PolygonE: Modeling N-ary Relational Data as Gyro-Polygons in Hyperbolic Space

Shiyao Yan[1,2,3,4], Zequn Zhang[1,2], Xian Sun[1,2], Guangluan Xu[1,2], Shuchao Li[1,2], Qing Liu[1,2], Nayu Liu[1,2,3,4], Shensi Wang[1,2,3,4]

[1] Aerospace Information Research Institute, Chinese Academy of Sciences
[2] Key Laboratory of Network Information System Technology(NIST), Aerospace Information Research Institute
[3] University of Chinese Academy of Sciences
[4] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences
{yanshiyao19,liunayu18,wanshensi18}@mails.ucas.ac.cn, {sunxian,gluanxu,zqzhang1,lisc,liuqing1}@mail.ie.ac.cn

Abstract

N-ary relational knowledge base (KBs) embedding aims to map binary and beyond-binary facts into low-dimensional vector space simultaneously. Existing approaches typically decompose n-ary relational facts into subtuples, and they generally model n-ary relational KBs in Euclidean space. However, n-ary relational facts are semantically and structurally intact; decomposition leads to the loss of global information and undermines the semantical and structural integrity. Moreover, compared to the binary relational KBs, n-ary ones are characterized by more abundant and complicated hierarchy structures, which could not be well expressed in Euclidean space. To address the issues, we propose a gyro-polygon embedding framework to realize n-ary fact integrity keeping and hierarchy capturing, termed as PolygonE. Specifically, n-ary relational facts are modeled as gyro-polygons in the hyperbolic space, where we denote entities in facts as vertexes of gyro-polygons and relations as entity translocation operations. Importantly, we design a fact plausibility measuring strategy based on the vertex-gyrocentroid geodesic to optimize the relation-adjusted gyro-polygon. Experimental results demonstrate that PolygonE shows SOTA performance on all benchmark datasets and generalizes well on binary data. Finally, we also visualize the embedding to help comprehend PolygonE's awareness of hierarchies.

Introduction

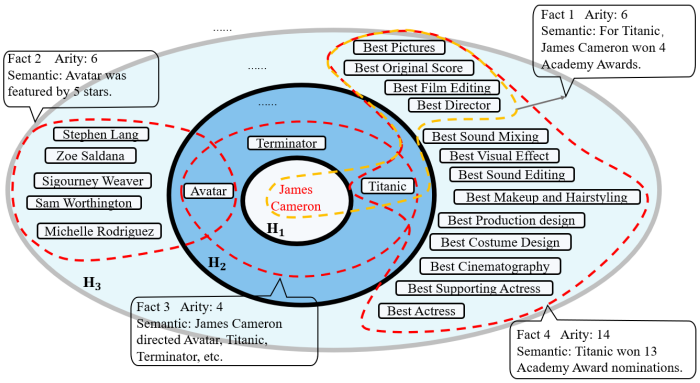
N-ary relational KBs comprise both binary and beyond-binary relational facts, among which binary relational facts in the form of (h,r,t) have been extensively explored in the past decade. In contrast, the beyond-binary ones are less studied (Ji et al. 2021). The latest research finds that beyond-binary relational facts contain abundant semantics and are closer to human-intelligent compared to their traditional binary counterparts (Wang et al. 2017). For the excellent potential that n-ary relational KBs showcase in NLP downstream tasks like textual entailment, question answering, and natural language understanding (Hogan et al. 2021), recently, lots of efforts (Rouces, de Melo, and Hose 2015; Zhang et al. 2018; Liu, Yao, and Li 2021) are poured into representing binary and beyond-binary facts simultaneously. And Existing works have seen gratifying progress in embedding n-ary relational KBs.

However, the best-performing models (Guan et al. 2020; Rosso, Yang, and Cudré-Mauroux 2020; Liu, Yao, and Li 2021) generally break down n-ary relational facts into subtuples (triples, pairs, quintuples, etc.). Such decomposition leads to both structural and semantical information loss (Rosso, Yang, and Cudré-Mauroux 2020). It makes sense to keep an n-ary relational fact intact. Still, some models (Liu, Yao, and Li 2020) work for single-arity data only, suffering from inflexibility.

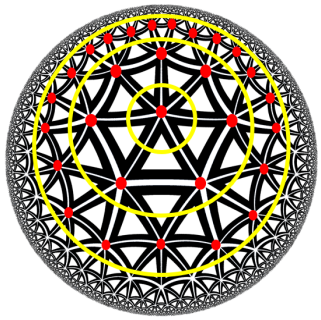
Moreover, we note that n-ary relational KBs are characterized by more abundant and complicated hierarchy structures with more entities involved in a single fact. Figure 1(a) shows several Wikipedia-exampled n-ary relational facts centered around the movie director *James Cameron*. We hierarchize the involved entities, from H_1 to H_3, the number of involved entities grows exponentially, which is in good coherence with the superlinear length growth (Ungar 2009) in a hyperbolic poincaré ball (as intuitively shown in Figure 1(b)). And it's noted that the higher the arity, the more pronounced the exponential characteristic. Existing works generally overlook such hierarchical information, which, if well captured, can benefit n-ary relational KB embeddings.

With the above consideration, we model n-ary relational facts as gyro-polygons (i.e., polygons in hyperbolic space), and propose a model termed as PolygonE where the polygon structure ensures fact integrity by representing fact semantic in the gyro-centroid, and it also ensures the model's applicability to arbitrary arity data. While hyperbolic space guarantees hierarchy capturing. Primarily, we represent entities in a fact as vertexes of a gyro-polygon and relations as entity translocations. Significantly, we optimize the relation-adjusted gyro-polygon by minimizing the vertex-gyrocentroid geodesic. Experimental results illustrate that PolygonE achieves excellent results on WikiPeople, JF17K, and FB-AUTO. And visualization of embeddings shows that PolygonE captures the hierarchies within n-ary relational KBs. To summarize, our contribution can be summarized as follows:

- We propose PolygonE where we model n-ary relational facts as gyro-polygons to ensure structural and semantical integrity, adjustability to arbitrary arity fact, and hierarchy capturing. To our knowledge, gyro-polygon structure has not been studied in knowledge base embedding.
- We design a fact plausibility scoring strategy based on



(a) N-ary Relational Facts Instances



(b) A hyperbolic poincaré ball

Figure 1: (a) are some Wikipedia-exampled n-ary relational facts centered around the film director James Cameron. To hierarchize the involved entities, H_1 are the top-level entity James Cameron, H_2 lists some representative works of James Cameron (Fact 1), H_3 are entities denoting the starring (Fact2) or awarding (Fact3 and Fact4) information of the films in H_2. From H_1 to H_3, the number of involved entities grows exponentially, which is in good coherence with the superlinear growth in a poincaré ball (as shown in (b)). And the higher the arity, the more pronounced the exponential characteristic. Inspired by this, we leverage hyperbolic poincaré ball to capture the exponential feature within n-ary KBs.

the vertex-gyrocentroid geodesic to evaluate whether an entity is compatible with the whole fact, which is leveraged to optimize the relation-adjusted gyro-polygons.

- Extensive experiments show that our PolygonE realizes excellent performance. Comparison with binary models demonstrates its generalizability, and visualization shows PolygonE's awareness of hierarchy.

Related Work

Binary Relational Knowledge Base Embedding Categorized by scoring function, binary KB embedding methods mainly fall into 3 families. Translational models (Bordes et al. 2013; Wang et al. 2014; Lin et al. 2015) typically map entities onto a latent vector space via translation operations, and calculate fact validity via specific distance metrics. Deep models like ConvE (Dettmers et al. 2018), R-GCN (Schlichtkrull et al. 2018), CompGCN (Vashishth et al. 2020) demonstrate excellent expressiveness, but suffer from relatively high complexity brought by the tremendous parameters. Typical bilinear models (Nickel, Tresp, and Kriegel 2011; Yang et al. 2015; Trouillon et al. 2016) treat entity as vector and relation as matrix. Zhang et al. (2019); Nguyen et al. (2020) represent entities as quaternions, while Balazevic, Allen, and Hospedales (2019b) introduce tucker decomposition, Kazemi and Poole (2018) introduce inverse relation embedding in SimplE. In hyperbolic space, Murp (Balazevic, Allen, and Hospedales 2019a) largely surpass its Euclidean counterparts, ATTH (Chami et al. 2020) explores logic patterns, Wang et al. (2021b); Chen et al. (2021) utilize hyperbolic neural networks. These hyperbolic models are based on strong binary scoring functions, incapacitates them from representing n-ary facts.

N-ary Relational Knowledge Base Embedding Pioneering works like m-TransH (Wen et al. 2016) and RAE (Zhang et al. 2018) directly extend TransH (Lin et al. 2015) from binary to n-ary case, inheriting the weak expressiveness of TransH. Multilinear model HypE (Fatemi et al. 2020) embeds entities with positional convolutional filters and evaluates facts with multilinear product. While GETD (Liu, Yao, and Li 2020) intakes tensor ring decomposition to tucker, but it works on single-arity fact only. S2S (Di, Yao, and Chen 2021) then extends it to mixed arity facts. Deep models NaLP (Guan et al. 2019) treats N-ary relational facts as role-entity pairs and measures facts relying on FCNs, t-Nalp+ (Guan et al. 2021) further considers type information. NeuInfer (Guan et al. 2020) and HINGE (Rosso, Yang, and Cudré-Mauroux 2020) tear apart n-ary facts into a primary triple and several entity-role pairs, of which the compatibility is modeled by CNN network. StarE (Galkin et al. 2020) mainly focuses on triples. These best-performing deep models bring drastic performance boast, but they generally introduce decomposition, undermining fact structure and semantic. While RAM (Liu, Yao, and Li 2021) seeks to model the semantic roles in n-ary relations and GRAN Wang et al. 2021a treats n-ary relational KBs as a heterogeneous graph. Nevertheless, none of them notices the hierarchical anatomy within n-ary relational KBs.

Preliminaries

N-ary Relational KBs N-ary facts are ubiquitous, but the representation is not unified. The 3 widely used ones are single relation $\{r, e_1, e_2, ..., e_i, ..., e_n\}$ (Wen et al. 2016), role-value pairs $\{e_1, r_1, ..., e_i, r_i, ..., e_n, r_n\}$ (Guan et al. 2019), and triple+pairs $\{h, r, t...., r_i, e_i, ..., r_{n-2}, e_{n-2}\}$ (Rosso, Yang, and Cudré-Mauroux 2020). We follow the third one. Formally, given an n-ary relational KB $\mathcal{B}$ with a set of entities $\mathcal{E}$ and a set of relations $\mathcal{R}$, we represent a fact $\mathcal{F}$ in $\mathcal{B}$ in the form of $\mathcal{F}: (e_h, r, e_t, ...r_i, e_i, ..., r_{n-2}, e_{n-2})$, where $r \in \mathcal{R}$, $e_i \in \mathcal{E}$, and $n \geq 2$. n indicates the number of enti-

ties participating in the tuple $\mathcal{F}$. In case $n = 2$, $\mathcal{F}$ is a binary fact. If $n > 2$, $\mathcal{F}$ is a beyond-binary fact, then (e_h, r, e_t) is taken as primary triple.

Poincaré Ball Before diving into gyro-polygon, we first briefly introduce Poincaré ball and some algebraic operations in hyperbolic space. A d-dimensional Poincaré ball $(\mathbb{H}^d, g^{\mathbb{H}})$ is a real and smooth manifold $\mathbb{H}^d := \left\{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|^2 < 1\right\}$ accompanied with a Riemannian metric $g^{\mathbb{H}}$, where $g^{\mathbb{H}} = (\lambda_{\mathbf{x}})^2 g^{\mathbb{E}}$ *and* $\lambda_{\mathbf{x}} = \frac{2}{(1-\|\mathbf{x}\|^2)}$. $g^{\mathbb{E}} = \mathbf{I}_d$ and $\|\mathbf{x}\|^2$ are the Euclidean identity metric tensor and the Euclidean norm, respectively, while $\lambda_{\mathbf{x}}$ is the conformal factor between the Euclidean metric and the hyperbolic metric. Basic algebraic operations like addition and multiplication in Euclidean space cannot be directly applied in hyperbolic space, gyrovector spaces (Ungar 2009; Ganea, Bécigneul, and Hofmann 2018) provide corresponding equivalence of these algebraic operations in hyperbolic space. For a Poincaré ball of radius c, some basic operations are summarized in Appendix A. we list Möbius addition ($\oplus_c$), exponential map ($\exp^c_{\mathbf{x}}$), logarithmic map ($\log^c_{\mathbf{x}}$), matrix-vector product ($\otimes_c$), and Möbius half in Appendix A. Moreover, the distance between two points $\mathbf{x}, \mathbf{y}$ in a poincaré ball is given by:

$$d_{\mathbb{H}}(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{c}} \tanh^{-1}\left(\sqrt{c}\,\|-\mathbf{x} \oplus_c \mathbf{y}\|\right), \tag{1}$$

The inverse hyperbolic tangent function ($\tanh^{-1}$) brings the exponential length growth in a Poincaré ball. Without loss of generality, radius c is set to 1 in practice.

Gyro-polygon Polygon in Euclidean space is defined by a finite number of straight-line segments connected to form a closed polygonal chain. In full analogy, gyro-polygon is the equivalence of polygon in gyrovector space where gyro-line segments replace straight-line segments (Ungar 2009). Figure 2 are two examples of gyro-polygons. As shown in Figure 2(a), $\mathbf{ABC}$ is a gyro-trigon. In Figure 2(b), $\mathbf{ABCD}$ is a gyro-tetragon. Since n-ary relational KBs contain both binary and beyond binary facts, we utilize gyro-line segments to represent 2-ary facts, while gyro-polygons are employed to represent those beyond-binary facts. Similar to Euclidean space, the gyro-midpoint of $\mathbf{x}, \mathbf{y}$ is given by:

$$M(\mathbf{x}, \mathbf{y}) = \mathbf{x} \oplus_c (\frac{1}{2} \otimes_c (-\mathbf{x} \oplus_c \mathbf{y}), \tag{2}$$

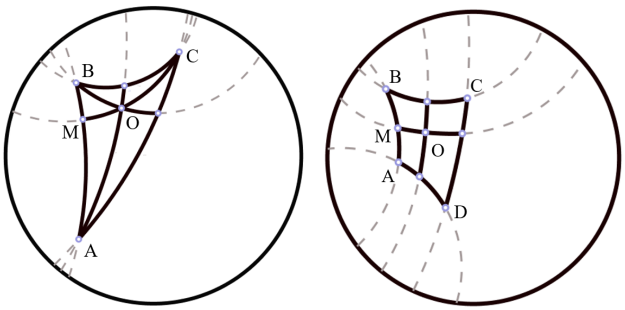
while the gyro-centroid of the gyro-polygon with $\mathbf{e}_1, \cdots, \mathbf{e}_n$ being the vertexes is given by (Ungar 2009):

$$O(\mathbf{e}_1, ...\mathbf{e}_n) = \frac{1}{2} \otimes_c \frac{\sum_{i=1}^{n} 2\gamma^2_{\mathbf{e}_i} \mathbf{e}_i}{\sum_{i=1}^{n} (2\gamma^2_{\mathbf{e}_i} - 1)}, \tag{3}$$

where $\gamma_{\mathbf{x}} = \sqrt{\frac{1}{1-\|\mathbf{x}\|^2}}$ is the gamma factor in relativity theory. With gamma factor, Möbius half is represented as:

$$\frac{1}{2} \otimes_c \mathbf{x} = \frac{\gamma_{\mathbf{x}}}{1 + \gamma_{\mathbf{x}}} \mathbf{x}. \tag{4}$$

Hence by (3) and (4), the lantent vector for gyro-centroid can be obtained.



(a) Gyro-trigon (b) Gyro-tetragon

Figure 2: Instances of gyro-polygons. Left is a gyro-trigon, right is a gyro-tetragon. In both (a) and (b), O represents the gyrocentroid, M is the gyro-midpoint.

PolygonE

To retain the structure and semantic integrity, and capture hierarchy, we propose PolygonE where 1) n-ary facts are modeled as gyro-polygons in hyperbolic poincaré ball, 2) and a vertex-gyrocentroid based plausibility function is designed to optimize the gyro-polygon.

Modeling Facts as Gyro-polygons

To keep facts intact, we model facts as whole gyro-polygons and leverage the gyro-centroid to represent the semantic. In polygonE, entities in an n-ary relational fact $\mathcal{F} : (e_h, r, e_t, ...r_i, e_i, ..., r_{n-2}, e_{n-2})$ are first initialized as random d-dimensional vectors $\mathbf{e}_h,\ \mathbf{e}_t,\ ...\ \mathbf{e}_i, ...\mathbf{e}_{n-2}$ in a gyro-polygon. As illustrated in Figure 3, the vertexes of the gyro-polygon $\mathbf{ABCDE}$ stand for the stochastically initialized latent vector of the entities in the 5-ary fact $\mathcal{F} : (A, r, B, r_1, C, r_2, D, r_3, E)$. Meanwhile, relations are interpreted as the operation of moving an entity (translation or rotation). For the sake of clarity, we only visualize one relation in Figure 3. As depicted, entity $\mathbf{A}$ is transferred to $\mathbf{A}_{pos}$ by a relation $\mathbf{r}$. Similarly, other entities in the facts are translocated by their corresponding relations in the same manner. Translation and rotation are the two essential physical movements, inspired by this, we translate or rotate each entity to obtain more accurate gyro-polygon or gyro-line segment. The intuition behind this is to get relevant entities closer to form smaller gyro-polygon or shorter gyro-line segment. From pre-experiment, in terms of the primary triple (e_h, r, e_t), head entity $\mathbf{e}_h$ is rotated by a relation-specific matrix $R \in \mathbb{R}^{d \times d}$, while tail entity $\mathbf{e}_t$ is translated by a relation $\mathbf{r} \in \mathbb{R}^d$, i.e.,

$$\mathbf{e}_h^{'} = \mathbf{R} \otimes_c \mathbf{e}_t, \quad e_t^{'} = \mathbf{e}_t \oplus_c \mathbf{r}. \tag{5}$$

The rest entities $\mathbf{e}_i, (i \in [0, n-2))$ are translocated by relation vector $\mathbf{r}_i$:

$$\mathbf{e}_i^{'} = \mathbf{e}_i \oplus_c \mathbf{r}_i. \tag{6}$$

After being translocated by corresponding relations, entities lean to get closer to form a smaller gyro-polygon. Ideally, when all entities move to somewhere identical, the relation-adjusted gyro-polygon would degenerate to a point. In this way, the intrinsic semantic is contained in the gyrocentroid.

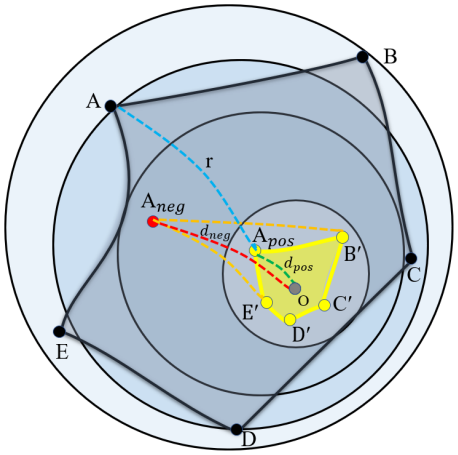


Figure 3: Illustration of PolygonE. Take a 5-ary facts as an instance, the 5 entities in the fact are first randomly initialized as $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}$. After being translocated by corresponding relation, entities tend to get closer to each other to form a smaller gyro-polygon as shown as $\mathbf{A_{pos}B'C'D'E'}$. Ideally, when all entities move to somewhere identical, the gyro-polygon would degenerate to a point.

Vertex-gyrocentroid Plausibility Measuring

To optimize the relation-adjusted gyro-polygon, the perimeter or area of the gyro-polygon are seemingly good metrics. But considering that 1) there isn't an explicit integration method that can directly give the area of an arbitrary gyro-polygon; 2) perimeter involves multiple times gyro-line segment length calculation, bringing both fact decomposition and high computation expense. We quit the two metrics.

Instead, we utilize the vertex-gyrocentroid geodesic length as the metric to evaluate whether an entity is compatible with the whole fact, which is different from existing decomposition-based models (Guan et al. 2020; Rosso, Yang, and Cudré-Mauroux 2020) that only consider the compatibility between an entity and the primary triple before a final aggregation.

Specifically, as shown in Figure 3, the shorter distance from vertex $\mathbf{A_{pos}}$ to the gyrocentroid $\mathbf{O}$ than that from $\mathbf{A_{neg}}$ to $\mathbf{O}$ suggests that the vertex-gyrocentroid metric can separate positive samples from negative ones. In fact, this metric avoids direct area or perimeter calculation, but can reflect area or perimeter. A shorter vertex-gyrocentroid geodesic length always means a smaller area and shorter perimeter. Significantly, after entities in $\mathcal{F}$: $(e_h, r, e_t, ...r_i, e_i, ..., r_{n-2}, e_{n-2})$ are translocated by corresponding relations following equation (5) and (6), according to equation (3), the gyrocentroid $\mathbf{O}$ is given by

$$\mathbf{O} = \frac{1}{2} \otimes_c \frac{2\gamma^2_{\mathbf{e}'_h}\mathbf{e}'_h + 2\gamma^2_{\mathbf{e}'_t}\mathbf{e}' + \sum_{i=1}^{n-2} 2\gamma^2_{\mathbf{e}'_i}\mathbf{e}'_i}{(2\gamma^2_{\mathbf{e}'_h} - 1) + (2\gamma^2_{\mathbf{e}'_t} - 1) + \sum_{i=1}^{n-2}(2\gamma^2_{\mathbf{e}'_i} - 1)}. \tag{7}$$

Following the Möbius half in equation (4), the gyrocentroid $\mathbf{O}$ can be obtained. To this end, we can draw the geodesic length from a translocated entity e' (e'_h, e'_t or any e'_i) to the gyrocentroid $\mathbf{O}$ by the distance function in equation (1), i.e.,

$$d_G = \frac{2}{\sqrt{c}} \tanh^{-1}\left(\sqrt{c}\left\|-\mathbf{O}\oplus_c \mathbf{e}'\right\|\right), \tag{8}$$

based on which, we give the global plausibility score as:

$$S_{Global}(\mathcal{F}) = -d_G^2 + b_h + b_t + \sum_{i=1}^{n-2} b_i, \tag{9}$$

where b is bias coming along with entity movement. From equation (9), we can learn that a smaller d_G brings a higher global plausibility score. In binary cases, gyro-polygon degenerates to gyro-line segment, while gyrocentroid degenerates to gyromidpoint, which we term as $\mathbf{M}$, can be obtained following equation (2), i.e.,

$$\mathbf{M} = M(\mathbf{e}'_h, \mathbf{e}'_t) = \mathbf{e}'_h \oplus_c (\frac{1}{2} \otimes_c (-\mathbf{e}'_h \oplus_c \mathbf{e}'_t). \tag{10}$$

The geodesic length between $\mathbf{M}$ and a translocated entity $\mathbf{e}'$ ($\mathbf{e}'_h$ *or* $\mathbf{e}'_t$), is obtained by:

$$d_B = \frac{2}{\sqrt{c}} \tanh^{-1}\left(\sqrt{c}\left\|-\mathbf{M}\oplus_c \mathbf{e}'\right\|\right), \tag{11}$$

from which a plausibility function for binary relational facts can be obtained by:

$$S_{Binary}(\mathcal{F}) = -d_B^2 + b_h + b_t. \tag{12}$$

Since the primary triple also plays significant roles in beyond-binary fact (Rosso, Yang, and Cudré-Mauroux 2020), we further incorporate the binary score S_{Binary} into S_{Global}. The final score for fact $\mathcal{F}$ are defined as below:

$$S(\mathcal{F}) = \alpha S_{Binary}(\mathcal{F}) + \beta S_{Global}(\mathcal{F}). \tag{13}$$

In case $\mathcal{F}$ is a binary relational fact, β is set to 0. When $\beta = 0$ and $\alpha = 4$, PolygonE can generalize to Murp. When the gyro-vector operations turn to their Euclidean analogs, PolygonE can generalize to TransE or RotatE.

Training and Optimization

As a widely used method for data augmentation (Kazemi and Poole 2018), reciprocal relations (e'_t, r^{-1}, e'_h) are introduced for each binary relational fact and each primary triple in beyond-binary fact. And for each fact in training set, n_{neg} negative samples are generated by corrupting an entity domain with randomly selected entities from $\mathcal{E}$. Cross-entropy loss function are applied to optimize the model, which is given by:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log\left(p_i\right) + \left(1 - y_i\right)\log\left(1 - p_i\right)\right), \tag{14}$$

where N denotes the total number of facts in training set, y_i is a binary indicator suggesting whether a fact is genuine or not. $p_i = sigmoid(S(\mathcal{F}))$ is the predicted probability for fact $\mathcal{F}$. Parameters in PolygonE are learned via Riemannian stochastic gradient descent (RSGD Bonnabel (2013)).

Dataset	$\|\mathcal{E}\|$	Arity	$\|\mathcal{R}\|$	#Train	#Valid	#Test	#2-ary	#3-ary	#4-ary	#$\geq$ 5-ary
WikiPeople	47,765	2-9	707	305,725	38,223	38,281	337,914	25,820	15,188	3,307
JF17K	28,645	2-6	322	61,104	15,275	24,568	54,627	34,544	9,509	2,267
FB-AUTO	3,388	2,4,5	8	6,778	2,225	2,180	3,786	0	125	7,212
WN18	40,943	2	18	141,442	5,000	5,000	141,442	-	-	-
FB15K	14,951	2	1,345	484,142	50,000	59,071	141,442	-	-	-

Table 1: Statistics of Datasets

Experiments

Experimental Setup

Dataset Knowledge base completion (KBC) experiments are conducted on JF17K (Wen et al. 2016), WikiPeople (Guan et al. 2019), and FB-AUTO (Fatemi et al. 2020). Binary relational facts in the three datasets, WN18,and FB15K are tested in binary relational KBC experiments. Details of the three datasets are given in Table 1.

Metric Mean Reciprocal Rank (MRR) and Hit@k ($k = 1, 3, 10$) are utilized for evaluation following the filter setting (Bordes et al. 2013). All entity domains are evaluated.

Baseline We compare PolygonE with several strongest models, including RAE, NaLP, tNaLP+, HypE , HINGE, NeuInfer, S2S, and RAM. While StarE focuses on triple only, thus it's excluded in comparison. In binary case, we compare with several best-performing models, including TransE, SimplE, RotatE, and Tucker and RAM. Hyperbolic models Murp and ATTH are also compared.

Hyper-parameters Embedding dimensions are set to 50 for a fair comparison with RAM. Other hyper-parameters are chosen from grid search. Concretely, learning rate η is selected from $\{10, 15, 30, 50, 100\}$, batch size n_{batch} are chosen from $\{64, 128, 256\}$, number of negative samples n_{neg} are selected from $\{25, 50, 100\}$. α and β in equation (13) are integers sampled from $\{1, 2, 3, 4, 5, 6\}$. Experiments are implemented on a single NVIDIA RTX 3080 GPU.

N-ary Relational KBC Results

Overall results of n-ary relational KBC on benchmark datasets are reported in Table 2. It is noted that PolygonE achieves significant performance boost on all the three benchmark datasets in comparison with all the baselines. PolygonE surpasses S2S, RAM, and other models that treat entities as equally important, that's because PolygonE takes into consideration the validity of the primary triple in the scoring function. It is also observed that NeuInfer, NaLP, tNaLP+, and HINGE show relatively weak expressiveness. Beyond-binary relational facts in these models are neither decomposed into role-entity pairs or quadruples, which could undermine structural and semantical information within facts. Our proposed PolygonE keeps n-ary fact intact by leveraging the gyro-centroid to represent the intrinsic semantic without introducing decomposition, therefore showing casing better expressive power.

With regard to the break-down performance on single-arity data, the same observation can be obtained from Figure 4. Conclusion also can be drawn that PolygonE consistently outperforms the baseline models. It's noticed that RAM lags far behind NeuInfer on WikiPeople concerning 3-ary and 4-ary relational facts, which Liu, Yao, and Li (2021) attribute to the unbalance of the dataset (Binary relational facts are in the overwhelming majority). However, our model surpasses NeuInfer on 3-ary data and achieves competitive results on 4-ary data, showcasing strong robustness.

Effectiveness of Vertex-gyrocentroid Metric

To validate the vertex-gyrocentroid-based metric's effectiveness and explain PolygonE's advantage in modeling high-arity data, we design two variants of PolygonE, i.e., perimeterE and PolygonE(Eu). PerimeterE optimizes the gyro-polygon with perimeter, PolygonE(Eu) replaces all gyro-vector operations with corresponding Euclidean equivalences. As the results shown in Table 3, PerimeterE is largely inferior to PolygonE. One plausible explanation can be that only one entity is corrupted during training, but lengths of all edges are summed. As illustrated in Figure 3, only the two edges connected to the corrupted entity domain are influenced by the negative entity; the rest $n-2$ edges stay the same. With the addition of the rest $n-2$ edges, the gap between positive and negative sample keep the same, i.e.,

$$\begin{aligned} &P(A_{neg}B^{'}C^{'}D^{'}E^{'}) - P(A_{pos}B^{'}C^{'}D^{'}E^{'}) \\ &= (A_{neg}E^{'} + A_{neg}B^{'}) - (A_{pos}E^{'} + A_{pos}B^{'}), \end{aligned} \tag{15}$$

where $P(\cdot)$ is a function calculating perimeter. However, the divergence could be offset by the rest $n-2$ edges. PolygonE calculates the vertex-gyrocentroid geodesic, avoiding such offset brought by redundant edges. Moreover, to get the perimeter requires multiple times calculation of gyro-line segment, which also brings about decomposition.

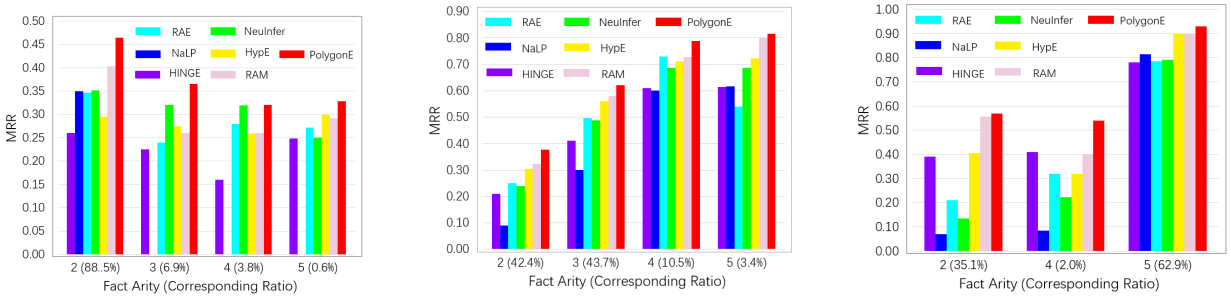
In Table 3, it's obtained that both PolygonE and PolygonE(Eu) outperform the best baseline RAM. To see the breakdown performance in Table 4, PolygonE(Eu) doesn't

	JF17K			FB-AUTO		
Model	MRR	Hit@10	Hit@1	MRR	Hit@10	Hit@1
RAM	0.539	0.690	0.463	0.830	0.876	0.803
PerimeterE	0.498	0.674	0.401	0.824	0.884	0.792
PolygonE(Eu)	<u>0.548</u>	<u>0.695</u>	<u>0.474</u>	<u>0.832</u>	<u>0.896</u>	<u>0.807</u>
PolygonE	**0.565**	**0.708**	**0.485**	**0.858**	**0.921**	**0.826**

Table 3: PolygonE and its variants' performance on JF17K and FB-AUTO.

Model	WikiPeople MRR	WikiPeople Hit@10	WikiPeople Hit@3	WikiPeople Hit@1	JF17K MRR	JF17K Hit@10	JF17K Hit@3	JF17K Hit@1	FB-AUTO MRR	FB-AUTO Hit@10	FB-AUTO Hit@3	FB-AUTO Hit@1
RAE	0.253	0.463	0.343	0.118	0.396	0.561	0.433	0.312	0.703	0.854	0.764	0.614
NaLP	0.338	0.466	0.364	0.272	0.386	0.517	0.413	0.386	0.672	0.774	0.712	0.611
tNaLP+	0.339	0.473	0.369	0.269	0.449	0.598	0.484	0.370	-	-	-	-
HINGE	0.333	0.477	0.361	0.259	0.473	0.618	0.490	0.397	0.678	0.774	0.772	0.630
NeuInfer	0.350	0.467	0.381	0.282	0.451	0.604	0.484	0.373	0.737	0.805	0.755	0.700
HyPE	0.292	0.502	0.375	0.162	0.507	0.669	0.550	0.421	0.804	0.856	0.824	0.774
S2S	0.372	0.533	0.439	0.277	0.528	0.690	0.570	0.457	-	-	-	-
RAM	0.380	0.539	0.445	0.279	0.539	0.690	0.573	0.463	0.830	0.876	0.851	0.803
PolygonE	**0.431**	**0.568**	**0.454**	**0.334**	**0.565**	**0.708**	**0.601**	**0.485**	**0.858**	**0.921**	**0.871**	**0.826**

Table 2: N-ary relational KBC results on benchmark datasets. Best results are in bold and second best are underlined. Results of HypE on FB-AUTO and results of NeuInfer, S2S, and tNaLP+ are copies from original papers. Others are copied from RAM.



(a) WikiPeople (b) JF17K (c) FB-AUTO

Figure 4: Breakdown Performance of MRR on single-arity data.

Model	JF17K 2ary 42.4%	JF17K 3ary 43.7%	JF17K 4ary 10.5%	JF17K 5ary 3.4%	FB-AUTO 2ary 35.1%	FB-AUTO 4ary 2.0%	FB-AUTO 5ary 62.9%
RAM	0.337	0.579	0.722	.801	0.557	0.400	.903
PolygonE(Eu)	0.362	**0.617**	0.735	0.722	0.562	0.404	0.901
PolygonE	**0.364**	0.612	**0.787**	**0.812**	**0.565**	**0.540**	**0.925**

Table 4: Breakdown MRR of PolygonE and PolygonE(Eu) on JF17K and FB-AUTO.

show an obvious advantage to RAM in 4-ary and 5-ary data. We believe it can surpass RAM in terms of the overall MRR mainly due to its good performance on binary and 3-ary data. PolygonE brings more performance gain to RAM than PolygonE(Eu), which can be explained by PolygonE's expressiveness for high-arity data. We note that PolygonE and PolygonE(Eu) are on par with each other in terms of binary and 3-ary facts, but PolygonE significantly outperforms PolygonE(Eu) in 4-ary and 5-ary facts. This result tallies with our illustration in Figure 1 where we show that the higher the arity, the more pronounced the exponential characteristic. PolygonE is built in hyperbolic space, therefore demonstrating a better understanding of high-arity data.

Binary Relational KBC Results

To show PolygonE's generalizability to binary relational data, we compare it with several representative binary models. Results of binary relational KBC are listed in Table 5. PolygonE brings apparent performance gain on WikiPeople and JF17K. In terms of FB-AUTO, it also vastly outperforms all baselines except for a marginal improvement compared with RAM. Concerning WN18 and FB15K, PolygonE sees competitive performance compared with the strongest baselines. And PolygonE can outperform its hyperbolic binary counterparts Murp and ATTH as well. The above analysis indicates that PolygonE generalizes well on binary data, the gyro-line segment can effectively model binary facts.

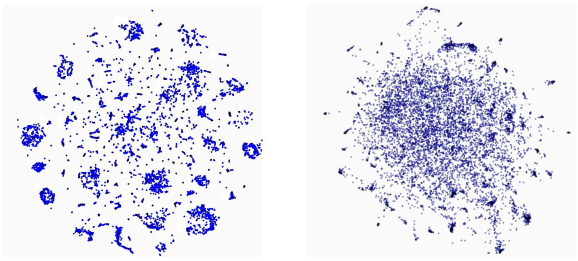
Awareness of Hierarchical Anatomy

To help comprehend hierarchy anatomy, the embeddings of entities in WikiPeople are shown in Figure 5. From Figure 5(a), it can be intuitively observed that in PolygonE, more entities are prone to lying near the boundary of the ball, which is in good coherence with the exponential length growth in Poincaré ball (Figure 1(b)). While in RAM, entities distribute more randomly, and no obvious geometry can be learned. The sparsity in the center and density in the boundary suggest PolygonE well captures the hierarchy structure.

To further show the hierarchies, we calculate the distance to origin and accumulate degrees for all entities in FB-AUTO. Comparison between PolygonE and RAM is shown in Figure 6 where a scatter means an entity. Generally, only a tiny number of entities lie at the top hierarchical level

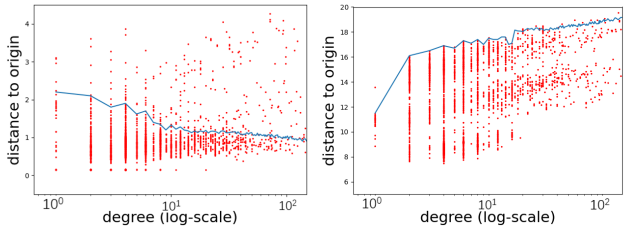
Model	WikiPeople			JF17K			FB-AUTO			WN18			FB15K		
	MRR	Hit@10	Hit@1	MRR	Hit@10	Hit@1	MRR	Hit@10	Hit@1	MRR	Hit@10	Hit@1	MRR	Hit@10	Hit@1
TransE	0.312	0.574	0.146	0.276	0.495	0.167	0.313	0.562	0.132	0.495	0.943	0.113	0.463	0.749	0.297
SimplE	0.326	0.449	0.249	0.333	0.512	0.244	0.510	0.621	0.450	0.942	0.947	0.939	0.727	0.838	0.660
RotatE	0.422	0.519	0.285	0.304	0.496	0.244	0.470	0.577	0.408	0.949	0.959	0.944	0.797	0.884	0.746
Tucker	0.429	0.538	0.365	0.333	0.512	0.244	0.510	0.621	0.450	0.953	0.958	0.949	0.795	0.892	0.741
RAM	0.408	0.568	0.303	0.337	0.523	0.246	0.557	0.649	0.507	0.947	0.952	0.943	0.803	0.882	0.756
GETD	-	-	-	-	-	-	-	-	-	0.948	0.954	0.944	0.824	0.888	0.787
S2S	-	-	-	-	-	-	-	-	-	**0.955**	**0.963**	**0.949**	**0.850**	**0.910**	0.820
Murp	0.451	0.577	0.388	0.348	0.550	0.261	0.546	0.644	0.490	0.945	0.952	0.940	0.803	0.885	0.720
ATTH	0.366	0.557	0.259	0.327	0.528	0.244	0.562	0.651	**0.519**	0.942	0.953	0.935	0.806	0.883	0.722
PolygonE	**0.462**	**0.596**	**0.401**	**0.364**	**0.578**	**0.279**	**0.565**	**0.678**	0.512	0.953	0.956	0.949	0.824	0.891	0.783

Table 5: Binary relational KBC results on benchmark datasets. Best results are in boldface and second best are underlined.



(a) Embedding of PolygonE (b) Embedding of RAM

Figure 5: Embeddings of WikiPeople learned by PolygonE and RAM. T-SNE is employed to realize dimension reduction. Then entities are visualized according to the distance to the origin. We mark some clusters in (a).



(a) PolygonE (b) RAM

Figure 6: Scatter plot of entity-origin distance v.s. node degrees on FB-AUTO. The x-coordinate stands for entity degree, y-coordinate denotes the distance between entity embedding and the origin. The curve shows the tendency that distance to origin changes with degrees.

in a hierarchy structure. And these high-hierarchical entities are involved in more facts, i.e., they are few in number, high in degree. PolygonE can capture such hierarchy; it is obtained from Figure 6(a), high-degree entities are fewer in number and tend to lie closer to the root, while low-degree entities tend to be far from the origin. In Figure 6(b), entity-origin length is positively correlated with entity degree, which means RAM does not well learn the hierarchical structure.

Translocation	MRR	Hit@10	Hit@3	Hit@1
T(all)	.791	.901	.831	.731
R(all)	.721	.850	.792	.685
T(tail) R(rest)	.741	.860	.804	.712
T(head) R(rest)	.752	.873	.812	.735
R(tail) T(rest)	.837	.905	.850	.796
R(head) T(rest)	**.858**	**.921**	**.871**	**.826**

Table 6: Influence of translocations on FB-AUTO, T means translation, R stands for rotation.

Influence of different translocations

Translation and rotation are the two basic movements in the physical world, in this part, we explore the influence of different translation and rotation operations on each entity on FB-AUTO dataset. As shown in Table 6, rotating the head and translating the rest performs the best. PolygonE adopts this setting in experiments. Translating all entities, rotating all entities, or other settings will all bring performance degradation. Rotating all brings the most significant performance drop. We study other parameters in the Appendix.

Conclusion

In this paper, we propose an n-ary relational KB embedding method named PolygonE, where we embed n-ary facts as gyro-polygons in hyperbolic Poincaré ball. Gyro-polygon helps retain structural and semantical information. Moreover, we devise the vertex-gyrocentoid optimization goal, which is effective for fact plausibility measuring. Our approach significantly surpasses current SOTA methods, generalizes well to binary data, and captures the hierarchy anatomy. We will explicitly explore entity type in future work.

References

Balazevic, I.; Allen, C.; and Hospedales, T. M. 2019a. Multi-relational Poincaré Graph Embeddings. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 4465–4475.

Balazevic, I.; Allen, C.; and Hospedales, T. M. 2019b. TuckER: Tensor Factorization for Knowledge Graph Completion. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 5184–5193. Association for Computational Linguistics.

Bonnabel, S. 2013. Stochastic Gradient Descent on Riemannian Manifolds. *IEEE Trans. Autom. Control.*, 58(9): 2217–2229.

Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-relational Data. In Burges, C. J. C.; Bottou, L.; Ghahramani, Z.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 2787–2795.

Chami, I.; Wolf, A.; Juan, D.; Sala, F.; Ravi, S.; and Ré, C. 2020. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 6901–6914. Association for Computational Linguistics.

Chen, W.; Han, X.; Lin, Y.; Zhao, H.; Liu, Z.; Li, P.; Sun, M.; and Zhou, J. 2021. Fully Hyperbolic Neural Networks. *CoRR*, abs/2105.14686.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2D Knowledge Graph Embeddings. In McIlraith, S. A.; and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 1811–1818. AAAI Press.

Di, S.; Yao, Q.; and Chen, L. 2021. Searching to Sparsify Tensor Decomposition for N-ary Relational Data. In Leskovec, J.; Grobelnik, M.; Najork, M.; Tang, J.; and Zia, L., eds., *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, 4043–4054. ACM / IW3C2.

Fatemi, B.; Taslakian, P.; Vázquez, D.; and Poole, D. 2020. Knowledge Hypergraphs: Prediction Beyond Binary Relations. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 2191–2197. ijcai.org.

Galkin, M.; Trivedi, P.; Maheshwari, G.; Usbeck, R.; and Lehmann, J. 2020. Message Passing for Hyper-Relational Knowledge Graphs. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 7346–7359. Association for Computational Linguistics.

Ganea, O.; Bécigneul, G.; and Hofmann, T. 2018. Hyperbolic Neural Networks. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 5350–5360.

Guan, S.; Jin, X.; Guo, J.; Wang, Y.; and Cheng, X. 2020. NeuInfer: Knowledge Inference on N-ary Facts. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 6141–6151. Association for Computational Linguistics.

Guan, S.; Jin, X.; Guo, J.; Wang, Y.; and Cheng, X. 2021. Link Prediction on N-ary Relational Data Based on Relatedness Evaluation. *IEEE Transactions on Knowledge and Data Engineering*, PP(99): 1–1.

Guan, S.; Jin, X.; Wang, Y.; and Cheng, X. 2019. Link Prediction on N-ary Relational Data. In Liu, L.; White, R. W.; Mantrach, A.; Silvestri, F.; McAuley, J. J.; Baeza-Yates, R.; and Zia, L., eds., *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 583–593. ACM.

Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; de Melo, G.; Gutiérrez, C.; Kirrane, S.; Gayo, J. E. L.; Navigli, R.; Neumaier, S.; Ngomo, A. N.; Polleres, A.; Rashid, S. M.; Rula, A.; Schmelzeisen, L.; Sequeda, J. F.; Staab, S.; and Zimmermann, A. 2021. Knowledge Graphs. *ACM Comput. Surv.*, 54(4): 71:1–71:37.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Yu, P. S. 2021. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *IEEE Transactions on Neural Networks and Learning Systems*, 1–21.

Kazemi, S. M.; and Poole, D. 2018. SimplE Embedding for Link Prediction in Knowledge Graphs. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 4289–4300.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Bonet, B.; and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, 2181–2187. AAAI Press.

Liu, Y.; Yao, Q.; and Li, Y. 2020. Generalizing Tensor Decomposition for N-ary Relational Knowledge Bases. In Huang, Y.; King, I.; Liu, T.; and van Steen, M., eds., *WWW*

'20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020, 1104–1114. ACM / IW3C2.

Liu, Y.; Yao, Q.; and Li, Y. 2021. Role-Aware Modeling for N-ary Relational Knowledge Bases. In Leskovec, J.; Grobelnik, M.; Najork, M.; Tang, J.; and Zia, L., eds., *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, 2660–2671. ACM / IW3C2.

Nguyen, D. Q.; Vu, T.; Nguyen, T. D.; and Phung, D. 2020. QuatRE: Relation-Aware Quaternions for Knowledge Graph Embeddings. *CoRR*, abs/2009.12517.

Nickel, M.; Tresp, V.; and Kriegel, H. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. In Getoor, L.; and Scheffer, T., eds., *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, 809–816. Omnipress.

Rosso, P.; Yang, D.; and Cudré-Mauroux, P. 2020. Beyond Triplets: Hyper-Relational Knowledge Graph Embedding for Link Prediction. In Huang, Y.; King, I.; Liu, T.; and van Steen, M., eds., *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, 1885–1896. ACM / IW3C2.

Rouces, J.; de Melo, G.; and Hose, K. 2015. FrameBase: Representing N-Ary Relations Using Semantic Frames. In Gandon, F.; Sabou, M.; Sack, H.; d'Amato, C.; Cudré-Mauroux, P.; and Zimmermann, A., eds., *The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015. Proceedings*, volume 9088 of *Lecture Notes in Computer Science*, 505–521. Springer.

Schlichtkrull, M. S.; Kipf, T. N.; Bloem, P.; van den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling Relational Data with Graph Convolutional Networks. In Gangemi, A.; Navigli, R.; Vidal, M.; Hitzler, P.; Troncy, R.; Hollink, L.; Tordai, A.; and Alam, M., eds., *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, 593–607. Springer.

Sun, Z.; Deng, Z.; Nie, J.; and Tang, J. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex Embeddings for Simple Link Prediction. In Balcan, M.; and Weinberger, K. Q., eds., *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, 2071–2080. JMLR.org.

Ungar, A. A. 2009. *A Gyrovector Space Approach to Hyperbolic Geometry*. Synthesis Lectures on Mathematics & Statistics. Morgan & Claypool Publishers.

Vashishth, S.; Sanyal, S.; Nitin, V.; and Talukdar, P. P. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wang, Q.; Mao, Z.; Wang, B.; and Guo, L. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.*, 29(12): 2724–2743.

Wang, Q.; Wang, H.; Lyu, Y.; and Zhu, Y. 2021a. Link Prediction on N-ary Relational Facts: A Graph-based Approach. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, 396–407. Association for Computational Linguistics.

Wang, S.; Wei, X.; dos Santos, C. N.; Wang, Z.; Nallapati, R.; Arnold, A. O.; Xiang, B.; Yu, P. S.; and Cruz, I. F. 2021b. Mixed-Curvature Multi-Relational Graph Neural Network for Knowledge Graph Completion. In Leskovec, J.; Grobelnik, M.; Najork, M.; Tang, J.; and Zia, L., eds., *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, 1761–1771. ACM / IW3C2.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In Brodley, C. E.; and Stone, P., eds., *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, 1112–1119. AAAI Press.

Wen, J.; Li, J.; Mao, Y.; Chen, S.; and Zhang, R. 2016. On the Representation and Embedding of Knowledge Bases beyond Binary Relations. In Kambhampati, S., ed., *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, 1300–1307. IJCAI/AAAI Press.

Yang, B.; Yih, W.; He, X.; Gao, J.; and Deng, L. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zhang, R.; Li, J.; Mei, J.; and Mao, Y. 2018. Scalable Instance Reconstruction in Knowledge Bases via Relatedness Affiliated Embedding. In Champin, P.; Gandon, F.; Lalmas, M.; and Ipeirotis, P. G., eds., *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, 1185–1194. ACM.

Zhang, S.; Tay, Y.; Yao, L.; and Liu, Q. 2019. Quaternion Knowledge Graph Embeddings. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2731–2741.

Appendices

A Gyro-vector Space

Algebraic operations such as addition and scalar product which are straightforward in the Euclidean space cannot be directly applied in hyperbolic space. While Gyrovector

Operations	Symbols	Formalizations	Descriptions
Möbius addition	$\oplus_c$	$\mathbf{x}\oplus_c\mathbf{y}=\frac{\left(1+2c\langle\mathbf{x},\mathbf{y}\rangle+c\|\mathbf{y}\|^2\right)\mathbf{x}+\left(1-c\|\mathbf{x}\|^2\right)\mathbf{y}}{1+2c\langle\mathbf{x},\mathbf{y}\rangle+c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}$	Vector Addition
Exponential map	$\exp_{\mathbf{x}}^c(\cdot)$	$\exp_{\mathbf{x}}^c(\mathbf{v})=\mathbf{x}\oplus_c\left(\tanh\left(\sqrt{c}\frac{\lambda_{\mathbf{x}}^c\|\mathbf{v}\|}{2}\right)\frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}\right)$	Map from Euclidean to Hyperbilic space
Logarithmic map	$\log_{\mathbf{x}}^c(\cdot)$	$\log_{\mathbf{x}}^c(\mathbf{v})=\frac{2}{\sqrt{c}\lambda_{\mathbf{x}}^c}\tanh^{-1}\left(\sqrt{c}\left\|-\mathbf{x}\oplus_c\mathbf{v}\right\|\right)\frac{-\mathbf{x}\oplus_c\mathbf{v}}{\|-\mathbf{x}\oplus_c\mathbf{v}\|}$	Map from Hyperbolic to Euclidean space
Matrix-vector product	$\otimes_c$	$\mathbf{M}\otimes_c\mathbf{x}=\exp_0^c\left(\mathbf{M}log_0^c(\mathbf{x})\right)$	Matrix-vector multiplication
Möbius half	$\otimes_c$	$\frac{1}{2}\otimes_c\mathbf{x}=\frac{\gamma_{\mathbf{x}}}{1+\gamma_{\mathbf{x}}}\mathbf{x},\gamma_{\mathbf{x}}=\sqrt{\frac{1}{1-\|\mathbf{x}\|^2}}$	$\gamma_{\mathbf{x}}$: gamma factor in relativity theory
Gyromidpoint	M	$M(\mathbf{x},\mathbf{y})=\mathbf{x}\oplus_c\left(\frac{1}{2}\otimes_c\left(-\mathbf{x}\oplus_c\mathbf{y}\right)\right.$	midpoint for gyro-line segment
Gyrocentroid	O	$O(e_1,...e_n)=\frac{1}{2}\otimes_c\frac{\sum_{i=1}^n 2\gamma_{e_i}^2 e_i}{\sum_{i=1}^n(2\gamma_{e_i}^2-1)}$	Centroid of gyro-polygon
Distance function	$d_{\mathbb{H}}(\cdot,\cdot)$	$d_{\mathbb{H}}(\mathbf{x},\mathbf{y})=\frac{2}{\sqrt{c}}\tanh^{-1}\left(\sqrt{c}\left\|-\mathbf{x}\oplus_c\mathbf{y}\right\|\right).$	Distance between gyro-vectors

Table 7: Summary of Operations in gyro-vector space

spaces allow for the formalization of these operations in hyperbolic space. We list some basic operations in the gyro-vector space in Table 7.

B Hyperparameter Study

This section will study the influence brought by dimensions and α, β in the scoring function. The optimal settings are reported in Table 8.

	WikiPeople	JF17K	FB-AUTO
batch size	128	128	128
learning rate	30	30	10
negative sample	100	100	50
dimension	50	50	50
α	4	4	4
β	4	4	6

Table 8: Optimal Hyperparameters.

Dimensions

As we can obtain from Figure 7, increasing dimension from 20 to 50 brings noticeable performance gain. While when the dimension is over 50, it's observed that MRR grows slowly or fluctuates. PolygonE can achieve good performance with very low dimension, it surpasses the baseline RAM with 20 dims. In main context, for a fair comparison with RAM, the dimension is set to 50. Actually, with 200 dims or 500 dims, PolygonE can achieve even better results than reported.

α, β in scoring function

When $\beta = 0$, given 2 points e_1,e_2, and the gyro-midpoint O. Murp outputs score as $s_1 = -d^2(e_1, e_2) + b_1 + b_2$. while $s_2 = -\alpha * d^2(e_1, O) + b_1 + b_2$ is the score for PolygonE. $d(e_1, e_2) = 2 * d(e_1, O)$, $\alpha = 4 \rightarrow s_1 = s_2$, then PolygonE generalizes to Murp.

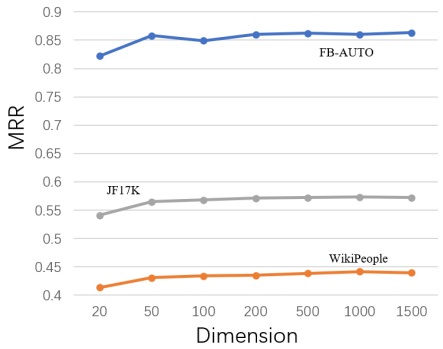


Figure 7: MRR of PolygonE with different dimensions.

For fair comparison with Murp, α is set to 4 in experiments. Under this setting, we explore PolyonE's performance on the three n-ary relational KBs. According to results in Figure 8, the best α for FB-AUTO, JF17K, and WikiPeople is set to 4, 4, 6, respectively.

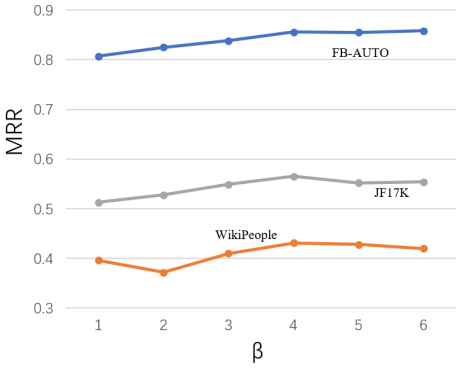


Figure 8: MRR of PolygonE with different β in scoring function.