbolic CS Decomposition [149])**.** *Let* $\mathbf{Q}$ *be J-orthogonal and assume that* $q \leq p$*. Then there are orthogonal matrices* $\mathbf{U}_1, \mathbf{V}_1 \in \mathbb{R}^{p\times p}$ *and* $\mathbf{U}_2, \mathbf{V}_2 \in \mathbb{R}^{q\times q}$ *s.t.*

$$\mathbf{Q} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{C} & \mathbf{0} & \mathbf{S} \\ \mathbf{0} & I_{p-q} & \mathbf{0} \\ \mathbf{S} & \mathbf{0} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2^T \end{bmatrix}, \tag{3.9}$$

where $\mathbf{C} = \operatorname{diag}(c_1, \ldots, c_q), \mathbf{S} = \operatorname{diag}(s_1, \ldots, s_q)$ *and* $\mathbf{C}^2 - \mathbf{S}^2 = \mathbf{I}_q$*. For cases where* $q > p$*, the decomposition can be defined analogously. For simplicity, we only consider* $q \leq p$*.*

Geometrically, the J-orthogonal matrix is decomposed into various geometric operators. The orthogonal matrices $\mathbf{U}_1, \mathbf{V}_1$ represent circular rotation or reflection (depending on their determinant) [2] in the space dimension, while $\mathbf{U}_2, \mathbf{V}_2$ represent circular rotation or reflection in the time dimension. The intermediate matrix that is uniquely determined by $\mathbf{C}, \mathbf{S}$, denotes a hyperbolic rotation (analogous to the "circular rotation") across the space and time dimensions. Fig. 3.7 shows a 2-dimensional example of circular rotation and hyperbolic rotation.

2 Depending on the determinant, a orthogonal matrix $\mathbf{U}$ denotes a rotation iff $\det(\mathbf{U}) = 1$ or a reflection iff $\det(\mathbf{U}) = -1$

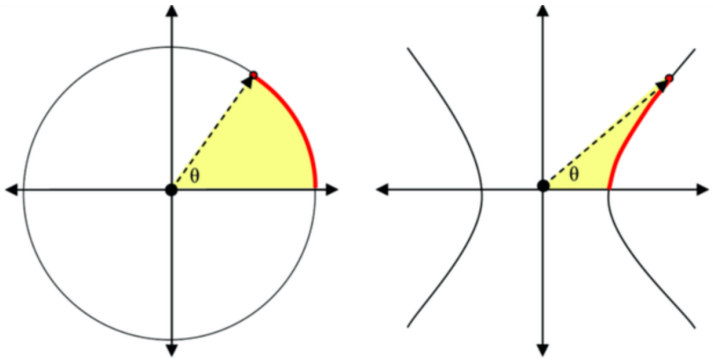


Figure 3.7: A two-dimensional example of circular rotation (*left*) and hyperbolic rotation (*right*), with θ being a angle.[3]

It is worth noting that both circular rotation/reflection and hyperbolic rotation are important operations for KG embeddings. On the one hand, circular rotations/reflections are able to model complex relational patterns including inversion, composition, symmetry, or anti-symmetry. Besides, these relational patterns usually form some non-hierarchies (e.g., cycles). Hence, circular rotations/reflections inherently encode non-hierarchical graph structures. Hyperbolic rotation, on the other hand, is able to model hierarchies, i.e., by connecting entities at different levels of hierarchies. Therefore, this decomposition shows that J-orthogonal transformation is powerful for representing both relational patterns and graph structures.

3.2.2.2 *Relation Parameterization*

Circular Rotation/Reflection. Parameterizing circular rotation or reflection via orthogonal matrices is non-trivial and there are some *trivialization* approaches such as using Cayley Transform [140]. However, such parameterization requires $\mathcal{O}(d^2)$ parameter complexity. To simplify the complexity, we consider Given transformations denoted by 2×2 matrices. Suppose the number of dimension p, q are even, circular rotation and reflection can be denoted by block-diagonal matrices of the form, given as

$$\begin{aligned} \mathbf{Rot}\left(\Theta_r\right) &= \operatorname{diag}\left(\mathbf{G}^{+}\left(\theta_{r,1}\right), \ldots, \mathbf{G}^{+}\left(\theta_{r,\frac{p+q}{2}}\right)\right) \\ \mathbf{Ref}\left(\Phi_r\right) &= \operatorname{diag}\left(\mathbf{G}^{-}\left(\phi_{r,1}\right), \ldots, \mathbf{G}^{-}\left(\phi_{r,\frac{p+q}{2}}\right)\right) \\ \text{where} \quad \mathbf{G}^{\pm}(\theta) &:= \begin{bmatrix} \cos(\theta) & \mp\sin(\theta) \\ \sin(\theta) & \pm\cos(\theta) \end{bmatrix} \end{aligned}, \tag{3.10}$$

where $\Theta_r := \left(\theta_{r,i}\right)_{i \in \left\{1, \ldots \frac{p+q}{2}\right\}}$ and $\Phi_r := \left(\phi_{r,i}\right)_{i \in \left\{1, \ldots \frac{p+q}{2}\right\}}$ are relation-specific parameters.

3 Source: https://en.m.wikipedia.org/wiki/File:Planar_rotations.png

Although circular rotation is theoretically able to infer symmetric patterns [168] (i.e., by setting rotation angle $\theta = \pi$ or $\theta = 0$), circular reflection can more effectively represent symmetric relations since their second power is the identity. AttH [168] combines circular rotations and circular reflections by using an attention mechanism learned in the tangent space, which requires additional parameters. We also combine circular rotation and circular reflection operators but in a different way. Since the J-orthogonal matrix is decomposed into two rotation/reflection matrices, we set the first matrix in Eq. (3.9) to be circular rotation while the third part to be circular reflection matrices, given by

$$\mathbf{U}_{\mathbf{\Theta_r}} = \begin{bmatrix} \mathbf{Rot}\left(\Theta_{r_p}\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{Rot}\left(\Theta_{r_q}\right) \end{bmatrix}, \quad \mathbf{V}_{\mathbf{\Phi_r}} = \begin{bmatrix} \mathbf{Ref}\left(\Phi_{r_p}\right) & \mathbf{0} \\ \mathbf{0} & \mathbf{Ref}\left(\Phi_{r_q}\right) \end{bmatrix}, \tag{3.11}$$

Clearly, the parameterization of circular rotation and reflection in Eq. (3.10), as well as the combination of them in Eq.(3.11), lose a certain degree of freedom of J-orthogonal transformation. However, it 1) results in a linear ($\mathcal{O}(d)$) memory complexity of relational embeddings; 2) significantly reduces the risk of overfitting; and 3) is sufficiently expressive to model complex relational patterns as well as graph structures. This is similar to many other Euclidean models, such as SimplE [83], that sacrifice some degree of freedoms of the multiplicative model (i.e., RESCAL [123]) parameterized by quadratic matrices while pursuing a linearly complex, less overfitting and highly expressive relational model.

Hyperbolic Rotation. The hyperbolic rotation matrix is parameterized by two diagonal matrices $\mathbf{C}, \mathbf{S}$ that satisfy the condition $\mathbf{C}^2 - \mathbf{S}^2 = \mathbf{I}$. The hyperbolic rotation matrix can be seen as a generalization of the 2×2 hyperbolic rotation given by $\begin{bmatrix} \cosh(\mu) & \sinh(\mu) \\ \sinh(\mu) & \cosh(\mu) \end{bmatrix}$, where the trigonometric functions sinh and cosh are hyperbolic versions of the sin and cos functions. Clearly, it satisfies the condition $\cosh(\mu)^2 - \sinh(\mu)^2 = 1$. Analogously, to satisfy the condition $\mathbf{C}^2 - \mathbf{S}^2 = \mathbf{I}$, we parameterize $\mathbf{C}, \mathbf{S}$ by diagonal matrices

$$\mathbf{C}(\mu) = \operatorname{diag}\left(\cosh(\mu_1), \ldots, \cosh(\mu_q)\right), \tag{3.12}$$

$$\mathbf{S}(\mu) = \operatorname{diag}\left(\sinh(\mu_1), \ldots, \sinh(\mu_q)\right), \tag{3.13}$$

where $\mu = (\mu_1, \cdots, \mu_q)$ is the parameter of hyperbolic rotation to learn. Therefore, the hyperbolic rotation matrix can be denoted by

$$\mathbf{H}_{\mu_r} = \begin{bmatrix} \operatorname{diag}\left(\cosh(\mu_{r,1}), \ldots, \cosh(\mu_{r,q})\right) & \mathbf{0} & \operatorname{diag}\left(\sinh(\mu_{r,1}), \ldots, \sinh(\mu_{r,q})\right) \\ \mathbf{0} & I_{p-q} & \mathbf{0}, \\ \operatorname{diag}\left(\sinh(\mu_{r,1}), \ldots, \sinh(\mu_{r,q})\right) & \mathbf{0} & \operatorname{diag}\left(\cosh(\mu_{r,1}), \ldots, \cosh(\mu_{r,q})\right) \end{bmatrix}, \tag{3.14}$$

Given the parameterization of each component, the final transformation function of relation r is given by

$$f_r = \mathbf{U}_{\theta_r} \mathbf{H}_{\mu_r} \mathbf{V}_{\Phi_r}. \tag{3.15}$$

Notably, the combination of circular rotation/reflection and hyperbolic rotation covers various kinds of geometric transformations, including circular, elliptic, parabolic, and hyperbolic transformations (See Fig. 3.6(a)). Hence, our relational embedding is able to work with all corresponding geometrical spaces.

3.2.2.3 *Objective and Manhattan-like Distance*

Objective Function. Given f_r and entity embeddings e, we design a score function for each triplet (h, r, t) as

$$s(h, r, t) = -d^2_{\mathbb{U}} \left(f_r \left(\mathbf{e}_h \right), \mathbf{e}_t \right) + b_h + b_t + \delta, \tag{3.16}$$

where $f_r \left(\mathbf{e}_h \right) = \mathbf{U}_{\theta_r} \mathbf{H}_{\text{-}_r} \mathbf{V}_{\Phi_r} \mathbf{e}_\mathbf{h}$, and $\mathbf{e}_h, \mathbf{e}_t \in \mathbb{U}^{p,q}$ are the embeddings of head entity h and tail entity t, $b_h, b_t \in \mathbf{R}^d$ are entity-specific biases, and each bias defines an entity-specific *sphere of influence* [12] surrounding the center point. δ is a global margin hyper-parameter. $d_{\mathbb{U}}(\cdot)$ is a function that quantifies the nearness/distance between two points in the ultrahyperbolic manifold.

Manhattan-like Distance. Defining a proper distance $d_{\mathbb{U}}(\cdot)$ in the ultrahyperbolic manifold is non-trivial. Different from Riemannian manifolds that are geodesically connected, ultrahyperbolic manifolds are not geodesically connected, and there exist *broken cases* in which the geodesic distance is not defined [97]. Some approximation approaches [97, 190] are proposed and satisfy some of the axioms of a classic metric (e.g., symmetric premetric). However, these distances suffer from the lack of the identity of indiscernibles, that is, one may have $\mathbf{d}_{\mathbb{U}}(\mathbf{x}, \mathbf{y}) = 0$ for some distinct points $\mathbf{x} \neq \mathbf{y}$. This is not a problem for metric learning that learns to preserve the pair-wise distances [97, 190]. However, our preliminary experiments find that the geodesic distance lacks the identity of indiscernibles results in unstable and non-convergent training. We conjecture this is due to the fact that the target of KG embedding is different from the graph embedding aiming at preserving pair-wise distance. KG embedding aims at satisfying $f_r(\mathbf{e}_h) \approx \mathbf{e}_t$ for each positive triple (h, r, t) while not for negative triples. Hence, we need to retain the *identity of indiscernibles*, that is, $\mathbf{d}_{\mathbb{U}}(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$.

To address this issue, we propose a novel Manhattan-like distance function, which is defined by a composition of a spherical distance

and a hyperbolic distance. Fig. 3.6 shows the Manhattan-like distance. Formally, given two points $\mathbf{x}, \mathbf{y} \in \mathbb{U}^{p,q}$, we first define a projection $\rho_{\mathbf{x}}(\mathbf{y})$ of $\mathbf{y}$ on the *circular conic section* crossing $\mathbf{x}$, such that $\rho_{\mathbf{x}}(\mathbf{y})$ and $\mathbf{x}$ share the same space dimension while $\rho_{\mathbf{x}}(\mathbf{y})$ and y lie on a hyperbolic subspace.

$$\rho_{\mathbf{x}}(\mathbf{y}) = \begin{pmatrix} \mathbf{x_p} \\ \alpha \mathbf{y}_q \frac{\|x_p\|}{\|y_p\|} \end{pmatrix}, \tag{3.17}$$

This projection makes that $\mathbf{x}$ and $\rho_{\mathbf{x}}(\mathbf{y})$ are connected by a purely space-like geodesic while $\rho_{\mathbf{x}}(\mathbf{y}), \mathbf{y}$ are connected by a purely time-like geodesic. The distance function of $\mathbb{U}$ hence can be defined as a Manhattan-like distance, i.e., the sum of the two distances, given as

$$d_{\mathbb{U}}(\mathbf{x}, \mathbf{y}) = \min\{d_{\mathbb{S}}(\mathbf{y}, \rho_{\mathbf{y}}(\mathbf{x})) + d_{\mathbb{H}}(\rho_{\mathbf{y}}(\mathbf{x}), \mathbf{x}), \tag{3.18}$$

$$d_{\mathbb{S}}(\mathbf{x}, \rho_{\mathbf{x}}(\mathbf{y})) + d_{\mathbb{H}}(\rho_{\mathbf{x}}(\mathbf{y}), \mathbf{y})\}, \tag{3.19}$$

where $d_{\mathbb{S}}$ and $d_{\mathbb{H}}$ are spherical and hyperbolic distances, respectively, which are well-defined and maintain the identity of indiscernibles.

3.2.2.4 *Optimization*

For each triplet (h, r, t), we create k negative samples by randomly corrupting its head or tail entity. The probability of a triple is calculated as $p = \sigma(s(h, r, t))$ where $\sigma(.)$ is a sigmoid function. We minimize the binary cross entropy loss, given as

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \left(\log p^{(i)} + \sum_{j=1}^{k} \log \left(1 - \tilde{p}^{(i,j)} \right) \right), \tag{3.20}$$

where $p^{(i)}$ and $\tilde{p}^{(i,j)}$ are the probabilities for positive and negative triplets respectively, and N is the number of samples.

Notably, directly optimizing the embeddings in ultrahyperbolic manifold is challenging. The issue is caused by the fact that there exist some points that cannot be connected by a geodesic in the manifold (hence no tangent direction for gradient descent). One way to sidestep the problem is to define entity embeddings in the Euclidean space and use a *diffeomorphism* to map the points into the manifold. In particular, we consider the following diffeomorphism.

Theorem 5. *[Diffeomorphism [190]] For any point $\mathbf{x} \in \mathbb{U}_{\alpha}^{p,q}$, there exists a diffeomorphism $\psi : \mathbb{U}_{\alpha}^{p,q} \to \mathbb{R}^p \times \mathbb{S}_{\alpha}^{q}$ that maps $\mathbf{x}$ into the product manifolds of a sphere and the Euclidean space. The mapping and its inverse are given by*

$$\psi(\mathbf{x}) = \begin{pmatrix} \mathbf{s} \\ \alpha \frac{\mathbf{t}}{\|\mathbf{t}\|} \end{pmatrix}, \quad \psi^{-1}(\mathbf{z}) = \begin{pmatrix} \mathbf{v} \\ \frac{\sqrt{\alpha^2 + \|\mathbf{v}\|^2}}{\alpha} \mathbf{u} \end{pmatrix}, \tag{3.21}$$

where $\mathbf{x} = \begin{pmatrix} \mathbf{s} \\ \mathbf{t} \end{pmatrix} \in \mathbb{U}_\alpha^{p,q}$ with $\mathbf{s} \in \mathbb{R}^p$ and $\mathbf{t} \in \mathbb{R}_*^q$. $\mathbf{z} = \begin{pmatrix} \mathbf{v} \\ \mathbf{u} \end{pmatrix} \in \mathbb{R}^p \times \mathbb{S}_\alpha^q$ with $\mathbf{v} \in \mathbb{R}^p$ and $\mathbf{u} \in \mathbb{S}_\alpha^q$.

With these mappings, any vector $\mathbf{x} \in \mathbb{R}^p \times \mathbb{R}_*^q$ can be mapped to $\mathbb{U}_\alpha^{p,q}$ by a double projection $\varphi = \psi^{-1} \circ \psi$. Note that since the diffeomorphism is differential and bijective, the canonical chain rule can be exploited to perform standard gradient descent optimization.

3.2.3 *Theoretical Analysis*

In this section, we provide some theoretical analyses of UltraE and the connections with related approaches.

3.2.3.1 *Complexity Analysis*

To make the model scalable to the size of the current KGs and keep up with their growth, a KG embedding model should have linear time and parameter complexity [18, 83]. In our case, the number of relation parameters of circular rotation, circular reflection, and hyperbolic rotation grows linearly with the dimensionality given by $p + q$. The total number of parameters is then $\mathcal{O}((N_e + N_r) \times d)$, where N_e and N_r are the numbers of entities and relations and $d = p + q$ is the embedding dimensionality. Similar to TransE [19] and RotH [26], UltraE has time complexity $\mathcal{O}(d)$. The additional cost is proportional to the number of relations, which is usually much smaller than the number of entities.

3.2.3.2 *Inference Patterns and Subsumption*

Following proposition shows that UltraE can infer various relational patterns

Proposition 2. *UltraE can infer symmetry, anti-symmetry, inversion, and composition relations.*

Detailed proofs and parameter settings of these inference patterns are given in the Appendix.

UltraE has close connections with some existing hyperbolic KG embedding methods, including HyboNet [31], RotH/RefH [26], and MuRP [12]. Essentially, we have the following two propositions (full derivations are in the Appendix).

Proposition 3. *UltraE, if parameterized by a full J-orthogonal matrix, generalizes HyboNet to support arbitrary signature.*

That is, HyboNet is the case of UltraE where $q = 1$.

Proposition 4. *HyboNet subsumes MuRP and RotH/RefH.*

3.2.4 *Empirical Evaluation*

In this section, we evaluate the performance of UltraE on link prediction in three KGs that contain both hierarchical and non-hierarchical relations. We systematically study the major components of our framework and show that (1) UltraE outperforms Euclidean and non-Euclidean baselines on embedding KGs with heterogeneous topologies, especially in low-dimensional cases (Sec. 3.2.4.2); (2) the signature of embedding space works as a knob for controlling the geometry, and hence influences the performance of UltraE (Sec. 3.2.4.3); (3) UltraE is able to improve the embeddings of relations with heterogeneous topologies (Sec. 3.2.4.3); and (4) the combination of rotation and reflection outperforms a single operator (Sec. 3.2.4.3).

3.2.4.1 *Experiment Setup*

Dataset. We use three standard benchmarks: WN18RR [19], a subset of WordNet containing 11 lexical relationships, FB15k-237 [19], a subset of Freebase containing general world knowledge, and YAGO3-10 [107], a subset of YAGO3 containing information of relationships between people. All three datasets contain hierarchical (e.g., *partOf*) and non-hierarchical (e.g., *similarTo*) relations, and some of which contain relational patterns like symmetry (e.g., *isMarriedTo*). For each KG, we follow the standard data augmentation protocol [94] and the same train/valid/test splitting as used in [94] for fair comparision. Following the previous work [26], we use the global graph curvature

Table 3.6: The statistics of KGs, where ξ_G measures the tree-likeness (the lower the ξ_G is, the more tree-like the KG is).

Dataset	#entities	#relations	#triples	ξ_G
WN18RR	41**k**	11	93**k**	-2.54
FB15k-237	15**k**	237	310**k**	-0.65
YAGO3-10	123**k**	37	1**M**	-0.54

[63] to measure the geometric properties of the datasets. The statistics of datasets are summarized in Table 3.6. As we can see, all datasets are globally hierarchical (i.e., the curvature is negative) but none of which is a pure tree structure. Comparatively, WN18RR is more hierarchical than FB15k-237 and YAGO3-10 since it has a smaller global graph curvature.

Evaluation protocol. Two popular ranking-based metrics are reported: 1) mean reciprocal rank (MRR), the mean of the inverse of the true entity ranking in the prediction; and 2) hit rate $H@K$ ($K \in \{1, 3, 10\}$), the percentage of the correct entities appearing in the top K ranked entities. As a standard, we report the metrics in the filtered setting [19], i.e., when calculating the ranking during evaluation, we filter out all true triples in the training set, since predicting a low rank for these triples should not be penalized.

Hyperparameters. For each KG, we explore batch size $\in \{500, 1000\}$, global margin $\in \{2, 4, 6, 8\}$ and learning rate $\in \{3e-3, 5e-3, 7e-3\}$ in the validation set. The negative sampling size is fixed to 50. The maximum number of epochs is set to 1000. The radius of curvature α is fixed to 1 since our model does not need relation-specific curvatures but is able to learn relation-specific mappings in the ultrahyperbolic manifold. The signature of the product manifold is set as the same as [172].

Baselines. Our baselines are divided into two groups:

- **Euclidean models**. 1) TransE [19], the first translational model; 2) RotatE [151], a rotation model in a complex space; 3) DistMult [192], a multiplicative model with a diagonal relational matrix; 4) ComplEx [156], an extension of DisMult in a complex space; 5) QuatE [22], a generalization of complex KG embedding in a hypercomplex space ; 6) 5⋆E that models a relation as five transformation functions; 7) MuRE [12], a Euclidean model with a diagonal relational matrix.

- **Non-Euclidean models.** 1) MuRP [12], a hyperbolic model with a diagonal relational matrix; 2) MuRS, a spherical analogy of MuRP; 3) RotH/RefH [26], a hyperbolic embedding with

Table 3.7: Link prediction results (%) on WN18RR, FB15k-237 and YAGO3-10 for low-dimensional embeddings ($d = 32$) in the filtered setting. The first group of models are Euclidean models, the second groups are non-Euclidean models, and MuRMP is a mixed-curvature baseline. RotatE, MuRE, MuRP, RotH, RefH and AttH results are taken from [26]. RotatE results are reported without self-adversarial negative sampling. The best score and best baseline are in **bold** and underlined, respectively.

	WN18RR				FB15k-237				YAGO3-10			
Model	MRR	H@ 1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE	36.6	27.4	43.3	51.5	29.5	21.0	32.2	46.6	-	-	-	-
RotatE	38.7	33.0	41.7	49.1	29.0	20.8	31.6	45.8	-	-	-	-
ComplEx	42.1	39.1	43.4	47.6	28.7	20.3	31.6	45.6	33.6	25.9	36.7	48.4
QuatE	42.1	39.6	43.0	46.7	29.3	21.2	32.0	46.0	-	-	-	-
5⋆E	44.9	41.8	46.2	51.0	32.3	24.0	35.5	50.1	-	-	-	-
MuRE	45.8	42.1	47.1	52.5	31.3	22.6	34.0	48.9	28.3	18.7	31.7	47.8
MuRP	46.5	42.0	48.4	54.4	32.3	23.5	35.3	50.1	23.0	15.0	24.7	39.2
RotH	47.2	42.8	49.0	55.3	31.4	22.3	34.6	49.7	39.3	30.7	43.5	55.9
RefH	44.7	40.8	46.4	51.8	31.2	22.4	34.2	48.9	38.1	30.2	41.5	53.0
AttH	46.6	41.9	48.4	55.1	32.4	23.6	35.4	50.1	39.7	31.0	43.7	56.6
MuRMP	47.0	42.6	48.3	54.7	31.9	23.2	35.1	50.2	39.5	30.8	42.9	56.6
UltraE (q=2)	48.1	43.4	50.0	55.4	33.1	24.1	35.5	50.3	39.5	31.2	43.9	56.8
UltraE (q=4)	**48.8**	**44.0**	**50.3**	**55.8**	33.4	24.3	36.0	51.0	40.0	31.5	44.3	57.0
UltraE (q=6)	48.3	42.5	49.1	55.5	**33.8**	**24.7**	**36.3**	**51.4**	**40.5**	**31.8**	**44.7**	**57.2**
UltraE (q=8)	47.5	42.3	49.0	55.1	32.6	24.6	36.2	51.0	39.4	31.3	43.4	56.5

rotation or reflection; 4) AttH [26], a combination of RotH and RefH by attention mechanism; 5) MuRMP [172], a generalization of MuRP in the product manifold.

We compare UltraE with varied signatures (time dimensions). Since for all KGs, the hierarchies are much more dominant than cyclicity and we assume that both space and time dimension are even numbers, we set the time dimension to be a relatively small value (i.e., $q = 2, 4, 6, 8$ and $q = 20, 40, 80, 160$ for low-dimension settings and high-dimension settings, respectively) for comparison. A full possible setting of time dimension with $q \leq p$ is studied in Sec. 3.2.4.3.

3.2.4.2 *Overall Results*

Low-dimensional Embeddings. Following previous approaches [26], we first evaluate UltraE in the low dimensional setting ($d = 32$). Table 3.7 shows the performance of UltraE and the baselines. Overall, it is clear that UltraE with varying time dimension ($q = 2, 4, 6, 8$) improves the performance of all methods. UltraE, even with only 2 time dimension, consistently outperforms all baselines, suggesting that the heterogeneous structure imposed by the pseudo-Riemannian geometry leads to better representations. In particular, the best performance of WN18RR is achieved by UltraE ($q = 4$) while the best performances of FB15k-237 and YAGO3-10 are achieved by UltraE ($q = 6$). We believe that this is because WN18RR is more hierarchical than FB15k-237

Table 3.8: Link prediction results (%) on WN18RR, FB15k-237 and YAGO3-10 for high-dimensional embeddings (best for $d \in \{200, 400, 500\}$) in the filtered setting. RotatE, MuRE, MuRP, RotH, RefH and AttH results are taken from [26]. RotatE results are reported without self-adversarial negative sampling. The best score and best baseline are in **bold** and underlined, respectively.

	WN18RR				FB15k-237				YAGO3-10			
Model	MRR	H@ 1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE	48.1	43.3	48.9	57.0	34.2	24.0	37.8	52.7	-	-	-	-
DistMult	43.0	39.0	44.0	49.0	24.1	15.5	26.3	41.9	34.0	24.0	38.0	54.0
RotatE	47.6	42.8	49.2	57.1	33.8	24.1	37.5	53.3	49.5	40.2	55.0	67.0
ComplEx	48.0	43.5	49.5	57.2	35.7	26.4	39.2	54.7	56.9	49.8	60.9	70.1
QuatE	48.8	43.8	50.8	58.2	34.8	24.8	38.2	55.0	-	-	-	-
5⋆E	50.0	**45.0**	51.0	59.0	**37.0**	**28.0**	**40.0**	56.0	-	-	-	-
MuRE	47.5	43.6	48.7	55.4	33.6	24.5	37.0	52.1	53.2	44.4	58.4	69.4
MuRP	48.1	44.0	49.5	56.6	33.5	24.3	36.7	51.8	35.4	24.9	40.0	56.7
RotH	49.6	44.9	51.4	58.6	34.4	24.6	38.0	53.5	57.0	49.5	61.2	70.6
RefH	46.1	40.4	48.5	56.8	34.6	25.2	38.3	53.6	57.6	50.2	61.9	**71.1**
AttH	48.6	44.3	49.9	57.3	34.8	25.2	38.4	54.0	56.8	49.3	61.2	70.2
MuRMP	48.1	44.1	49.6	56.9	35.8	27.3	39.4	56.1	49.5	44.8	59.1	69.8
UltraE (q=20)	48.5	44.2	50.0	57.3	34.9	25.1	38.5	54.1	56.9	49.5	61.0	70.3
UltraE (q=40)	**50.1**	**45.0**	**51.5**	**59.2**	35.1	27.5	**40.0**	56.0	57.5	49.8	62.0	70.8
UltraE (q=80)	49.7	44.8	51.2	58.5	36.8	27.6	**40.0**	**56.3**	**58.0**	**50.6**	**62.3**	**71.1**
UltraE (q=160)	48.6	44.5	50.3	57.4	35.4	26.0	39.0	55.5	57.0	49.5	61.8	70.5

and YAGO3-10, validating our conjecture that the number of time dimensions controls the geometry of the embedding space. Besides, we observed that the mixed-curvature baseline MuRMP does not consistently improve the hyperbolic methods. We conjecture that this is because MuRMP cannot properly model relational patterns.

High-dimensional Embeddings Table 3.8 shows the results of link prediction in high dimensions (best for $d \in \{200, 400, 500\}$). Overall, UltraE achieves either better or competitive results against a variety of other models. In particular, we observed that there is no significant performance gain among hyperbolic methods and mixed-curvature methods against Euclidean-based methods. We conjecture that this is because when the dimension is sufficiently large, both Euclidean and hyperbolic geometries have sufficient ability to represent complex hierarchies in KGs. However, UltraE roughly outperforms all compared approaches, with the only exception of 5⋆E achieving competitive results. Again, the performance gain is not as significant as in the low-dimension cases, which further validates the hypothesis that KG embeddings are not sensitive to the choice of embedding space with high dimensions. The additional performance gain might be obtained from the flexibility of inference of the relational patterns.

3.2.4.3 *Parameter Sensitivity*

The Effect of Dimensionality. To investigate the effect of dimensionality, we conduct experiments on WN18RR and compare UltraE ($q = 4$) against various state-of-the-art counterparts with varying dimension-

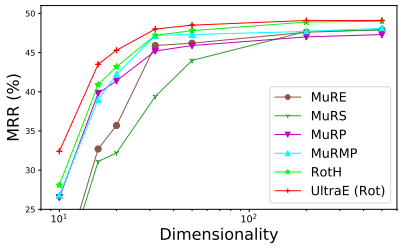


Figure 3.8: The MRR of various methods on WN18RR, with $d \in \{10, 16, 20, 32, 50, 200, 500\}$. UltraE is implemented with only rotation and $q = 4$. The results of MuRE, MuRS and MuRMP are taken from [172] with $d \in \{10, 15, 20, 40, 100, 200, 500\}$. All results are averaged over 10 runs.

ality. For a fair comparison with RotH that only considers rotation, we only use rotation for the implementation of UltraE, denoted by UltraE (Rot). Fig. 3.8 shows the results obtained by averaging over 10 runs. It clearly shows that the mixed-curvature method MuRMP outperforms its counterparts (MuRE, MuRP) with a single geometry, showcasing the limitation of a single homogeneous geometry on capturing the intrinsic heterogeneous structures. However, RotH performs slightly better than MuRMP, especially in high dimensionality, we conjecture that this is due to the capability of RotH on inferring relational patterns. UltraE achieves further improvements across a broad range of dimensions, suggesting the benefits of ultrahyperbolic manifold for modeling relation-specific geometries as well as inferring relational patterns.

The effect of signature. We study the influence of the signature on WN18RR by setting a varying number of time dimensions under the condition of $d = p + q = 32, p \geq q$. Fig. 3.9 shows that in all three benchmarks, by increasing q, the performance grows first and starts to decline after reaching a peak, which is consistent with our hypothesis that the signature acts as a knob for controlling the geometric properties. One might also note that compared with hyperbolic baselines (the dashed horizontal lines), the performance gain for WN18RR is relatively smaller than those of FB15k-237 and YAGO3-10. We conjecture that this is because WN18RR is more hierarchical than FB15k-237 and YAGO3-10, and the hyperbolic embedding performs already well. This assumption is further validated by the fact that the best time dimension of WN18RR ($q = 4$) is smaller than that of FB15k-237 and YAGO3-10 ($q = 6$).

The Effect of Relation Types. In this part, we investigate the per-relationship performance of UltraE on WN18RR. Similar to RotE and RotH that only consider rotation, we consider UltraE (Rot) as before. Two metrics that describe the geometric properties of each relation are reported, including global graph curvature and Krackhardt

Table 3.9: Comparison of H@10 for WN18RR relations. Higher $\mathbf{Khs}_G$ and lower ξ_G mean more hierarchical structure. UltraE is implemented by rotation and with best signature $(4, 28)$.

Relation	$\mathbf{Khs}_G$	ξ_G	RotE	RotH	UltraE (Rot)
member meronym	1.00	-2.90	32.0	39.9	41.3
hypernym	1.00	-2.46	23.7	27.6	28.6
has part	1.00	-1.43	29.1	34.6	36.0
instance hypernym	1.00	-0.82	48.8	52.0	53.2
member of domain region	1.00	-0.78	38.5	36.5	43.3
member of domain usage	1.00	-0.74	45.8	43.8	50.3
synset domain topic of	0.99	-0.69	42.5	44.7	46.3
also see	0.36	-2.09	63.4	70.5	73.5
derivationally related form	0.07	-3.84	96.0	96.8	97.1
similar to	0.07	-1.00	100.0	100.0	100.0
verb group	0.07	-0.50	97.4	97.4	98.0

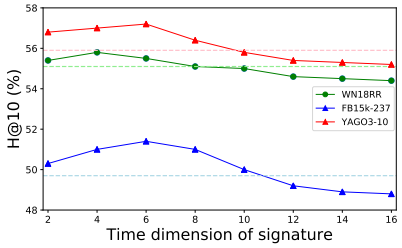


Figure 3.9: The performance (H@10) of UltraE with varied signature (time dimensions) under the condition of $d = p + q = 32, q \leq p$ on WN18RR. The dashed horizontal lines denote the results of RotH. As q increases, the performance first increases and starts to decrease after reaching a peak.

hierarchy score [26], for which higher $\mathbf{Khs}_G$ and lower ξ_G means more hierarchical. As shown in Table 3.9, although RotH outperforms RotE on most of the relation types, the performance is not on par with RotE on relations "member of domain region" and "member of domain usage". UltraE (Rot), however, consistently outperforms both RotE and RotH on all relations, with significant performance gains on relations "member of domain region " and "member of domain usage " that RotH fails on. The overall observation also verifies the flexibility and effectiveness of the proposed method in dealing with heterogeneous topologies of KGs.

The Effect of Rotation and Reflection. To investigate the role of rotation and reflection, we compare UltraE against its two variants: UltraE with only rotation (UltraE (Rot)) and UltraE with only reflection (UltraE (Ref)). Table 3.10 shows the per-relationship results on YAGO3-10. We observe that UltraE with rotation performs better on anti-symmetric relations while UltraE with reflection performs better on symmetric relations, suggesting that reflection is more suitable for representing symmetric patterns. On almost all relations including

Table 3.10: Comparison of H@10 on YAGO3-10 relations. UltraE (Rot) and UltraE (Ref) are implemented by only rotation and reflection, respectively. We choose the best signature $(6, 26)$.

Relation	Anti-symmetric	Symmetric	UltraE (Rot)	UltraE (Ref)	UltraE
hasNeighbor	×	✓	75.3	**100.0**	**100.0**
isMarriedTo	×	✓	94.0	94.4	**100.0**
actedIn	✓	×	14.7	12.7	**15.3**
hasMusicalRole	✓	×	43.5	37.0	**46.0**
directed	✓	×	51.5	45.3	**56.8**
graduatedFrom	✓	×	26.8	16.3	**27.5**
playsFor	✓	×	67.2	64.0	**66.8**
wroteMusicFor	✓	×	**28.4**	18.8	27.9
hasCapital	✓	×	**73.2**	68.3	**73.2**
dealsWith	×	×	30.4	29.7	**43.6**
isLocatedIn	×	×	41.5	39.8	**42.8**

relations that are neither symmetric nor anti-symmetric, except for "wroteMusicFor", UltraE outperforms both rotation or reflection variants, showcasing that combining multiple operators can learn more expressive representations.

3.2.5 *Conclusion*

We proposes UltraE, an ultrahyperbolic KG embedding method in a pseudo-Riemannian manifold that interleaves hyperbolic and spherical geometries, allowing for simultaneously modeling multiple hierarchical and non-hierarchical structures in KGs. We derive a relational embedding by exploiting the pseudo-orthogonal transformation, which is decomposed into various geometric operators including circular rotations/reflections and hyperbolic rotations, allowing for inferring complex relational patterns in KGs. We propose a Manhattan-like distance that measures the nearness of points in the ultrahyperbolic manifold. The embeddings are optimized by standard gradient descent thanks to the differentiable and bijective mapping. We discuss theoretical connections of UltraE with other hyperbolic methods. On three standard KG datasets, UltraE outperforms many previous Euclidean and non-Euclidean counterparts, especially in low-dimensional settings.

4

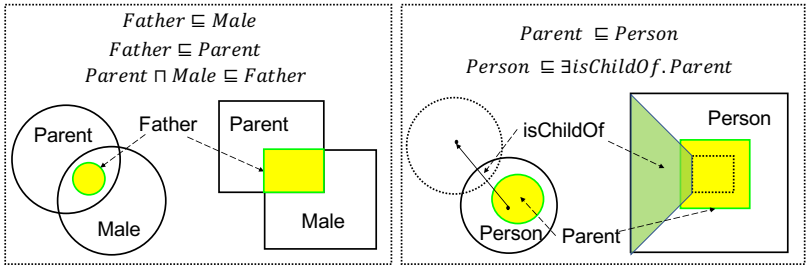
GEOMETRIC EMBEDDING OF LOGICAL STRUCTURES

Recently, increasing efforts are put into learning continual representations for symbolic knowledge bases (KBs). However, these approaches either only embed the data-level knowledge or suffer from inherent limitations when dealing with concept-level knowledge, i.e., they cannot faithfully model the logical structure present in the KBs. We present BoxEL, a geometric KB embedding approach that allows for better capturing the logical structure in the description logic $\mathcal{EL}^{++}$. BoxEL models concepts in a KB as axis-parallel *boxes* that are suitable for modeling concept intersection, entities as points inside boxes, and relations between concepts/entities as *affine transformations*. We show theoretical guarantees (*soundness*) of BoxEL for preserving logical structure. Namely, the learned model of BoxEL embedding with loss 0 is a (logical) model of the KB. Experimental results on (plausible) subsumption reasonings and a real-world application for protein-protein prediction show that BoxEL outperforms traditional knowledge graph embedding methods as well as state-of-the-art $\mathcal{EL}^{++}$ embedding approaches.

4.1 MOTIVATION AND BACKGROUND

Knowledge bases (KBs) provide a *conceptualization* of objects and their relationships, which are of great importance in many applications like biomedical and intelligent systems [41, 134]. KBs are often expressed using description logics (DLs) [8], a family of languages allowing for expressing domain knowledge via logical statements (a.k.a axioms). These logical statements are divided into two parts: 1) an ABox consisting of *assertions* over instances, i.e., factual statements like $\mathsf{isFatherOf}(\mathsf{John}, \mathsf{Peter})$; 2) a TBox consisting of logical statements constraining concepts, e.g., $\mathsf{Parent} \sqsubseteq \mathsf{Person}$.

KBs not only provide clear semantics in the application domains but also enable (classic) reasoners [82, 147] to perform logical inference, i.e., making implicit knowledge explicit. Existing reasoners are highly optimized and scalable but they are limited to only computing classical logical entailment but not designed to perform inductive (analogical)



(a) Ball and Box embedding (b) Translation and affine transformation

Figure 4.1: Two counterexamples of ball embedding and its relational transformation. (a) Ball embedding cannot express concept equivalence Parent ⊓ Male ≡ Father with intersection operator. (b) The *translation* cannot model relation (e.g. isChildOf) between Person and Parent when they should have different volumes. These two issues can be solved by *box* embedding and modelling relation as *affine transformation* among boxes, respectively.

reasoning and cannot handle noisy data. Embedding based methods, which map the objects in the KBs into a low dimensional vector space while keeping the similarity, have been proposed to complement the classical reasoners and shown remarkable empirical performances on performing (non-classical) analogical reasonings.

Most KB embeddings methods [169] focus on embedding data-level knowledge in ABoxes, a.k.a., knowledge graph embeddings (KGEs). However, KGEs cannot preserve concept-level knowledge expressed in TBoxes. Recently, embedding methods for KBs expressed in DLs have been explored. Prominent examples include $\mathcal{EL}^{++}$ [92] that supports conjunction and full existential quantification, and $\mathcal{ALC}$ [127] that further supports logical negation. We focus on $\mathcal{EL}^{++}$, an underlying formalism of the OWL2 EL profile of the Web Ontology Language [60], which has been used in expressing various biomedical ontologies [41, 134]. For embedding $\mathcal{EL}^{++}$ KBs, several approaches such as Onto2Vec [144] and OPA2Vec [145] have been proposed. These approaches require annotation data and do not model logical structure explicitly. Geometric representations, in which the objects are associated with geometric objects such as balls [92] and convex cones [127], provide a high expressiveness on embedding logical properties. For $\mathcal{EL}^{++}$ KBs, ELEm [92] represents concepts as open n-balls and relations as simple translations. Although effective, ELEm suffers from several major limitations:

- Balls are not closed under intersection and cannot faithfully represent concept intersections. For example, the intersection of two concepts Parent ⊓ Male, that is supposed to represent Father, is not a ball (see Fig. 4.1(a)). Therefore, the concept equivalence Parent ⊓ Male ≡ Father cannot be captured in the embedding.

- The relational embedding with simple *translation* causes issues for embedding concepts with varying sizes. E.g., Fig. 4.1(b) shows the embeddings of the axiom $\exists \mathsf{isChildOf}.\mathsf{Person} \sqsubseteq \mathsf{Parent}$ assuming the existence of another axiom $\mathsf{Parent} \sqsubseteq \mathsf{Person}$. In this case, it is impossible to *translate* the larger concept Person to the smaller one Parent,[1] as it does not allow for scaling the size.

- ELEm does not distinguish between entities in ABox and concepts in TBox, but rather regards ABox axioms as special cases of TBox axioms. This simplification cannot fully express the logical structure, e.g., an entity must have minimal volume.

To overcome these limitations, we consider modeling concepts in the KB as *boxes* (i.e., axis-aligned hyperrectangles), encoding entities as points inside the boxes that they should belong to, and the relations as the *affine transformation* between boxes and/or points. Fig. 4.1(a) shows that the box embedding has closed form of intersection and the *affine transformation* (Fig. 4.1(b)) can naturally capture the cases that are not possible in ELEm. In this way, we present BoxEL for embedding $\mathcal{EL}^{++}$ KBs, in which the interpretation functions of $\mathcal{EL}^{++}$ theories in the KB can be represented by the geometric transformations between boxes/points. We formulate BoxEL as an optimization task by designing and minimizing various loss terms defined for each logical statement in the KB. We show theoretical guarantee (*soundness*) of BoxEL in the sense that if the loss of BoxEL embedding is 0, then the trained model is a (logical) model of the KB. Experiments on (plausible) subsumption reasoning over three ontologies and predicting protein-protein interactions show that BoxEL outperforms previous approaches.

4.2 BOXEL FOR EMBEDDING $\mathcal{EL}^{++}$ KNOWLEDGE BASES

In this section, we first present the geometric construction process of $\mathcal{EL}^{++}$ with box embedding and affine transformation, followed by a discussion of the geometric interpretation. Afterward, we describe the BoxEL embedding by introducing proper loss function for each ABox and TBox axiom. Finally, an optimization method is described for the training of BoxEL.

1 Under the *translation* setting, the embeddings will simply become $\mathsf{Parent} \equiv \mathsf{Person}$, which is obviously not what we want as we can express $\mathsf{Parent} \not\equiv \mathsf{Person}$ with $\mathcal{EL}^{++}$ by propositions like $\mathsf{Children} \sqcap \mathsf{Parent} \sqsubseteq \bot$, $\mathsf{Children} \sqsubseteq \mathsf{Person}$ and $\mathsf{Children}(a)$

4.2.1 *Geometric Construction*

We consider a KB $(\mathcal{A}, \mathcal{T})$ consisting of an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$ where $\mathcal{T}$ has been normalized as explained before. Our goal is to associate entities (or individuals) with points and concepts with boxes in $\mathbb{R}^n$ such that the axioms in the KB are respected.

To this end, we consider two functions m_w, M_w parameterized by a parameter vector w that has to be learned. Conceptually, we consider points as boxes of volume 0. This will be helpful later to encode the meaning of axioms for points and boxes in a uniform way. Intuitively, $m_w : N_I \cup N_C \to \mathbb{R}^n$ maps individual and concept names to the lower left corner and $M_w : N_I \cup N_C \to \mathbb{R}^n$ maps them to the upper right corner of the box that represents them. For individuals $a \in N_I$, we have $m_w(a) = M_w(a)$, so that it is sufficient to store only one of them. The *box associated with* C is defined as

$$\text{Box}_w(C) = \{x \in \mathbb{R}^n \mid m_w(C) \leq x \leq M_w(C)\}, \tag{4.1}$$

where the inequality is defined component-wise.

Note that boxes are closed under intersection, which allows us to compute the volume of the intersection of boxes. The lower corner of the box $\text{Box}_w(C) \cap \text{Box}_w(D)$ is $\max(m_w(C), m_w(D))$ and the upper corner is $\min(M_w(C), M_w(D))$, where minimum and maximum are taken component-wise. The volume of boxes can be used to encode axioms in a very concise way. However, as we will describe later, one problem is that points have volume 0. This does not allow distinguishing empty boxes from points. To show that our encoding correctly captures the logical meaning of axioms, we will consider a *modified volume* that assigns a non-zero volume to points and some empty boxes. The (modified) volume of a box is defined as

$$\text{MVol}(\text{Box}_w(C)) = \prod_{i=1}^{n} \max(0, M_w(C)_i - m_w(C)_i + \epsilon), \tag{4.2}$$

where $\epsilon > 0$ is a small constant. A point now has volume ϵ^n. Some empty boxes can actually have arbitrarily large modified volume. For example the 2D-box with lower corner $(0, 0)$ and upper corner $(-\frac{\epsilon}{2}, N)$ has volume $\frac{\epsilon \cdot N}{2}$. While this is not meaningful geometrically, it does not cause any problems for our encoding because we only want to ensure that boxes with zero volume are empty (and not points). In practice, we will use *softplus volume* as approximation (see Sec. 4.2.2.3).

We associate every role name $r \in N_r$ with an affine transformation denoted by $T_w^r(x) = D_w^r x + b_w^r$, where D_w^r is an $(n \times n)$ diagonal matrix

with non-negative entries and $b^r_w \in \mathbb{R}^n$ is a vector. In a special case where all diagonal entries of D^r_w are 1, $T^r_w(x)$ captures translations. Note that relations have been represented by translation vectors analogous to TransE in [92]. However, this necessarily means that the concept associated with the range of a role has the same size as its domain. This does not seem very intuitive, in particular, for N-to-one relationships like *has_nationality* or *lives_in* that map many objects to the same object. Note that $T^r_w(\text{Box}_w(C)) = \{T^r_w(x) \mid x \in \text{Box}_w(C)\}$ is the box with lower corner $T^r_w(m_w(C))$ and upper corner $T^r_w(M_w(C))$. To show this, note that $m_w(C) < M_w(C)$ implies $D^r_w m_w(C) \leq D^r_w M_w(C)$ because D^r_w is a diagonal matrix with non-negative entries. Hence, $T^r_w(m_w(C)) = D^r_w m_w(C) + b^r_w \leq D^r_w M_w(C) + b^r_w = T^r_w(M_w(C))$. For $m_w(C) \geq M_w(C)$, both $\text{Box}_w(C)$ and $T^r_w(\text{Box}_w(C))$ are empty.

Overall, we have the following parameters:

- for every individual name $a \in N_I$, we have n parameters for the vector $m_w(a)$ (since $m_w(a) = M_w(A)$, we have to store only one of m_w and M_w),
- for every concept name $C \in N_C$, we have $2n$ parameters for the vectors $m_w(C)$ and $M_w(C)$,
- for every role name $r \in N_r$, we have $2n$ parameters. n parameters for the diagonal elements of D^r_w and n parameters for the components of b^r_w.

As we explained informally before, w summarizes all parameters. The overall number of parameters in w is $n \cdot (|N_I| + 2 \cdot |N_C| + 2 \cdot |N_r|)$.

4.2.2 *Geometric Interpretation*

The next step is to encode the axioms in our KB. However, we do not want to do this in an arbitrary fashion, but, ideally, in a way that gives us some analytical guarantees. [92] made an interesting first step by showing that their encoding is *sound*. In order to understand soundness, it is important to know that the parameters of the embedding are learnt by minimizing a loss function that contains a loss term for every axiom. Soundness then means that if the loss function yields 0, then the KB is satisfiable. Recall that satisfiability means that there is an interpretation that satisfies all axioms in the KB. Ideally, we should be able to construct such an interpretation directly from our embedding. This is indeed what the authors in [92] did. The idea is that points in the vector space make up the domain of the

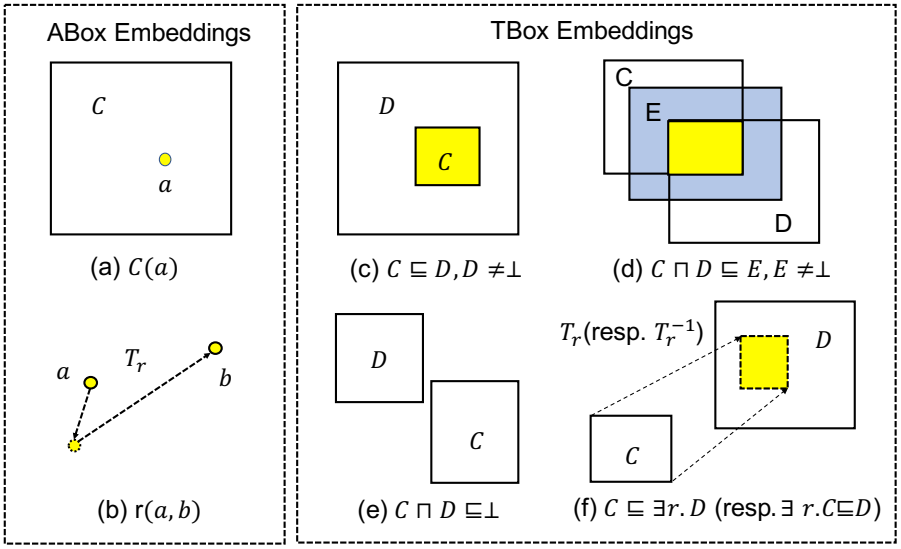


Figure 4.2: The geometric interpretation of logical statements in ABox (*left*) and TBox (*right*) expressed by DL $\mathcal{EL}^{++}$ with BoxEL embeddings. The concepts are represented by *boxes*, entities are represented by *points* and relations are represented by *affine transformations*. T_r and T_r^{-1} denote the transformation function of relation r and its inverse function, respectively.

interpretation, the points that lie in regions associated with concepts correspond to the interpretation of this concept and the interpretation of roles correspond to translations between points like in TransE. In our context, geometric interpretation can be defined as follows.

Definition 18 (Geometric Interpretation)**.** *Given a parameter vector w representing an $\mathcal{EL}^{++}$ embedding, the corresponding* geometric interpretation $\mathcal{I}_w = (\Delta^{\mathcal{I}_w}, \cdot^{\mathcal{I}_w})$ *is defined as follows:*

1. $\Delta^{\mathcal{I}_w} = \mathbb{R}^n$,
2. *for every concept name* $C \in N_C$, $C^{\mathcal{I}_w} = \text{Box}_w(C)$,
3. *for every role* $r \in N_R$, $r^{\mathcal{I}_w} = \{(x, y) \in \Delta^{\mathcal{I}_w} \times \Delta^{\mathcal{I}_w} \mid T_w^r(x) = y\}$,
4. *for every individual name* $a \in N_I$, $a^{\mathcal{I}_w} = m_w(a)$.

We will now encode the axioms by designing one loss term for every axiom in a normalized $\mathcal{EL}^{++}$ KB, such that the axiom is satisfied by the geometric interpretation when the loss is 0.

4.2.2.1 *ABox Embedding*

ABox contains concept assertions and role assertions. We introduce the following two loss terms that respect the geometric interpretations.
Concept Assertion Geometrically, a concept assertion $C(a)$ asserts that the point $m_w(a)$ is inside the box $\text{Box}_w(C)$ (see Fig. 4.2(a)). This can be expressed by demanding $m_w(C) \leq m_w(a) \leq M_w(C)$ for every component. The loss $\mathcal{L}_{C(a)}(w)$ is defined by

$$\begin{aligned}\mathcal{L}_{C(a)}(w) = &\sum_{i=1}^{n} \|\max(0, m_w(a)_i - M_w(C)_i)\|_2 \\ &+ \sum_{i=1}^{n} \|\max(0, m_w(C)_i - m_w(a)_i)\|_2 \,. \end{aligned} \tag{4.3}$$

Role Assertion Geometrically, a role assertion $r(a, b)$ means that the point $m_w(a)$ should be mapped to $m_w(b)$ by the transformation T^r_w (see Fig. 4.2(b)). That is, we should have $T^r_w(m_w(a)) = m_w(b)$. We define a loss term

$$\mathcal{L}_{r(a,b)}(w) = \|T^r_w(m_w(a)) - m_w(b)\|_2 \,. \tag{4.4}$$

It is clear from the definition that when the loss terms are 0, the axioms are satisfied in their geometric interpretation.

Proposition 5. *We have*

1. *If* $\mathcal{L}_{C(a)}(w) = 0$, *then* $\mathcal{I}_w \models C(a)$,
2. *If* $\mathcal{L}_{r(a,b)}(w) = 0$, *then* $\mathcal{I}_w \models r(a, b)$.

4.2.2.2 *TBox Embedding*

For the TBox, we define loss terms for the four cases in the normalized KB. Before doing so, we define an auxiliary function that will be used inside these loss terms.

Definition 19 (Disjoint measurement)**.** *Given two boxes* B_1, B_2, *the disjoint measurement can be defined by the (modified) volumes of* B_1 *and the intersection box* $B_1 \cap B_2$,

$$\text{Disjoint}(B_1, B_2) = 1 - \frac{\text{MVol}(B_1 \cap B_2)}{\text{MVol}(B_1)}. \tag{4.5}$$

We have the following guarantees.

Lemma 1. 1. $0 \leq \text{Disjoint}(B_1, B_2) \leq 1$,

2. $\text{Disjoint}(B_1, B_2) = 0$ *implies* $B_1 \subseteq B_2$,

3. $\text{Disjoint}(B_1, B_2) = 1$ *implies* $B_1 \cap B_2 = \emptyset$.

NF1: Atomic Subsumption An axiom of the form $C \sqsubseteq D$ geometrically means that $\text{Box}_w(C) \subseteq \text{Box}_w(D)$ (see Fig. 4.2(c)). If $D \neq \bot$, we consider the loss term

$$\mathcal{L}_{C \sqsubseteq D}(w) = \text{Disjoint}(\text{Box}_w(C), \text{Box}_w(D)). \tag{4.6}$$

For the case $D = \bot$ where C is not a nominal, e.g., $C \sqsubseteq \bot$, we define the loss term

$$\mathcal{L}_{C \sqsubseteq \bot}(w) = \max(0, M_w(C)_0 - m_w(C)_0 + \epsilon). \tag{4.7}$$

If C is a nominal, the axiom is inconsistent and our model can just return an error.

Proposition 6. *If* $\mathcal{L}_{C \sqsubseteq D}(w) = 0$, *then* $\mathcal{I}_w \models C \sqsubseteq D$, *where we exclude the inconsistent case* $C = \{a\}, D = \bot$.

NF2: Conjunct Subsumption An axiom of the form $C \sqcap D \sqsubseteq E$ means that $\text{Box}(C) \cap \text{Box}(D) \subseteq \text{Box}(E)$ (see Fig. 4.2(d)). Since $\text{Box}(C) \cap \text{Box}(D)$ is a box again, we can use the same idea as for NF1. For the case $E \neq \bot$, we define the loss term as

$$\mathcal{L}_{C \sqcap D \sqsubseteq E}(w) = \text{Disjoint}(\text{Box}_w(C) \cap \text{Box}_w(D), \text{Box}_w(E)). \tag{4.8}$$

For $E = \bot$, the axiom states that C and D must be disjoint. The disjointedness can be interpreted as the volume of the intersection of the associated boxes being 0 (see Fig. 4.2(e)). However, just using the volume as a loss term may not work well because a minimization algorithm may minimize the volume of the boxes instead of the volume of their intersections. Therefore, we normalize the loss term by dividing by the volume of the boxes. Given by

$$\mathcal{L}_{C \sqcap D \sqsubseteq \bot}(w) = \frac{\text{MVol}(\text{Box}_w(C) \cap \text{Box}_w(D))}{\text{MVol}(\text{Box}_w(C)) + \text{MVol}(\text{Box}_w(D))}. \tag{4.9}$$

Proposition 7. *If* $\mathcal{L}_{C \sqcap D \sqsubseteq E}(w) = 0$, *then* $\mathcal{I}_w \models C \sqcap D \sqsubseteq E$, *where we exclude the inconsistent case* $a \sqcap a \sqsubseteq \bot$ *(that is,* $C = D = \{a\}, E = \bot$*).*

NF3: Right Existential Next, we consider axioms of the form $C \sqsubseteq \exists r.D$. Note that $\exists r.D$ describes those entities that are in relation r with an entity from D. Geometrically, those are points that are mapped to points in $\text{Box}_w(D)$ by the affine transformation corresponding to r. $C \sqsubseteq \exists r.D$ then means that every point in $\text{Box}_w(C)$ must be mapped to a point in $\text{Box}_w(D)$, that is the mapping of $\text{Box}_w(C)$ is contained in $\text{Box}_w(D)$ (see Fig. 4.2(f)). Therefore, the encoding comes again down to encoding a subset relationship as before. The only difference to the first normal form is that $\text{Box}_w(C)$ must be mapped by the affine transformation T^r_w. These considerations lead to the following loss term

$$\mathcal{L}_{C \sqsubseteq \exists r.D}(w) = \text{Disjoint}(T^r_w(\text{Box}_w(C)), \text{Box}_w(D)). \tag{4.10}$$

Proposition 8. *If* $\mathcal{L}_{C \sqsubseteq \exists r.D}(w) = 0$, *then* $\mathcal{I}_w \models C \sqsubseteq \exists r.D$.

NF4: Left Existential Axioms of the form $\exists r.C \sqsubseteq D$ can be treated symmetrically to the previous case (see Fig. 4.2(f)). We only consider the case $D \neq \bot$ and define the loss

$$\mathcal{L}_{\exists r.C \sqsubseteq D}(w) = \text{Disjoint}(T^{-r}_w(\text{Box}_w(C)), \text{Box}_w(D)), \tag{4.11}$$

where T^{-r}_w is the inverse function of T^r_w that is defined by $T^{-r}_w(x) = D^{-r}_w x - D^{-r}_w b^r_w$, where D^{-r}_w is obtained from D^r_w by replacing all diagonal elements with their reciprocal. Strictly speaking, the inverse only exists if all diagonal entries of D^r_w are non-zero. However, we assume that the entries that occur in a loss term of the form $\mathcal{L}_{\exists r.C \sqsubseteq D}(w)$ remain non-zero in practice when we learn them iteratively.

Proposition 9. *If* $\mathcal{L}_{\exists r.C \sqsubseteq D}(w) = 0$, *then* $\mathcal{I}_w \models \exists r.C \sqsubseteq D$.

4.2.2.3 *Optimization*

Softplus Approximation For optimization, while the computation of the volume of boxes is straightforward, using a precise *hard volume* is known to cause problems when learning the parameters using gradient descent algorithms, e.g. there is no training signal (gradient flow) when box embeddings that should overlap but become disjoint [43, 98, 131]. To mitigate the problem, we approximate the volume of boxes by the *softplus volume* [131] due to its simplicity.

$$\text{SVol}\left(\text{Box}_w\left(C\right)\right) = \prod_{i=1}^{n} \text{Softplus}_t\left(M_w\left(C\right)_i - m_w\left(C\right)_i\right) \tag{4.12}$$

where t is a temperature parameter. The softplus function is defined as $\text{softplus}_t(x) = t \log\left(1 + e^{x/t}\right)$, which can be regarded as a smoothed

version of the ReLu function ($\max\{0, x\}$) used for calculating the volume of *hard boxes*. In practice, the *softplus volume* is used to replace the *modified volume* in Eq.(4.2) as it empirically resolves the same issue that point has zero volume.

Regularization We add a regularization term in Eq.(4.13) to all non-empty boxes to encourage that the boxes lie in the unit box $[0, 1]^n$.

$$\lambda = \sum_{i=1}^{n} \max(0, M_w(C)_i - 1 + \epsilon) + \max(0, -m_w(C)_i - \epsilon) \qquad (4.13)$$

In practice, this also avoids numerical stability issues. For example, to minimize a loss term, a box that should have a fixed volume could become very *slim*, i.e. some side lengths be extremely large while others become extremely small.

Negative Sampling In principle, the embeddings can be optimized without negatives. However, we empirically find that the embeddings will be highly overlapped without negative sampling. e.g. for role assertion $r(a, b)$, a and b will simply become the same point. We generate negative samples for the role assertion $r(a, b)$ by randomly replacing one of the head or tail entity. Finally, we sum up all the loss terms, and learn the embeddings by minimizing the loss with Adam optimizer [84].

4.3 EMPIRICAL EVALUATION

4.3.1 *A Proof-of-concept Example*

We begin by first validating the model in modeling a toy ontology–family domain [92], which is described by the following axioms:[2]

$\mathsf{Male} \sqsubseteq \mathsf{Person}$	$\mathsf{Female} \sqsubseteq \mathsf{Person}$
$\mathsf{Father} \sqsubseteq \mathsf{Male}$	$\mathsf{Mother} \sqsubseteq \mathsf{Female}$
$\mathsf{Father} \sqsubseteq \mathsf{Parent}$	$\mathsf{Mother} \sqsubseteq \mathsf{Parent}$
$\mathsf{Female} \sqcap \mathsf{Male} \sqsubseteq \bot$	$\mathsf{Female} \sqcap \mathsf{Parent} \sqsubseteq \mathsf{Mother}$
$\mathsf{Male} \sqcap \mathsf{Parent} \sqsubseteq \mathsf{Father}$	$\exists \mathsf{hasChild.Person} \sqsubseteq \mathsf{Parent}$
$\mathsf{Parent} \sqsubseteq \mathsf{Person}$	$\mathsf{Parent} \sqsubseteq \exists\ \mathsf{hasChild.Person}$
$\mathsf{Father}(\mathrm{Alex})$	$\mathsf{Father}(\mathrm{Bob})$
$\mathsf{Mother}(\mathrm{Marie})$	$\mathsf{Mother}(\mathrm{Alice})$

We set the dimension to 2 to visualize the embeddings. Fig. 4.3 shows that the generated embeddings accurately encode all of the axioms. In

2 Compared with the example given in [92], we add additional concept assertion statements that distinguish entities and concepts:

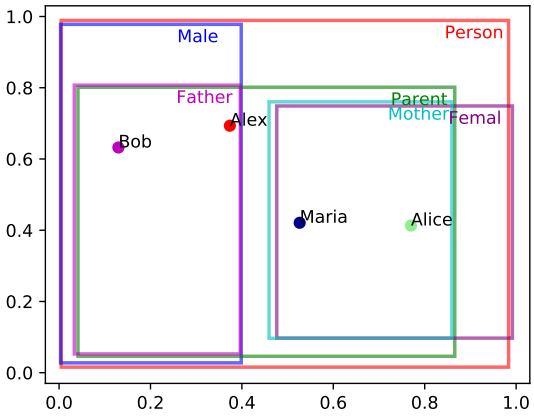


Figure 4.3: BoxEL embeddings in the family domain.

particular, the embeddings of Father and Mother align well with the conjunction Parent ⊓ Male and Parent ⊓ Female, respectively, which is impossible to be achieved by ELEm.

4.3.2 *Subsumption Reasoning*

We evaluate the effectiveness of BoxEL on (plausible) subsumption reasoning (also known as ontology completion). The problem is to predict whether a concept is subsumed by another one. For each subsumption pair $C \sqsubseteq D$, the scoring function can be defined by

$$P(C \sqsubseteq D) = \frac{\text{MVol}(\text{Box}(C) \cap \text{Box}(D))}{\text{MVol}(\text{Box}(C))}. \tag{4.14}$$

Note that such subsumption relations are not necessary to be (logically) entailed by the input KB, e.g., a subsumption relation can be plausibly inferred by $P(C \sqsubseteq D) = 0.9$, allowing for non-classical plausible reasoning. While the subsumption reasoning does not need negatives, we add an additional regularization term for non-subsumption axiom. In particular, for each atomic subsumption axiom $C \sqsubseteq D$, we generate a non-subsumption axiom $C \not\sqsubseteq D'$ or $C' \not\sqsubseteq D$ by randomly replacing one of the concepts C and D. Note that this does not produce regular negative samples as the generated concepts pair does not have to be disjoint. Thus, the loss term for non-subsumption axiom cannot be simply defined by $\mathcal{L}_{C \not\sqsubseteq D'} = 1 - \mathcal{L}_{C \sqsubseteq D'}$. Instead, we define the loss term as $\mathcal{L}_{C \not\sqsubseteq D'} = \phi(1 - \mathcal{L}_{C \sqsubseteq D'})$ by multiplying a small positive constant ϕ that encourages splitting the non-subsumption concepts while does not encourage them to be disjoint. If $\phi = 1$, the loss would encourage the non-subsumption concepts to be disjoint. We

Table 4.1: Summary of classes, relations and axioms in different ontologies. NF_i represents the i^{th} normal form.

Ontology	GO	GALEN	ANATOMY
Classes	45895	24353	106363
Relations	9	1010	157
NF1	85480	28890	122142
NF2	12131	13595	2121
NF3	20324	28118	152289
NF4	12129	13597	2143
Disjoint	30	0	184

Table 4.2: The accuracies achieved by the embeddings of different approaches in terms of geometric interpretation of the classes in various ontologies.

	ELEm	EmEL^{++}	BoxEL
GO	0.250	0.415	**0.489**
GALEN	0.480	0.345	**0.788**
ANATOMY	0.069	0.215	**0.453**

empirically show that $\phi = 1$ produces worse performance as we do not want non-subsumption concepts to be disjoint.

Datasets We use three biomedical ontologies as our benchmark. 1) *Gene Ontology (GO)* [72] integrates the representation of genes and their functions across all species. 2) *GALEN* [134] is a clinical ontology. 3) *Anatomy* [115] is a ontology that represents linkages of different phenotypes to genes. Table 4.1 summarizes the statistical information of these datasets. The subclass relations are split into training set (70%), validation set (20%) and testing set (10%), respectively.

Evaluation protocol Two strategies can be used to measure the effectiveness of the embeddings. 1) Ranking based measures rank the probability of C subsumed by all concepts. We evaluate and report four ranking based measures. Hits@10, Hits@100 describe the fraction of true cases that appear in the first 10 and 100 test cases of the sorted rank list, respectively. Mean rank computes the arithmetic mean over all individual ranks (i.e. MR $= \frac{1}{|\mathcal{I}|} \sum_{\text{rank} \in \mathcal{I}} \text{rank}$, where *rank* is the individual rank), while AUC computes the area under the ROC curve. 2) Accuracy based measure is a stricter criterion, for which the prediction is true if and only if the subclass box is exactly inside the superclass box (even not allowing the subclass box slightly outside the superclass box). We use this measure as it evaluates the performance of embeddings on retaining the underlying characteristics of ontology in vector space. We only compare ELEm and EmEL^{++} as KGE baselines fail in this setting (KGEs cannot preserve the ontology).

Implementation details The ontology is normalized into standard normal forms, which comprise a set of axioms that can be used as the *positive samples*. Similar to previous works [92], we perform normalization using the OWL APIs and the APIs offered by the jCel reasoner [18]. The hyperparameter for negative sampling is set to $\phi = 0.05$. For ELEm and EmEL^{++}, the embedding size is searched from $n = [50, 100, 200]$ and margin parameter is searched from $\gamma = [-0.1, 0, 0.1]$. Since box embedding has double the number

Table 4.3: The ranking based measures of embedding models for sumbsumtion reasoning on the testing set. $*$ denotes the results from [114].

Dataset	Metric	TransE$*$	TransH$*$	DistMult$*$	ELEm	EmEL^{++}	**BoxEL**
GO	Hits@10	0.00	0.00	0.00	0.09	0.10	0.03
	Hits@100	0.00	0.00	0.00	0.16	0.22	0.08
	AUC	0.53	0.44	0.50	0.70	0.76	0.81
	Mean Rank	-	-	-	13719	11050	8980
GALEN	Hits@10	0.00	0.00	0.00	0.07	0.10	0.02
	Hits@100	0.00	0.00	0.00	0.14	0.17	0.03
	AUC	0.54	0.48	0.51	0.64	0.65	0.85
	Mean Rank	-	-	-	8321	8407	3584
ANATOMY	Hits@10	0.00	0.00	0.00	0.18	0.18	0.03
	Hits@100	0.01	0.00	0.00	0.38	0.40	0.04
	AUC	0.53	0.44	0.49	0.73	0.76	0.91
	Mean Rank	-	-	-	28564	24421	10266

of parameters of ELEm and EmEL^{++}, we search the embedding size from $n = [25, 50, 100]$ for BoxEL. We summarize the best performing hyperparameters in our supplemental material. All experiments are evaluated with 10 random seeds and the mean results are reported for numerical comparisons.

Baselines We compare the state-of-the-art $\mathcal{EL}^{++}$ embeddings (ELEm) [92], the first geometric embeddings of $\mathcal{EL}^{++}$, as well as the extension EmEL^{++} [114] that additionally considers the role inclusion and role chain embedding, as our major baselines. For comparison with classical methods, we also include the reported results of three classical KGEs in [114], including TransE [19], TransH [173] and DistMult [192].

Results Table 4.3 summarizes the ranking based measures of embedding models. We first observe that both ELEm and EmEL^{++} perform much better than the three standard KGEs (TransE, TransH, and DistMult) on all three datasets, especially on hits@k for which KGEs fail, showcasing the limitation of KGEs and the benefits of geometric embeddings on encoding logic structures. EmEL^{++} performs slightly better than ELEm on all three datasets. Overall, our model BoxEL outperforms ELEm and EmEL^{++}. In particular, we find that for Mean Rank and AUC, our model achieves significant performance gains on all three datasets. Note that Mean Rank and AUC have theoretical advantages over hits@k because hits@k is sensitive to any model performance changes while Mean Rank and AUC reflect the average performance, demonstrating that BoxEL achieves better average performance. Table 4.2 shows the accuracies of different embeddings in terms of the geometric interpretation of the classes in various ontologies. It clearly demonstrates that BoxEL outperforms ELEm and EmEL^{++} by a large margin, showcasing that BoxEL preserves the underlying ontology characteristics in vector space better than ELEm and EmEL^{++} that use ball embeddings.

4.3.3 *Protein-Protein Interactions*

Dataset We use a biomedical knowledge graph built by [92] from Gene Ontology (TBox) and STRING database (ABox) to conduct this task. Gene Ontology contains information about the functions of proteins, while STRING database consists of the protein-protein interactions. We use the protein-protein interaction data of yeast and human organisms, respectively. For each pair of proteins (P_1, P_2) that exists in STRING, we add a role assertion $\mathsf{interacts}(P1, P2)$. If protein P is associated with the function F, we add a membership axiom $\{P\} \sqsubseteq \exists \mathsf{hasFunction}.F$, the membership assertion can be regarded as a special case of NF3, in which P is a point (i.e. zero-volume box). The interaction pairs of proteins are split into training (80%), testing (10%) and validation (10%) sets. To perform prediction for each protein pair (P_1, P_2), we predict whether the role assertion $\mathsf{interacts}(P_1, P_2)$ hold. This can be measured by Eq.(4.15).

$$P(\mathsf{interacts}(P_1, P_2)) = \left\| T_w^{\mathsf{interacts}}(m_w(P_1)) - m_w(P_2) \right\|_2 . \tag{4.15}$$

where $T_w^{\mathsf{interacts}}$ is the affine transformation function for relation $\mathsf{interacts}$. For each positive interaction pair $\mathsf{interacts}(P_1, P_2)$, we generate a corrupted negative sample by randomly replacing one of the head and tail proteins.

Baselines We consider ELEm [92] and EmEL^{++} [114] as our two major baselines as they have been shown outperforming the traditional KGEs. We also report the result of Onto2Vec [144] that treats logical axioms as a text corpus and OPA2Vec [145] that combines logical axioms with annotation properties. Besides, we report the results of two semantic similarity measures: Resnik's similarity and Lin's similarity in [92]. For KGEs, we compare TransE [**BordesUGWY13**]) and BoxE [35]. We report the hits@10, hits@100, mean rank and AUC (area under the ROC curve) as explained before for numerical comparison. Both raw ranking measures and filtered ranking measures that ignore the triples that are already known to be true in the training stage are reported. Baseline results are taken from the standard benchmark developed by [93].[3]

Overall Results Table 4.4 and Table 4.5 summarize the performance of protein-protein prediction in yeast and human organisms, respectively. We first observe that similarity based methods (SimResnik and SimLin) roughly outperform TransE, showcasing the limitation of classical knowledge graph embeddings. BoxE roughly outperforms TransE as it does encode some logical properties. The geometric methods

3 https://github.com/bio-ontology-research-group/machine-learning-with-ontologies

Table 4.4: Prediction performance on protein-protein interaction (yeast).

Method	Raw Hits@10	Filtered Hits@10	Raw Hits@100	Filtered Hits@100	Raw Mean Rank	Filtered Mean Rank	Raw AUC	Filtered AUC
TransE	0.06	0.13	0.32	0.40	1125	1075	0.82	0.83
BoxE	0.08	0.14	0.36	0.43	633	620	0.85	0.85
SimResnik	**0.09**	0.17	0.38	0.48	758	707	0.86	0.87
SimLin	0.08	0.15	0.33	0.41	875	825	0.8	0.85
ELEm	0.08	0.17	0.44	0.62	451	394	0.92	0.93
EmEL^{++}	0.08	0.16	0.45	0.63	451	397	0.90	0.91
Onto2Vec	0.08	0.15	0.35	0.48	641	588	0.79	0.80
OPA2Vec	0.06	0.13	0.39	0.58	523	467	0.87	0.88
BoxEL	**0.09**	**0.20**	**0.52**	**0.73**	**423**	**379**	**0.93**	**0.94**

Table 4.5: Prediction performance on protein-protein interaction (human).

Method	Raw Hits@10	Filtered Hits@10	Raw Hits@100	Filtered Hits@100	Raw Mean Rank	Filtered Mean Rank	Raw AUC	Filtered AUC
TransE	0.05	**0.11**	0.24	0.29	3960	3891	0.78	0.79
BoxE	0.05	0.10	0.26	0.32	2121	2091	0.87	0.87
SimResnik	0.05	0.09	0.25	0.30	1934	1864	0.88	0.89
SimLin	0.04	0.08	0.20	0.23	2288	2219	0.86	0.87
ELEm	0.01	0.02	0.22	0.26	1680	1638	0.90	0.90
EmEL^{++}	0.01	0.03	0.23	0.26	1671	1638	0.90	0.91
Onto2Vec	0.05	0.08	0.24	0.31	2435	2391	0.77	0.77
OPA2Vec	0.03	0.07	0.23	0.26	1810	1768	0.86	0.88
BoxEL (Ours)	**0.07**	0.10	**0.42**	**0.63**	**1574**	**1530**	**0.93**	**0.93**

ELEm and EmEL^{++} fail on the hits@10 measures and does not show significant performance gains on the hits@100 measures in human dataset. However, ELEm and EmEL^{++} outperform TransE, BoxE and similarity based methods on Mean Rank and AUC by a large margin, especially for the Mean Rank, showcasing the expressiveness of geometric embeddings. Onto2Vec and OPA2Vec achieve relatively better results than TransE and similarity based methods, but cannot compete ELEm and EmEL^{++}. We conjecture that this is due to the fact that they mostly consider annotation information but cannot encode the logical structure explicitly. Our method, BoxEL consistently outperforms all methods in hits@100, Mean Rank and AUC in both datasets, except the competitive results of hits@10, showcasing the better expressiveness of BoxEL.

4.3.4 *Ablation Studies*

Transformation vs Translation To study the contributions of using boxes for modeling concepts and using affine transformation for modeling relations, we conduct an ablation study by comparing relation embeddings with affine transformation (AffineBoxEL) and translation (TransBoxEL). The only difference of TransBox to the AffineBox is that TransBox does not associate a scaling factor for each relation. Table 4.6 clearly shows that TransBoxEL outperforms EmEL^{++}, showcasing the benefits of box modeling compared with ball modeling. While AffineBoxEL further improves TransBoxEL, demonstrating the advan-

Table 4.6: The performance of BoxEL with affine transformation (AffineBoxEL) and BoxEL with translation (TransBoxEL) on yeast protein-protein interaction.

Method	EmEL		TransBoxEL		AffineBoxEL	
	Raw	Filtered	Raw	Filtered	Raw	Filtered
Hits@10	0.08	0.17	0.04	0.18	**0.09**	**0.20**
Hits@100	0.44	0.62	0.54	0.68	**0.52**	**0.73**
Mean Rank	451	394	445	390	**423**	**379**
AUC	0.92	0.93	**0.93**	0.93	**0.93**	**0.94**

Table 4.7: The performance of BoxEL with point entity embedding and box entity embedding on yeast protein-protein interaction dataset.

Method	EmEL		BoxEL (boxes)		BoxEL (points)	
	Raw	Filtered	Raw	Filtered	Raw	Filtered
Hits@10	0.08	0.17	**0.09**	0.19	**0.09**	**0.20**
Hits@100	0.44	0.62	0.48	0.68	**0.52**	**0.73**
Mean Rank	451	394	450	388	**423**	**379**
AUC	0.92	0.93	0.92	0.93	**0.93**	**0.94**

tages of affine transformation. Hence, we could conclude that both of our proposed entity and relation embedding components boost the performance.

Entities as Points vs Boxes As mentioned before, distinguishing entities and concepts by identifying entities as points has better theoretical properties. Here, we study how this distinction influences the performance. For this purpose, we eliminate the ABox axioms by replacing each individual with a singleton class and rewriting relation assertions $r(a, b)$ and concept assertions $C(a)$ as $\{a\} \sqsubseteq \exists r.\{b\}$ and $\{a\} \sqsubseteq C$, respectively. In this case, we only have TBox embeddings and the entities are embedded as regular boxes. Table 4.7 shows that for hits@k, there is marginal significant improvement of point entity embedding over boxes entity embedding, however, point entity embedding consistently outperforms box entity embedding on Mean Rank and AUC, showcasing the benefits of distinguishing entities and concepts.

4.4 CONCLUSION

This work proposes BoxEL, a geometric KB embedding method that explicitly models the logical structure expressed by the theories of $\mathcal{EL}^{++}$. Different from the standard KGEs that simply ignore the analytical guarantees, BoxEL provides *soundness* guarantee for the underlying logical structure by incorporating background knowledge into machine learning tasks, offering a more reliable and logic-preserved fashion for KB reasoning. The empirical results further demonstrate that BoxEL outperforms previous KGEs and $\mathcal{EL}^{++}$ embedding approaches on subsumption reasoning over three ontologies and predicting protein-protein interactions in a real-world biomedical KB.

5

GEOMETRIC EMBEDDINGS OF STRUCTURED CONSTRAINTS IN MACHINE LEARNING

In this chapter, we introduce a geometric embedding approach for encoding structured constraints in a multi-label prediction problem, where the labels are organized under implication and mutual exclusion constraints. A major concern is to produce predictions that are logically consistent with these constraints. To do so, we formulate this problem as an *embedding inference* problem where the constraints are imposed onto the embeddings of labels by *geometric construction*. Particularly, we consider a hyperbolic Poincaré ball model in which we encode labels as Poincaré hyperplanes that work as linear decision boundaries. The hyperplanes are interpreted as convex regions such that the logical relationships (implication and exclusion) are geometrically encoded using *insideness* and *disjointedness* of these regions, respectively. We show theoretical groundings of the method for preserving logical relationships in the embedding space. Extensive experiments on 12 datasets show 1) significant improvements in mean average precision; 2) lower number of constraint violations; 3) an order of magnitude fewer dimensions than baselines.

5.1 MOTIVATION AND BACKGROUND

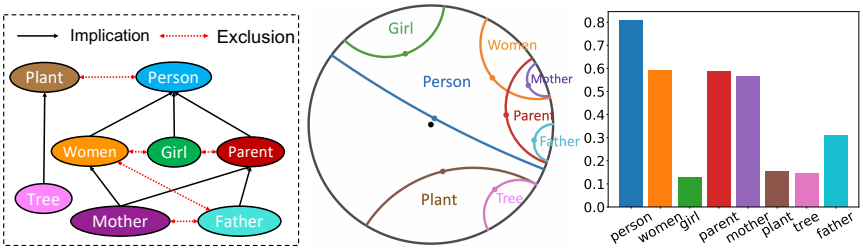
Structured multi-label prediction is a task aiming to associate every object with multiple labels that are semantically constrained in a structured manner (e.g., by implication and exclusion constraints). This task is of growing importance in many applications such as image annotation [47], text categorization, [90, 104] and functional genomics [91, 165], where the labels are organized in a directed acyclic graph (DAG) or an ontology. One of the central concerns of the task is to produce predictions that are *logically consistent* with the constraints of the labels. For example, a protein must be labeled to have the function *nucleic acid binding* if it is already labeled to have the function *RNA binding* (i.e., implication) and must not have the function *drug binding* (i.e., mutual exclusion).

Various works have been proposed to improve the prediction consistency [25, 58, 116, 130, 176]. One line of work called *label embedding*

aims to represent labels as low-dimensional vectors [14, 29]. A key disadvantage of the vector-based representations is that they only capture weak forms of correlation or "similarity' between labels, but do not strongly enforce the logical relationships. Another line of work [25, 58, 116, 176] imposes these logical constraints directly to the losses of neural networks. However, they do not explicitly learn the representations of labels and typically require a complete label taxonomy, which is not always available in and scalable to real-world settings [130].

Embedding-based inference [113], which imposes logical constraints directly to the label embeddings, is able to *inductively* infer the underlying label relationships from incomplete labelings [112]. Once all embeddings are adhering to the constraints, each label can be predicted independently without accessing the label relationships, which significantly reduces the computation cost during inference [113]. The key idea, which is inspired by the Venn diagram [112, 160] or set-theoretic semantics [157], is to represent each label as a convex region [113]. A prominent example is the multi-label box model (MBM) [130] that models label implications as box containments. However, MBM learns box-like decision boundaries, which are typically not compatible with standard classifiers (i.e., hyperplane margin-based models such as logistic regression [55]). Besides, box models suffer from a theoretical limitation, i.e., lower-way intersections enforce higher-way interactions [161]. Finally, current methods ignore the importance of constraining mutual exclusion, which is essential as otherwise, a model could trivially obtain zero implication violation by assigning the same score to all labels.

We consider a structured multi-label prediction problem with *implication* and *mutual exclusion* constraints that are jointly described by a hierarchy and exclusion (HEX) graph (see Figure 5.1(a) for an example). The key idea of our method is to transform the logical constraints into soft geometric constraints in the embedding space. In particular, we consider a hyperbolic Poincaré ball model that has demonstrated advantages in representing hierarchical data [194] and assign each label a Poincaré hyperplane that has several favorable theoretical properties in classification. Each Poincaré hyperplane can be interpreted as a convex region such that the *implication* and *mutual exclusion* are modeled by geometric *insideness* and *disjointness* between the corresponding regions, respectively. In this way, a multi-label classifier can be defined by measuring the confidence of an instance having a label as geometric *membership*. Unlike other hyperbolic region-based models such as hyperbolic cones [55] and hyperbolic disks [152], Poincaré hyperplane works as a linear decision boundary and can be seamlessly integrated into existing margin-based classifiers such as



(a) (b) (c)

Figure 5.1: (a) A HEX graph describing the logical relationships (implication and exclusion) between different labels; (b) The learned label embeddings (linear decision boundaries) in the Poincaré ball, where all constraints in the HEX graph are respected; (c) The prediction scores of a given instance of *mother* respect all constraints in the HEX graph, where each score is calculated as the confidence of the instance embedding being a member of the convex region of the corresponding label embedding.

hyperbolic logistic regression [55]. Figure 5.1(b) shows an example of the learned label representations that respect all the constraints given in Figure 5.1(a). We show theoretical groundings of the proposed method on modeling *implication* and *mutual exclusion*. Extensive experiments on 12 multi-label classification tasks show the model's capability to improve the mean average precision significantly while keeping the number of constraint violations low and requiring an order of magnitude fewer dimensions.

5.2 STRUCTURED MULTI-LABEL PREDICTION

Let $\mathcal{X} \subseteq \mathbb{R}^n$ denote an n-dimensional instance space and $\mathcal{L} = \{l_1, l_2, \ldots\}, |\mathcal{L}| \geq 2$ denote the finite set of possible labels. Given a set of N training examples $\mathcal{D} = \{(x_i, L_i) \mid 1 \leq i \leq N, x_i \in \mathcal{X}, L_i \subset \mathcal{L}\}$, *multi-label prediction* aims to learn a labeling function $f : \mathcal{X} \to 2^{\mathcal{L}}$ mapping from the instance space to the *powerset* of the label space, $f(x) \subset \mathcal{L}$.

Structured multi-label prediction additionally imposes a set of prior-known logical constraints over the labels, namely, the predictions must be logically consistent with these constraints. Analogous to Mirzazadeh et al. [113], we consider two forms of logical constraints between labels: implication and mutual exclusion. Specifically, an *implication* of the form $l_a \Rightarrow l_b$ imposes the constraint that whenever an instance is labeled as l_a then it must also be labeled as l_b, i.e., $l_a \Rightarrow l_b$ is a shorthand notation for $\forall x \in \mathcal{X}, l_a \in f(x) \Rightarrow l_b \in f(x)$. *Mutual exclusions* are constraints of the form $\neg l_a \vee \neg l_b$, implying that an instance cannot be simultaneously labeled as l_a and l_b, i.e., $\neg l_a \vee \neg l_b$ is a shorthand notation for $\forall x \in \mathcal{X}, l_a \notin f(x) \vee l_b \notin f(x)$. We can

concisely represent a set of implication and exclusion constraints with a hierarchy and exclusion (HEX) graph [47].

Definition 20 (HEX graph [47][1])**.** *A HEX graph $G = (V, E_h, E_e)$ is a graph consisting of a set of nodes $V = \{v_1, \ldots, v_n\}$, directed (hierarchy) edges $E_h \subseteq V \times V$, and undirected (exclusion) edges $E_e \subseteq V \times V$, such that the subgraph $G_h = (V, E_h)$ is a DAG and the subgraph $G_e = (V, E_e)$ has no self loop. Each node $v_i \in V$ represents the label l_i. A directed edge $(v_i, v_j) \in E_h$ represents the implication $l_i \Rightarrow l_j$, and an undirected edge $(v_i, v_j) \in E_e$ represents the exclusion $\neg l_i \vee \neg l_j$.*

Note that an arbitrary HEX graph may contain redundant edges. A hierarchy edge (v_i, v_j) is redundant when there is a path in G_h from v_i to v_j which does not contain the edge (v_i, v_j). Similarly, an exclusion edge (v_i, v_j) is redundant when there is another exclusion edge connecting their ancestors (or connecting one node's ancestor to the other node).

We can transform a HEX graph into an equivalent HEX graph by adding or removing redundant edges. In this paper, we only consider HEX graphs that have a minimal number of edges, we call such HEX graph a *minimal sparse* HEX graph (see Fig. 5.1(a) for an example). Given a minimal sparse HEX graph, we define the HEX-property as

Definition 21 (HEX-property)**.** *A labeling function f has the HEX property with respect to a HEX graph G if for all $x \in \mathcal{X}$, $f(x)$ respects all constraints represented by G.*

We also call such function *f logically consistent* w.r.t G. Given the HEX graph and the HEX-property, structured multi-label prediction is formally defined as a constrained optimization problem.

Definition 22 (Structured multi-label prediction)**.** *The structured multi-label prediction task with respect to a training set $\mathcal{D} = \{(x_i, L_i) \mid 1 \leq i \leq N, x_i \in \mathcal{X}, L_i \subset \mathcal{L}\}$, minimal HEX graph $G = (V, E_h, E_e)$, and multi-label prediction function f, is the task of learning f such that the function f minimizes $\sum_{(x_i, L_i) \in \mathcal{D}} \text{loss}(f(x_i), L_i)$, with* loss *a predefined function, while attempting to maintain the HEX-property with respect to G.*

Note that this definition allows for a *soft* interpretation of the constraints, meaning that the goal is to adhere to all of them, but we do

1 Deng et al. [47] use subsumption, which is the inverse relation of implication that we use here.

allow for loosening some if necessary. For example, a mutual exclusion constraint is allowed to loosen when an instance (e.g., image), though rarely happens, is simultaneously labeled as two mutual exclusive labels (e.g., *dog* and *cat*).

5.3 HYPERBOLIC EMBEDDING INFERENCE

We consider learning a real-valued ranking function $h : \mathcal{X} \times \mathcal{L} \mapsto [0, 1]$, where the output is interpreted as the confidence of an instance $x \in \mathcal{X}$ having a label $l \in \mathcal{L}$. Afterward, a binary multi-label classifier $f : \mathcal{X} \to 2^{\mathcal{L}}$ can be simply obtained by thresholding the ranking function with a threshold t, i.e., $f(x) = \{l \mid h(x, l) \geq t, \forall l \in \mathcal{L}\}$. The objective of h is to assign higher scores to positive instance-label pairs than that of negative instance-label pairs.

5.3.1 *Geometric construction*

Given an n-dimensional Poincaré ball $\mathbb{D}^n$, we associate each instance $x_i \in \mathcal{X}$ with a point in the Poincaré ball and associate each label $l_i \in \mathcal{L}$ with a Poincaré hyperplane, such that its corresponding positive and negative instances are correctly separated by the hyperplane.

Poincaré hyperplanes

Let $\mathbb{B}^n$ denote the set of n-balls in $\mathbb{R}^n$ whose boundaries $\partial\mathbb{B}^n$ intersect the Poincaré ball $\mathbb{D}^n$ perpendicularly. Poincaré hyperplanes are defined by $\partial\mathbb{B}^n \cap \mathbb{D}^n$ (see Fig. 5.2(a)) plus all linear subspaces going through the origin. For the former cases, a Poincaré hyperplane can be uniquely defined by its center point that has a minimal distance to the origin.

Definition 23. *Given a (center) point* $\mathbf{c} \in \mathbb{D}^n$ *where* $\mathbf{c} \neq \mathbf{0}$*, the Poincaré hyperplane is defined as*

$$H_{\mathbf{c}} = \left\{\mathbf{p} \in \mathbb{D}^n : g^{\mathbb{D}}\left(\log_{\mathbf{c}}(\mathbf{p}), \mathbf{c}\right) = 0\right\} \tag{5.1}$$

where $\mathbf{c}$ *is the center point and* $\mathbf{c} \in T_{\mathbf{c}}\mathbb{D}^n$ [2] *is the normal vector passing through the origin* $\mathbf{0}$*.*

2 In this paper, we distinguish normal vectors from regular points by adding an arrow on top of its letters.

Intuitively, this corresponds to the union of all geodesics passing through $\mathbf{c}$ while orthogonal to the normal vector $\mathbf{c} \in T_{\mathbf{c}}\mathbb{D}^n$. In the case where $\mathbf{c}$ is the center of the hyperplane, $\mathbf{c}$ must simultaneously pass through $\mathbf{c}$ and the origin. Hence, $\mathbf{c}$ can be simply taken as $\mathbf{c}$ without loss of generality. For the special case where $\mathbf{c} = \mathbf{0}$, the Poincaré hyperplanes are all linear subspaces (Euclidean planes) passing through the origin. In this paper, we exclude these special cases by assuming $\mathbf{c} \neq \mathbf{0}$.

Geometric intuition Essentially, the Poincaré hyperplane works as a linear decision boundary that separates the embedding space into two regions,[3] where the smaller region (i.e., convex hull) is interpreted as the space of positive samples while the other one is interpreted as the space of negative samples. Two reasons motivate us to model labels as Poincaré hyperplanes: 1) Modeling labels as hyperplanes has several desired theoretical advantages in margin-based classifiers. Our model shares the same philosophy as existing learning frameworks such as hyperbolic logistic regression [55] and hyperbolic SVM [37]; 2) More importantly, unlike Euclidean space that is flat, hyperbolic Poincaré ball is a curved space in which there are infinitely many non-parallel hyperplanes which do not intersect, implying that linear decision boundaries in hyperbolic space can capture more complicated set-theoretic interactions, such as implication and mutual exclusion.

Enclosing balls Given a Poincaré hyperplane $H_{\mathbf{c}}$, we call the corresponding n-ball $\mathbb{B}^n_{\mathbf{c}}$ that encloses $H_{\mathbf{c}}$ its enclosing n-ball. Formally, an enclosing n-ball $\mathbb{B}^n(\mathbf{o}, r)$ is defined by $\mathbb{B}^n(\mathbf{o}, r) = \{\mathbf{p} : \|\mathbf{p} - \mathbf{o}\| \leq r\}$, where $\mathbf{o} \in \mathbb{R}^n$ and r are the center point and the radius, respectively. Given $H_{\mathbf{c}}$, we have the following closed-form representation of $\mathbb{B}^n_{\mathbf{c}}$.

Proposition 10. *Given a Poincaré hyperplane $H_{\mathbf{c}}$ where $\mathbf{c} \neq \mathbf{0}$, there exists an n-ball $\mathbb{B}^n_{\mathbf{c}}(\mathbf{o}_{\mathbf{c}}, r_{\mathbf{c}})$ such that $H_{\mathbf{c}} \subset \mathbb{B}^n_{\mathbf{c}}(\mathbf{o}_{\mathbf{c}}, r_{\mathbf{c}})$, i.e., $H_{\mathbf{c}}$ is a subset of $\mathbb{B}^n_{\mathbf{c}}(\mathbf{o}_{\mathbf{c}}, r_{\mathbf{c}})$. $\mathbb{B}^n_{\mathbf{c}}$ is uniquely given by*

$$\mathbb{B}^n_{\mathbf{c}} = \mathbb{B}^n\left(\frac{(1 + \|\mathbf{c}\|^2)}{2\|\mathbf{c}\|}\mathbf{c}, \frac{1 - \|\mathbf{c}\|^2}{2\|\mathbf{c}\|}\right) \tag{5.2}$$

Proof sketch. The key idea is to solve a quadratic equation given by the fact that the radius of $\mathbb{B}^n_{\mathbf{c}}$, the radius of $\mathbb{D}^n$, and the distance from the center of $\mathbb{D}^n$ to the center of $\mathbb{B}^n_{\mathbf{c}}$ must satisfy the Pythagorean theorem [80]. Full proof is in the supplementary material.

3 Note that by using the metric in the Poincaré ball, each region has infinite (exponentially growing) volume.

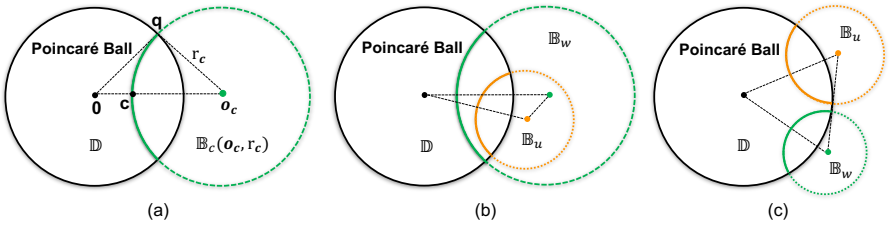


Figure 5.2: (a) A Poincaré hyperplane is defined as the intersection between the Poincaré ball $\mathbb{D}$ and the boundary of an n-ball $\mathbb{B}_{\mathbf{c}}$. The Poincaré hyperplane is uniquely parameterized by a center point $\mathbf{c}$, and the corresponding n-ball (its radius and center) can be uniquely determined by Proposition 11. (b) Label implication is interpreted as n-ball insideness. (c) Mutual exclusion is interpreted as n-ball disjointedness.

5.3.2 *Geometric interpretation*

Our main idea is to transform the logical relationships between labels into geometric relationships between their corresponding enclosing n-balls. In particular, the implication is modeled by the geometric insideness while the mutual exclusion is modeled by the geometric disjointness.

Implication The logical implication between two labels is interpreted as geometric relations between n-balls, i.e., n-ball insideness illustrated in Fig. 5.2(b). In particular, an n-ball $\mathbb{B}_{\mathbf{w}}\left(\mathbf{o}_{\mathbf{w}}, r_{\mathbf{w}}\right)$ contains $\mathbb{B}_{\mathbf{u}}\left(\mathbf{o}_{\mathbf{u}}, r_{\mathbf{u}}\right)$ if and only if $\|\mathbf{o}_{\mathbf{u}} - \mathbf{o}_{\mathbf{w}}\| + r_{\mathbf{u}} < r_{\mathbf{w}}$, and thus we can create an insideness loss defined by

$$\mathcal{L}_{\text{inside}}(\mathbb{B}_{\mathbf{u}}, \mathbb{B}_{\mathbf{w}}) = \max\{0, \|\mathbf{o}_{\mathbf{u}} - \mathbf{o}_{\mathbf{w}}\| + r_{\mathbf{u}} - r_{\mathbf{w}}\}. \tag{5.3}$$

Clearly, the insideness loss term satisfies the properties of correctness and transitivity

Lemma 2 (Correctness). $\mathbb{B}_{\mathbf{u}}$ *is inside of* $\mathbb{B}_{\mathbf{w}}$ *if and only if* $\mathcal{L}_{inside}(\mathbb{B}_{\mathbf{u}}, \mathbb{B}_{\mathbf{w}}) = 0$.

Lemma 3 (Transitivity). *If* $\mathcal{L}_{inside}(\mathbb{B}_{\mathbf{u}}, \mathbb{B}_{\mathbf{w}}) = 0$ *and* $\mathcal{L}_{inside}(\mathbb{B}_{\mathbf{w}}, \mathbb{B}_{\mathbf{v}}) = 0$, *we have* $\mathcal{L}_{inside}(\mathbb{B}_{\mathbf{u}}, \mathbb{B}_{\mathbf{v}}) \leq \mathcal{L}_{inside}(\mathbb{B}_{\mathbf{u}}, \mathbb{B}_{\mathbf{w}}) + \mathcal{L}_{inside}(\mathbb{B}_{\mathbf{w}}, \mathbb{B}_{\mathbf{v}}) \leq \mathcal{L}_{inside}(\mathbb{B}_{\mathbf{w}}, \mathbb{B}_{\mathbf{v}}) = 0$.

Mutual exclusion Similarly, we interpret mutual exclusion as geometric disconnectedness between n-balls illustrated in Fig. 5.2(c). $\mathbb{B}_{\mathbf{u}}$ disconnecting from $\mathbb{B}_{\mathbf{w}}$ can be measured by subtracting the distance

between their center points from the sum of their radii. Inversely, the corresponding loss is

$$\mathcal{L}_{\text{disjoint}}(\mathbb{B}_{\mathbf{u}}, \mathbb{B}_{\mathbf{w}}) = \max\{0, r_{\mathbf{w}} + r_{\mathbf{u}} - \|\mathbf{o}_{\mathbf{u}} - \mathbf{o}_{\mathbf{w}}\|\} \tag{5.4}$$

Again, the disjointedness loss term satisfies the correctness property

Lemma 4 (Correctness). $\mathbb{B}_{\mathbf{u}}$ *disconnects from* $\mathbb{B}_{\mathbf{w}}$ *if and only if* $\mathcal{L}_{disjoint}(\mathbb{B}_{\mathbf{u}}, \mathbb{B}_{\mathbf{w}}) = 0$.

5.3.3 *Classification and learning*

Given the embeddings of instances and labels, an instance can be classified by measuring the *geometric membership*, i.e., the confidence of a point $\mathbf{p} \in \mathbb{D}^n$ being inside the enclosing ball $\mathbb{B}$.

Membership and non-membership Formally, given an instance embedding $\mathbf{p} \in \mathbb{D}^n$ and a label embedding associated with an enclosing n-ball $\mathbb{B}_{\mathbf{c}}$. The confidence of an instance $\mathbf{p}$ being inside the enclosing n-ball $\mathbb{B}_{\mathbf{c}}$ can be measured by subtracting the distance between the center point of $\mathbb{B}_{\mathbf{c}}$ and $\mathbf{p}$ from the radius of $\mathbb{B}_{\mathbf{c}}$. The corresponding loss is defined as the inverse of the measure, given by

$$\mathcal{L}_{\text{membership}}\left(\mathbf{p}, \mathbb{B}_{\mathbf{c}}\left(\mathbf{o}_{\mathbf{c}}, r_{\mathbf{c}}\right)\right) = \max\{0, \|\mathbf{o}_{\mathbf{c}} - \mathbf{p}\| - r_{\mathbf{c}}\}. \tag{5.5}$$

Symmetrically, for negative instance-label relations, the loss of non-membership can be defined as

$$\mathcal{L}_{\text{non-membership}}\left(\mathbf{p}, \mathbb{B}_{\mathbf{c}}\left(\mathbf{o}_{\mathbf{c}}, r_{\mathbf{c}}\right)\right) = \max\{0, r_{\mathbf{c}} - \|\mathbf{o}_{\mathbf{c}} - \mathbf{p}\|\}. \tag{5.6}$$

Clearly, we have the following properties that follow directly from the definitions.

Lemma 5. *A point* $\mathbf{p}$ *is a member of* $\mathbb{B}_{\mathbf{c}}$ *if and only if* $\mathcal{L}_{membership}\left(\mathbf{p}, \mathbb{B}_{\mathbf{c}}\right) = 0$.

Lemma 6. *A point* $\mathbf{p}$ *is not a member of* $\mathbb{B}_{\mathbf{c}}$ *if and only if* $\mathcal{L}_{non\text{-}membership}\left(\mathbf{p}, \mathbb{B}_{\mathbf{c}}\right) = 0$.

Lemma 1-2, Lemma 3, Lemma 4-5 immediately follow the definitions of geometric insideness, disjointedness, and membership, respectively.

We aim to learn an encoder E_θ (i.e., a hyperbolic neural network whose designs depend on the datasets), where θ is the trainable

parameter, and a function $\mathcal{C}$ which maps labels to the center points of the corresponding Poincaré hyperplanes in the Poincaré ball.

Now, we define

$$h(x,l) = \sigma\left(\mathcal{L}_{\text{non-membership}}\left(E_\theta(x), C(l)\right) - \mathcal{L}_{\text{membership}}\left(E_\theta(x), C(l)\right)\right) \tag{5.7}$$

as our ranking function, where σ is the sigmoid function. The final classification function is then defined by $f(x) = \{l \mid h(x,l) \geq 0.5\}$. We call our classifier *hyperbolic multi-label embedding inference* (HMI). Given a HEX graph, HMI has the following guarantee.

Proposition 11 (HEX-property)**.** *The classification function f of HMI has the HEX property with respect to G if for every constraint in G, the corresponding loss term is* 0*.*

Learning with soft constraints

Let $\mathcal{D}^+ = \{(x_i, l_n) | (x_i, L_i) \in \mathcal{D}, l_n \in L_i\}$ be the set of positive instance-label pairs and $\mathcal{D}^- = \{(x_i, l_n) | (x_i, L_i) \in \mathcal{D}, l_n \in \mathcal{L}, l_n \notin L_i\}$ be the set of negative instance-label pairs. By combining the loss functions of membership, non-membership, insideness and disjointedness, the learning objective can be formulated as

$$\begin{aligned}\min_{\theta,\mathcal{C}} & \sum_{(x_i,l_n)\in\mathcal{D}^+} \mathcal{L}_{\text{membership}}\left(E_\theta\left(x_i\right), \mathbb{B}_{\mathcal{C}(l_n)}\right) \qquad (5.8)\\ & + \sum_{(x_i,l_n)\in\mathcal{D}^-} \mathcal{L}_{\text{non-membership}}\left(E_\theta\left(x_i\right), \mathbb{B}_{\mathcal{C}(l_n)}\right)\\ & + \lambda\left(\sum_{(v_i,v_j)\in E_h} \mathcal{L}_{inside}\left(\mathbb{B}_{\mathcal{C}(l_i)}, \mathbb{B}_{\mathcal{C}(l_j)}\right) + \sum_{(v_i,v_j)\in E_e} \mathcal{L}_{disjoint}\left(\mathbb{B}_{\mathcal{C}(l_i)}, \mathbb{B}_{\mathcal{C}(l_j)}\right)\right) \qquad (5.9)\end{aligned}$$

The first two terms are losses for positive and negative samples while the last two terms are implication and exclusion constraints, respectively, with λ being the penalty weight of the constraints.

The following corollary shows that our model has a strong inductive bias for preserving consistency.

Corollary 1. *Given a HEX graph G of labels, if the loss terms $\mathcal{L}_{inside}$ and $\mathcal{L}_{disjoint}$ are* 0*, then the learned prediction function is logically consistent.*

Classification via hyperbolic logistic regression A key advantage of our method is that the losses of constraints are compatible with other (margin-based) hyperbolic classifiers such as hyperbolic logistic regression (HLR) [55]

and hyperbolic support vector machine (HSVM) [37]. In our experiment we explore HLR, which formulates the *logits* as the distances from an instance to a Poincaré hyperplane of a label. That is, $h(x,l) = d\left(E_\theta\left(x\right), H_C\left(l\right)\right)$. $d(\mathbf{p}, H_\mathbf{c})$ has the following closed form:

$$d(\mathbf{p}, H_\mathbf{c}) = \sinh^{-1}\left(\frac{2|\langle(-\mathbf{c})\oplus\mathbf{p},\mathbf{c}\rangle|}{\left(1-\|(-\mathbf{c})\oplus\mathbf{p}\|^2\right)\|\mathbf{c}\|}\right) \tag{5.10}$$

where $\oplus$ is the Möbius addition [55]. The classifier is defined by $f(x) = \{l|\,\text{œ}\left(h\left(x,l\right)\right) \geq 0.5, \forall l \in \mathcal{L}\}$ where σ is the sigmoid function. We dub such classifier combined with HMI as HMI+HLR.

5.4 EVALUATION

5.4.1 *Experiment setup*

Datasets We consider 12 datasets that have been used for evaluating multi-label prediction methods [58, 130, 176]. These consist of 8 functional genomic datasets [39], 3 image annotation datasets [48, 49], and 1 text classification dataset [87]. All input features are pre-processed in the same way as described by Patel et al. [130]. For all datasets, the implication constraints (label taxonomy) are given. Following Mirzazadeh et al. [113] we add exclusion constraints between sibling nodes whenever this does not create a contradiction (i.e., they share no common descendant nodes). We also explore other strategies for deriving exclusions, but no significant difference was observed (see the supplement for an analysis). Similar to MBM [130] and its baselines, we sample 30% of the implications and exclusions constraints for training the model.

Hyperbolic encoder We adopt a simple hyperbolic linear layer as the instance encoder for all datasets. A single-layer hyperbolic fully-forward linear layer is defined by $f_{\theta=\{\mathbf{W},\mathbf{b}\}}(\mathbf{x}) = \tanh^{\otimes}\left(\mathbf{W}\otimes\mathbf{x}\oplus\mathbf{b}\right)$, with $\otimes$ being Möbius matrix-vector multiplication defined by $M\otimes\mathbf{x} = \tanh\frac{\|M\mathbf{x}\|}{\|\mathbf{x}\|}\tanh^{-1}\left(\|\mathbf{x}\|\right))\frac{M\mathbf{x}}{\|M\mathbf{x}\|}$, where $\mathbf{W}\in\mathbb{R}^{n\times d}$ is a trainable matrix and $\mathbf{x}$ is a point $\mathbf{x}\in\mathbb{D}^n, M\mathbf{x}\neq 0$. $\oplus$ denotes Möbius addition given by

$$\mathbf{x}\oplus\mathbf{y} = \frac{\left(1+2\langle\mathbf{x},\mathbf{y}\rangle+\|\mathbf{y}\|^2\right)\mathbf{x}+\left(1-\|\mathbf{x}\|^2\right)\mathbf{y}}{1+2\langle\mathbf{x},\mathbf{y}\rangle+\|\mathbf{x}\|^2\|\mathbf{y}\|^2} \tag{5.11}$$

and $\tanh^{\otimes}$ denotes an Möbius version of pointwise non-linearity given by $\tanh^{\otimes}(\mathbf{x}) = \exp_0\left(\tanh\left(\log_0(\mathbf{x})\right)\right)$, with $\exp_0$ and $\log_0$ being the exponential and logarithmic maps, see [55] for more details.

Baselines We compare our approach with both classical vector-based and state-of-the-art region-based embedding methods. In particular, we consider two vector-based models: 1) The multi-label vector model (MVM) [167], which encodes both inputs and labels as Euclidean vectors; 2) the multi-label hyperbolic model (MHM) used by Chen et al. [29], which represents inputs and labels as hyperbolic points; and two box models: 3) the non-

probabilistic box model (BoxE) [1] and 4) the probabilistic multi-label box model (MBM) [130] that encodes both instances and labels as axis-parallel hyper-rectangles. Besides, we compare with 5) hyperbolic logistic regression (HLR) [55] since it also encodes labels as Poincaré hyperplanes (but does not use geometric constraints). Furthermore, we compare with 6) C-HMCNN, a state-of-the-art non-embedding based method that injects hierarchy constraints directly into the loss function without embedding labels. A notable difference is that C-HMCNN needs the full hierarchy constraints as its input. Finally, we also implement HMI+HLR, a combination of our proposed constraints with HLR for an ablation study.

Implementation details We implement HMI, HLR and HMC-HLR using PyTorch [129] and train the models on NVIDIA A100 with 40GB memory. We train HMI, HLR and HMI+HLR using Riemannian Adam [13] optimizer implemented by the Geoopt library [88] with a batch size of 4. We also explore some larger batch sizes but it does not yield better results, which is also observed in Wehrmann et al.[176]. We set the dropout rate to 0.6 suggested by [176] to avoid the case that the model overfits the small training sets. We employ an early-stopping strategy with patience 20 to save training time. The results of other baselines are as reported by Patel et al.[130] that we closely follow. The learning rate is searched from $\{1e-4,\ 5e-4,\ 1e-3,\ 5e-3,\ 1e-2\}$. The penalty weight of the violation is searched from $\{1e-5,\ 5e-4,\ 1e-4,\ 5e-3,\ 1e-2\}$ and we also show its impact in an ablation. The best dimension per dataset is searched from $\{32, 64, 128, 256\}$, which is one order of magnitude lower than that used by Patel et al. [130] ($\{250, 500, 1000, 1750\}$). All methods have been run 10 times with random seeds and the average results are reported. We omit the standard deviations since they are in a very small range ($[2 \times 10^{-4}, 2.3 \times 10^{-3}]$).

Evaluation protocols In line with Patel et al. [130], we consider *Mean Average Precision (mAP)*,[4] which summarizes the information of precisions and recalls with varied thresholds. We also report two metrics that additionally take the constraints into account: 1) Constrained mAP (CmAP) is a variant of mAP that replaces the score of each label with the maximum scores of its descendant labels in the hierarchy [130]. 2) *Hierarchy Constraint Violation (HCV)* [130] measures the extent to which the label scores violate the implication constraints regardless of true labels for the instances. HCV is computed as $\text{HCV} = \frac{1}{|\mathcal{D}||E_h|} \sum_{k=1}^{|\mathcal{D}|} \sum_{(l_i,l_j)\in E_h} \mathbb{1}\left(h_i^k - h_j^k > 0\right)$, where h_i means the prediction score of label l_i. Clearly, a lower value of HCV implies higher consistency in the predictions.

5.4.2 *Main results*

As Table 5.1 shows, our method HMI either achieves the best (7-8/12 datasets) or competitive (4-5/12 datasets) performance (mAP and CmAP) over all compared methods. HMI outperforms all methods w.r.t the average ranking of mAP/CmAP, showcasing the advantages of HMI. We observed that the

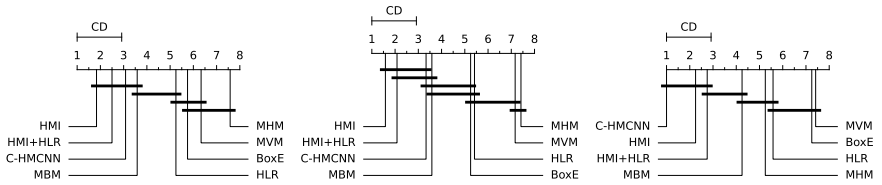
4 https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html

Table 5.1: Comparison of performance and consistency on 12 datasets, where underline indicates the best results over embedding-based methods, and **boldface** indicates the best results over all methods. We implemented HMI, HLR and HMI+HLR. Other results are taken from Patel et al. [130]. All metrics are averaged across 10 runs with random seeds (standard deviations are relatively small (in range $[2 \times 10^{-4}, 2.3 \times 10^{-3}]$) and are hence omitted).

Dataset	Metric	Ours		Embeddings					Non-embedding
		HMI	**HMI+HLR**	MVM	MHM	BoxE	MBM	HLR	C-HMCNN
ExprFUN	mAP ↑	**38.53**	38.50	37.94	31.90	37.30	38.42	37.98	38.41
	CmAP ↑	**38.72**	38.62	37.41	32.02	37.92	38.67	37.44	38.41
	HCV ↓	0.92	1.07	1.97	1.92	4.79	1.87	2.17	**0**
CellcycleFUN	mAP ↑	34.82	**34.84**	31.61	28.74	31.96	34.61	34.05	34.35
	CmAP ↑	34.90	**35.00**	31.33	28.89	32.70	34.78	34.11	34.35
	HCV ↓	1.30	1.32	3.45	1.78	4.02	1.35	2.30	**0**
DerisiFUN	mAP ↑	**36.71**	**36.71**	24.16	24.40	26.66	28.71	26.65	28.19
	CmAP ↑	**36.94**	36.89	24.35	24.52	26.96	28.88	26.83	28.19
	HCV ↓	0.73	0.87	4.01	0.85	2.27	1.43	2.30	**0**
SpoFUN	mAP ↑	**36.47**	36.44	24.21	26.57	27.97	29.62	28.29	29.18
	CmAP ↑	36.43	**36.54**	24.55	26.79	28.38	29.78	28.31	29.18
	HCV ↓	0.92	1.05	4.73	1.69	2.75	1.53	1.98	**0**
ExprGO	mAP ↑	48.63	48.50	44.97	40.52	46.75	48.45	**48.65**	48.61
	CmAP ↑	**48.68**	48.61	41.84	40.70	47.28	48.56	48.65	48.61
	HCV ↓	1.37	1.45	7.05	5.19	5.74	1.91	1.35	**0**
CellcycleGO	mAP ↑	45.58	45.51	44.19	39.74	43.08	44.93	40.28	**45.61**
	CmAP ↑	45.58	45.53	41.02	39.76	43.79	45.01	40.30	**45.61**
	HCV ↓	1.19	1.12	3.03	2.49	5.06	2.16	3.26	**0**
DerisiGO	mAP ↑	**42.31**	42.12	41.13	40.10	40.44	42.02	40.33	42.24
	CmAP ↑	**42.38**	42.28	38.21	40.20	40.73	42.12	40.35	42.24
	HCV ↓	0.86	0.99	3.46	2.02	3.16	1.13	2.31	**0**
SpoGO	mAP ↑	42.70	42.74	42.20	39.70	40.88	41.74	39.22	**42.77**
	CmAP ↑	42.76	**42.77**	39.04	39.77	41.27	41.54	39.26	**42.77**
	HCV ↓	0.95	1.20	2.77	1.90	3.89	1.80	2.33	**0**
Enron	mAP ↑	80.43	80.43	73.68	75.62	**80.44**	80.06	78.87	80.04
	CmAP ↑	**80.50**	80.47	66.87	75.68	80.46	80.05	78.94	80.04
	HCV ↓	**0**	**0**	2.53	0.36	0.20	0.03	0.04	**0**
Diatoms	mAP ↑	**79.19**	79.10	72.65	56.86	43.71	79.14	77.90	76.23
	CmAP ↑	**79.40**	79.36	72.18	56.07	45.16	79.23	78.07	76.23
	HCV ↓	0.17	0.18	19.20	5.55	6.39	0.34	6.36	**0**
Imclefo7a	mAP ↑	**90.67**	89.60	78.22	65.30	83.71	69.26	88.33	90.26
	CmAP ↑	**90.89**	89.71	77.46	66.01	84.73	69.48	88.45	90.26
	HCV ↓	0.20	0.19	22.86	4.75	12.73	2.40	1.77	**0**
Imclefo7d	mAP ↑	89.19	89.20	88.59	75.69	87.95	**89.56**	88.91	89.22
	CmAP ↑	90.00	90.02	86.87	76.95	88.93	**90.07**	87.38	89.22
	HCV ↓	0.37	0.36	11.02	7.56	11.93	5.66	6.88	**0**
Avg. Rank ↓	mAP	**1.75**	2.42	6.33	7.58	5.75	3.5	5.25	3.08
	CmAP	**1.58**	2.08	7.16	7.41	5.25	3.58	5.41	3.33
	HCV	2.25	2.75	7.42	5.25	7.25	4.25	5.58	**1.00**

CmAP is close to mAP, indicating that the model is adhering to the label constraints [130]. In terms of predictive consistency (HCV), HMI consistently achieves the best or the second-best results. Note that C-HMCNN always gets zero HCV because it exploits the complete hierarchy. HMI achieves competitive HCV, despite only using 30% of the hierarchy.

Statistical significance Following Patel et al. [130] and Giunchiglia and Lukasiewicz [58], we test the statistical significance of the performance across all datasets. First, we perform the Friedman test [46] and show that there exists a significant difference w.r.t. all metrics with p-values $\ll 0.05$. Next, we conduct the post-hoc Nemenyi test to verify the statistical differences of the average ranking. The critical diagram w.r.t the average ranking of mAP/CmAP is shown in Fig. 5.3, in which the methods that have no significant differences (significance level 0.05) are connected by a horizontal line. As shown in the diagrams, it is clear to conclude that



(a) mAP (b) CmAP (c) HCV

Figure 5.3: Critical diagrams of the post-hoc Nemenyi test across all 12 datasets.

r

Table 5.2: Results of Wilcoxon test over HMI against baselines.

Method	mAP	CmAP	CV
HMI vs C-HMCNN	5.8×10^{-4}	4.4×10^{-4}	5.0×10^{-3}
HMI vs MBM	3.3×10^{-4}	2.4×10^{-4}	4.9×10^{-4}
HMI vs HMI+HLR	2.3×10^{-2}	3.8×10^{-2}	9.7×10^{-1}

there is a statistically significant difference w.r.t mAPs/CmAPs of HMI and HMI+HLR against MVM, BoxE, MHM, and HLR but not the two strong baselines (MBM and C-HMCNN). We further perform the Wilcoxon test that considers not only the differences in rankings but also the numerical differences in the performance. The Wilcoxon test results show that there is a statistically significant difference between the mAPs/CmAPs of HMI and the two strong baselines with p-value $\ll 0.05$. In terms of HCV, our statistical significance test in Figure 5.3 and Table 5.2 shows that HMI and HMI+HLR significantly outperform MVM, BoxE, MHM, HLR, and MBM but not C-HMCNN since it has zero HCV. However, we observed that the predictive performance (mAP, CmAP) is not fully proportional to the HCV, e.g., HMI outperforms C-HMCNN w.r.t. mAP/CmAP on many of the datasets even though C-HMCNN has zero CV.

Classification via hyperbolic logistic regression To validate whether our proposed geometric constraints are able to improve hyperbolic logistic regression (HLR) [55], we implement HMI+HLR, a combination of our proposed constraints with HLR as described in Section 5.3.3. Table 5.1 show that HMI+HLR outperforms HLR with statistical confidence, showcasing that HMI is able to improve the predictive performance and consistency of HLR. However, there is no significant difference (with p-value larger than 0.05 in Table 5.2) between the two variants of our method (HMI and HMI+HLR).

5.4.3 *Ablation studies & parameter sensitivity.*

For further ablation, we introduce one additional metric. Exclusion Constraint Violation (ECV) measures, analogous to HCV, the fraction of the exclusion constraints violated by the predictions i.e., ECV $= \frac{1}{|\mathcal{D}||E_e|} \sum_{k=1}^{|\mathcal{D}|} \sum_{(l_i,l_j)\in E_e} \mathbb{1}\left(f_i^k \wedge f_j^k\right)$. We introduce this because HCV can be made zero trivially by

associating all labels with the same score. Hence, in the ablation study, we will show how the exclusion constraints (the results of ECV) complement HCV and influence the overall performance.

Impact of penalty weight Table 5.3 shows the results of HMI on "CellcycleFUN" and "CellcycleGO" dataset. We observed that with different penalty weights, the obtained results are slightly different. Even without penalty ($\lambda = 0$), the model already achieves acceptable results, in particular, it outperforms MVM, MHM, and BoxE, indicating that our hyperbolic model, to some extent, is capable of capturing label hierarchies without any explicit constraints. However, as Table 5.3 shows, a proper $\lambda = 0.001$, $\lambda = 0.005$ and $\lambda = 0.01$ indeed improves the performance and consistency. Finally, we observed that increasing λ to 0.1, though further improves consistency (HCV and ECV), does not further improve mAP and CmAP. We conjecture that this is because a large λ would encourage the model to "overfit" the given constraints while "underfitting" the classification loss.

Table 5.3: Impact of violation penalty weight λ on CellcycleFUN and CellcycleGO dataset.

Dataset	Metric	$\lambda = 0.0$	$\lambda = 0.001$	$\lambda = 0.005$	$\lambda = 0.01$	$\lambda = 0.1$
CellcycleFUN	mAP	33.87	34.78	**34.82**	34.76	32.28
	CmAP	34.03	34.83	**34.90**	34.85	33.75
	HCV	2.33	1.87	1.30	1.04	**0.75**
	ECV	4.33	3.77	2.40	1.67	**1.35**
CellcycleGO	mAP	40.26	41.47	**45.58**	45.56	41.28
	CmAP	39.87	42.05	45.58	**45.60**	40.75
	HCV	2.28	1.57	1.19	0.99	**0.86**
	ECV	3.98	3.27	2.17	1.71	**1.34**

Impact of implication & exclusion To study the roles of implication and exclusion. We implemented three variants of HMI by removing either implication or exclusion, or removing both of them. Table 5.4 depicts the results of these variants. It is clear that both implication and exclusion constraints improve the base model that has no constraints. When implication and exclusion are jointly constrained, the performance is significantly improved again. We also observed that implication and exclusion constraints, to some extent, do complement each other, e.g., by only using implication (resp. exclusion), the model archives lower ECV (resp. HCV). Finally, we observed that even without exclusion, our model still slightly outperforms MBM, showcasing the advantages of hyperbolic space for modeling hierarchies.

Table 5.4: Impact of implication and exclusion constraints on CellcycleFUN and CellcycleGO dataset.

Dataset	Metric	HMI	w/o implication	w/o exclusion	non constraints
CellcycleFUN	mAP	**34.82**	34.70	34.74	33.87
	CmAP	**34.90**	34.75	34.82	34.03
	HCV	1.30	2.34	1.45	2.33
	ECV	2.40	2.67	3.63	4.33
CellcycleGO	mAP	**45.58**	42.56	44.50	40.26
	CmAP	**45.58**	42.56	45.31	39.87
	HCV	1.19	2.16	1.73	2.28
	ECV	2.17	3.68	3.07	3.98

Impact of sampling ratio To study whether our method is able to preserve logical constraints from incomplete label constraints we compare the performance of HMI with different ratios for sampling the training constraints. As Figure 5.4(a) depicts, with zero sampling ratio, our method already achieves acceptable results. We conjecture that this is because some constraints can be learned from the data. However, Figure 5.4(a) clearly shows that including constraints indeed helps to improve the performance. Making the sampling ratio larger than 30-40% does not lead to a significant performance gain. We

conjecture that this is because certain ratio of training constraints is sufficient for inferring the full set of constraints.

Impact of embedding dimensionality We study how the choice of dimensionality affects performance. As Figure 5.4(b) depicts, HMI achieves acceptable results even in a very low dimension ($n \leq 100$). When increasing the dimension an order of magnitude ($n = 1000$), the performance grows only slightly. Note that all reported baselines achieved acceptable results with dimensions in $[500, 1000, 1750]$ (see hyperparameter settings in the Appendix of Patel et al. [130]). We conjecture that the reason we can achieve good performance with fewer dimensions is that the hyperbolic hyperplane is more suitable for representing hierarchical decision boundaries.

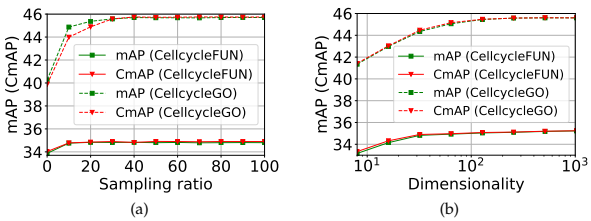


Figure 5.4: (a) The variation of performance w.r.t the sampling ratio. (b) The variation of performance w.r.t the embedding dimensions.

Comparison with MBM with only implication or without any constraint To faithfully study the advantages of hyperbolic hyperplane on modeling label relations than that of the box model (MBM), we also implement two versions of HMI by considering only (30%) implication constraints and without any constraint (sampling ratio= 0), respectively. Our Wilcoxon test in Table 5.5 shows that HMI with only implication and HMI without any constraint still outperform their corresponding counterparts of MBM on CmAP and HCV (with p-value < 0.05) while achieving comparable results on mAP (i.e., with better average ranks but without statistical significance, we believe this is because mAP is less sensitive to the constraints than CmAP).

Table 5.5: Results of Wilcoxon test on HMI against MBM in the settings where only implications are available and without any constraint. $-$ means no statistical difference between the compared methods.

Method	mAP	CmAP	CV
HMI (impl.) vs MBM (impl.)	$-$	2.4×10^{-4}	1.2×10^{-3}
HMI (no conts.) vs MBM (no conts.)	$-$	1.3×10^{-2}	6.1×10^{-3}

5.5 CONCLUSION

In this work, we focus on a structured multi-label prediction task whose output is supposed to respect the implication and exclusion constraints. We show that such a problem can be formulated in a hyperbolic Poincaré ball space whose linear decision boundaries (Poincaré hyperplanes) can be interpreted as convex regions. The implication and exclusion constraints are geometrically interpreted as insideness and disjointedness, respectively. Experiments on 12 datasets show significant improvements in mean average

precision and lower constraint violations, even with an order of magnitude fewer dimensions than baselines.

6

GEOMETRIC EMBEDDINGS OF HIGH-ORDER STRUCTURES

In this chapter, we introduce two geometric embeddings for high-order relational knowledge graphs. In Section 6.1, we introduce ShrinkE, a shrinking embedding for hyper-relational knowledge graphs. In Section 6.2, we introduce FactE, an embedding model for knowledge graphs with nested structures.

6.1 SHRINKING EMBEDDINGS FOR HYPER-RELATIONAL KNOWLEDGE GRAPHS

Link prediction on knowledge graphs (KGs) has been extensively studied on binary relational KGs, wherein each fact is represented by a triple. A significant amount of important knowledge, however, is represented by hyper-relational facts where each fact is composed of a primal triple and a set of qualifiers comprising a key-value pair that allows for expressing more complicated semantics. Although some recent works have proposed to embed hyper-relational KGs, these methods fail to capture essential inference patterns of hyper-relational facts such as qualifier monotonicity, qualifier implication, and qualifier mutual exclusion, limiting their generalization capability. To unlock this, we present *ShrinkE*, a geometric hyper-relational KG embedding method aiming to explicitly model these patterns. ShrinkE models the primal triple as a spatial-functional transformation from the head into a relation-specific box. Each qualifier "shrinks" the box to narrow down the possible answer set and, thus, realizes qualifier monotonicity. The spatial relationships between the qualifier boxes allow for modeling core inference patterns of qualifiers such as implication and mutual exclusion. Experimental results demonstrate ShrinkE's superiority on three benchmarks of hyper-relational KGs.

6.1.1 *Motivation and Background*

Link prediction on knowledge graphs (KGs) is a central problem for many KG-based applications [34, 105, 106, 188, 198]. Existing works [19, 151]s have mostly studied link prediction on binary relational KGs, wherein each fact is represented by a triple, e.g., (*Einstein*, *educated_at*, *University of Zurich*). In many popular KGs such as Freebase [17], however, a lot of important knowledge is not only expressed in triple-shaped facts, but also via facts about facts, which taken together are called hyper-relational facts. For

example, ((*Einstein*, *educated_at*, *University of Zurich*), {(*major*:*physics*), (*degree*:*PhD*)}) is a hyper-relational fact, where the primary triple (*Einstein*, *educated_at*, *University of Zurich*) is contextualized by a set of key-value pairs {(*major*:*physics*),(*degree*:*PhD*)}. Like much other related work, we follow the terminology established for Wikidata [164] and use the term *qualifiers* to refer to the key-value pairs.[1] The qualifiers play crucial roles in avoiding ambiguity issues. For instance, *Einstein* was *educated_at* several universities and the qualifiers for *degree* and *major* help distinguish them.

In order to predict links in hyper-relational KGs, pioneering works represent each hyper-relational fact as either an n-tuple in the form of $r(e_1, e_2, \cdots, e_n)$ [1, 52, 102] or a set of key-value pairs in the form of $\{(k_i : v_i)\}_{i=1}^{m}$ [65, 66, 103]. However, these modelings lose key structure information and are incompatible with the RDF-star schema [4] used by modern KGs, where both primal triples and qualifiers constitute the fundamental data structure. Recent works [64, 137] represent each hyper-relational fact as a primary triple coupled with a set of qualifiers that are compatible with RDF-star standards [4]. Link prediction is then achieved by modeling the validity of the primary triple and its compatibility with each annotated qualifier [64, 137]. More complicated graph encoders and decoders [53, 141, 170, 197] are proposed to further boost the performance. However, they require a relatively huge number of parameters that make them prone to overfitting.

To encourage generalization capability, KG embeddings should be able to model inference patterns, i.e., specifications of logical properties that may exist in KGs, which, if learned, empowers further principled inferences [1]. This has been extensively studied for binary relational KG embeddings [151, 156] but ignored for hyper-relational KGs in which not only primal triples but also qualifiers matter. One of the most important properties is qualifier monotonicity. Given a query, the answer set shrinks or at least does not expand as more qualifiers are added to the query expression. For example, a query $(Einstein, educated\_at, ?x)$ with a variable $?x$ corresponds to two answers $\{University\ of\ Zurich, ETH\ Zurich\}$, but a query $((Einstein, educated\_at, ?x), \{(degree : B.Sc.)\})$ extended by a qualifier for *degree* will only respond with $\{ETH\ Zurich\}$. Besides, different qualifiers might form logical relationships that the model must respect during inference including qualifier implication (e.g., adding a qualifier that is implicitly implied in the existing qualifiers does not change the truth of a fact) and qualifier mutual exclusion (e.g., adding any two mutually exclusive qualifiers to a fact leads to a contradiction).

In light of this, we propose ShrinkE, a hyper-relational embedding model that allows for modeling these inference patterns. ShrinkE embeds each entity as a point and models a primal triple as a spatio-functional transformation from the head entity to a relation-specific box that entails the possible tails. Each qualifier is modeled as a shrinking of the primal box to a qualifier box. The shrinking of boxes simulates the "monotonicity" of hyper-relational qualifiers, i.e., attaching qualifiers to a primal triple may only narrow down but never enlarges the answer set. The plausibility of a given fact is measured by a point-to-box function that judges whether the tail entity is inside the

1 Synonyms include statement-level *metadata* in RDF-star [4] and triple *annotation* in provenance communities [61].

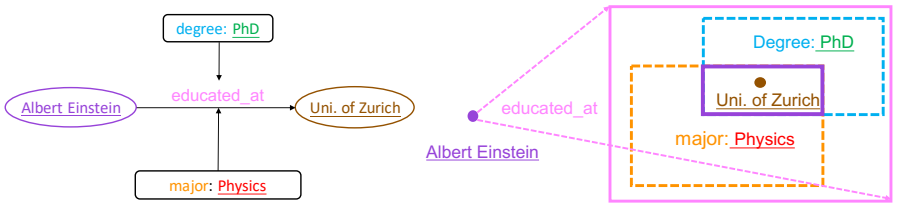


Figure 6.1: An illustration of the proposed idea. (a) A hyper-relational fact is composed of a primal triple and two key-value qualifiers, in which entities (values) are underlined while relations (keys) are not. (b) An illustration of the proposed hyper-relational KG embedding model ShrinkE. ShrinkE models the primal triple as a relation-specific transformation from the head entity to a query box (*purple*) that entails the possible answer entities. Each qualifier is modeled as a shrinking of the query box (*orange* and *cyan*) such that the shrinking box is a subset of the query box. The shrinking of the box can be viewed as a geometric interpretation of the monotonicity assumption that we follow. The final answer entities are supposed to be in the intersection box of all shrinking boxes.

intersection of all qualifier boxes. Moreover, since each qualifier is associated with a box, the spatial relationships between the qualifier boxes allow for modeling core inference patterns such as qualifier implication and mutual exclusion. We theoretically show the capability of ShrinkE on modeling various inference patterns including (fact-level) monotonicity, triple-level, and qualifier-level inference patterns. Empirically, ShrinkE achieves competitive performance on three benchmarks.

6.1.2 *Shrinking Embeddings for Hyper-Relational KGs*

We aim to design a scoring function $f(\cdot)$ taking the embeddings of facts as input so that the output values respect desired logical properties. To this end, we introduce primal triple embedding and qualifier embedding, respectively, as illustrated in Fig. 6.1.

6.1.2.1 *Primal Triple Embedding*

We represent each entity as a point $\mathbf{e} \in \mathbb{R}^d$. Each primal relation r is modeled as a spatio-functional transformation $\mathcal{B}_r : \mathbb{R}^d \to \text{Box}(d)$ that maps the head $\mathbf{e}_h \in \mathbb{R}^d$ to a d-dimensional box in $\text{Box}(d)$ with $\text{Box}(d)$ being the set of boxes in $\mathbb{R}^d$. Each box can be parameterized by a lower left point $\mathbf{m} \in \mathbb{R}^d$ and an upper right point $\mathbf{M} \in \mathbb{R}^d$, given by

$$\begin{aligned} &\text{Box}^d(\mathbf{m}, \mathbf{M}) = \\ &\{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{m}_i \leq \mathbf{x}_i \leq \mathbf{M}_i,\ i = 1, \cdots, d\}. \end{aligned} \tag{6.1}$$

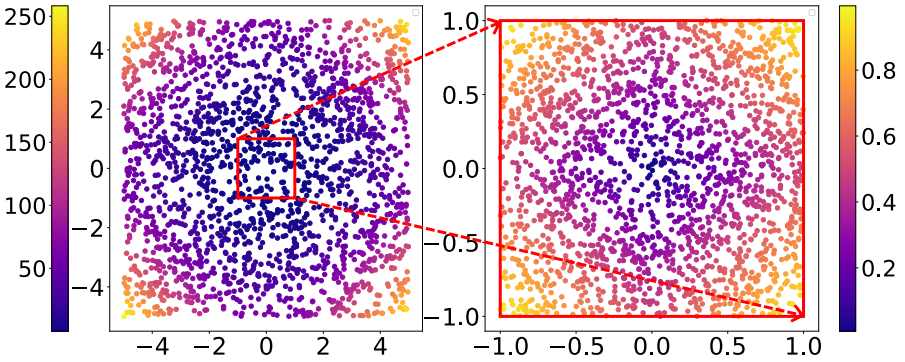


Figure 6.2: An illustration of the point-to-box distance. The distance (visualized by color maps) grows slowly when the point is inside of the box (*right*) while growing faster when the point is outside of the box (*left*).

We leave the superscript of Box^d away if it is clear from context. and call the transformed box a query box. Intuitively, all points in the query box correspond to the possible answer tail entities. Hence, the query box can be viewed as a geometric embedding of the answer set. Note that a query could result in an empty answer set. In order to capture such property, we do not exclude empty boxes that correspond to queries with empty answer set. Empty boxes are covered by the cases where there exists a dimension i such that $\mathbf{m}_i \geq \mathbf{M}_i$.

Point-to-box transform The spatio-functional point-to-box transformation $\mathcal{B}$ is composed of a relation-specific point transformation $\mathcal{H}_r : \mathbb{R}^d \to \mathbb{R}^d$ that transforms the head point $\mathbf{e}_h$ to a new point, and a relation-specific spanning that spans the transformed point to a box, formally given by

$$\mathcal{B}_r(\mathbf{e}_h) = \text{Box}(\mathcal{H}_r(\mathbf{e}_h) - \tau(\mathbf{ffi}_r), \mathcal{H}_r(\mathbf{e}_h) + \tau(\mathbf{ffi}_r)), \tag{6.2}$$

where $\mathbf{ffi}_r \in \mathbb{R}^n$ is a relation-specific spanning/offset vector, and $\tau_t(\mathbf{x}) = t \log\left(1 + e^{\mathbf{x}/t}\right)$ with t being a temperature hyperparameter, is a *softplus* function that enforces the spanned box to be non-empty.

The point transformation function $\mathcal{H}_r$ could be any functions that are used in other KG embedding models such as translation used in TransE [19] and rotations used in RotatE [151]. Hence, our model is highly flexible and effective at embedding primal triples. To allow for capturing multiple triple-level inference patterns such as symmetry, inversion, and composition, we combine translation and rotation, and formulate $\mathcal{H}_r$ as

$$\mathcal{H}_r(\mathbf{e}_h) = \Theta_r \mathbf{e}_h + \mathbf{b}_r \tag{6.3}$$

where Θ_r is a rotation matrix and $\mathbf{b}_r$ is a translation vector. We parameterize the rotation matrix by a block diagonal matrix $\Theta_r = \text{diag}\,\mathbf{G}\left(\theta_{r,1}\right), \ldots, \mathbf{G}\left(\theta_{r,\frac{d}{2}}\right)$, where

$$\mathbf{G}(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{6.4}$$

Point-to-box distance The validity of a primal triple (h, r, t) is then measured by judging whether the tail entity point $\mathbf{e}_t$ is geometrically inside of the query box. Given a query box $\text{Box}^n(\mathbf{m}, \mathbf{M})$ and an entity point $\mathbf{e} \in \mathbb{R}^d$, we denote the center point as $\mathbf{c} = \frac{\mathbf{m}+\mathbf{M}}{2}$. Let $|\cdot|$ denote the L1 norm and $\max()$ denote an element-wise maximum operation. The point-to-box distance is given by

$$\begin{aligned} D(\mathbf{e}, \text{Box}(\mathbf{m}, \mathbf{M})) &= \frac{|\mathbf{e} - \mathbf{c}|_1}{|\max(\mathbf{0}, \mathbf{M} - \mathbf{m})|_1} \\ &+ \left(|\mathbf{e} - \mathbf{m}|_1 + |\mathbf{e} - \mathbf{M}|_1 - |\max(\mathbf{0}, \mathbf{M} - \mathbf{m})|_1\right)^2 . \end{aligned} \tag{6.5}$$

Fig. 6.2 visualizes the distance function. Intuitively, in cases where the point is in the query box, the distance grows relatively slowly and inversely correlates with the box size. In cases where the point is outside the box, the distance grows fast.

6.1.2.2 *Qualifier Embedding*

Conceptually, qualifiers add information to given primary facts potentially allowing for additional inferences, but never for the retraction of inferences, reflecting the monotonicity of the representational paradigm. Corresponding to the non-declining number of inferences, the number of possible models for this representation shrinks, which can be intuitively reflected by a reduced size of boxes incurred by adding qualifiers.

Box Shrinking To geometrically mimic this property in the embedding space, we model each qualifier $(k : v)$ as a "shrinking" of the query box. Given a box $\text{Box}(\mathbf{m}, \mathbf{M})$, a shrinking is defined as a box-to-box transformation $\mathcal{S} : \text{Box} \rightarrow \text{Box}$ that potentially shrinks the volume of the box while not moving the resulting box outside of the source box. Let $\mathbf{L} = (\mathbf{M} - \mathbf{m})$ denote the side length vector, box shrinking is defined by

$$\begin{aligned} &\mathcal{S}_{r,k,v}\left(\text{Box}\left(\mathbf{m}, \mathbf{M}\right)\right) = \\ &\text{Box}\left(\mathbf{m} + \sigma\left(\mathbf{s}_{r,k,v}\right) \odot \mathbf{L}, \mathbf{M} - \sigma\left(\mathbf{S}_{r,k,v}\right) \odot \mathbf{L}\right), \end{aligned} \tag{6.6}$$

where $\mathbf{s}_{r,k,v} \in \mathbb{R}^n$ and $\mathbf{S}_{r,k,v} \in \mathbb{R}^n$ are the "shrinking" vectors for the lower left corner and the upper right corner, respectively. σ is a *sigmoid* function and $\odot$ is element-wise vector multiplication. The resulting box, including the case of empty box, is always inside the query box, i.e., $\mathcal{S}(\text{Box}^n(\mathbf{m}, \mathbf{M})) \subseteq \text{Box}^n(\mathbf{m}, \mathbf{M})$, which exactly resembles the qualifier monotonicity.

We use r, k, v as the indices of the shrinking vectors because the shrinking of the box should depend on the relatedness between the primal relation and the qualifier. For example, if a qualifier $(\textit{degree} : \textit{bachelor})$ is highly related to the primal relation *educated_at*, the scale of the shrinking vectors should be small as it adds a weak constraint to the triple. If the qualifier is unrelated to the primal relation, e.g., $(\textit{degree} : \textit{bachelor})$ and *born_in*, the shrinking might even enforce an empty box.

To learn the shrinking vectors, we leverage an MLP layer that takes the primal relation and key-value qualifier as input and outputs the shrinking

vectors defined by $\mathbf{s}_{r,k,v}, \mathbf{S}_{r,k,v} = \text{MLP}\left(\text{concat}\left(r_\theta, k_\theta, v_\theta\right)\right)$ where $r_\theta, k_\theta, v_\theta$ are the embeddings of r, k, v, respectively.

6.1.2.3 *Scoring function and learning.*

Scoring function

The score of a given hyper-relational fact is defined by

$$f\left(\left(\left(h, r, t\right), \mathcal{Q}\right)\right) = D(\mathbf{e}_t, \text{Box}_{\mathcal{Q}}(\mathbf{m}, \mathbf{M})), \tag{6.7}$$

where $\text{Box}_{\mathcal{Q}}(\mathbf{m}, \mathbf{M})$ denotes the target box that is calculated by the intersection of all shrinking boxes of the qualifier set $\mathcal{Q}$. The intersection of n boxes can be calculated by taking the maximum of lower left points of all boxes and taking the minimum of upper right points of all boxes, given by

$$\mathcal{I}(\text{Box}_1, \cdots, \text{Box}_n) = \text{Box}\left(\max_{i \in 1, \cdots, n} \mathbf{m}_i, \min_{i \in 1, \cdots, n} \mathbf{M}_i\right). \tag{6.8}$$

Note that if there is no intersection between boxes, this intersection operation still works as it results in an empty box. The intersection of boxes is a permutation-invariant operation, implying that perturbing the order of qualifiers does not change the plausibility of the facts.

Learning As a standard data augmentation strategy, we add reciprocal relations $\left(t', r^{-1}, h'\right)$ for the primary triple in each hyper-relational fact. For each positive fact in the training set, we generate n_{neg} negative samples by corrupting a subject/tail entity with randomly selected entities from $\mathcal{E}$. We adopt the cross-entropy loss to optimize the model via the Adam optimizer, which is given by

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \left(y_i \log\left(p_i\right) + \sum_{i=1}^{n_{\text{neg}}} \left(1 - y_i\right) \log\left(1 - p_i\right) \right), \tag{6.9}$$

where N denotes the total number of facts in the training set. y_i is a binary indicator denoting whether a fact is true or not. $p_i = \sigma(f(\mathcal{F}))$ is the predicted score of a fact $\mathcal{F}$ with σ being the *sigmoid* function.

6.1.3 *Theoretical Analysis*

Analyzing and modeling inference patterns is of great importance for KG embeddings because it enables generalization capability, i.e., once the patterns are learned, new facts that respect the patterns can be inferred. An inference pattern is a specification of a logical property that may exist in a KG, Formally, an inference pattern is a logical form $\psi \rightarrow \phi$ with ψ and ϕ being the body and head, implying that if the body is satisfied then the head must also be satisfied.

In this section, we analyze the theoretical capacity of ShrinkE for modeling inference patterns. All proofs of propositions are in Appendix A.5.

Fact-level inference pattern (monotonicity) The following proposition shows that ShrinkE is able to model monotonicity.

Proposition 12. *Given any two facts $\mathcal{F}_1 = (\mathcal{T}, \mathcal{Q}_1)$ and $\mathcal{F}_2 = (\mathcal{T}, \mathcal{Q}_2)$ where $\mathcal{Q}_2 \subseteq \mathcal{Q}_1$, i.e., $\mathcal{F}_2$ is a partial fact of $\mathcal{F}_1$, the output of the scoring function $f(\cdot)$ of ShrinkE satisfy the constraint $f(\mathcal{F}_2) \geq f(\mathcal{F}_1)$.*

Triple-level inference patterns Prominent triple-level inference patterns include symmetry $(h, r, t) \rightarrow (t, r, h)$, anti-symmetry $(h, r, t) \rightarrow \neg(h, r, t)$, inversion $(h, r_1, t) \rightarrow (t, r_2, h)$, composition $(e_1, r_1, e_2) \wedge (e_2, r_2, e_3) \rightarrow (e_1, r_3, e_3)$, relation implication $(h, r_1, t) \rightarrow (h, r_2, t)$, relation intersection $(h, r_1, t) \wedge (h, r_2, t) \rightarrow (h, r_3, t)$, and relation mutual exclusion $(h, r_1, t) \wedge (h, r_2, t) \rightarrow \bot$. All these triple-level inference patterns also exist in hyper-relational facts when their qualifiers are the same, e.g., hyper-relational symmetry means $((h, r, t), \mathcal{Q}) \rightarrow ((t, r, h), \mathcal{Q})$. Proposition 13 states that ShrinkE is able to infer all of them.

Proposition 13. *ShrinkE is able to infer hyper-relational symmetry, anti-symmetry, inversion, composition, relation implication, relation intersection, and relation exclusion.*

Qualifier-level inference pattern In hyper-relational KGs, inference patterns not only exist at the triple level but also at the level of qualifiers.

Definition 24 (qualifier implication)**.** *Given two qualifiers q_i and q_j, q_i is said to imply q_j, i.e., $q_i \rightarrow q_j$ iff for any fact $\mathcal{F} = (\mathcal{T}, \mathcal{Q})$, if attaching q_i to $\mathcal{Q}$ results in a true (resp. false) fact, then attaching q_j to $\mathcal{Q} \cup \{q_i\}$ also results in a true (resp. false) fact. Formally, $q_i \rightarrow q_j$ implies*

$$\forall\, \mathcal{T}, \mathcal{Q} : (\mathcal{T}, \mathcal{Q} \cup \{q_i\}) \rightarrow (T, Q \cup \{q_i, q_j\})\,. \tag{6.10}$$

Definition 25 (qualifier exclusion)**.** *Two qualifiers q_i, q_j are said to be mutually exclusive iff for any fact $\mathcal{F} = (\mathcal{T}, \mathcal{Q})$, by attaching q_i, q_j to the qualifier set of $\mathcal{F}$, the new fact $\mathcal{F}' = (\mathcal{T}, \mathcal{Q} \cup \{q_i, q_j\})$ is false, meaning that they lead to a contradiction, i.e., $q_i \wedge q_j \rightarrow \bot$. Formally, $q_i \wedge q_j \rightarrow \bot$ implies*

$$\forall\, \mathcal{T}, \mathcal{Q} \,:\, (T, Q \cup \{q_i, q_j\}) \rightarrow \bot \tag{6.11}$$

Note that if two qualifiers q_i, q_j are neither mutually exclusive nor forming implication pair, then q_i, q_j are said to be overlapping, a state between implication and mutual exclusion. Qualifier overlapping, in our case, can be captured by box intersection/overlapping. Qualifier overlapping itself does not form any logical property in the form of $\psi \rightarrow \phi$. However, when involving three qualifiers and two of them overlap, qualifier intersection can be modeled.

Table 6.1: Dataset statistics, where the columns indicate the number of all facts, hyper-relational facts with the number of qualifiers $m > 0$, entities, relations, and facts in train/dev/test sets, respectively.

	All facts	Higher-arity facts (%)	Entities	Relations	Train	Dev	Test
JF17K	100,947	46,320 (45.9%)	28,645	501	76,379	–	24,568
WikiPeople	382,229	44,315 (11.6%)	47,765	193	305,725	38,223	38,281
WD50k	236,507	32,167 (13.6%)	47,156	532	166,435	23,913	46,159
WD50K(33)	102,107	31,866 (31.2%)	38,124	475	73,406	10,568	18,133
WD50K(66)	49,167	31,696 (64.5%)	27,347	494	35,968	5,154	8,045
WD50K(100)	31,314	31,314 (100%)	18,792	279	22,738	3,279	5,297

Definition 26 (qualifier intersection)**.** *A qualifier q_k is said to be an intersection of two qualifiers q_i, q_j iff for any fact $\mathcal{F} = (\mathcal{T}, \mathcal{Q})$, if attaching q_i, q_j to $\mathcal{Q}$ results in a true (resp. false) fact, then by replacing $\{q_i, q_j\}$ with q_k, the truth value of the fact does not change. Namely, $q_i \wedge q_j \to q_k$ implies*

$$\forall\, \mathcal{T}, \mathcal{Q} : \big(T, \mathcal{Q} \cup \{q_i, q_j\}\big) \to (\mathcal{T}, \mathcal{Q} \cup \{q_k\})\,. \tag{6.12}$$

Apparently, qualifier intersection $q_i \wedge q_j \to q_k$ necessarily implies qualifier implications $q_i \to q_k$ and $q_j \to q_k$. Hence, qualifier intersection can be viewed as a combination of two qualifier implications, and this can be generalized to $q_1 \wedge q_2 \wedge \cdots \to q_k$. Proposition 14 shows that ShrinkE is able to infer qualifier implication, exclusion, and composition.

Proposition 14. *ShrinkE is able to infer qualifier implication, mutual exclusion, and intersection.*

6.1.4 *Evaluation*

In this section, we evaluate the effectiveness of ShrinkE on hyper-relational link prediction tasks.

6.1.4.1 *Experimental Setup*

Datasets. We conduct link prediction experiment on three hyper-relational KGs: JF17K [177], WikiPeople [66], and WD50k [53]. JF17K is extracted from Freebase while WikiPeople and WD50k are extracted from Wikidata. In WikiPeople and WD50k, only 11.6% and 13.6% of the facts, respectively, contain qualifiers, while the remaining facts contain only triples (after dropping statements containing literals in WikiPeople, only 2.6% facts contain qualifiers). For better comparison, we also consider three splits of WD50K that contain a higher percentage of triples with qualifiers. The three splits are WD50K(33), WD50K(66), and WD50K(100), which contain 33%, 66%, and 100% facts with qualifiers, respectively. Statistics of the datasets are given in Table 6.1. We conjecture that the performance on WikiPeople and WD50k will be dominated by the scores of triple-only facts while the performance on the variants of WD50k will be dominated by the modeling of qualifiers. [53]

Table 6.2: Link prediction results on three benchmarks with the number in the parentheses denoting the ratio of facts with qualifiers. Baseline results are taken from [53].

Method	WikiPeople (2.6)			JF17K (45.9)			WD50K (13.6)		
	MRR	H@1	H@10	MRR	H@ 1	H@ 10	MRR	H@ 1	H@ 10
m-TransH	0.063	0.063	0.300	0.206	0.206	0.463	–	–	–
RAE	0.059	0.059	0.306	0.215	0.215	0.469	–	–	–
NaLP-Fix	0.420	0.343	0.556	0.245	0.185	0.358	0.177	0.131	0.264
NeuInfer	0.350	0.282	0.467	0.451	0.373	0.604	–	–	–
HINGE	0.476	0.415	0.585	0.449	0.361	0.624	0.243	0.176	0.377
Transformer	0.469	0.403	0.586	0.512	0.434	0.665	0.264	0.194	0.401
BoxE	0.395	0.293	0.503	0.560	0.472	0.722	–	–	–
StarE	**0.491**	0.398	**0.648**	0.574	0.496	0.725	**0.349**	0.271	**0.496**
ShrinkE	0.485	**0.431**	0.601	**0.589**	**0.506**	**0.749**	0.345	**0.275**	0.482

detected a data leakage issue of JF17K, i.e., about 44.5% of the test facts share the same primal triple as the train facts. To alleviates this issue, WD50K removes all facts from train/validation sets that share the same primal triple with test facts. We conjecture that WD50K will be a more challenging benchmark than JF17K and WikiPeople. Besides, WD50K still contains only a small percentage (13.6%) of facts that contain qualifiers. Since JF17K does not provide a validation set, we split 20% of facts from the training set as the validation set.

Environments and hyperparameters We implement ShrinkE with Python 3.9 and Pytorch 1.11, and train our model on one Nvidia A100 GPU with 40GB of VRAM. We use Adam optimizer with a batch size of 128 and an initial learning rate of 0.0001. For negative sampling, we follow the strategy used in StarE [53] by randomly corrupting the head or tail entity in the primal triple. Different from HINGE [137] and NeuInfer [64] that score all potential facts one by one that takes an extremely long time for evaluation, ShrinkE ranks each target answer against all candidates in a single pass and significantly reduces the evaluation time. We search the dimensionality from $[50, 100, 200, 300]$ and the best one is 200. We set the temperature parameter to be $t = 1.0$. We use the label smoothing strategy and set the smoothing rate to be 0.1. We repeat all experiments for 5 times with different random seeds and report the average values, the error bars are relatively small and are omitted.

Baselines We compare ShrinkE against various models, including m-TransH [177], RAE [199], NaLP-Fix [137], HINGE [137], NeuInfer [64], BoxE [**boxE**], Transformer and StarE [53]. Note that we exclude Hy-Transformer [197], GRAN [170] and QUAD [141] for comparison because 1) they are heavily based on StarE and Transformer; and 2) they leverage auxiliary training tasks, which can also be incorporated into our framework and we leave as one future work.

Evaluation We strictly follow the settings of [53], where the aim is to predict a missing head/tail entity in a hyper-relational fact. We consider the widely used ranking-based metrics for link prediction: mean reciprocal rank (MRR) and H@K (K=1,10). For ranking calculation, we consider the filtered setting by filtering the facts existing in the training and validation sets [19].

Table 6.3: Link prediction results on WD50K splits with the number in the parentheses denoting the ratio of facts with qualifiers. Baseline results are taken from [53].

Method	WD50K (33)			WD50K (66)			WD50K (100)		
	MRR	H@1	H@10	MRR	H@ 1	H@ 10	MRR	H@ 1	H@ 10
NaLP-Fix	0.204	0.164	0.277	0.334	0.284	0.423	0.458	0.398	0.563
HINGE	0.253	0.190	0.372	0.378	0.307	0.512	0.492	0.417	0.636
Transformer	0.276	0.227	0.371	0.404	0.352	0.502	0.562	0.499	0.677
StarE	0.331	0.268	**0.451**	0.481	0.420	0.594	0.654	0.588	0.777
ShrinkE	**0.336**	**0.272**	0.449	**0.511**	**0.422**	**0.611**	**0.695**	**0.629**	**0.814**

Table 6.4: Example pairs of qualifiers with implication relations (body $\rightarrow$ head). $X \in$ [*Eric Schmidt*, *Mark Zuckerberg*, *Dustin Moskovitz*, *Larry Page*] denotes a CEO name of a company. $Y \in [112, 115, 113, \cdots]$ and $Z \in [912, 18, 192, \cdots]$ are emergency numbers involving *police* and *fire department*, respectively. Qualifier exclusion pairs are ubiquitous and are hence omitted.

body	head
(residence: Monte Carlo)	(country, Monaco)
(residence: Belgrade)	(country, Serbia)
(owned_by: X)	(of, voting interest)
(emergency phone number: Y)	(has_use, police)
(emergency phone number: Z)	(has_use, fire department)
(used_by: software)	(via, operating_system)

6.1.4.2 *Main Results and Analysis*

Table 6.2 and Table 6.3 summarize the performances of all approaches on the six datasets. Overall, ShrinkE achieves either the best or the second-best results against all baselines, showcasing the expressivity and capability of ShrinkE on hyper-relational link prediction. In particular, We observe that ShrinkE outperforms all baselines on JF17K and the three variants of WD50K with a high ratio of facts containing qualifiers while achieving highly competitive results on WikiPeople and the original version of WD50K that contain fewer facts with qualifiers. Interestingly, we find that the performance gains increase when increasing the ratio of facts containing qualifiers. On WD50K (100) where 100% facts contain qualifiers, the performance gain of ShrinkE is most significant across all metrics (6.2%, 6.9%, and 4.7% improvements over MRR, H@1, and H@10, respectively). We believe this is because that ShrinkE is excellent at modeling qualifiers due to its explicit modeling of inference patterns, while other models relying on tremendous parameters tend to be overfitting.

Case analysis Table 6.4 shows some examples of qualifier implication pairs recovered by our learned embeddings. Note that exclusions pairs are ubiquitous (i.e., most of the random qualifiers are mutually exclusive) and hence we do not analyze them. We find that some qualifier implications happen when they are about geographic information and involve geographic inclusion such as *Monte Carlo* is in *Monaco*. Interestingly, we find that qualifiers associated with key *owned_by* imply (*of, voting interest*), and qualifiers with key

Method	MRR	H@ 1	H@ 10
ShrinkE (w/o translation)	0.583	0.495	0.729
ShrinkE (w/o rotation)	0.581	0.497	0.724
ShrinkE (w/o shrinking)	0.571	0.490	0.711
ShrinkE	**0.589**	**0.506**	**0.749**

Table 6.5: The performance of ShrinkE by removing one relational component on JF17K.

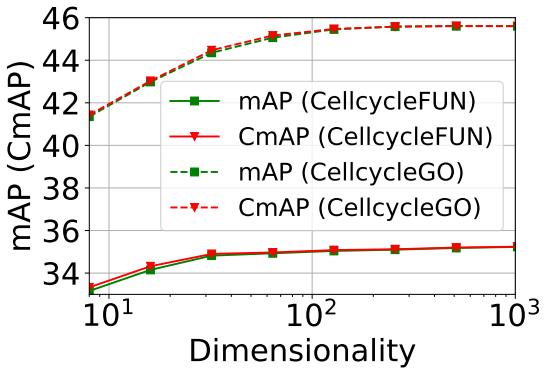


Figure 6.3: Performance of ShrinkE with different dimensions $d = [4, 8, 16, 32, 64, 128, 256]$ on JF17K.

emergency phone number imply (*has_use*, *police*) or (*has_use*, file department), which conceptually make sense.

6.1.4.3 *Ablations and Parameter Sensitivity*

Impact of relational components To determine the importance of each component in relational modeling, we conduct an ablation study by considering three versions of ShrinkE in which one of the components (translation, rotation, and shrinking) is removed. Table 6.5 shows that the removal of each component of the relational transformation leads to a degradation in performance, validating the importance of each component. In particular, by removing the qualifier shrinking, which is the main contribution of our framework, the performance reduces 3% and 5% in MRR and H@10, respectively, showcasing the usefulness of modeling qualifiers as shrinking. The removals of translation and rotation both result in around 1% and 2% reduction in MRR and H@10, respectively.

Impact of dimensionality We conduct experiments on JF17K under a varied number of dimensions $d = [4, 8, 16, 32, 64, 128, 256]$. As Fig. 6.3 depicts, the performance increases when increasing the number of dimensions. However, the growth trend gradually flattens with the increase of dimensions and it achieves comparable performance when the dimension is higher than 128.

6.1.4.4 *Discussion*

Comparison with neural network models Heavy neural network models such as GRAN [170] and QUAD [141] are built on relational GNNs and/or Transformers and require a large number of parameters. In contrast, ShrinkE is a neuro-symbolic model that requires only one MLP layer and a much smaller number of parameters. The logical modelling of ShrinkE makes it more explainable than GNN-based and Transformer-based methods.

Comparison with other box embeddings in KGs ShrinkE is the first to not only represent hyper-relational facts, but also explicitly model the logical properties of these facts. SrinkE is different from previous box embedding methods [1] of KGs in three key modules: 1) our point-to-box transform function modelling triple inference patterns; 2) a new point-to-box distance function; and 3) we introduce box shrinking to model qualifier-level inference patterns. Moreover, we provide a comprehensive theoretical analysis of ShrinkE on modelling various logical properties.

6.1.5 *Conclusion*

We present a novel hyper-relational KG embedding model ShrinkE. ShrinkE models a primal triple as a spatio-functional transformation while modeling each qualifier as a shrinking that monotonically narrows down the answer set. We proved that ShrinkE is able to spatially infer core inference patterns at different levels including triple-level, fact-level, and qualifier-level. Experimental results on three benchmarks demonstrate the advantages of ShrinkE in predicting hyper-relational links.

6.2 MODELING RELATIONSHIPS BETWEEN FACTS FOR KNOWLEDGE GRAPH REASONING

Reasoning with knowledge graphs (KGs) has primarily focused on triple-shaped facts. Recent advancements have been explored to enhance the semantics of these facts by incorporating more potent representations, such as hyper-relational facts. However, these approaches are limited to *atomic facts*, which describe a single piece of information. This paper extends beyond *atomic facts* and delves into *nested facts*, represented by quoted triples where subjects and objects are triples themselves (e.g., ((*BarackObama*, *holds_position*, *President*), *succeed_by*, (*DonaldTrump*, *holds_position*, *President*))). These nested facts enable the expression of complex semantics like *situations* over time and *logical patterns* over entities and relations. In response, we introduce FactE, a novel KG embedding approach that captures the semantics of both atomic and nested factual knowledge. FactE represents each atomic fact as a 1×3 matrix, and each nested relation is modeled as a 3×3 matrix that rotates the 1×3 atomic fact matrix through matrix multiplication. Each element of the matrix is represented as a complex number in the generalized 4D hypercomplex space, including (spherical) quaternions, hyperbolic quater-

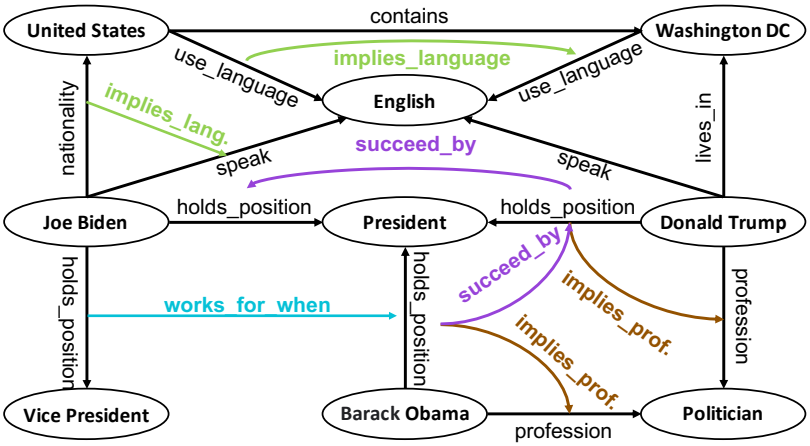


Figure 6.4: An example of a nested factual KG consisting of 1) a set of atomic facts describing the relationship between entities and 2) a set of nested facts describing the relationship between atomic facts. Nested factual relations are colored and they either describe situations in/over time (e.g., *succeed_by* and *works_for_when*) or logical patterns (e.g., *implies_profession* and *implies_language*).

nions, and split-quaternions. Through thorough analysis, we demonstrate the embedding's efficacy in capturing diverse logical patterns over nested facts, surpassing the confines of first-order logic-like expressions. Our experimental results showcase FactE's significant performance gains over current baselines in triple prediction and conditional link prediction. The code is attached as supplemental material and will be made publicly available.

6.2.1 *Motivation and Background*

Knowledge graphs (KGs) depict relationships between entities, commonly through triple-shaped facts such as (*JoeBiden*, *holds_position*, *VicePresident*). KG embeddings map entities and relations into a lower-dimensional vector space while retaining their relational semantics. This empowers the effective inference of missing relationships between entities directly from their embeddings. Prior research [19, 151, 156] has primarily centered on embedding triple-shaped facts and predicting the missing elements of these triples. Yet, to augment the triple-shaped representations, recent endeavors explore knowledge that extends beyond these triples. For instance, n-ary facts [52, 102] describe relationships between multiple entities, and hyper-relational facts [53, 137] augment primal triples with key-value qualifiers that provide contextual information. These approaches allow for expressing complex semantics and enable answering more sophisticated queries with additional knowledge [2].

However, these beyond-triple representations typically focus only on relationships between entities that jointly define an *atomic fact*, overlooking the significance of relationships that describe multiple facts together. Indeed, within a KG, each atomic fact may have a relationship with another atomic

fact. Consider the following two atomic facts: T_1=(*JoeBiden*, *holds_position*, *VicePresident*) and T_2=(*BarackObama*, *holds_position*, *President*). We can depict the scenario where *JoeBiden* held the position of *VicePresident* under the *President BarackObama* using a triple $(T_1, works\_for\_when, T_2)$. Such a fact about facts is referred to as a *nested fact* [2] and the relation connecting these two facts is termed a *nested relation*. Fig. 6.4 provides an illustration of a KG containing both atomic and nested facts.

These nested relations play a crucial role in expressing complex semantics and queries in two ways: 1) **Expressing situations involving facts in or over time**. This facilitates answering complex queries that involve multiple facts. For example, KG embeddings face challenges when addressing queries like "*Who was the president of the USA after* DonaldTrump*?*" because the query about the primary fact (?, *holds_position*, *President*) depends on another fact (*DonaldTrump*, *holds_position*, *President*). As depicted in Fig. 6.4, *succeed_by* conveys such temporal situation between these two facts, allowing the direct response to the query through conditional link prediction; 2) **Expressing logical patterns (implications) using a non-first-order logical form** $\psi \xrightarrow{\hat{r}} \phi$. As illustrated in Fig. 6.4, (*Location A*, *uses_language*, *Language B*) $\overset{implies\_language}{\rightarrow}$ (*Location in A*, *uses_language*, *Language B*) represents a logical pattern, as it holds true for all pairs of (*Location A*, *Location in A*). Modeling such logical patterns is crucial as it facilitates generalization. Once these patterns are learned, new facts adhering to these patterns can be inferred. A recent study [38] explored link prediction over nested facts.[3] However, their method embeds facts using a multilayer perceptron (MLP), which fails to capture essential logical patterns and thus has limited generalization capabilities.

In this paper, we introduce FactE, an innovative approach designed to embed the semantics of both atomic facts and nested facts that enable representing temporal situations and logical patterns over facts. FactE represents each atomic fact as a 1×3 hypercomplex matrix, with each element signifying a component of the atomic fact. Furthermore, each nested relation is modeled through a 3×3 hypercomplex matrix that rotates the 1×3 atomic fact matrix via a matrix-multiplicative Hamilton product. Our matrix-like modeling for facts and nested relations demonstrates the capacity to encode diverse logical patterns over nested facts. The modeling of these logical patterns further enables efficient modeling of logical rules that extend beyond the first-order-logic-like expressions (e.g., Horn rules). Moreover, we propose a more general hypercomplex embedding framework that extends the quaternion embedding [200] to include hyperbolic quaternions and split-quaternions. This generalization of hypercomplex space allows for expressing rotations over hyperboloid, providing more powerful and distinct inductive biases for embedding complex structural patterns (e.g., hierarchies). Our experimental findings on triple prediction and conditional link prediction showcase the remarkable performance gain of FactE.

2 This is also called a *quoted triple* in RDF star [28].

3 In their work, the KG is referred to as a bi-level KG, and the term "high-level facts" is synonymous with nested facts.

6.2.2 *Related Work*

Describing relationships between facts Rule-based approaches [45, 68, 109, 124, 138, 193] consider relationships between facts, but they are confined to first-order-logic-like expressions (i.e., Horn rules), i.e., $\forall e_1, e_2, e_3 : (e_1, r_1, e_2) \wedge (e_2, r_2, e_3) \Rightarrow (e_1, r_3, e_3)$, where there must exist a path connecting e_1, e_2, and e_3 in the KG. Notably, [38] marked an advancement by examining KG embeddings with relationships between facts as nested facts, denoted as $(x, r_1, y) \overset{\hat{r}}{\Rightarrow} (p, r_2, q)$. The proposed embeddings (i.e., BiVE-Q and BiVE-B[4]) concatenate the embeddings of the head, relation, and tail, subsequently embedding them via an MLP. However, such modeling does do not explicitly capture crucial logical patterns over nested facts, which bear significant importance in KG embeddings [151, 156].

Algebraic and geometric embeddings Algebraic embeddings like QuatE [200] and BiQUE [67] represent relations as algebraic operations and score triples using inner products. They can be viewed as a unification of many earlier functional [19] and multiplication-based [156] models. Geometric embeddings like hyperbolic embeddings [12, 26] further extend the functional models to non-Euclidean hyperbolic space, enabling the representation of hierarchical relations.

6.2.3 *FactE: Embedding Atomic and Nested Facts*

6.2.3.1 *Unified Hypercomplex Embeddings*

We first extend QuatE [200], a KG embedding in 4D hypercomplex quaternion space, into a more general 4D hypercomplex number system including three variations: (spherical) quaternions, hyperbolic quaternions, and split quaternions. Each of these 4D hypercomplex numbers is composed of one real component and three imaginary components denoted by $s + xi + yj + zk$ with $s, x, y, z \in \mathbb{R}$ and i, j, k being the three imaginary parts. The distinctive feature among these hyper-complex number systems lies in their multiplication rules of the imaginary components.

(Spherical) quaternions $\mathcal{Q}$ follow the multiplication rules:

$$\begin{aligned} &i^2 = j^2 = k^2 = 1, \\ &ij = k = -ji, jk = i = -kj, ki = j = -ik. \end{aligned} \tag{6.13}$$

4 Note that BiVE-B, despite being described as based on the biquaternian–BiQUE, employs quaternion space with an additional translation component based on our analysis of the code.

Hyperbolic quaternions $\mathcal{H}$ follow the multiplication rules:

$$\begin{aligned} &i^2 = -1, j^2 = k^2 = 1, \\ &ij = k, jk = -i, ki = j,\ ji = -k, kj = i, ik = -j. \end{aligned} \tag{6.14}$$

Split quaternions $\mathcal{S}$ follow the multiplication rules:

$$\begin{aligned} &i^2 = -1, j^2 = k^2 = 1, \\ &ij = k, jk = -i, ki = j,\ ji = -k, kj = i, ik = -j. \end{aligned} \tag{6.15}$$

Geometric intuitions The distinctions in the multiplication rules of various hypercomplex numbers give rise to different geometric spaces that provide suitable inductive biases for representing different types of relations. Specifically, spherical quaternions, hyperbolic quaternions, and split quaternions with the same norm c correspond to 4D hypersphere, Lorentz model of hyperbolic space (i.e., the upper part of the double-sheet hyperboloid), and pseudo-hyperboloid (i.e., one-sheet hyperboloid, with curvature $\sqrt{c}$, respectively. These are denoted as follows:

$$\begin{aligned} &|\mathcal{Q}| = s^2 + x^2 + y^2 + z^2 = c > 0 \text{ (hypersphere)} \\ &|\mathcal{H}| = s^2 - x^2 - y^2 - z^2 = c > 0 \text{ (Lorentz hyperbolic space)} \\ &|\mathcal{S}| = s^2 + x^2 - y^2 - z^2 = c > 0 \text{ (pseudo-hyperboloid)}. \end{aligned} \tag{6.16}$$

The 3D versions of them are shown in Fig. 6.5. These spaces have well-known characteristics: spherical spaces are adept at modeling cyclic relations [172], hyperbolic spaces provide geometric inductive biases for hierarchical relations [26], and the pseudo-hyperboloid [189] offers a balance between spherical and hyperbolic spaces, making it suitable for embedding both cyclic and hierarchical relations. Moreover, by representing relations as geometric rotations over these spaces (i.e., Hamilton product), fundamental logical patterns such as symmetry, inversion, and compositions can be effectively inferred [26, 189, 200]. Our proposed embeddings can be viewed as a unification of previous approaches that leverages these geometric inductive biases in these geometric spaces within a single geometric algebraic framework.

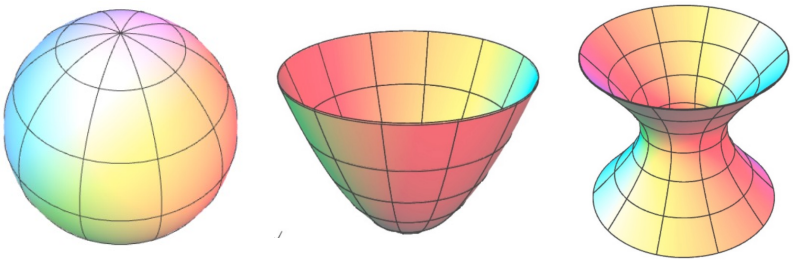
For convenience, we parameterize each entity and relation as a Cartesian product of d 4D hypercomplex numbers $\mathbf{s} + \mathbf{x}i + \mathbf{y}j + \mathbf{z}k$, where $\mathbf{s}, \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^d$. This enables us to define all algebraic operations involving these hypercomplex vectors in an element-wise manner.

6.2.3.2 *Atomic Fact Embeddings*

Each atomic relation is represented by a rotation hypercomplex vector $\mathbf{r}_\theta$ and a translation hypercomplex vector $\mathbf{r}_b$. For a given triple (h, r, t), we apply the following operation:

$$\mathbf{h}' = (\mathbf{h} \oplus \mathbf{r}_b) \otimes \mathbf{r}_\theta, \tag{6.17}$$

where $\oplus$ and $\otimes$ stand for addition and Hamilton product between hypercomplex numbers, respectively. The addition involves an element-wise sum of



(a) Hypersphere (b) Lorentz hyperbolic space (c) Pseudo-hyperboloid

Figure 6.5: Visualization of the hypersphere, Lorentz hyperbolic space, and pseudo-hyperboloid in 3D space.

each hypercomplex component. The Hamilton product rotates the head entity. To ensure proper rotation on the unit sphere, we normalize the rotation hypercomplex number $\mathbf{r}_\theta = \mathbf{s}_r^\theta + \mathbf{x}_r^\theta i + \mathbf{y}_r^\theta j + \mathbf{z}_r^\theta k$ by $\mathbf{r}_\theta = \frac{\mathbf{s}_r^\theta + \mathbf{x}_r^\theta \boldsymbol{i} + \mathbf{y}_r^\theta \boldsymbol{j} + \mathbf{z}_r^\theta \boldsymbol{k}}{\sqrt{\mathbf{s}_r^{\theta 2} + \mathbf{x}_r^{\theta 2} + \mathbf{y}_r^{\theta 2} + \mathbf{z}_r^{\theta 2}}}$. Hamilton product is defined by combining the components of the hypercomplex numbers.

$$
\begin{aligned}
\mathbf{h}' &= \mathbf{h} \otimes \mathbf{r}_\theta \\
&= \left(\mathbf{s}_h \circ \mathbf{s}_r^\theta \circ 1 + \mathbf{x}_h \circ \mathbf{x}_r^\theta \circ i^2 + \mathbf{y}_h \circ \mathbf{y}_r^\theta \circ j^2 + \mathbf{z}_h \circ \mathbf{z}_r^\theta \circ k^2\right) \\
&+ \left(\mathbf{s}_h \circ \mathbf{x}_r^\theta \circ i + \mathbf{x}_h \circ \mathbf{s}_r^\theta \circ i + \mathbf{y}_h \circ \mathbf{z}_r^\theta \circ jk + \mathbf{z}_h \circ \mathbf{y}_r^\theta \circ kj\right) \\
&+ \left(\mathbf{s}_h \circ \mathbf{y}_r^\theta \circ j + \mathbf{x}_h \circ \mathbf{z}_r^\theta \circ ik + \mathbf{y}_h \circ \mathbf{s}_r^\theta \circ j + \mathbf{z}_h \circ \mathbf{x}_r^\theta \circ ik\right) \\
&+ \left(\mathbf{s}_h \circ \mathbf{z}_r^\theta \circ k + \mathbf{x}_h \circ \mathbf{y}_r^\theta \circ ij + \mathbf{y}_h \circ \mathbf{x}_r^\theta \circ ij + \mathbf{z}_h \circ \mathbf{s}_r^\theta \circ k\right) \\
&= \mathbf{s}_{h'} + \mathbf{x}_{h'} i + \mathbf{y}_{h'} j + \mathbf{z}_{h'} k,
\end{aligned}
\tag{6.18}
$$

where the multiplication of imaginary components follows the rules (Eq.1-3) of the chosen hypercomplex systems.

The scoring function $\phi(h, r, t)$ is defined as:

$$
\phi(h, r, t) = \langle \mathbf{h}', \mathbf{t} \rangle = \langle \mathbf{s}_{h'}, \mathbf{s}_t \rangle + \langle \mathbf{x}_{h'}, \mathbf{x}_t \rangle + \langle \mathbf{y}_{h'}, \mathbf{y}_t \rangle + \langle \mathbf{z}_{h'}, \mathbf{z}_t \rangle , \tag{6.19}
$$

where $\langle \cdot, \cdot \rangle$ represents the inner product.

6.2.3.3 *Nested Fact Embeddings*

To represent an atomic fact (h, r, t) without losing information, we embed each atomic triple as a 1×3 matrix, where each column corresponds to the embedding of the respective element. Consequently, we have $\mathbf{T}_i = [\mathbf{h}_i, \mathbf{r}_i, \mathbf{t}_i]$.

To embed various shapes of nested relations between T_i and T_j with relation $\widehat{r}$, we model each nested relation using a 1×3 translation matrix and a 3×3 rotation matrix, where each element is a 4D hypercomplex number.

Specifically, we first translate the head triple $\mathbf{T}_i$ with $\widehat{\mathbf{r}}_b$, followed by applying a matrix-like rotation of $\widehat{\mathbf{r}}_\theta$, as defined by

$$\mathbf{T}i' = (\mathbf{T}i \oplus_{1\times 3} \widehat{\mathbf{r}}_b) \otimes_{3\times 3} \widehat{\mathbf{r}}_\theta, \tag{6.20}$$

where the matrix addition $\oplus_{1\times 3}$ is performed through an element-wise summation of the hypercomplex components within the matrices. The matrix-like Hamilton product $\otimes_{3\times 3}$ is defined as a product akin to matrix multiplication:

$$\begin{aligned} \mathbf{T}'_i = \mathbf{T}_i \otimes_{3\times 3} \widehat{\mathbf{r}}_\theta = \begin{bmatrix} \mathbf{h}_i \\ \mathbf{r}_i \\ \mathbf{t}_i \end{bmatrix}^\top \times \begin{bmatrix} \widehat{\mathbf{r}}^\theta_{11} & \widehat{\mathbf{r}}^\theta_{12} & \widehat{\mathbf{r}}^\theta_{13} \\ \widehat{\mathbf{r}}^\theta_{21} & \widehat{\mathbf{r}}^\theta_{22} & \widehat{\mathbf{r}}^\theta_{23} \\ \widehat{\mathbf{r}}^\theta_{31} & \widehat{\mathbf{r}}^\theta_{32} & \widehat{\mathbf{r}}^\theta_{33} \end{bmatrix} = \\ \begin{bmatrix} \mathbf{h}_i \otimes \widehat{\mathbf{r}}^\theta_{11} + \mathbf{r}_i \otimes \widehat{\mathbf{r}}^\theta_{21} + \mathbf{t}_i \otimes \widehat{\mathbf{r}}^\theta_{31} \\ \mathbf{h}_i \otimes \widehat{\mathbf{r}}^\theta_{12} + \mathbf{r}_i \otimes \widehat{\mathbf{r}}^\theta_{22} + \mathbf{t}_i \otimes \widehat{\mathbf{r}}^\theta_{32} \\ \mathbf{h}_i \otimes \widehat{\mathbf{r}}^\theta_{13} + \mathbf{r}_i \otimes \widehat{\mathbf{r}}^\theta_{23} + \mathbf{t}_i \otimes \widehat{\mathbf{r}}^\theta_{33} \end{bmatrix}^\top = \begin{bmatrix} \mathbf{h}'_i \\ \mathbf{r}'_i \\ \mathbf{t}'_i \end{bmatrix}^\top, \end{aligned} \tag{6.21}$$

where $\otimes$ is the Hamilton product.

Remarks This matrix-like modeling of nested facts provides flexibility to capture diverse shapes of logical patterns inherent in nested relations. In essence, different shapes of situations or patterns can be effectively modeled by manipulating the 3×3 rotation matrix. For instance, relational implications can be represented using a diagonal matrix, while inversion can be captured using an anti-diagonal matrix. See **theoretical justification** for further analysis.

To assess the plausibility of the nested fact $(T_i, \widehat{r}, T_j)$, we calculate the inner product between the transformed head $\mathbf{T}'_i$ fact and the tail fact $\mathbf{T}_j$ as:

$$\rho(T_i, \widehat{r}, T_j) = \langle \mathbf{T}'_i, \mathbf{T}_j \rangle, \tag{6.22}$$

where $\langle \cdot, \cdot \rangle$ denotes the matrix inner product.

Learning objective

We sum up the loss of atomic fact embedding $\mathcal{L}_{\text{atomic}}$, the loss of nested fact embedding $\mathcal{L}_{\text{meta}}$, and additionally the loss term $\mathcal{L}_{\text{aug}}$ for augmented triples generated by random walking as used in [38]. The overall loss is defined as

$$\mathcal{L} = \mathcal{L}_{\text{atomic}} + \lambda_1 \mathcal{L}_{\text{nested}} + \lambda_2 \mathcal{L}_{\text{aug}} \tag{6.23}$$

where λ_1 and λ_2 are the weight hyperparameters indicating the importance of each loss. Negative sampling is applied by randomly replacing one of the head or tail entity/triple. These losses are defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{atomic}} &= \sum_{(h,r,t)\in\mathcal{T}} g\left(-\phi\left(h,r,t\right)\right) + \sum_{g((h',r',t'))\notin\mathcal{T}} g\left(\phi\left(h',r',t'\right)\right) \\ \mathcal{L}_{\text{aug}} &= \sum_{(h,r,t)\in\mathcal{T}'} g\left(-\phi\left(h,r,t\right)\right) + \sum_{(h',r',t')\notin\mathcal{T}'} g\left(\phi(h',r',t')\right) \\ \mathcal{L}_{\text{nested}} &= \sum_{(T_i,\widehat{r},T_j)\notin\widehat{\mathcal{T}}} g\left(-\rho(T_i,\widehat{r},T_j)\right) + \sum_{(T'_i,\widehat{r},T'_j)\notin\widehat{\mathcal{T}}} g\left(\rho(T'_i,\widehat{r},T'_j)\right), \end{aligned} \tag{6.24}$$

where $g = \log(1 + \exp(x))$ and $\mathcal{T}'$ is the set of augmented triples.

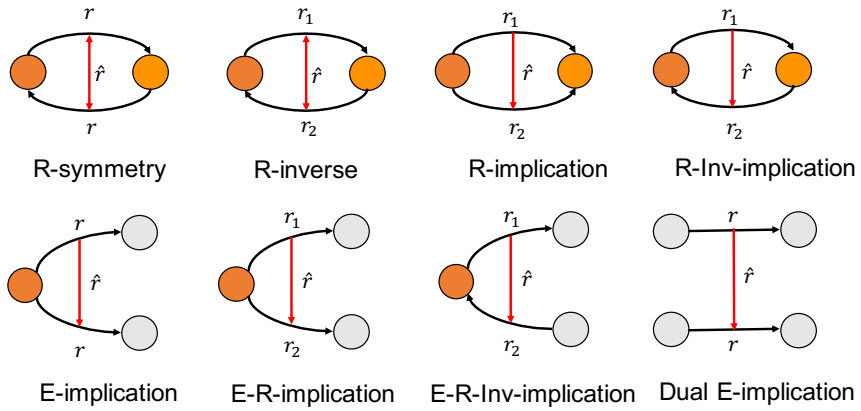


Figure 6.6: A structural illustration of different shapes of logical patterns, where the colored circles are free variables.

6.2.4 *Theoretical Justification*

Modeling logical patterns is of great importance for KG embeddings because it enables generalization, i.e., once the patterns are learned, new facts that respect the patterns can be inferred. A logical pattern is a logical form $\psi \rightarrow \phi$ with ψ and ϕ being the body and head, implying that if the body is satisfied then the head must also be satisfied.

First-order-logic-like logical patterns Existing KG embeddings studied logical patterns expressed in the first-order-logic-like form. Prominent examples include symmetry $\forall h, t\colon (h, r, t) \rightarrow (t, r, h)$, anti-symmetry $\forall h, t\colon (h, r, t) \rightarrow \neg(h, r, t)$, inversion $\forall h, t\colon (h, r_1, t) \rightarrow (t, r_2, h)$ and composition $\forall e_1, e_2, e_3\colon (e_1, r_1, e_2) \wedge (e_2, r_2, e_3) \rightarrow (e_1, r_3, e_3)$.

Proposition 15. *FactE can infer symmetry, anti-symmetry, inversion, and composition, regardless of the specific choices of hypercomplex number systems.*

This proposition holds because FactE subsumes ComplEx [156] (i.e., 4D complex numbers generalize 2D complex numbers).

Logical patterns over nested facts We extend the vanilla logical patterns in KGs to include nested facts. This can be expressed in a non-first-order-logic-like form $\psi \xrightarrow{\widehat{r}} \phi$.

- **Relational symmetry (R-symmetry):** an atomic relation r is symmetric w.r.t a nested relation $\widehat{r}$ if $\forall x, y \in \mathcal{E}, \langle x, r, y\rangle \overset{\widehat{r}}{\leftrightarrow} \langle y, r, x\rangle$.
- **Relational inverse (R-inverse):** two atomic relations r_1 and r_2 are inverse w.r.t a nested relation $\widehat{r}$ if $\forall x, y \in \mathcal{E}, (\langle x, r_1, y\rangle \overset{\widehat{r}}{\leftrightarrow} \langle y, r_2, x\rangle)$.

Table 6.6: Exemplary logical patterns of nested facts.

Pattern	$\widehat{r}$	triple template
R-Symmetry	EquivalentTo	(Person A, IsMarriedTo, Person B) (Person B, IsMarriedTo, Person A)
R-Inverse	EquivalentTo	(Location A, UsesLanguage, Language B) (Language B, IsSpokenIn, Location A)
R-Implication	ImpliesLocation	(Country A, CapitalIsLocatedIn, City B) (Country A, Contains, City B)
R-Inv-Implication	ImpliesLocation	(Organization A, Headquarter, Location B) (Location B, Contains, Organization A)
E-Implication	ImpliesTimeZone	(Location A, TimeZone, Time Zone B) (Location in A, TimeZone, Time Zone B)
E-R-Implication	ImpliesProfession	(Person A, HoldsPosition, Government Position B) (Person A, IsA, Politician)
E-R-Inv-Implication	ImpliesProfession	(Work A, CinematographyBy, Person B) (Person B, IsA, Cinematographer)
Dual E-Implication	ImpliesLocation	(Location A, Contains, Location B) (Location containing A, Contains, Location in B)

- **Relational implication (R-implication):** an atomic relation r_1 implies a atomic relation r_2 w.r.t a nested relation $\widehat{r}$ if $\forall x, y \in \mathcal{E}, (\langle x, r_1, y\rangle \xrightarrow{\widehat{r}} \langle x, r_2, y\rangle)$.

- **Relational inverse implication (R-Inv-implication):** an atomic relation r_1 inversely implies an atomic relation r_2 w.r.t a nested relation $\widehat{r}$ if $\forall x, y \in \mathcal{E}, (\langle x, r_1, y\rangle \xrightarrow{\widehat{r}} \langle y, r_2, x\rangle)$.

- **Entity implication (E-implication):** an entity x_1 (resp. y_1) implies entity x_2 (resp. y_2) w.r.t an atomic relation r and a nested relation $\widehat{r}$ if $\forall y \in \mathcal{E}, (\langle x_1, r, y\rangle \xrightarrow{\widehat{r}} \langle x_2, r, y\rangle)$ (resp. $\forall x \in \mathcal{E}, (\langle x, r, y_1\rangle \xrightarrow{\widehat{r}} \langle x, r, y_2\rangle)$).

- **Entity relational implication (E-R-implication):** an entity x_1 and relation r_1 (resp. y_1 and relation r_1) implies entity x_2 and relation r_2 (resp. y_2 and relation r_2) w.r.t a nested relation $\widehat{r}$ if $\forall y \in \mathcal{E}, (\langle x_1, r_1, y\rangle \xrightarrow{\widehat{r}} \langle x_2, r_2, y\rangle)$ (resp. $\forall x \in \mathcal{E}, (\langle x, r_1, y_1\rangle \xrightarrow{\widehat{r}} \langle x, r_2, y_2\rangle)$).

- **Entity relational inverse implication (E-R-Inv-implication):** an entity x_1 and relation r_1 (resp. y_1 and relation r_1) inversely implies entity x_2 and relation r_2 (resp. y_2 and relation r_2) w.r.t a nested relation $\widehat{r}$ if $\forall y \in \mathcal{E}, (\langle x_1, r_1, y\rangle \xrightarrow{\widehat{r}} \langle y, r_2, x_2\rangle)$ (resp. $\forall x \in \mathcal{E}, (\langle x, r_1, y_1\rangle \xrightarrow{\widehat{r}} \langle y_2, r_2, x\rangle)$).

- **Dual Entity implication (Dual E-implication):** an entity pair (x_1, x_2) implies another entity pair (y_1, y_2) iff both (x_1, y_1) and (x_2, y_2) satisfy E-implication.

Fig. 6.6 illustrates the structure of the introduced patterns and Table 6.6 presents exemplary patterns of nested facts.

Proposition 16. *FactE can infer R-symmetry, R-inverse, R-implication, R-Inv-implication, E-implication, E-R-implication, E-R-Inv-implication, and Dual E-implication.*

Table 6.7: Statistics of $\widehat{G} = (V, R, \mathcal{T}, \widehat{\mathcal{R}}, \widehat{\mathcal{T}})$. $|\mathcal{T}|'$ denotes the number of atomic triples involved in the nested triples.

	$\lvert V\rvert$	$\lvert R\rvert$	$\lvert\mathcal{T}\rvert$	$\lvert\widehat{\mathcal{R}}\rvert$	$\lvert\widehat{\mathcal{T}}\rvert$	$\lvert\mathcal{T}\rvert'$
FBH	14,541	237	310,117	6	27,062	33,157
FBHE	14,541	237	310,117	10	34,941	33,719
DBHE	12,440	87	68,296	8	6,717	8,206

To infer different logical patterns via different free variables, we can set some elements of the relation matrix to be zero-valued or one-valued complex numbers. For example, the implication and inverse implication relations can be inferred by setting the matrix to be diagonal or anti-diagonal. See Appendix for details.

6.2.5 *Experimental Results*

6.2.5.1 *Experiment Setup*

Datasets We utilize three benchmark KGs: FBH, FBHE, and DBHE, that contain nested facts and are constructed by [38]. FBH and FBHE are based on FB15K237 from Freebase [16] while DBHE is based on DB15K from DBpedia [6]. FBH contains only nested facts that can be inferred from the triple facts, e.g., *prerequisite_for* and *implies_position*, while FBHE and DBHE further contain externally-sourced knowledge crawled from Wikipedia articles, e.g., *next_almaMater* and *transfers_to*. The authors of [38] spent six weeks manually defining these nested facts and adding them to the KGs. Besides, we employ the same data augmentation strategies as used in the original paper, i.e., adding reverse relations and reversed triple to the training set, and using random walks to augment plausible triples. The dataset details are presented in Table 6.7. We split $\mathcal{T}$ and $\widehat{\mathcal{T}}$ into training, validation, and test sets in an 8:1:1 ratio.

Baselines We consider BiVE-Q and BiVE-B [38] as our major baselines as they are specifically designed for KGs with nested facts and have demonstrated significant improvements over triple-based methods. We also compare some rule-based approaches as they indirectly consider relations between facts in first-order-logic-like expression, including Neural-LP [193], DRUM [138], and AnyBURL [109]. We further include QuatE [200] and BiQUE [67] as they are the SoTA triple-based methods and they are also based on 4D hypercomplex numbers. However, these triple-based methods do not directly apply to the nested facts. Following [38], we create a new triple-based KG G_T where the atomic facts are converted into entities and nested facts are converted into triples (see Appendix for details). For our approach, we implement three variants of FactE: FactE-Q (using quaternions), FactE-H (using hyperbolic quaternions), FactE-S (split quaternions), as well as their counterparts with translations: FactE-QB, FactE-HB, and FactE-SB. In the Appendix, we also extend BiVE-Q and BiVE-B to other hypercomplex numbers: BiVE-H, BiVE-HB BiVE-S, and BiVE-SB for further comparison. We employ three standard

Table 6.8: Results of triple prediction. Shaded numbers are better results than the best baseline. The best scores are boldfaced and the second best scores are underlined. * denotes results taking from [38].

	FBH			*FBHE*			*DBHE*		
	MR (↓)	MRR (↑)	Hit@10 (↑)	MR (↓)	MRR (↑)	Hit@10 (↑)	MR (↓)	MRR (↑)	Hit@10 (↑)
QuatE*	145603.8	0.103	0.114	94684.4	0.101	0.209	26485.0	0.157	0.179
BiQUE*	81687.5	0.104	0.115	61015.2	0.135	0.205	19079.4	0.163	0.185
Neural-LP*	115016.6	0.070	0.073	90000.4	0.238	0.274	21130.5	0.170	0.209
DRUM*	115016.6	0.069	0.073	90000.3	0.261	0.274	21130.5	0.166	0.209
AnyBURL*	108079.6	0.096	0.108	83136.8	0.191	0.252	20530.8	0.177	0.214
BiVE-Q	6.20	0.855	0.941	8.35	0.711	0.866	3.63	0.687	0.958
BiVE-B	8.63	0.833	0.924	9.53	0.705	0.860	4.66	0.718	0.945
FactE-Q (Ours)	6.56	0.863	0.953	5.77	0.811	0.943	3.51	0.809	0.960
FactE-H (Ours)	4.69	0.858	0.964	3.99	0.781	0.943	2.65	0.806	0.969
FactE-S (Ours)	3.87	0.867	0.977	3.60	0.795	0.947	2.55	0.809	0.966
FactE-QB (Ours)	6.04	0.898	0.958	5.55	0.845	0.947	2.54	0.847	0.973
FactE-HB (Ours)	3.82	0.899	0.971	3.53	0.828	0.955	2.62	0.842	0.972
FactE-SB (Ours)	**3.34**	**0.922**	**0.982**	**3.05**	**0.851**	**0.962**	**2.07**	**0.862**	**0.984**

Table 6.9: Results of conditional link prediction. Shaded numbers are better results than the best baseline. The best scores are boldfaced and the second best scores are underlined. * denotes results taking from [38].

	FBH			*FBHE*			*DBHE*		
	MR (↓)	MRR (↑)	Hit@10 (↑)	MR (↓)	MRR (↑)	Hit@10 (↑)	MR (↓)	MRR (↑)	Hit@10 (↑)
QuatE*	163.7	0.346	0.494	1546.4	0.124	0.189	551.6	0.208	0.309
BiQUE*	111.0	0.423	0.641	90.1	0.387	0.617	29.5	0.378	0.677
Neural-LP*	185.9	0.433	0.648	146.2	0.466	0.716	32.2	0.517	0.756
DRUM*	262.7	0.394	0.555	207.6	0.413	0.620	49.0	0.470	0.732
AnyBURL*	228.5	0.380	0.563	166.0	0.418	0.607	81.7	0.403	0.594
BiVE-Q	4.33	0.826	0.948	6.56	0.761	0.886	2.69	0.852	0.971
BiVE-B	5.34	0.836	0.940	7.49	0.761	0.872	2.91	0.858	0.967
FactE-Q (Ours)	1.70	0.930	0.986	2.89	0.863	0.948	**1.68**	0.930	0.987
FactE-H (Ours)	1.68	0.909	0.987	2.87	0.843	0.945	1.82	0.912	0.986
FactE-S (Ours)	1.54	0.925	**0.991**	3.04	0.850	0.941	1.76	0.910	0.988
FactE-QB (Ours)	1.71	**0.935**	0.987	3.00	0.865	0.949	1.70	**0.931**	0.986
Fact-HB (Ours)	1.60	0.924	0.989	2.76	0.855	0.950	1.92	0.918	0.981
FactE-SB (Ours)	**1.52**	0.934	**0.991**	**2.61**	**0.867**	**0.951**	1.72	0.919	**0.990**

metrics: Filtered MR (Mean Rank), MRR (Mean Reciprocal Rank), and Hit@10. We report the mean performance over 10 random seeds for each method, and the relatively small standard deviations are omitted.

Implementation details We implement the framework based on OpenKE [5] and the code [6]. We train our methods on triple prediction and evaluate them on other tasks. The dimensionality is set to be $d = 200$. We reuse the hyperparameters as used in [38] and do not perform further hyperparameter searches. The detailed hyperparameter settings can be found in the Appendix.

6.2.5.2 *Main Results*

Triple prediction

5 https://github.com/thunlp/OpenKE
6 https://github.com/bdi-lab/BiVE/

Table 6.8 presents the results of triple prediction. First, it shows that all triple-based approaches yield relatively modest results compared to BiVE-Q and BiVE-B, designed specifically for KGs with nested facts. Our approach, FactE-Q, the quaternionic version, already outperforms the baselines across most metrics. Particularly notable are the pronounced enhancements in FBHE and DBHE, with MRR improvements of 14.1% and 17.7% respectively, underscoring the efficacy of the proposed FactE model. Furthermore, FactE-H and FactE-S demonstrate heightened performance over FactE-Q across various evaluation metrics, particularly in terms of MR. This highlights the advantages that hyperbolic quaternions and split quaternions offer over standard quaternions. Impressively, the split quaternionic version attains the highest performance, followed closely by the hyperbolic quaternionic variant. Moreover, through the incorporation of a hypercomplex translation component, FactE-QB, Fact-HB, and FactE-SB consistently outperform their non-translation counterparts, illustrating the advantages of combining multiple transformations (rotation and translation) within the hypercomplex space.

Conditional link prediction Table 6.9 shows the outcomes of conditional link prediction. It is evident that all three FactE variants substantially outperform the two SoTA baselines, BiVE-Q and BiVE-B, across all datasets. Notably, the best FactE variant surpasses the baselines by 11.8%, 13.9%, and 8.5% in terms of MRR for FBH, FBHE, and DBHE, respectively. This remarkable performance gain underscores the effectiveness of the proposed method. Similar to the trends observed in triple prediction, the incorporation of translation components in FactE-QB, Fact-HB, and Fact-SB leads to further improvements over their counterparts without translation components. This reaffirms the advantages gained from the integration of multiple hypercomplex transformations. Intriguingly, we noticed that varying hypercomplex number systems yield the best performance on different datasets, contrasting the observations from triple prediction. We conjecture that this stems from the inherent variance in inductive biases offered by different hypercomplex number systems, making them more suitable for certain datasets over others. We believe the choices of spaces can be linked to a hyperparameter that offers flexibility in adapting to diverse dataset characteristics.

Base link prediction Table 6.10 illustrates the results of base link prediction. Among our approaches, namely FactE-Q, FactE-H, and FactE-S, we observe competitive or improved results in comparison to SoTA embedding-based and rule-based methods on the FBHE and DBHE datasets. The best performance is achieved by FactE-QB, which outperforms the baselines across a majority of metrics. This outcome substantiates the fact that the incorporation of nested facts into triple-based KGs indeed enhances the inference capabilities for base link prediction.

6.2.5.3 *Ablation Analysis*

Embedding analysis of logical patterns To verify whether the learned embeddings capture the inference of logical patterns over nested facts, we visualized the real part of the embeddings of the 8 relations in DBHE. The

Table 6.10: Results of base link prediction. The best scores are boldfaced and the second best scores are underlined. * denotes results taking from [38].

	FBHE			*DBHE*		
	MR (↓)	MRR (↑)	Hit@10 (↑)	MR (↓)	MRR (↑)	Hit@10 (↑)
QuatE*	139.0	0.354	0.581	409.6	0.264	0.440
BiQUE*	134.9	0.356	0.583	**376.6**	0.274	**0.446**
Neural-LP*	1942.5	0.315	0.486	2904.8	0.233	0.357
DRUM*	1945.6	0.317	0.490	2904.7	0.237	0.359
AnyBURL*	342.0	0.310	0.526	879.1	0.220	0.364
BiVE-Q	136.13	0.369	0.603	827.18	0.271	0.428
BiVE-B	136.54	<u>0.370</u>	<u>0.607</u>	795.59	0.274	0.422
FactE-Q (Ours)	<u>131.72</u>	0.365	0.605	749.75	<u>0.284</u>	**0.446**
Fact-H (Ours)	153.00	0.349	0.593	868.82	0.266	0.423
FactE-S (Ours)	149.64	0.350	0.592	895.85	0.272	0.432
FactE-QB (Ours)	**130.13**	**0.371**	**0.608**	751.18	**0.289**	<u>0.443</u>
Fact-HB (Ours)	155.74	0.353	0.594	801.76	0.271	0.423
FactE-SB (Ours)	149.73	0.355	0.594	827.89	0.273	0.431

Table 6.11: Ablation study on the nested fact embeddings for base link prediction. Best results are boldfaced.

	FBHE			*DBHE*		
	MR (↓)	MRR (↑)	Hit@10 (↑)	MR (↓)	MRR (↑)	Hit@10 (↑)
FactE-Q ($\lambda_1 = 0$)	135.38	**0.368**	0.604	799.97	0.281	0.431
FactE-Q ($\lambda_1 = 0.5$)	**131.72**	0.365	**0.605**	**749.75**	**0.284**	**0.446**
FactE-H ($\lambda_1 = 0$)	154.75	0.347	0.589	922.34	0.267	0.420
FactE-H ($\lambda_1 = 0.5$)	**153.00**	**0.349**	**0.593**	**868.82**	0.266	**0.423**
FactE-S ($\lambda_1 = 0$)	151.84	0.347	0.589	910.16	**0.272**	0.427
FactE-S ($\lambda_1 = 0.5$)	**149.64**	**0.350**	**0.592**	**895.85**	**0.272**	**0.432**

analysis of the embeddings yields insightful observations. As shown in Fig. 6.7, the lower left element and upper right element of the embedding *EquivalentTo* are 1, showcasing that *EquivalentTo* predominantly adheres to R-symmetry or R-inverse. On the other hand, the upper left element and lower right element of the *ImpliesLang.* are 1, affirming its alignment with the R-implication rule. Similarly, the embeddings of *NextAlmaM.*, *TransfersTo*, and *ImpliesGenre* indicate high adherence to E-implicationsas as only one of the corners is 1. We find that the embedding of relation *ImpliesProf.* does not have a significant pattern. We conjecture that this is because *ImpliesProf.* follows many rule patterns and there exists no global solution that satisfies all rules. See the Appendix for the statistics of the logical patterns in the datasets.

Influence of nested fact embeddings To evaluate the influence of nested fact embeddings, we perform a comparison by excluding the loss associated with nested fact embeddings (i.e., setting $\lambda_1 = 0$). The outcomes presented in Table 6.11 underscore the significant enhancements achieved by incorporating nested fact embeddings, particularly evident in the improvements in MR and H@10 for DBHE.

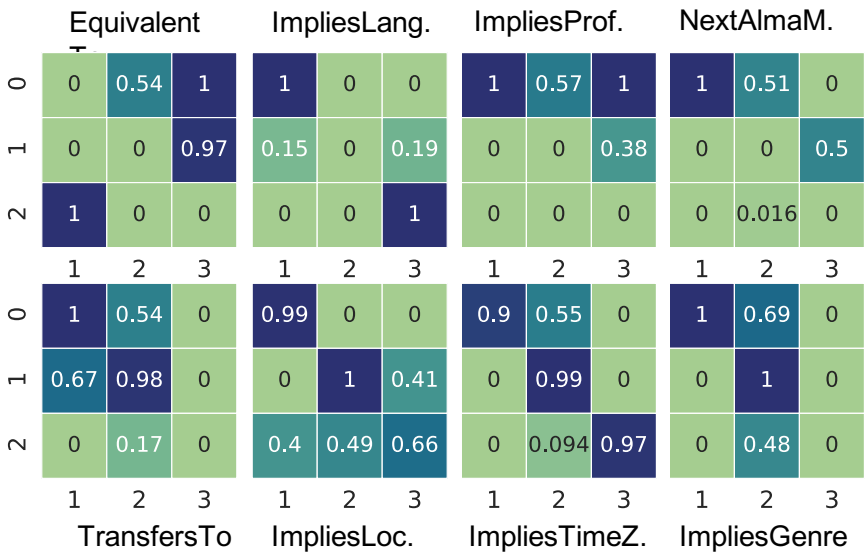


Figure 6.7: The visualization of the average of the real component embeddings of the 8 nested relations in DBHE.

Table 6.12: Performance per relation on triple prediction. Freq. indicates the number of nested facts in the test set.

$\widehat{r}$	Freq.	FactE-Q	FactE-QB	FactE-H	FactE-HB	FactE-S	FactE-SB
Equiv.To	98	0.994	**0.997**	0.992	**0.997**	**0.997**	0.995
ImpliesLang.	29	0.671	0.602	**0.680**	0.662	0.614	0.622
ImpliesProf.	210	0.807	0.916	0.830	0.935	0.832	**0.936**
ImpliesLocat.	163	0.929	0.869	0.893	0.810	0.929	**0.958**
ImpliesTime.	44	0.305	0.297	0.307	**0.329**	0.290	0.293
ImpliesGenre	84	0.726	0.762	0.719	0.741	0.742	**0.796**
NextAlmaM.	14	0.770	**0.812**	0.689	0.688	0.751	0.795
Transf.To	29	**0.977**	0.952	0.964	0.949	0.921	0.953

Relation-specific performance In Table 6.12, we present the performance results for each relation within the DBHE dataset. Notably, the diverse hypercomplex number systems lead to optimal performance for different relations. This reiterates our conjecture that distinct benefits are offered by varying hypercomplex number systems, catering to the specific characteristics of different relation types. Remarkably, our findings reveal that the incorporation of a hypercomplex translation component (as seen in FactE-QB, FactE-HB, and FactE-SB) notably enhances the embeddings of relations such as *ImpliesProf.* and *ImpliesGenre* across all variants of hypercomplex number systems. However, this does not extend to relations like *ImpliesLocat.* and *ImpliesLang.*, suggesting a more complex relationship between these specific relations and the hypercomplex translation components.

6.2.6 *Conclusion*

This paper considers a novel perspective by extending traditional atomic factual knowledge representation to include nested factual knowledge. This enables the representation of both temporal situations and logical patterns

that go beyond conventional first-order logic expressions (Horn rules). Our proposed approach, FactE, presents a family of hypercomplex embeddings capable of embedding both atomic and nested factual knowledge. This framework effectively captures essential logical patterns that emerge from nested facts. Empirical evaluation demonstrates the substantial performance enhancements achieved by FactE compared to existing baseline methods. Additionally, our generalized hypercomplex embedding framework unifies previous algebraic (e.g., quaternionic) and geometric (e.g., hyperbolic) embedding methods, offering versatility in embedding diverse relation types.

7

CONCLUSION, LIMITATIONS, AND FUTURE WORKS

7.1 CONCLUSION

Relational data offer a structured representation of real-world knowledge that has been applied in a wide range of applications. Many types of relational data, such as social networks, knowledge graphs, and ontologies, are incomplete and noisy due to the process of human curation. Relational representation learning aims to address these issues by mapping these relational objects into continual low-dimensional vector space such that the relational structure is preserved and missing relational knowledge can be inferred by the analogical or the similarity structure among the embedding vectors.

However, mapping relational data to a continual vector space is more challenging than the embeddings of image and text data that typically require only preservation of "similarity" between data objects. Relational data, besides requiring capturing similarity, exhibit various discrete properties that cannot be easily captured by the plain vectors in Euclidean space. In this dissertation, we consider geometric relational embeddings that map relational objects as geometric objects that faithfully model the discrete properties inherent in the relational data. In particular, we made the following contributions.

CHAPTER 3 introduces pseudo-Riemannian manifold embeddings. We propose a QGCN, pseudo-Riemannian graph convolutional networks (GCNs) that generalizes GCN in the pseudo-Riemannian manifolds. This GCN defines node embeddings in the pseudo-Riemannian manifolds allowing for capturing both hierarchical and cyclic structural patterns. Besides, we propose a pseudo-Riemannian knowledge graph embedding method that models entities in pseudo-Riemannian manifold and relations as pseudo-orthogonal transformation. The proposed KG embedding allows for simultaneous modeling of graph structural patterns (hierarchices and cycles) and relational patterns (symmstry, inversion, composition).

CHAPTER 4 introduces BoxEL, a geometric embedding that allows for better capturing the logical structure in a knowledge base that is expressed in the Description Logic EL++. BoxEL models concepts as boxes and models relations as affine transformation. This modeling faithfully models the intersectional closure and the complex relational mapping between concepts.

CHAPTER 5 introduces HMI, a hyperbolic embedding method that explicitly encodes the class hierarchy and exclusion relations for structured multi-label prediction. Embedding such relational constraints of classes onto embedding space is useful as it encourages the predictions to be coherent to the given relational constraints.

CHAPTER 6 introduces two high-order relational embeddings: 1) We propose ShrinkE, a hyper-relational embedding model that allows for modeling logical patterns over qualifiers, including monotonicity and qualifier-level logical patterns (ie..g, qualifier implication and qualifier exclusion); 2) We propose NestE, an innovative approach designed to embed the semantics of both atomic facts and nested facts that enable representing temporal situations and logical patterns over facts.

7.2 LIMITATIONS AND FUTURE WORKS

The current geometric embeddings still suffer from several limitations.

- **Shallow nature of geometric embeddings:** Although geometric embeddings have found applications in various relational reasoning tasks, many of these methods learn embeddings in a shallow manner. This becomes problematic when the input relational objects contain rich multi-modal features, such as images and text descriptions. In such cases, geometric embeddings should learn a neural network encoder that takes images or text as input and outputs the corresponding geometric objects.

- **Lack of freely chosen geometric inductive biases:** Different geometric inductive biases may be suitable for various inference tasks. Typically, geometric inductive biases are chosen based on prior human knowledge about the tasks. However, when a task requires the joint consideration of multiple properties, developing a geometric relational embedding that simultaneously captures these diverse properties remains highly challenging.

- **Increased computational and parameter requirements:** Geometric embeddings often necessitate additional computation or memory. For instance, hyperbolic embeddings involve the computation of exponential and logarithmic maps. Box embeddings require two embedding vectors, storing the lower-left corner and the upper-right corner, respectively.

In the future, we plan to explore the following directions to address and enhance our research:

- **Geometric inductive biases for imperfect learning settings.** Real-world data rarely align with perfection, and many existing methods

assume balanced training data, leading to biased predictions, often referred to as minority collapse. This poses challenges in generalizing to real-world scenarios. We plan to develop geometric relational biases explicitly incorporating inductive biases to promote equality. An example of such a bias is the maximum class separation. Generalizing these biases to geometric embeddings is non-trivial and requires specific geometric design considerations.

- **Heterogeneous hierarchies** Most current geometric embedding methods can only encode one hierarchy relation (e.g., *is_a*). However, a real-world KG might simultaneously contain multiple hierarchical relations (e.g., *is_a* and *has_part*) [131]. We plan to develop embedding models that simultaneously encode multiple hierarchies.

- **Injecting relational constraints into Large Language Models (LLMs).** Relational knowledge graphs offer explicit factual knowledge that complements the inherent lack of factual knowledge in LLMs. By learning explicit geometric KG embeddings, complex relational knowledge, such as logical rules, can be preserved. However, LLMs do not guarantee the presence of such factual knowledge, leading to hallucination, especially in the case of domain-specific data like biomedical and healthcare information. In the future, we plan to inject relational constraints into LLMs.

- **Learning vector-symbolic representations.** The fusion of knowledge representation (symbolic methods) and deep learning (neural methods) is particularly intriguing, leveraging the strengths of both approaches—the reliability of knowledge representations and the effectiveness of deep learning. We aim to develop vector-symbolic representations that carry symbolic meanings while serving as continuous vectors suitable for input into neural networks. An example of such an approach is the Vector Symbolic Architecture (VSA) [86, 139], which has seen limited exploration in the machine learning community.

- **Hybrid semantic search.** LLMs are shaping the future of data management by transforming unstructured data into numerical vectors stored in a Vector Database. However, these embeddings are structureless and do not carry any explicit symbolic information (e.g., item attributes) that plays essential roles in querying unstructured data. With geometric embedding, we may perform hybrid query that combines similarity search with structured attribute filtering. This can be done by transforming attribute information into geometric vector space.

- **Efficiency improvements.** We plan to investigate strategies for enhancing the efficiency of geometric embeddings by reducing additional computational and memory requirements. This includes exploring optimizations for specific geometric embedding methods, potentially streamlining processes and making them more scalable for real-world applications.

- **Flexible geometric inductive biases.** We may explore to devise methods that allow for the incorporation of freely chosen geometric inductive biases. This involves exploring techniques that dynamically adapt to different inference tasks, especially those requiring the joint consideration of multiple properties, to improve the flexibility and adaptability of geometric relational embeddings.

- **Applications in biomedical sciences.** Geometric embeddings benefit biomedical science in modeling various aspects of biomedical networks: 1) biomedical networks exhibit hierarchical structures, e.g., the structures of ICD-9 codes, and MeSH terms; 2) biomedical knowledge graphs are heterogeneous and many of the relations exhibits complex relational patterns (e.g., symmetry of drug-drug interaction); 3) biomedical ontologies and taxonomies have logical structures and the prediction models must be consistent with these logical structures. My future work would be exploring novel relational embeddings that are suitable for biomedical scenarios.

A

PROOF OF THEOREMS

A.1 PROOF OF THEOREMS OF QGCN

A.1.1 *Proof of Theorem 1*

Theorem 1 (Theorem 4.1 in [97]). *For any point $\mathbf{x} \in \mathcal{Q}_\beta^{s,t}$, there exists a diffeomorphism $\psi : \mathcal{Q}_\beta^{s,t} \to \mathbb{S}_1^t \times \mathbb{R}^s$ that maps $\mathbf{x}$ into the product manifolds of an unit sphere and the Euclidean space, the mapping and its inverse are given by,*

$$\psi(\mathbf{x}) = \begin{pmatrix} \frac{1}{\|t\|}\mathbf{t} \\ \frac{1}{\sqrt{|\beta|}}\mathbf{s} \end{pmatrix}, \quad \psi^{-1}(\mathbf{z}) = \sqrt{|\beta|} \begin{pmatrix} \sqrt{1+\|\mathbf{v}\|^2}\mathbf{u} \\ \mathbf{v} \end{pmatrix}, \tag{A.1}$$

where $\mathbf{x} = \begin{pmatrix} \mathbf{t} \\ \mathbf{s} \end{pmatrix} \in \mathcal{Q}_\beta^{s,t}$ with $\mathbf{t} \in \mathbb{R}_*^{t+1}$ and $\mathbf{s} \in \mathbb{R}^s$. $\mathbf{z} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \in \mathbb{S}_1^t \times \mathbb{R}^s$ with $\mathbf{u} \in \mathbb{S}_1^t$ and $\mathbf{v} \in \mathbb{R}^s$.

Please refer Appendix C.5 in [97] for proof of this theorem.

A.1.2 *Proof of Theorem 2*

Theorem 2. *For any point $\mathbf{x} \in \mathcal{Q}_\beta^{s,t}$, there exists a diffeomorphism $\psi : \mathcal{Q}_\beta^{s,t} \to \mathbb{S}_{-\beta}^t \times \mathbb{R}^s$ that maps $\mathbf{x}$ into the product manifolds of a sphere and the Euclidean space, the mapping and its inverse are given by,*

$$\psi(\mathbf{x}) = \begin{pmatrix} \sqrt{|\beta|}\frac{\mathbf{t}}{\|\mathbf{t}\|} \\ \mathbf{s} \end{pmatrix} \quad , \quad \psi^{-1}(\mathbf{z}) = \begin{pmatrix} \frac{\sqrt{|\beta|+\|\mathbf{v}\|^2}}{\sqrt{|\beta|}}\mathbf{u} \\ \mathbf{v} \end{pmatrix}, \tag{A.2}$$

where $\mathbf{x} = \begin{pmatrix} \mathbf{t} \\ \mathbf{s} \end{pmatrix} \in \mathcal{Q}_\beta^{s,t}$ with $\mathbf{t} \in \mathbb{R}_*^t$ and $\mathbf{s} \in \mathbb{R}^s$. $\mathbf{z} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \in \mathbb{S}_{-\beta}^t \times \mathbb{R}^s$ with $\mathbf{u} \in \mathbb{S}_{-\beta}^t$ and $\mathbf{v} \in \mathbb{R}^s$.

Proof. It is easy to show that the ψ and ψ^{-1} are smooth functions as they only involve with a linear mapping with a constant scaling vector. Hence, we only need to show that $\psi\left(\psi^{-1}(\mathbf{z})\right) = \mathbf{z}$ and $\psi^{-1}(\psi(\mathbf{x})) = \mathbf{x}$. Here, we consider space dimensions and time dimensions separately.

For space dimensions, the mapping of the space dimensions of $\mathbf{x}$ to the space dimensions of $\mathbf{z}$ is an identity function (i.e. $\mathbf{v} = \mathbf{s}, \mathbf{s} = \mathbf{v}$). Thus, we only need to show the invertibility of the mappings taking time dimensions as inputs. For time dimensions, we first show that:

$$\begin{aligned}\psi^{-1}(\psi(\mathbf{t})) &= \frac{\sqrt{|\beta| + \|\mathbf{v}\|^2}}{\sqrt{|\beta|}}\sqrt{|\beta|}\frac{\mathbf{t}}{\|\mathbf{t}\|} = \sqrt{|\beta| + \|\mathbf{v}\|^2}\frac{\mathbf{t}}{\|\mathbf{t}\|} \\ &= \sqrt{|\beta| + \|\mathbf{s}\|^2}\frac{\mathbf{t}}{\|\mathbf{t}\|} = \|\mathbf{t}\|\frac{\mathbf{t}}{\|\mathbf{t}\|} = \mathbf{t}. \end{aligned} \tag{A.3}$$

Note that the last equality can be inferred by the fact that $\mathbf{x} \in \mathcal{Q}_\beta^{s,t}$ and $\beta < 0$. We then show that:

$$\psi(\psi^{-1}(\mathbf{u})) = \sqrt{|\beta|}\frac{\sqrt{|\beta| + \|\mathbf{v}\|^2}}{\sqrt{|\beta|}}\frac{\mathbf{u}}{\|\frac{\sqrt{|\beta|+\|\mathbf{v}\|^2}}{\sqrt{|\beta|}}\mathbf{u}\|} = \sqrt{|\beta|}\frac{\mathbf{u}}{\|\mathbf{u}\|} = \mathbf{u}. \tag{A.4}$$

Note that the last equality can be inferred by the fact that $\mathbf{u} \in \mathbb{S}^t_{-\beta}$ and $\beta < 0$. □

A.1.3 *Proof of Theorem 3*

Theorem 3. *For any reference point* $\mathbf{x} = \begin{pmatrix} \mathbf{t} \\ \mathbf{s} \end{pmatrix} \in \mathcal{Q}_\beta^{s,t}$ *with space dimension* $\mathbf{s} = \mathbf{0}$*, the induced tangent space of* $\mathcal{Q}_\beta^{s,t}$ *is equal to the tangent space of its diffeomorphic manifold* $\mathbb{S}^t_{-\beta} \times \mathbb{R}^s$*, namely,* $\mathcal{T}_{\psi(\mathbf{x})}(\mathbb{S}^t_{-\beta} \times \mathbb{R}^s) = \mathcal{T}_\mathbf{x}\mathcal{Q}_\beta^{s,t}$.

Proof. For any point $\mathbf{x} = \begin{pmatrix} \mathbf{t} \\ \mathbf{s} \end{pmatrix} \in \mathcal{Q}_\beta^{s,t}$ with $\mathbf{s} = \mathbf{0}$, the corresponding point in the diffeomorphic manifold is $\psi(\mathbf{x}) = \begin{pmatrix} \sqrt{|\beta|}\frac{\mathbf{t}}{\|\mathbf{t}\|} \\ \mathbf{0} \end{pmatrix}$. Based on the definition of tangent space, for any tangent vector $\boldsymbol{\xi} \in \mathcal{T}_\mathbf{x}\mathcal{Q}_\beta^{s,t}$, $\langle \mathbf{x}, \boldsymbol{\xi} \rangle_t = 0$ implies $-\mathbf{t}\boldsymbol{\xi}_t + \mathbf{0}\boldsymbol{\xi}_s = 0$, which means $\mathbf{t}\boldsymbol{\xi}_t = 0$. Thus, $\langle \psi(\mathbf{x}), \boldsymbol{\xi} \rangle_t = 0$. Based on the definition of tangent space, $\boldsymbol{\xi} \in \mathcal{T}_{\psi(\mathbf{x})}(\mathbb{S}^t_{-\beta} \times \mathbb{R}^s)$. □

A.1.4 *Proof of Theorem 4*

Theorem 4. *For any point* $\mathbf{x} \in \mathcal{Q}_\beta^{s,t}$*, the union of the normal neighborhood of* $\mathbf{x}$ *and the normal neighborhood of its antipodal point* $-\mathbf{x}$ *cover the entire manifold. Namely,* $\mathcal{U}_\mathbf{x} \cup \mathcal{U}_{-\mathbf{x}} = \mathcal{Q}_\beta^{s,t}$.

Proof. For any point $\mathbf{x} \in \mathcal{Q}_\beta^{s,t}$ and $\mathbf{y} \notin \mathcal{U}_x$. Based on the definition of normal neighborhood, $\langle \mathbf{x}, \mathbf{y} \rangle_t \geq |\beta| \rightarrow \langle -\mathbf{x}, \mathbf{y} \rangle_t \leq |\beta|$. Thus, $\mathbf{y} \in \mathcal{U}_{-x}$, and $\mathcal{U}_\mathbf{x} \cup \mathcal{U}_{-\mathbf{x}} = \mathcal{Q}_\beta^{s,t}$. □

A.2 PROOF OF PATTERN INFERENCE OF ULTRAE

UltraE can naturally infer relation patterns including symmetry, anti-symmetry, inversion and composition. As discussed above, the defined relation transformation $f_r = U_{\theta_r} B_{\mu_r} V_{\Phi_r}$ consists of three operations, including a circular rotation, a hyperbolic rotation, and a circular reflection. The three operation matrices can all be identified as identity matrices. Therefore, there are several different combinations of parameter settings to meet the above inferred requirements, demonstrating the comprehensive capability of the proposed UltraE on encoding relational patterns. For the sake of proof, we assume B_{μ_r} is an identity matrix $\mathbf{I}$, and $\Theta_r, \Phi_r \in [-\pi, \pi)$.

Proposition 1. *Let r be a symmetric relation such that for each triple (e_h, r, e_t), its symmetric triple (e_t, r, e_h) also holds. This symmetric property of r can be encoded into UltraE.*

Proof. If r is a symmetric relation, by taking the $B_{\mathbf{b}_r} = \mathbf{I}$ and $U_{\Theta_r} = \mathbf{I}$, we have

$$\mathbf{e}_h = f_r(\mathbf{e}_t) = V_{\Phi_r}\mathbf{e}_t,\ \mathbf{e}_t = f_r(\mathbf{e}_h) = V_{\Phi_r}\mathbf{e}_h \Rightarrow V_{\Phi_r}^2 = \mathbf{I}$$

which holds true when $\Phi_r = 0$ or $\Phi_r = -\pi$. □

Proposition 2. *Let r be an anti-symmetric relation such that for each triple (e_h, r, e_t), its symmetric triple (e_t, r, e_h) is not true. This anti-symmetric property of r can be encoded into UltraE.*

Proof. If r is an anti-symmetric relation, by taking the $B_{\mathbf{b}_r} = \mathbf{I}$ and $U_{\Theta_r} = \mathbf{I}$, we have

$$\mathbf{e}_h = f_r(\mathbf{e}_t) = V_{\Phi_r}\mathbf{e}_t,\ \mathbf{e}_t = f_r(\mathbf{e}_h) = V_{\Phi_r}\mathbf{e}_h \Rightarrow \mathbf{e}_h = \mathbf{e}_t$$

which holds true when $\Phi_r \neq 0$ and $\Phi_r \neq -\pi$. □

Proposition 3. *Let r_1 and r_2 be inverse relations such that for each triple (e_h, r_1, e_t), its inverse triple (e_t, r_2, e_h) is also true. This inverse property of r_1 and r_2 can be encoded into UltraE.*

Proof. If r_1 and r_2 are inverse relations, by taking the $B_{\mathbf{b}_{r_1}} = B_{\mathbf{b}_{r_2}} = \mathbf{I}$ and $V_{\Phi_{r_1}} = V_{\Phi_{r_2}} = \mathbf{I}$, we have

$$\mathbf{e}_h = f_{r_1}(\mathbf{e}_t) = U_{\Theta_{r_1}}\mathbf{e}_t,\ \mathbf{e}_t = f_{r_2}(\mathbf{e}_h) = U_{\Theta_{r_2}}\mathbf{e}_h \Rightarrow U_{\Theta_{r_1}} U_{\Theta_{r_2}} = \mathbf{I}$$

which holds true when $\Theta_{r_1} + \Theta_{r_2} = 0$. □

Proposition 4. *Let relation r_1 be composed of r_2 and r_3 such that triple (e_h, r_1, e_t) exists when (e_h, r_2, e_t) and (e_h, r_3, e_t) exist. This composition property can be encoded into UltraE.*

Proof. If r_1 is composed of r_2 and r_3, by taking the $B_{\mathbf{b}_{r_1}} = B_{\mathbf{b}_{r_2}} = \mathbf{I}$ and $V_{\Phi_{r_1}} = V_{\Phi_{r_2}} = \mathbf{I}$, we have

$$\begin{aligned} &\mathbf{e}_h = f_{r_1}(\mathbf{e}_t) = U_{\Theta_{r_1}}\mathbf{e}_t,\ \mathbf{e}_h = f_{r_2}(\mathbf{e}_t) = U_{\Theta_{r_2}}\mathbf{e}_t, \\ &\mathbf{e}_h = f_{r_3}(\mathbf{e}_t) = U_{\Theta_{r_3}}\mathbf{e}_t \ \Rightarrow U_{\Theta_{r_1}} = U_{\Theta_{r_2}}U_{\Theta_{r_3}} \end{aligned} \tag{A.5}$$

which holds true when $\Theta_{r_1} = \Theta_{r_2} + \Theta_{r_3}$ or $\Theta_{r_1} = \Theta_{r_2} + \Theta_{r_3} + 2\pi$ or $\Theta_{r_1} = \Theta_{r_2} + \Theta_{r_3} - 2\pi$. □

Connections with Hyperbolic Methods. UltraE has close connections with some existing hyperbolic KG embedding methods, including HyboNet [31], RotH/RefH [26], and MuRP [12]. To show this, we first introduce Lorentz transformation.

Definition 27. *Lorentz transformation is a pseudo-orthogonal transformation with signature* $(p, 1)$.

HyboNet [31] embeds entities as points in a Lorentz space and models relations as Lorentz transformations. According to Definition 27, we have the following proposition.

Proposition 5. *UltraE, if parameterized by a full J-orthogonal matrix, generalizes HyboNet to support arbitrary signatures.*

That is, HyboNet is the case of UltraE (with full J-orthogonal matrix parameterization) where $q = 1$.

By exploiting the polar decomposition [133], a Lorentz transformation matrix $\mathbf{T}$ can be decomposed into $\mathbf{T} = \mathbf{R}_\mathbf{U}\mathbf{R}_\mathbf{b}$, where

$$\mathbf{R}_\mathbf{U} = \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \mathbf{R}_\mathbf{b} = \begin{bmatrix} \left(\mathbf{I} + \mathbf{b}\mathbf{b}^\top\right)^{\frac{1}{2}} & \mathbf{b}^\top \\ \mathbf{b} & \sqrt{1 + \|\mathbf{b}\|_2^2} \end{bmatrix}, \tag{A.6}$$

where $\mathbf{R}_\mathbf{U}$ is an orthogonal matrix. In Lorentz geometry, $\mathbf{R}_\mathbf{U}$ and $\mathbf{R}_\mathbf{b}$ are called Lorentz rotation and Lorentz boost, respectively. $\mathbf{R}_\mathbf{U}$ represents rotation or reflection in space dimension (without changing the time dimension), while $\mathbf{R}_\mathbf{b}$ denotes a hyperbolic rotation across the time dimension and each space dimension. [153] established an equivalence between Lorentz boost and Möbius addition (or hyperbolic translation). Hence, HyboNet inherently models each relation as a combination of a rotation/reflection and a hyperbolic translation.

RotH/RefH [26], interestingly, also models each relation as a combination of a rotation/reflection and a hyperbolic translation that is implemented by Möbius addition. Hence, HyboNet subsumes RotH/RefH,[1] where the

1 Note that RotH/RefH consider Poincaré Ball while HyboNet considers Lorentz model. The subsumption still holds since Poincaré Ball is isometric to the Lorentz model.

equivalence cannot hold because the rotation/reflection of RotH/RefH is parameterized by the Givens rotation/reflection [26].

MuRP [12] models relations as a combination of Möbius multiplication (with diagonal matrix) and Möbius addition. Note that [31] established a fact that a Lorentz rotation is equivalent to Möbius multiplication, and [153] proved that Lorentz boost is equivalent to Möbius addition. Hence, HyboNet subsumes MuRP, where the equivalence cannot hold because the Möbius multiplication in MuRP is parameterized by a diagonal matrix.

To sum up, UltraE generalizes HyboNet to allow for arbitrary signature (p, q), while HyboNet subsumes RotH/RefH and MuRP.

A.3 PROOFS OF THEOREMS OF BOXEL

Proposition 6. *We have*

1. *If* $\mathcal{L}_{C(a)}(w) = 0$, *then* $\mathcal{I}_w \models C(a)$,
2. *If* $\mathcal{L}_{r(a,b)}(w) = 0$, *then* $\mathcal{I}_w \models r(a, b)$.

Proposition 6 follows directly from the definitions.

Lemma 1. 1. $0 \leq \text{Disjoint}(B_1, B_2) \leq 1$,

2. $\text{Disjoint}(B_1, B_2) = 0$ *implies* $B_1 \subseteq B_2$,
3. $\text{Disjoint}(B_1, B_2) = 1$ *implies* $B_1 \cap B_2 = \emptyset$.

Proof. 1. Since $B_1 \cap B_2 \subseteq B_1$, $\frac{\text{MVol}(B_1 \cap B_2)}{\text{MVol}(B_1)} \leq 1$. The modified volume is also non-negative, so that the fraction is non-negative and $0 \leq \text{Disjoint}(B_1, B_2) \leq 1$.

2. If $\text{Disjoint}(B_1, B_2) = 0$ then we must have $\text{MVol}(B_1 \cap B_2) = 1$ and therefore $\text{MVol}(B_1 \cap B_2) = \text{MVol}(B_1) \neq 0$. Since $B_1 \cap B_2 \subseteq B_1$, this is only possible if $B_1 \cap B_2 = B_1$, but this implies that $B_1 \subseteq B_2$.

3. If $\text{Disjoint}(B_1, B_2) = 1$, we must have $\text{MVol}(B_1 \cap B_2) = 0$. By definition of the modified volume this is only possible if $B_1 \cap B_2 = \emptyset$. □

Proposition 7. *If* $\mathcal{L}_{C \sqsubseteq D}(w) = 0$, *then* $\mathcal{I}_w \models C \sqsubseteq D$, *where we exclude the inconsistent case* $C = \{a\}, D = \bot$.

Proof. For $D \neq \bot$, the claim follows from Lemma 1. For $D = \bot$, $\mathcal{L}_{C \sqsubseteq \bot}(w) = 0$ implies that $\text{Box}_w(C) = \emptyset$ and the claim is trivially true. □

Proposition 8. *If $\mathcal{L}_{C\sqcap D\sqsubseteq E}(w) = 0$, then $\mathcal{I}_w \models C \sqcap D \sqsubseteq E$, where we exclude the inconsistent case $a \sqcap a \sqsubseteq \bot$ (that is, $C = D = \{a\}, E = \bot$).*

Proof. We have to show that for every $x \in \text{Box}_w(C)$, there is a $y \in \text{Box}_w(D)$ such that $T^r_w(x) = y$. Note that $\text{Disjoint}(T^r_w(\text{Box}_w(C)), \text{Box}_w(D)) = 0$ implies that $T^r_w(\text{Box}_w(C)) \subseteq \text{Box}_w(D)$ according to Lemma 1. Since $x \in \text{Box}_w(C)$, we have $T^r_w(x) = y \in \text{Box}_w(D)$. □

Proposition 9. *If $\mathcal{L}_{C\sqsubseteq\exists r.D}(w) = 0$, then $\mathcal{I}_w \models C \sqsubseteq \exists r.D$.*

The proof of Proposition 4 is analogous to the proof of Proposition 3.

Proposition 10. *If $\mathcal{L}_{\exists r.C\sqsubseteq D}(w) = 0$, then $\mathcal{I}_w \models \exists r.C \sqsubseteq D$.*

Proof. We have to show that if $T^{-r}_w(x) = y$ and $y \in \text{Box}_w(C)$, then $x \in \text{Box}_w(D)$. $\text{Disjoint}(T^{-r}_w(\text{Box}_w(C)), \text{Box}_w(D)) = 0$ implies that $T^{-r}_w(\text{Box}_w(C)) \subseteq \text{Box}_w(D)$ according to Lemma 1. Since $y \in \text{Box}_w(C)$, we have $x = T^{-r}_w(y) \in \text{Box}_w(D)$. □

A.4 PROOF OF THEOREMS OF HMI

Proposition 11. *Given a Poincaré hyperplane $H_{\mathbf{c}}$ where $\mathbf{c} \neq \mathbf{0}$, there exists an n-ball $\mathbb{B}_{\mathbf{c}}(\mathbf{o}_{\mathbf{c}}, r_{\mathbf{c}})$ such that $H_{\mathbf{c}} \subset \mathbb{B}_{\mathbf{c}}(\mathbf{o}_{\mathbf{c}}, r_{\mathbf{c}})$, i.e., $H_{\mathbf{c}}$ is a subset of $\mathbb{B}_{\mathbf{c}}(\mathbf{o}_{\mathbf{c}}, r_{\mathbf{c}})$. $\mathbb{B}_{\mathbf{c}}$ is uniquely given by*

$$\mathbb{B}^n_{\mathbf{c}} = \mathbb{B}^n\left(\frac{(1+\|\mathbf{c}\|^2)}{2\|\mathbf{c}\|}\mathbf{c}, \frac{1-\|\mathbf{c}\|^2}{2\|\mathbf{c}\|}\right) \tag{A.7}$$

Proof. Since c is the center point of the Poincaré hyperplane, the vector $\overrightarrow{c}$ must be a normal vector of the tangent space $T_c\mathbb{B}^n$ of $\mathbb{B}^n$ at c. Let q be one of the point that the Poincaré hyperplane and the Poincaré ball intersect at. Then, the radius of $\mathbb{B}_{\mathbf{c}}(\mathbf{o}_{\mathbf{c}}, r_{\mathbf{c}})$, the radius of $\mathbb{D}^n$, and the distance from the centers of $\mathbb{D}^n$ to the center of $\mathbb{B}_{\mathbf{c}}(\mathbf{o}_{\mathbf{c}}, r_{\mathbf{c}})$ must satisfy the Pythagorean theorem, i.e., the three Euclidean distances $d(\mathbf{0}, q)$, $d(q, \mathbf{o}_{\mathbf{c}})$ and $d(\mathbf{o}_{\mathbf{c}}, \mathbf{0})$ must satisfy

$$d(\mathbf{0}, \mathbf{q})^2 + d(\mathbf{q}, \mathbf{o}_{\mathbf{c}})^2 = d(\mathbf{o}_{\mathbf{c}}, \mathbf{0})^2 = \left(d(\mathbf{0}, \mathbf{c}) + d(\mathbf{c}, \mathbf{o}_{\mathbf{c}})\right)^2. \tag{A.8}$$

Since we have $d(\mathbf{c}, \mathbf{o}_{\mathbf{c}}) = d(\mathbf{q}, \mathbf{o}_{\mathbf{c}}) = r_{\mathbf{c}}$, by solving this quadratic equation, we have $r_{\mathbf{c}} = \frac{1-\|\mathbf{c}\|^2}{2\|\mathbf{c}\|}$. Since $\mathbf{o}_{\mathbf{c}} = \mathbf{c}(1 + \frac{r_{\mathbf{c}}}{d(\mathbf{0},\mathbf{c})})$, we have $\mathbf{o}_{\mathbf{c}} = \mathbf{c}\frac{(1+\|\mathbf{c}\|^2)}{2\|\mathbf{c}\|}$. Thus, $\mathbb{B}_{\mathbf{c}} = \mathbb{B}\left(\mathbf{o}_{\mathbf{c}} = \mathbf{c}\frac{(1+\|\mathbf{c}\|^2)}{2\|\mathbf{c}\|}, r_{\mathbf{c}} = \frac{1-\|\mathbf{c}\|^2}{2\|\mathbf{c}\|}\right)$. □

Proposition 12 (HEX-property)**.** *The classification function f has the HEX property with respect to G if and only if for any constraint in G, the corresponding loss term is 0.*

Proof. Note that the loss term of the constraint being 0 implies that the corresponding constraint is respected. Our loss terms clearly connect the HEX property. That is, for any point $\mathbf{p} \in D^n$ and a pair of enclosing n-balls $(\mathbb{B}_w, \mathbb{B}_u)$, $\mathcal{L}_{\text{membership}}(p, \mathbb{B}_w) \geq \mathcal{L}_{\text{membership}}(p, \mathbb{B}_u)$ for all $(\mathbb{B}_w, \mathbb{B}_u)$ where $\mathcal{L}_{\text{inside}}(\mathbb{B}_w, \mathbb{B}_u) = 0$ and $\neg\mathcal{L}_{\text{membership}}(p, \mathbb{B}_w) \vee \neg\mathcal{L}_{\text{membership}}(p, \mathbb{B}_u)$ for all $(\mathbb{B}_w, \mathbb{B}_u)$ where $\mathcal{L}_{\text{disjoint}}(\mathbb{B}_u, \mathbb{B}_w) = 0$. According to the definition of HEX-property, f has the HEX property with respect to G if and only if the corresponding loss term of the corresponding constraint is 0. □

Corollary 1. *Given a HEX graph G of labels and if the loss of the embeddings is 0, then the learned prediction function is logically consistent with respect to G.*

Proof. Note that the loss terms $\mathcal{L}_{\text{inside}}, \mathcal{L}_{\text{disjoint}}, \mathcal{L}_{\text{membership}}, \mathcal{L}_{\text{non-membership}}$ in Eq.7 are all non-negative. Hence, the loss being 0 implies that all losses are zeros (all constraints are satisfied). According to the definition of consistency, the prediction function is consistent. □

A.5 PROOF OF PROPOSITIONS OF SHRINKE

Proposition 13. *Given any two facts $\mathcal{F}_1 = (\mathcal{T}, \mathcal{Q}_1)$ and $\mathcal{F}_2 = (\mathcal{T}, \mathcal{Q}_2)$ where $\mathcal{Q}_2 \subseteq \mathcal{Q}_1$, i.e., $\mathcal{F}_2$ is a partial fact of $\mathcal{F}_1$, the output of the scoring function $f(\cdot)$ of ShrinkE satisfy the constraint $f(\mathcal{F}_2) \geq f(\mathcal{F}_1)$, which implies Eq.(2.8).*

Proof. We first prove that the resulting box of $\mathcal{F}_2$ subsumes the resulting box of $\mathcal{F}_2$. Since the primal triple of $\mathcal{F}_1$ and $\mathcal{F}_2$ are the same (let assume it is $\mathcal{T} = (h, r, t)$), the spanned boxes of the two facts are $\mathcal{H}_r(\mathbf{e}_h)$. Since $\mathcal{Q}_2 \subseteq \mathcal{Q}_1$, the final shrunken box of $\mathcal{F}_1$ must be a subset of the shrunken box of $\mathcal{F}_2$. Hence, we have,

$$\text{Box}_{\mathcal{F}_2} \subseteq \text{Box}_{\mathcal{F}_1}. \tag{A.9}$$

Given the tail entity t whose embedding is denoted by $\mathbf{e}_t$, we consider three cases of its position.

1) If $\mathbf{e}_t$ is inside the small box $\text{Box}_{\mathcal{F}_2}$, then $\mathbf{e}_t$ must also be inside $\text{Box}_{\mathcal{F}_1}$ since $\text{Box}_{\mathcal{F}_2} \subseteq \text{Box}_{\mathcal{F}_1}$. Note that our point-to-box function is monotonically increasing w.r.t. the increase of distance from the tail point to the center of box. Hence, we will have $D(\mathbf{e}, \text{Box}_{\mathcal{F}_2}) \geq D(\mathbf{e}, \text{Box}_{\mathcal{F}_1})$, implying $f(\mathcal{F}_2) \geq f(\mathcal{F}_1)$.

2) If $\mathbf{e}_t$ is outside the small box $\text{Box}_{\mathcal{F}_2}$ but inside in the larger $\text{Box}_{\mathcal{F}_1}$, according to the definition of the point-to-box distance function, we immediately have $D(\mathbf{e}, \text{Box}_{\mathcal{F}_2}) \geq D(\mathbf{e}, \text{Box}_{\mathcal{F}_1})$, implying $f(\mathcal{F}_2) \geq f(\mathcal{F}_1)$.

3) If $\mathbf{e}_t$ is outside the larger box $\text{Box}_{\mathcal{F}_1}$,, then $\mathbf{e}_t$ must also be outside $\text{Box}_{\mathcal{F}_2}$ since $\text{Box}_{\mathcal{F}_2} \subseteq \text{Box}_{\mathcal{F}_1}$. Note that our point-to-box function is monotonically decreasing w.r.t. the increase of volume of box. Hence, we will have $D(\mathbf{e}, \text{Box}_{\mathcal{F}_2}) \geq D(\mathbf{e}, \text{Box}_{\mathcal{F}_1})$, implying $f(\mathcal{F}_2) \geq f(\mathcal{F}_1)$. □

Proposition 14. *ShrinkE is able to infer hyper-relational symmetry, anti-symmetry, inversion, composition, hierarchy, intersection, and exclusion.*

We first prove that ShrinkE is able to infer symmetry, anti-symmetry, inversion, and composition. For the sake of proof, we assume $\theta_r \in [-\pi, \pi)$. We prove them by proving Lemma B.1-4 one by one.

Lemma 2 (Symmetry)**.** *Let r be a symmetric relation such that for each triple (e_h, r, e_t), its symmetric triple (e_t, r, e_h) also holds. This symmetric property of r can be modeled by ShrinkE.*

Proof. If r is a symmetric relation, by taking the $\mathbf{ffi}_r = \mathbf{0}$, $\mathbf{b}_r = \mathbf{0}$, and $\mathbf{\Theta}_r = \text{diag}\left(\mathbf{G}\left(\grave{}_{r,1}\right), \ldots, \mathbf{G}\left(\grave{}_{r,\frac{d}{2}}\right)\right)$, where $\mathbf{G}(\theta)$ is a 2×2 diagonal matrix, we have

$$\begin{aligned} &\mathbf{e}_h = f_r\left(\mathbf{e}_t\right) = \mathbf{\Theta}_r \mathbf{e}_t,\ \mathbf{e}_t = f_r\left(\mathbf{e}_h\right) = \mathbf{\Theta}_r \mathbf{e}_h \\ &\Rightarrow \mathbf{\Theta}_r^2 = \mathbf{I} \end{aligned}$$

which holds true when $\grave{}_{r,i} = \mathbf{0}$ or $\grave{}_{r,i} = -\text{ß}$ for $i = 1, \cdots, \frac{d}{2}$. □

Lemma 3 (Anti-symmetry)**.** *Let r be an anti-symmetric relation such that for each triple (e_h, r, e_t), its symmetric triple (e_t, r, e_h) is not true. This anti-symmetric property of r can be modeled by ShrinkE.*

Proof. If r is a anti-symmetric relation, by taking the $\mathbf{ffi}_r = \mathbf{0}$, $\mathbf{b}_r = \mathbf{0}$, and $\mathbf{\Theta}_r = \text{diag}\left(\mathbf{G}\left(\grave{}_{r,1}\right), \ldots, \mathbf{G}\left(\grave{}_{r,\frac{d}{2}}\right)\right)$, where $\mathbf{G}(\theta)$ is a 2×2 diagonal matrix, we have

$$\begin{aligned} &\mathbf{e}_h \neq f_r\left(\mathbf{e}_t\right) = \mathbf{\Theta}_r \mathbf{e}_t,\ \mathbf{e}_t = f_r\left(\mathbf{e}_h\right) = \mathbf{\Theta}_r \mathbf{e}_h \\ &\Rightarrow \mathbf{\Theta}_r^2 \neq \mathbf{I} \end{aligned}$$

which holds true when $\grave{}_{r,i} \neq \mathbf{0}$ or $\grave{}_{r,i} \neq -\text{ß}$ for $i = 1, \cdots, \frac{d}{2}$. □

Lemma 4 (Inversion)**.** *Let r_1 and r_2 be inverse relations such that for each triple (e_h, r_1, e_t), its inverse triple (e_t, r_2, e_h) is also true. This inverse property of r_1 and r_2 can be modeled by ShrinkE.*

Proof. If r_1 and r_2 are inverse relations, by taking the $\mathbf{ffi}_r = \mathbf{0}$, $\mathbf{b}_r = \mathbf{0}$, and $\mathbf{\Theta}_r = \text{diag}\left(\mathbf{G}\left(\grave{}_{r,1}\right), \ldots, \mathbf{G}\left(\grave{}_{r,\frac{d}{2}}\right)\right)$, where $\mathbf{G}(\theta)$ is a 2×2 diagonal matrix, we have

$$\begin{aligned} &\mathbf{e}_t = f_{r_1}(\mathbf{e}_h) = \Theta_{r_1}\mathbf{e}_h,\ \mathbf{e}_h = f_{r_2}(\mathbf{e}_t) = \Theta_{r_2}\mathbf{e}_h \\ &\Rightarrow \Theta_{r_1}\Theta_{r_2} = \mathbf{I} \end{aligned}$$

which holds true when for $\theta_{r_1,i} r_1 + \theta_{r_2,i} = 0$ for $i = 1, \cdots, \frac{d}{2}$. □

Lemma 5 (Composition). *Let relation r_1 be composed of r_2 and r_3 such that triple (e_1, r_1, e_3) exists when (e_1, r_2, e_2) and (e_2, r_3, e_3) exist. This composition property can be modeled by ShrinkE.*

Proof. If r_1 is composed of r_2 and r_3, by taking the $\mathbf{ffi}_r = \mathbf{0}$, $\mathbf{b}_r = \mathbf{0}$, and $\mathbf{\Theta}_r = \text{diag}\left(\mathbf{G}\left(\grave{}_{r,1}\right), \ldots, \mathbf{G}\left(\grave{}_{r,\frac{d}{2}}\right)\right)$, where $\mathbf{G}(\theta)$ is a 2×2 diagonal matrix, we have

$$\begin{aligned} &\mathbf{e}_3 = f_{r_1}(\mathbf{e}_1) = \Theta_{r_1}\mathbf{e}_1,\ \mathbf{e}_2 = f_{r_2}(\mathbf{e}_1) = \Theta_{r_2}\mathbf{e}_1, \\ &\mathbf{e}_3 = f_{r_3}(\mathbf{e}_2) = \Theta_{r_3}\mathbf{e}_2 \ \Rightarrow \Theta_{r_1} = \Theta_{r_2}\Theta_{r_3} \end{aligned}$$

which holds true when $\theta_{r_1,i} = \theta_{r_2,i} + \theta_{r_3,i}$ or $\theta_{r_1,i} = \theta_{r_2,i} + \theta_{r_3,i} + 2\pi$ or $\theta_{r_1,i} = \theta_{r_2,i} + \theta_{r_3,i} - 2\pi$ for $i = 1, \cdots, \frac{d}{2}$. □

We now prove that ShrinkE is able to infer relation implication, exclusion and intersection.

Lemma 6 (Relation implication). *Let $r_1 \to r_2$ form a hierarchy such that for each triple (e_h, r_1, e_t), (e_h, r_2, e_t) also holds. This hierarchy property $r_1 \to r_2$ can be modeled by ShrinkE.*

Proof. If $r_1 \to r_2$, by taking $\mathcal{T}_{r_1} = \mathcal{T}_{r_2}$, i.e., $\boldsymbol{\delta}_{r_1} = \boldsymbol{\delta}_{r_2}$ and $\Theta_{r_1} = \Theta_{r_2}$, we have, $(e_h, r_1, e_t) \to (e_h, r_2, e_t)$ implies that the spanning box of query $(e_h, r_1, x?)$ is subsumed by the spanning box of query $(e_h, r_2, x?)$. i.e., $\text{Box}(\mathcal{H}_{r_1}(e_h) - \sigma(\boldsymbol{\delta}_{r_1}), \mathcal{H}_{r_1}(e_h) + \sigma(\boldsymbol{\delta}_{r_1})) \subseteq \text{Box}(\mathcal{H}_{r_1}(e_h) - \sigma(\boldsymbol{\delta}_{r_2}), \mathcal{H}_{r_1}(e_h) + \sigma(\boldsymbol{\delta}_{r_2}))$, which holds true when $\boldsymbol{\delta}_{r_1} \leq \boldsymbol{\delta}_{r_2}$. □

Lemma 7 (Relation exclusion). *Let r_1, r_2 be mutually exclusive, that is, (e_h, r_1, e_t), (e_h, r_2, e_t) can not be simultaneously hold. This mutual exclusion property $r_1 \wedge r_2 \to \perp$ can be modeled by ShrinkE.*

Proof. If $r_1 \wedge r_2 \to \perp$, we have $(e_h, r_1, e_t) \wedge (e_h, r_2, e_t) \to \perp$, which implies that the spanning box of query $(e_h, r_1, x?)$ and the spanning box of query $(e_h, r_2, x?)$ are mutually exclusive, i.e., $\text{Box}(\mathcal{H}_{r_1}(e_h) - \sigma(\boldsymbol{\delta}_{r_1}), \mathcal{H}_{r_1}(e_h) + \sigma(\boldsymbol{\delta}_{r_1})) \cap \text{Box}(\mathcal{H}_{r_1}(e_h) - \sigma(\boldsymbol{\delta}_{r_2}), \mathcal{H}_{r_1}(e_h) + \sigma(\boldsymbol{\delta}_{r_2})) \to \perp$ □

Lemma 8 (Relation intersection)**.** *Let r_3 be a intersection of r_1, r_2, that is, if (e_h, r_1, e_t) and (e_h, r_2, e_t) hold, then (e_h, r_3, e_t) also holds. This intersection property $r_1 \wedge r_2 \rightarrow r_3$ can be modeled by ShrinkE.*

Proof. Note that box is closed under intersection and this property can be view as a combination of two pairs of relation implication. Hence, the proof is similar to the proof of Lemma A.5. □

Proposition 15. *ShrinkE is able to infer qualifier implication, mutual exclusion, and intersection.*

Proof. Since each qualifier is associated with a box, the implication and mutual exclusion relationships between qualifiers can be modeled by their geometric relationships, i.e., box entailment and box disjointedness, respectively, between their corresponding boxes. Qualifier intersection can be modeled by enforcing the box of one qualifier to be inside the intersection of the boxes of another two qualifiers. □

A.6 THEORETICAL JUSTIFICATIONS OF NESTE

We extend the vanilla logical patterns in KGs to include nested facts. This can be expressed in a non-first-order-logic-like form $\psi \xrightarrow{\widehat{r}} \phi$.

- **Relational symmetry (R-symmetry):** an atomic relation r is symmetric w.r.t a nested relation $\widehat{r}$ if $\forall x, y \in \mathcal{E}, \langle x, r, y\rangle \xleftrightarrow{\widehat{r}} \langle y, r, x\rangle$.
- **Relational inverse (R-inverse):** two atomic relations r_1 and r_2 are inverse w.r.t a nested relation $\widehat{r}$ if $\forall x, y \in \mathcal{E}, (\langle x, r_1, y\rangle \xleftrightarrow{\widehat{r}} \langle y, r_2, x\rangle)$.
- **Relational implication (R-implication):** an atomic relation r_1 implies a atomic relation r_2 w.r.t a nested relation $\widehat{r}$ if $\forall x, y \in \mathcal{E}, (\langle x, r_1, y\rangle \xrightarrow{\widehat{r}} \langle x, r_2, y\rangle)$.
- **Relational inverse implication (R-Inv-implication):** an atomic relation r_1 inversely implies an atomic relation r_2 w.r.t a nested relation $\widehat{r}$ if $\forall x, y \in \mathcal{E}, (\langle x, r_1, y\rangle \xrightarrow{\widehat{r}} \langle y, r_2, x\rangle)$.
- **Entity implication (E-implication):** an entity x_1 (resp. y_1) implies entity x_2 (resp. y_2) w.r.t an atomic relation r and a nested relation $\widehat{r}$ if $\forall y \in \mathcal{E}, (\langle x_1, r, y\rangle \xrightarrow{\widehat{r}} \langle x_2, r, y\rangle)$ (resp. $\forall x \in \mathcal{E}, (\langle x, r, y_1\rangle \xrightarrow{\widehat{r}} \langle x, r, y_2\rangle)$).
- **Entity relational implication (E-R-implication):** an entity x_1 and relation r_1 (resp. y_1 and relation r_1) implies entity x_2 and relation r_2 (resp. y_2 and relation r_2) w.r.t a nested relation $\widehat{r}$ if $\forall y \in \mathcal{E}, (\langle x_1, r_1, y\rangle \xrightarrow{\widehat{r}} \langle x_2, r_2, y\rangle)$ (resp. $\forall x \in \mathcal{E}, (\langle x, r_1, y_1\rangle \xrightarrow{\widehat{r}} \langle x, r_2, y_2\rangle)$).

- **Entity relational inverse implication (E-R-Inv-implication):** an entity x_1 and relation r_1 (resp. y_1 and relation r_1) inversely implies entity x_2 and relation r_2 (resp. y_2 and relation r_2) w.r.t a nested relation $\widehat{r}$ if $\forall y \in \mathcal{E}, (\langle x_1, r_1, y\rangle \xrightarrow{\widehat{r}} \langle y, r_2, x_2\rangle)$ (resp. $\forall x \in \mathcal{E}, (\langle x, r_1, y_1\rangle \xrightarrow{\widehat{r}} \langle y_2, r_2, x\rangle)$).

- **Dual Entity implication (Dual E-implication):** an entity pair (x_1, x_2) implies another entity pair (y_1, y_2) iff both (x_1, y_1) and (x_2, y_2) satisfy E-implication.

Proposition 16. *NestE can infer R-symmetry, R-inverse, R-implication, R-Inv-implication, E-implication, E-R-implication, E-R-Inv-implication, and Dual E-implication.*

To infer different logical patterns via different free variables, we can set some elements of the relation matrix to be zero-valued or one-valued complex numbers. We prove proposition 16 by showing the following special solutions for each case of pattern.

Proposition 17. *Relational symmetry can be inferred by setting*

$$\mathbf{R} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{R}_{22} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{A.10}$$

where $\mathbf{r} \otimes \mathbf{R}_{22} = \mathbf{r}$ *implying* $\mathbf{R}_{22} = \mathbf{1}$.

Proposition 18. *Relational inversion can be inferred by setting*

$$\mathbf{R} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{R}_{22} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{A.11}$$

where $\mathbf{r}_1 \otimes \mathbf{R}_{22} = \mathbf{r}_2$ *and* $\mathbf{r}_2 \otimes \mathbf{R}_{22} = \mathbf{r}_1$ *implying* $\mathbf{R}_{22} = \mp 1$.

Proposition 19. *Relational implication can be inferred by setting*

$$\mathbf{R} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{A.12}$$

where $\mathbf{r}_1 \otimes \mathbf{R}_{22} = \mathbf{r}_2$.

Proposition 20. *Relational inverse implication can be inferred by setting*

$$\mathbf{R} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{R}_{22} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{A.13}$$

where $\mathbf{r}_1 \otimes \mathbf{R}_{22} = \mathbf{r}_2$.

Proposition 21. *Entity implication can be inferred by setting*

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \mathbf{0} \\ \mathbf{R}_{21} & \mathbf{R}_{21} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{A.14}$$

where

$$\mathbf{x}_1 \otimes \mathbf{R}_{11} + \mathbf{r} \otimes \mathbf{R}_{21} = \mathbf{x}_2 \tag{A.15}$$
$$\mathbf{x}_1 \otimes \mathbf{R}_{12} + \mathbf{r} \otimes \mathbf{R}_{22} = \mathbf{r} \tag{A.16}$$

or

$$\mathbf{R} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{22} & \mathbf{R}_{23} \\ \mathbf{0} & \mathbf{R}_{32} & \mathbf{R}_{33} \end{bmatrix} \tag{A.17}$$

where

$$\mathbf{r} \otimes \mathbf{R}_{32} + \mathbf{y}_1 \otimes \mathbf{R}_{33} = \mathbf{y}_2 \tag{A.18}$$
$$\mathbf{r} \otimes \mathbf{R}_{22} + \mathbf{y}_1 \otimes \mathbf{R}_{32} = \mathbf{r} \tag{A.19}$$

Proposition 22. *Entity inverse implication can be inferred by setting*

$$\mathbf{R} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{R}_{13} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{R}_{31} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{A.20}$$

where $\mathbf{x}_1 \otimes \mathbf{R}_{31} = \mathbf{x}_2$. *Or,*

$$\mathbf{R} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{R}_{13} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{R}_{31} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{A.21}$$

where $\mathbf{y}_1 \otimes \mathbf{R}_{33} = \mathbf{y}_2$.

LIST OF FIGURES

BIBLIOGRAPHY

[1] Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. "BoxE: a box embedding model for knowledge base completion." In: *NeurIPS* 33 (2020), pp. 9649–9661.

[2] Dimitrios Alivanistos, Max Berrendorf, Michael Cochez, and Mikhail Galkin. "Query Embedding on Hyper-Relational Knowledge Graphs." In: *ICLR*. OpenReview.net, 2022.

[3] Ben Andrews and Christopher Hopper. *The Ricci flow in Riemannian geometry: a complete proof of the differentiable 1/4-pinching sphere theorem*. springer, 2010.

[4] Dörthe Arndt, Jeen Broekstra, Bob DuCharme, Ora Lassila, Peter F. Patel-Schneider, Eric Prud'hommeaux, Jr. Ted Thibodeau, and Bryan Thompson. "RDF-star and SPARQL-star." In: *Final Community Group Report*. Ed. by Olaf Hartig, Pierre-Antoine Champin, Gregg Kellogg, and Andy Seaborne. 2021. URL: https://www.w3.org/2021/12/rdf-star.html.

[5] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. "Gene ontology: tool for the unification of biology." In: *Nature genetics* 25.1 (2000), pp. 25–29.

[6] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. "DBpedia: A Nucleus for a Web of Open Data." In: *ISWC/ASWC*. Vol. 4825. Lecture Notes in Computer Science. Springer, 2007, pp. 722–735.

[7] Franz Baader, Sebastian Brandt, and Carsten Lutz. "Pushing the EL envelope." In: *IJCAI*. Vol. 5. 2005, pp. 364–369.

[8] Franz Baader, Diego Calvanese, Deborah McGuinness, Peter Patel-Schneider, Daniele Nardi, et al. *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003.

[9] Gregor Bachmann, Gary Bécigneul, and Octavian Ganea. "Constant Curvature Graph Convolutional Networks." In: *ICML*. Vol. 119. PMLR, 2020, pp. 486–496.

[10] Yushi Bai, Zhitao Ying, Hongyu Ren, and Jure Leskovec. "Modeling Heterogeneous Hierarchies with Relation-specific Hyperbolic Cones." In: *NeurIPS* 34 (2021).

[11] Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. "Multi-relational Poincaré Graph Embeddings." In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett. 2019, pp. 4465–4475. URL: https://proceedings.neurips.cc/paper/2019/hash/f8b932c70d0b2e6bf071729a4fa68dfc-Abstract.html.

[12] Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. "Multi-relational Poincaré Graph Embeddings." In: *NeurIPS*. 2019, pp. 4465–4475.

[13] Gary Bécigneul and Octavian-Eugen Ganea. "Riemannian Adaptive Optimization Methods." In: *ICLR (Poster)*. OpenReview.net, 2019.

[14] Wei Bi and James T. Kwok. "MultiLabel Classification on Tree- and DAG-Structured Hierarchies." In: *ICML*. Omnipress, 2011, pp. 17–24.

[15] Marián Boguñá, Ivan Bonamassa, Manlio De Domenico, Shlomo Havlin, Dmitri Krioukov, and M Ángeles Serrano. "Network geometry." In: *Nature Reviews Physics* (2021), pp. 1–22.

[16] Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. "Freebase: a collaboratively created graph database for structuring human knowledge." In: *SIGMOD Conference*. ACM, 2008, pp. 1247–1250.

[17] Kurt Bollacker, Robert Cook, and Patrick Tufts. "Freebase: A shared database of structured general human knowledge." In: *AAAI*. Vol. 7. 2007, pp. 1962–1963.

[18] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. "Irreflexive and Hierarchical Relations as Translations." In: *CoRR* abs/1304.7158 (2013).

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

[20] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. "Spectral Networks and Locally Connected Networks on Graphs." In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. URL: http://arxiv.org/abs/1312.6203.

[21] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. "Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences." In: *WWW*. ACM, 2019, pp. 151–161.

[22] Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. "Dual Quaternion Knowledge Graph Embeddings." In: *AAAI*. AAAI Press, 2021, pp. 6894–6902.

[23] Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. "Geometry Interaction Knowledge Graph Embeddings." In: *AAAI*. 2022.

[24] Remzi Celebi, Hüseyin Uyar, Erkan Yasar, Özgür Gümüs, Oguz Dikenelli, and Michel Dumontier. "Evaluation of knowledge graph embedding approaches for drug-drug interaction prediction in realistic settings." In: *BMC Bioinform.* 20.1 (2019), p. 726.

[25] Ricardo Cerri, Rodrigo C. Barros, and André Carlos Ponce de Leon Ferreira de Carvalho. "Hierarchical multi-label classification using local neural networks." In: *J. Comput. Syst. Sci.* 80.1 (2014), pp. 39–56.

[26] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. "Low-Dimensional Hyperbolic Knowledge Graph Embeddings." In: *ACL*. 2020, pp. 6901–6914.

[27] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. "Hyperbolic Graph Convolutional Neural Networks." In: *NeurIPS*. 2019, pp. 4869–4880.

[28] Pierre-Antoine Champin. "RDF-star: Paving the Way to the Next generation of Linked Data." In: *ERCIM News* 2022.128 (2022).

[29] Boli Chen, Xin Huang, Lin Xiao, Zixin Cai, and Liping Jing. "Hyperbolic Interaction Model for Hierarchical Multi-Label Classification." In: *AAAI*. AAAI Press, 2020, pp. 7496–7503.

[30] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. "Simple and Deep Graph Convolutional Networks." In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1725–1735. URL: http://proceedings.mlr.press/v119/chen20v.html.

[31] Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. "Fully Hyperbolic Neural Networks." In: *CoRR* abs/2105.14686 (2021).

[32] Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. "Fully hyperbolic neural networks." In: *ACL* (2022).

[33] Xuelu Chen, Michael Boratko, Muhao Chen, Shib Sankar Dasgupta, Xiang Lorraine Li, and Andrew McCallum. "Probabilistic Box Embeddings for Uncertain Knowledge Graph Reasoning." In: *NAACL-HLT*. ACL, 2021, pp. 882–893.

[34] Yankai Chen, Menglin Yang, Yingxue Zhang, Mengchen Zhao, Ziqiao Meng, Jianye Hao, and Irwin King. "Modeling Scale-free Graphs with Hyperbolic Geometry for Knowledge-aware Recommendation." In: *WSDM*. ACM, 2022, pp. 94–102.

[35] Tejas Chheda, Purujit Goyal, Trang Tran, Dhruvesh Patel, Michael Boratko, Shib Sankar Dasgupta, and Andrew McCallum. "Box Embeddings: An open-source library for representation learning using geometric structures." In: *EMNLP (Demos)*. 2021.

[36] Ara Cho, Junha Shin, Sohyun Hwang, Chanyoung Kim, Hongseok Shim, Hyojin Kim, Hanhae Kim, and Insuk Lee. "WormNet v3: a network-assisted hypothesis-generating server for Caenorhabditis elegans." In: *Nucleic acids research* 42.W1 (2014), W76–W82.

[37] Hyunghoon Cho, Benjamin DeMeo, Jian Peng, and Bonnie Berger. "Large-margin classification in hyperbolic space." In: *The 22nd international conference on artificial intelligence and statistics*. PMLR. 2019, pp. 1832–1840.

[38] Chanyoung Chung and Joyce Jiyoung Whang. "Learning Representations of Bi-Level Knowledge Graphs for Reasoning beyond Link Prediction." In: *AAAI*. 2023.

[39] Amanda Clare. "Machine learning and data mining for yeast functional genomics." PhD thesis. University of Wales, Aberystwyth, 2003.

[40] James R Clough and Tim S Evans. "Embedding graphs in Lorentzian spacetime." In: *PloS one* 12.11 (2017), e0187301.

[41] Gene Ontology Consortium. "Gene ontology consortium: going forward." In: *Nucleic acids research* 43.D1 (2015), pp. D1049–D1056.

[42] Jindou Dai, Yuwei Wu, Zhi Gao, and Yunde Jia. "A Hyperbolic-to-Hyperbolic Graph Convolutional Network." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 154–163.

[43] Shib Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Li, and Andrew McCallum. "Improving local identifiability in probabilistic box embeddings." In: *NeurIPS* 33 (2020), pp. 182–192.

[44] Michaël Defferrard, Nathanaël Perraudin, Tomasz Kacprzak, and Raphael Sgier. "DeepSphere: towards an equivariant graph-based spherical CNN." In: *ICLR 2019 Workshop on Representation Learning on Graphs and Manifolds* (2019).

[45] Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. "Lifted Rule Injection for Relation Embeddings." In: *EMNLP*. The Association for Computational Linguistics, 2016, pp. 1389–1399.

[46] Janez Demsar. "Statistical Comparisons of Classifiers over Multiple Data Sets." In: *J. Mach. Learn. Res.* 7 (2006), pp. 1–30.

[47] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. "Large-Scale Object Classification Using Label Relation Graphs." In: *ECCV (1)*. Springer, 2014, pp. 48–64.

[48] Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Saso Dzeroski. "Hierarchical annotation of medical images." In: *Pattern Recognit.* 44.10-11 (2011), pp. 2436–2449.

[49] Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Saso Dzeroski. "Hierarchical classification of diatom images using ensembles of predictive clustering trees." In: *Ecol. Informatics* 7.1 (2012), pp. 19–29.

[50] Tiansi Dong, Christian Bauckhage, Hailong Jin, Juanzi Li, Olaf Cremers, Daniel Speicher, Armin B. Cremers, and Jörg Zimmermann. "Imposing Category Trees Onto Word-Embeddings Using A Geometric Construction." In: *ICLR (Poster)*. OpenReview.net, 2019.

[51] Peng Fang, Arijit Khan, Siqiang Luo, Fang Wang, Dan Feng, Zhenli Li, Wei Yin, and Yuchao Cao. "Distributed Graph Embedding with Information-Oriented Random Walks." In: *Proc. VLDB Endow.* 16.7 (2023), pp. 1643–1656.

[52] Bahare Fatemi, Perouz Taslakian, David Vázquez, and David Poole. "Knowledge Hypergraphs: Prediction Beyond Binary Relations." In: *IJCAI*. ijcai.org, 2020, pp. 2191–2197.

[53] Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. "Message Passing for Hyper-Relational Knowledge Graphs." In: *EMNLP (1)*. Association for Computational Linguistics, 2020, pp. 7346–7359.

[54] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. "Hyperbolic Neural Networks." In: *Advances in neural information processing systems*. 2018, pp. 5350–5360.

[55] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. "Hyperbolic entailment cones for learning hierarchical embeddings." In: *ICML*. PMLR. 2018, pp. 1646–1655.

[56] Tingran Gao, Lek-Heng Lim, and Ke Ye. "Semi-Riemannian manifold optimization." In: *CoRR* (2018).

[57] Fabio Giannoni, Paolo Piccione, and Rosella Sampalmieri. "On the geodesical connectedness for a class of semi-Riemannian manifolds." In: *Journal of mathematical analysis and applications* 252.1 (2000), pp. 444–476.

[58] Eleonora Giunchiglia and Thomas Lukasiewicz. "Coherent Hierarchical Multi-Label Classification Networks." In: *NeurIPS*. 2020.

[59] David Gleich, Leonid Zhukov, and Pavel Berkhin. "Fast parallel PageRank: A linear system approach." In: *Yahoo! Research Technical Report YRL-2004-038, available via http://research. yahoo. com/publication/YRL-2004-038. pdf* 13 (2004), p. 22.

[60] Bernardo Cuenca Graua, Ian Horrocksa, Boris Motika, Bijan Parsiab, Peter Patel-Schneiderc, and Ulrike Sattlerb. "Web semantics: Science, services and agents on the World Wide Web." In: *Web Semantics: Science, Services and Agents on the World Wide Web* 6 (2008), pp. 309–322.

[61] Todd J Green, Grigoris Karvounarakis, and Val Tannen. "Provenance semirings." In: *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 2007, pp. 31–40.

[62] Mikhael Gromov. "Hyperbolic groups." In: *Essays in group theory*. Springer, 1987, pp. 75–263.

[63] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. "Learning Mixed-Curvature Representations in Product Spaces." In: *ICLR (Poster)*. OpenReview.net, 2019.

[64] Saiping Guan, Xiaolong Jin, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. "NeuInfer: Knowledge Inference on N-ary Facts." In: *ACL*. Association for Computational Linguistics, 2020, pp. 6141–6151.

[65] Saiping Guan, Xiaolong Jin, Jiafeng Guo, Yuanzhuo none Wang, and Xueqi Cheng. "Link prediction on n-ary relational data based on relatedness evaluation." In: *TKDE* (2021).

[66] Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. "Link Prediction on N-ary Relational Data." In: *WWW*. ACM, 2019, pp. 583–593.

[67] Jia Guo and Stanley Kok. "BiQUE: Biquaternionic Embeddings of Knowledge Graphs." In: *EMNLP (1)*. Association for Computational Linguistics, 2021, pp. 8338–8351.

[68] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. "Jointly Embedding Knowledge Graphs and Logical Rules." In: *EMNLP*. The Association for Computational Linguistics, 2016, pp. 192–202.

[69] Víctor Gutiérrez-Basulto and Steven Schockaert. "From Knowledge Graph Embedding to Ontology Embedding? An Analysis of the Compatibility between Vector Space Representations and Rules." In: *KR*. AAAI Press, 2018, pp. 379–388.

[70] William L. Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive Representation Learning on Large Graphs." In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett. 2017, pp. 1024–1034. URL: https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html.

[71] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. "DyERNIE: Dynamic Evolution of Riemannian Manifold Embeddings for Temporal Knowledge Graph Completion." In: *EMNLP*. 2020, pp. 7301–7316.

[72] MA Harris, J Clark, A Ireland, J Lomax, M Ashburner, R Foulger, K Eilbeck, S Lewis, B Marshall, C Mungall, et al. "The Gene Ontology (GO) database and informatics resource Nucleic Acids Research, 32." In: *D258–D261* (2004).

[73] Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. "Learning to represent knowledge graphs with Gaussian embedding." In: *CIKM*. 2015, pp. 623–632.

[74] Nicholas J. Higham. "*J*-Orthogonal Matrices: Properties and Generation." In: *SIAM Rev.* 45.3 (2003), pp. 504–519.

[75] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. "Knowledge graphs." In: *ACM Computing Surveys* 54.4 (2021), pp. 1–37.

[76] Gang Hua, Matthew A. Brown, and Simon A. J. Winder. "Discriminant Embedding for Local Image Descriptors." In: *ICCV*. IEEE Computer Society, 2007, pp. 1–8.

[77] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. "Knowledge Graph Embedding Based Question Answering." In: *WSDM*. ACM, 2019, pp. 105–113.

[78] Rana Hussein, Dingqi Yang, and Philippe Cudré-Mauroux. "Are Meta-Paths Necessary?: Revisiting Heterogeneous Graph Embeddings." In: *CIKM*. ACM, 2018, pp. 437–446.

[79] Roshni G Iyer, Yunsheng Bai, Wei Wang, and Yizhou Sun. "Dual-Geometric Space Embedding Model for Two-View Knowledge Graphs." In: *SIGKDD*. 2022.

[80] Richard V Kadison. "The Pythagorean theorem: I. The finite case." In: *Proceedings of the National Academy of Sciences* 99.7 (2002), pp. 4178–4184.

[81] Md. Rezaul Karim, Till Döhmen, Michael Cochez, Oya Beyan, Dietrich Rebholz-Schuhmann, and Stefan Decker. "DeepCOVIDExplainer: Explainable COVID-19 Diagnosis from Chest X-ray Images." In: *BIBM*. IEEE, 2020, pp. 1034–1037.

[82] Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik. "The Incredible ELK - From Polynomial Procedures to Efficient Reasoning with EL Ontologies." In: *J. Autom. Reason.* 53.1 (2014), pp. 1–61.

[83] Seyed Mehran Kazemi and David Poole. "SimplE Embedding for Link Prediction in Knowledge Graphs." In: *NeurIPS*. 2018, pp. 4289–4300.

[84] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization." In: *ICLR (Poster)*. 2015.

[85] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks." In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=SJU4ayYgl.

[86] Denis Kleyko, Dmitri A. Rachkovskij, Evgeny Osipov, and Abbas Rahimi. "A Survey on Hyperdimensional Computing aka Vector Symbolic Architectures, Part I: Models and Data Transformations." In: *ACM Comput. Surv.* 55.6 (2023), 130:1–130:40.

[87] Bryan Klimt and Yiming Yang. "The Enron Corpus: A New Dataset for Email Classification Research." In: *ECML*. Vol. 3201. Lecture Notes in Computer Science. Springer, 2004, pp. 217–226.

[88] Max Kochurov, Rasul Karimov, and Serge Kozlukov. "Geoopt: Riemannian Optimization in PyTorch." In: *CoRR* abs/2005.02819 (2020).

[89] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. "Hyperbolic geometry of complex networks." In: *Physical Review E* 82.3 (2010), p. 036106.

[90] Ranjay Krishna et al. "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations." In: *Int. J. Comput. Vis.* 123.1 (2017), pp. 32–73.

[91] Maxat Kulmanov and Robert Hoehndorf. "DeepGOPlus: improved protein function prediction from sequence." In: *Bioinformatics* 36.2 (2020), pp. 422–429.

[92] Maxat Kulmanov, Wang Liu-Wei, Yuan Yan, and Robert Hoehndorf. "EL Embeddings: Geometric Construction of Models for the Description Logic EL++." In: *IJCAI*. ijcai.org, 2019, pp. 6103–6109.

[93] Maxat Kulmanov, Fatima Zohra Smaili, Xin Gao, and Robert Hoehndorf. "Semantic similarity and machine learning with ontologies." In: *Briefings Bioinform.* 22.4 (2021).

[94] Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. "Canonical Tensor Decomposition for Knowledge Base Completion." In: *ICML*. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 2869–2878.

[95] Gerhard Lakemeyer and Bernhard Nebel. "Foundations of Knowledge Representation and Reasoning." In: *ECAI Workshop on Knowledge Representation and Reasoning*. Vol. 810. Lecture Notes in Computer Science. Springer, 1992, pp. 1–12.

[96] Julian Laub and Klaus-Robert Müller. "Feature discovery in non-metric pairwise data." In: *The Journal of Machine Learning Research* 5 (2004), pp. 801–818.

[97] Marc T. Law and Jos Stam. "Ultrahyperbolic Representation Learning." In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/123b7f02433572a0a560e620311a469c-Abstract.html.

[98] Xiang Li, Luke Vilnis, Dongxu Zhang, Michael Boratko, and Andrew McCallum. "Smoothing the Geometry of Probabilistic Box Embeddings." In: *ICLR*. OpenReview.net, 2019.

[99] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. "Learning Entity and Relation Embeddings for Knowledge Graph Completion." In: *AAAI*. AAAI Press, 2015, pp. 2181–2187.

[100] Hanxiao Liu, Yuexin Wu, and Yiming Yang. "Analogical Inference for Multi-relational Embeddings." In: *ICML*. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 2168–2178.

[101] Qi Liu, Maximilian Nickel, and Douwe Kiela. "Hyperbolic Graph Neural Networks." In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett. 2019, pp. 8228–8239. URL: https://proceedings.neurips.cc/paper/2019/hash/103303dd56a731e377d01f6a37badae3-Abstract.html.

[102] Yu Liu, Quanming Yao, and Yong Li. "Generalizing Tensor Decomposition for N-ary Relational Knowledge Bases." In: *WWW*. ACM / IW3C2, 2020, pp. 1104–1114.

[103] Yu Liu, Quanming Yao, and Yong Li. "Role-Aware Modeling for N-ary Relational Knowledge Bases." In: *WWW*. ACM / IW3C2, 2021, pp. 2660–2671.

[104] Federico López, Benjamin Heinzerling, and Michael Strube. "Fine-Grained Entity Typing in Hyperbolic Space." In: *RepL4NLP@ACL*. Association for Computational Linguistics, 2019, pp. 169–180.

[105] Jiaying Lu, Jiaming Shen, Bo Xiong, Wenjing Ma, Steffen Staab, and Carl Yang. "HiPrompt: Few-Shot Biomedical Knowledge Fusion via Hierarchy-Oriented Prompting." In: *SIGIR*. ACM, 2023.

[106] Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. "Neural network-based question answering over knowledge graphs on word and character level." In: *WWW*. 2017, pp. 1211–1220.

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

[108] Julian J. McAuley and Jure Leskovec. "Learning to Discover Social Circles in Ego Networks." In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. 2012, pp. 548–556.

[109] Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. "Anytime Bottom-Up Rule Learning for Knowledge Graph Completion." In: *IJCAI*. ijcai.org, 2019, pp. 3137–3143.

[110] Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance M. Kaplan, and Jiawei Han. "Spherical Text Embedding." In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett. 2019, pp. 8206–8215. URL: https://proceedings.neurips.cc/paper/2019/hash/043ab21fc5a1607b381ac3896176dac6-Abstract.html.

[111] George A. Miller. "WordNet: A Lexical Database for English." In: *Commun. ACM* 38.11 (1995), pp. 39–41.

[112] Farzaneh Mirzazadeh. "Solving Association Problems with Convex Co-embedding." In: (2017).

[113] Farzaneh Mirzazadeh, Siamak Ravanbakhsh, Nan Ding, and Dale Schuurmans. "Embedding Inference for Structured Multilabel Prediction." In: *NIPS*. 2015, pp. 3555–3563.

[114] Sutapa Mondal, Sumit Bhatia, and Raghava Mutharaju. "EmEL++: Embeddings for EL++ Description Logic." In: *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*. Vol. 2846. CEUR Workshop Proceedings. CEUR-WS.org, 2021.

[115] Christopher J Mungall, Carlo Torniai, Georgios V Gkoutos, Suzanna E Lewis, and Melissa A Haendel. "Uberon, an integrative multi-species anatomy ontology." In: *Genome biology* 13.1 (2012), pp. 1–20.

[116] Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. "Hierarchical Losses and New Resources for Fine-grained Entity Typing and Linking." In: *ACL (1)*. Association for Computational Linguistics, 2018, pp. 97–109.

[117] Mojtaba Nayyeri, Sahar Vahdati, Can Aykul, and Jens Lehmann. "5* Knowledge Graph Embeddings with Projective Transformations." In: *AAAI*. Vol. 35. 2021, pp. 9064–9072.

[118] Mojtaba Nayyeri, Bo Xiong, Majid Mohammadi, Mst. Mahfuja Akter, Mirza Mohtashim Alam, Jens Lehmann, and Steffen Staab. "Knowledge Graph Embeddings using Neural Ito Process: From Multiple Walks to Stochastic Trajectories." In: *ACL (Findings)*. Association for Computational Linguistics, 2023, pp. 7165–7179.

[119] Mojtaba Nayyeri, Chengjin Xu, Franca Hoffmann, Mirza Mohtashim Alam, Jens Lehmann, and Sahar Vahdati. "Knowledge graph representation learning using ordinary differential equations." In: *EMNLP*. 2021, pp. 9529–9548.

[120] Maximilian Nickel and Douwe Kiela. "Poincaré Embeddings for Learning Hierarchical Representations." In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett. 2017, pp. 6338–6347. URL: https://proceedings.neurips.cc/paper/2017/hash/59dfa2df42d9e3d41f5b02bfc32229dd-Abstract.html.

[121] Maximilian Nickel and Douwe Kiela. "Poincaré Embeddings for Learning Hierarchical Representations." In: *NIPS*. 2017, pp. 6338–6347.

[122] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. "A Three-Way Model for Collective Learning on Multi-Relational Data." In: *ICML*. Omnipress, 2011, pp. 809–816.

[123] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. "A Three-Way Model for Collective Learning on Multi-Relational Data." In: *ICML*. Omnipress, 2011, pp. 809–816.

[124] Guanglin Niu, Yongfei Zhang, Bo Li, Peng Cui, Si Liu, Jingyang Li, and Xiaowei Zhang. "Rule-Guided Compositional Representation Learning on Knowledge Graphs." In: *AAAI*. AAAI Press, 2020, pp. 2950–2958.

[125] Barrett O'neill. "Semi-Riemannian geometry with applications to relativity." In: *Academic pres* 103 (1983).

[126] Özgü r Lü tfü Özçep, Mena Leemhuis, and Diedrich Wolter. "Cone semantics for logics with negation." In: *IJCAI*. 2021, pp. 1820–1826.

[127] Özgür Lütfü Özçep, Mena Leemhuis, and Diedrich Wolter. "Cone Semantics for Logics with Negation." In: *IJCAI*. ijcai.org, 2020, pp. 1820–1826.

[128] Zhe Pan and Peng Wang. "Hyperbolic Hierarchy-Aware Knowledge Graph Embedding for Link Prediction." In: *EMNLP*. 2021, pp. 2941–2948.

[129] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. "Pytorch: An imperative style, high-performance deep learning library." In: *Advances in neural information processing systems* 32 (2019).

[130] Dhruvesh Patel, Pavitra Dangati, Jay-Yoon Lee, Michael Boratko, and Andrew McCallum. "Modeling label space interactions in multi-label classification using box embeddings." In: *ICLR*. 2021.

[131] Dhruvesh Patel and Shib Sankar. "Representing joint hierarchies with box embeddings." In: *AKBC* (2020).

[132] Alexandrin Popescul and Lyle H Ungar. "Statistical relational learning for link prediction." In: *IJCAI workshop on learning statistical models from relational data*. Vol. 2003. 2003.

[133] John G Ratcliffe, S Axler, and KA Ribet. *Foundations of hyperbolic manifolds*. Vol. 149. Springer, 1994.

[134] Alan L Rector, Jeremy E Rogers, and Pam Pole. "The GALEN high level ontology." In: *Medical Informatics Europe'96*. IOS Press, 1996, pp. 174–178.

[135] Hongyu Ren, Weihua Hu, and Jure Leskovec. "Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings." In: *ICLR*. 2019.

[136] Hongyu Ren and Jure Leskovec. "Beta embeddings for multi-hop logical reasoning in knowledge graphs." In: *NeurIPS* 33 (2020), pp. 19716–19726.

[137] Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. "Beyond Triplets: Hyper-Relational Knowledge Graph Embedding for Link Prediction." In: *WWW*. ACM / IW3C2, 2020, pp. 1885–1896.

[138] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. "DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs." In: *NeurIPS*. 2019, pp. 15321–15331.

[139] Kenny Schlegel, Peer Neubert, and Peter Protzel. "A comparison of vector symbolic architectures." In: *Artif. Intell. Rev.* 55.6 (2022), pp. 4523–4555.

[140] Ron Shepard, Scott R Brozell, and Gergely Gidofalvi. "The representation and parametrization of orthogonal matrices." In: *The Journal of Physical Chemistry A* 119.28 (2015), pp. 7924–7939.

[141] Harry Shomer, Wei Jin, Juan-Hui Li, Yao Ma, and Jiliang Tang. "Learning Representations for Hyper-Relational Knowledge Graphs." In: *CoRR* abs/2208.14322 (2022).

[142] Aaron Sim, Maciej Wiatrak, Angus Brayne, Páidí Creed, and Saee Paliwal. "Directed Graph Embeddings in Pseudo-Riemannian Manifolds." In: *Thirty-eighth International Conference on Machine Learning* (2021).

[143] Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. "Mixed-curvature Variational Autoencoders." In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=S1g6xeSKDS.

[144] Fatima Zohra Smaili, Xin Gao, and Robert Hoehndorf. "Onto2Vec: joint vector-based representation of biological entities and their ontology-based annotations." In: *Bioinform.* 34.13 (2018), pp. i52–i60.

[145] Fatima Zohra Smaili, Xin Gao, and Robert Hoehndorf. "OPA2Vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction." In: *Bioinform.* 35.12 (2019), pp. 2133–2140.

[146] Daniel Spiegel. "The hopf-rinow theorem." In: *Notes available online* (2016).

[147] Andreas Steigmiller, Thorsten Liebig, and Birte Glimm. "Konclude: System description." In: *J. Web Semant.* 27-28 (2014), pp. 78–85.

[148] George Stephanopoulos and Chonghun Han. "Intelligent systems in process engineering: A review." In: *Computers & Chemical Engineering* 20.6-7 (1996), pp. 743–791.

[149] Michael Stewart and Paul Van Dooren. "On the Factorization of Hyperbolic and Unitary Transformations into Rotations." In: *SIAM J. Matrix Anal. Appl.* 27.3 (2005), pp. 876–890.

[150] Ke Sun, Jun Wang, Alexandros Kalousis, and Stéphane Marchand-Maillet. "Space-Time Local Embeddings." In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. Ed. by Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett. 2015, pp. 100–108. URL: https://proceedings.neurips.cc/paper/2015/hash/7cbbc409ec990f19c78c75bd1e06f215-Abstract.html.

[151] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space." In: *ICLR (Poster)*. OpenReview.net, 2019.

[152] Ryota Suzuki, Ryusuke Takahama, and Shun Onoda. "Hyperbolic disk embeddings for directed acyclic graphs." In: *ICML*. PMLR. 2019, pp. 6066–6075.

[153] Puoya Tabaghi and Ivan Dokmanic. "On Procrustes Analysis in Hyperbolic Space." In: *IEEE Signal Process. Lett.* 28 (2021), pp. 1120–1124.

[154] Yanchao Tan, Hang Lv, Zihao Zhou, Wenzhong Guo, Bo Xiong, Weiming Liu, Chaochao Chen, Shiping Wang, and Carl Yang. "Logical Relation Modeling and Mining in Hyperbolic Space for Recommendation." In: *The 40th IEEE International Conference on Data Engineering* (2024).

[155] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. "Representing Text for Joint Embedding of Text and Knowledge Bases." In: *EMNLP*. The Association for Computational Linguistics, 2015, pp. 1499–1509.

[156] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. "Complex Embeddings for Simple Link Prediction." In: *ICML*. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 2071–2080.

[157] Jean Van Heijenoort. "Set-theoretic semantics." In: *Studies in Logic and the Foundations of Mathematics*. Vol. 87. Elsevier, 1977, pp. 183–190.

[158] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. "Graph Attention Networks." In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=rJXMpikCZ.

[159] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. "Order-embeddings of images and language." In: *ICLR* (2016).

[160] John Venn. "I. On the diagrammatic and mechanical representation of propositions and reasonings." In: *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 10.59 (1880), pp. 1–18.

[161] Luke Vilnis. "Geometric Representation Learning." In: *Doctoral dissertations. nr. 2146 University of Massachusetts Amherst* (2021), pp. 116–119. URL: https://doi.org/10.7275/20638273.

[162] Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. "Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures." In: *ACL*. 2018, pp. 263–272.

[163] Luke Vilnis and Andrew McCallum. "Word Representations via Gaussian Embedding." In: *ICLR*. 2015.

[164] Denny Vrandečić and Markus Krötzsch. "Wikidata: a free collaborative knowledgebase." In: *Communications of the ACM* 57.10 (2014), pp. 78–85.

[165] Thi Thuy Duong Vu and Jaehee Jung. "Protein function prediction with gene ontology: from traditional to deep learning models." In: *PeerJ* 9 (2021), e12019.

[166] Feiyang Wang, Zhongbao Zhang, Li Sun, Junda Ye, and Yang Yan. "DiriE: Knowledge Graph Embedding with Dirichlet Distribution." In: *WWW*. 2022, pp. 3082–3091.

[167] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. "Joint Embedding of Words and Labels for Text Classification." In: *ACL (1)*. Association for Computational Linguistics, 2018, pp. 2321–2331.

[168] Kai Wang, Yu Liu, Dan Lin, and Michael Sheng. "Hyperbolic Geometry is Not Necessary: Lightweight Euclidean-Based Models for Low-Dimensional Knowledge Graph Embeddings." In: *EMNLP (Findings)*. Association for Computational Linguistics, 2021, pp. 464–474.

[169] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. "Knowledge graph embedding: A survey of approaches and applications." In: *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017), pp. 2724–2743.

[170] Quan Wang, Haifeng Wang, Yajuan Lyu, and Yong Zhu. "Link Prediction on N-ary Relational Facts: A Graph-based Approach." In: *ACL/IJCNLP (Findings)*. Vol. ACL/IJCNLP 2021. Findings of ACL. Association for Computational Linguistics, 2021, pp. 396–407.

[171] Shen Wang, Xiaokai Wei, Cicero Nogueira Nogueira dos Santos, Zhiguo Wang, Ramesh Nallapati, Andrew Arnold, Bing Xiang, Philip S Yu, and Isabel F Cruz. "Mixed-curvature multi-relational graph neural network for knowledge graph completion." In: *WWW*. 2021, pp. 1761–1771.

[172] Shen Wang, Xiaokai Wei, Cícero Nogueira dos Santos, Zhiguo Wang, Ramesh Nallapati, Andrew O. Arnold, Bing Xiang, Philip S. Yu, and Isabel F. Cruz. "Mixed-Curvature Multi-Relational Graph Neural Network for Knowledge Graph Completion." In: *WWW*. ACM / IW3C2, 2021, pp. 1761–1771.

[173] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. "Knowledge Graph Embedding by Translating on Hyperplanes." In: *AAAI*. AAAI Press, 2014, pp. 1112–1119.

[174] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world'networks." In: *nature* 393.6684 (1998), pp. 440–442.

[175] Melanie Weber and Maximilian Nickel. "Curvature and representation learning: Identifying embedding spaces for relational data." In: *NeurIPS* (2018).

[176] Jonatas Wehrmann, Ricardo Cerri, and Rodrigo C. Barros. "Hierarchical Multi-Label Classification Networks." In: *ICML*. Vol. 80. PMLR. PMLR, 2018, pp. 5225–5234.

[177] Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. "On the Representation and Embedding of Knowledge Bases beyond Binary Relations." In: *IJCAI*. IJCAI/AAAI Press, 2016, pp. 1300–1307.

[178] Thomas James Willmore. *An introduction to differential geometry*. Courier Corporation, 2013.

[179] Richard C Wilson, Edwin R Hancock, Elżbieta Pekalska, and Robert PW Duin. "Spherical and hyperbolic embeddings of data." In: *IEEE transactions on pattern analysis and machine intelligence* 36.11 (2014), pp. 2255–2269.

[180] Felix Wu, Amauri H. Souza Jr., Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. "Simplifying Graph Convolutional Networks." In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 6861–6871. URL: http://proceedings.mlr.press/v97/wu19e.html.

[181] Han Xiao, Minlie Huang, and Xiaoyan Zhu. "TransG: A Generative Model for Knowledge Graph Embedding." In: *ACL*. 2016, pp. 2316–2325.

[182] Bo Xiong, Michael Cochez, Mojtaba Nayyeri, and Steffen Staab. "Hyperbolic Embedding Inference for Structured Multi-Label Prediction." In: *NeurIPS*. 2022.

[183] Bo Xiong, Michael Cochez, Mojtaba Nayyeri, and Steffen Staab. "Hyperbolic Embedding Inference for Structured Multi-Label Prediction." In: *Advances in Neural Information Processing Systems* (2022).

[184] Bo Xiong, Mojtaba Nayyer, Shirui Pan, and Steffen Staab. "Shrinking Embeddings for Hyper-Relational Knowledge Graphs." In: *ACL*. International Committee on Computational Linguistics, 2023.

[185] Bo Xiong, Mojtaba Nayyeri, Ming Jin, Yunjie He, Michael Cochez, Shirui Pan, and Steffen Staab. "Geometric Relational Embeddings: A Survey." In: *CoRR* abs/2304.11949 (2023).

[186] Bo Xiong, Mojtaba Nayyeri, Linhao Luo, Zihao Wang, Shirui Pan, and Steffen Staab. "NestE: Modeling Nested Relational Structures for Knowledge Graph Reasoning." In: *AAAI*. 2024.

[187] Bo Xiong, Nico Potyka, Trung-Kien Tran, Mojtaba Nayyeri, and Steffen Staab. "Faithful Embeddings for EL++ Knowledge Bases." In: *ISWC*. Springer. 2022, pp. 22–38.

[188] Bo Xiong, Nico Potyka, Trung-Kien Tran, Mojtaba Nayyeri, and Steffen Staab. "Faithful Embeddings for EL++ Knowledge Bases." In: *ISWC*. Vol. 13489. Lecture Notes in Computer Science. Springer, 2022, pp. 22–38.

[189] Bo Xiong, Shichao Zhu, Mojtaba Nayyeri, Chengjin Xu, Shirui Pan, Chuan Zhou, and Steffen Staab. "Ultrahyperbolic Knowledge Graph Embeddings." In: *KDD*. ACM, 2022, pp. 2130–2139.

[190] Bo Xiong, Shichao Zhu, Nico Potyka, Shirui Pan, Chuan Zhou, and Steffen Staab. "Pseudo-Riemannian Graph Convolutional Networks." In: *NeurIPS*. 2022.

[191] Chengjin Xu, Fenglong Su, Bo Xiong, and Jens Lehmann. "Time-aware Entity Alignment using Temporal Relational Attention." In: *WWW*. ACM, 2022, pp. 788–797.

[192] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. "Embedding Entities and Relations for Learning and Inference in Knowledge Bases." In: *ICLR (Poster)*. 2015.

[193] Fan Yang, Zhilin Yang, and William W. Cohen. "Differentiable Learning of Logical Rules for Knowledge Base Reasoning." In: *NIPS*. 2017, pp. 2319–2328.

[194] Menglin Yang, Zhihao Li, Min Zhou, Jiahong Liu, and Irwin King. "Hicf: Hyperbolic informative collaborative filtering." In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, pp. 2212–2221.

[195] Menglin Yang, Min Zhou, Jiahong Liu, Defu Lian, and Irwin King. "HRCF: Enhancing collaborative filtering via hyperbolic geometric regularization." In: *Proceedings of the ACM Web Conference 2022*. 2022, pp. 2462–2471.

[196] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. "Revisiting Semi-Supervised Learning with Graph Embeddings." In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. Ed. by Maria-Florina Balcan and Kilian Q. Weinberger. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 40–48. URL: http://proceedings.mlr.press/v48/yanga16.html.

[197] Donghan Yu and Yiming Yang. "Improving Hyper-Relational Knowledge Graph Completion." In: *CoRR* abs/2104.08167 (2021).

[198] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. "Collaborative knowledge base embedding for recommender systems." In: *SIGKDD*. 2016, pp. 353–362.

[199] Richong Zhang, Junpeng Li, Jiajie Mei, and Yongyi Mao. "Scalable Instance Reconstruction in Knowledge Bases via Relatedness Affiliated Embedding." In: *WWW*. ACM, 2018, pp. 1185–1194.

[200] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. "Quaternion Knowledge Graph Embeddings." In: *NeurIPS*. 2019, pp. 2731–2741.

[201] Yiding Zhang, Xiao Wang, Chuan Shi, Nian Liu, and Guojie Song. "Lorentzian Graph Convolutional Networks." In: *International World Wide Web Conference Committee*. 2021.

[202] Zhanqiu Zhang, Jie Wang, Jiajun Chen, Shuiwang Ji, and Feng Wu. "ConE: Cone embeddings for multi-hop reasoning over knowledge graphs." In: *NeurIPS* 34 (2021).

[203] Shichao Zhu, Shirui Pan, Chuan Zhou, Jia Wu, Yanan Cao, and Bin Wang. "Graph Geometry Interaction Learning." In: *Advances in Neural Information Processing Systems*. 2020.

DECLARATION

Erklärung über die Eigenständigkeit der Dissertation

Ich versichere, dass ich die vorliegende Arbeit mit dem Titel „Geometric relational embeddings“ selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe; aus fremden Quellen entnommene Passagen und Gedanken sind als solche kenntlich gemacht.

Declaration of authorship

I hereby certify that the dissertation entitled “Geometric relational embeddings” is entirely my own work except where otherwise indicated. Passages and ideas from other sources have been clearly indicated.

Stuttgart, Sep. 2, 2024

Bo Xiong