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

Shiyao Yan\textsuperscript{1,2,3}, Zequn Zhang\textsuperscript{*1,2}, Xian Sun\textsuperscript{1,2}, Guanglun Xu\textsuperscript{1,2}, Shuchao Li\textsuperscript{1,2}, Qing Liu\textsuperscript{1,2}, Nayu Liu\textsuperscript{1,2,3}, Shensi Wang\textsuperscript{1,2,3}

\textsuperscript{1}Aerospace Information Research Institute, Chinese Academy of Sciences
\textsuperscript{2}Key Laboratory of Network Information System Technology(NIST), Aerospace Information Research Institute
\textsuperscript{3}School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences

\texttt{yan_shiyao19@mails.ucas.ac.cn, zqzhang1@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 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 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 semantical 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

\textsuperscript{*}Corresponding Author.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
the vertex-gyrocentroid geodesic to evaluate whether an entity is compatible with the whole fact, which is lever-
ageed to optimize the relation-adjusted gyro-polynomials.

- 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 meth-
ods mainly fall into 3 families. Translational models (Bordes et al. 2013; Wang et al. 2014; Lin et al. 2015) typi-
cally 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 quater-

nions, while Balazevic, Allen, and Hospedales (2019b) in-
troduce tucker decomposition, Kazemi and Poole (2018) in-
troduce 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 func-
tions, 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) em-
beds 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, tNalp+ (Guan et al. 2021) further considers type information. Neulnfer (Guan et al. 2020) and HINE (Rosso, Yang, and Cudré-Mauroux 2020) tear apart n-ary facts into a primary triple and several entity-role pairs, of which the compati-
bility 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 se-

matic. 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 \( \{r, e_1, e_2, \ldots, e_n\} \) (Wen et al. 2016), role-
value pairs \( \{e_1, r_1, \ldots, e_i, r_i, \ldots, e_n, r_n\} \) (Guan et al. 2019), and triple+ pairs \( \{h, r, i_1, e_1, \ldots, r_{n-2}, e_{n-2}\} \) (Rosso, Yang, and Cudré-Mauroux 2020). We follow the third one. Formally, given an n-ary relational KB \( B \) with a set of enti-
ties \( E \) and a set of relations \( R \), we represent a fact \( F \) in \( B \) in the form of \( F : (e_h, r, e_i, \ldots, r_{n-2}, e_{n-2}) \), where \( r \in R, e_i \in E \), and \( n \geq 2 \). \( n \) indicates the number of enti-
ties participating in the tuple $F$. In case $n = 2$, $F$ is a binary fact. If $n > 2$, $F$ 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 $\mathbb{H}^d := \{ x \in \mathbb{R}^d : \|x\|^2 < 1 \}$ accompanied with a Riemannian metric $g^\mathbb{H}$, where $g^\mathbb{H} = (\lambda_x)^2 g^\mathbb{E}$ and $\lambda_x = \frac{2}{1 - \|x\|^2}$. $g^\mathbb{E} = \mathbb{I}_d$ and $\|x\|^2$ are the Euclidean identity metric tensor and the Euclidean norm, respectively, while $\lambda_x$ 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 $(\exp_c)$, logarithmic map $(\log_c)$, matrix-vector product $(\otimes_c)$, and Möbius half in Appendix A. Moreover, the distance between two points $x, y$ in a Poincaré ball is given by:

$$d_2(x, y) = \frac{2}{\sqrt{c}} \tanh^{-1} \left( \sqrt{c} \|x \oplus_c y\| \right),$$

(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), ABC is a gyro-trigon. In Figure 2(b), 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 $x, y$ is given by:

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

(2)

while the gyro-centroid of the gyro-polygon with $e_1, \ldots, e_n$ being the vertexes is given by (Ungar 2009):

$$O(e_1, ... e_n) = \frac{1}{2} \otimes_c \sum_{i=1}^{n} 2\gamma_x^2 e_i \frac{2\gamma_x^2 e_i}{\sum_{i=1}^{n} (2\gamma_x^2 e_i - 1)},$$

(3)

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

$$\frac{1}{2} \otimes_c x = \gamma_x \frac{\|x\|^2}{1 + \gamma_x} x.$$

(4)

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

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 $\mathcal{F} : (e_h, r, e_t, r_1, \ldots, r_n, e_{n-2})$ are first initialized as random d-dimensional vectors $e_h, e_t, \ldots, e_{n-2}$ in a gyro-polygon. As illustrated in Figure 3, the vertexes of the gyro-polygon 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 $A$ is transferred to $A_{pos}$ by a relation $r$. 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 $e_h$ is rotated by a relation-specific matrix $R \in \mathbb{R}^{d \times d}$, while tail entity $e_t$ is translated by a relation $r \in \mathbb{R}^d$, i.e.,

$$e'_h = R \otimes_c e_h, \quad e'_t = e_t \oplus_c r.$$

(5)

The rest entities $e_i, (i \in [0, n-2])$ are translocated by relation vector $r_i$:

$$e'_i = e_i \oplus_c r_i.$$  

(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.
Following the Möbius half in equation (4), the gyrocentroid $O$ can be obtained. To this end, we can draw the geodesic length from a translocated entity $e'$ ($e'_1, e'_2, \ldots, e'_{n-2}$) to the gyrocentroid $O$ by the distance function in equation (1), i.e.,

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

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

$$S_{Global}(F) = -d_G^2 + b_h + b_t + \sum_{i=1}^{n-2} b_i,$$  \hfill (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 $M$, can be obtained following equation (2), i.e.,

$$M = M(e'_h, e'_i) = e'_h \oplus_c \frac{1}{2} \oplus_c (-e'_h \oplus_c e'_i).$$  \hfill (10)

The geodesic length between $M$ and a translocated entity $e'$ ($e'_h$ or $e'_i$), is obtained by:

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

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

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

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

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

In case $F$ is a binary relational fact, $\beta$ 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 ($e_i, r^{-1}, e_i'$) 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:

$$L = -\frac{1}{N} \sum_{i=1}^{N} (y_i \log (p_i) + (1 - y_i) \log (1 - p_i)),$$  \hfill (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. $p_i = \text{sigmoid}(S(F))$ is the predicted probability for fact $F$. Parameters in PolygonE are learned via Riemannian stochastic gradient descent (RSGD Bonnabel (2013)).
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.

**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, SimpE, 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 nbatch are chosen from {64, 128, 256}, number of negative samples nneg 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 semantic 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 gyropolygon with perimeter, PolygonE(Eu) replaces all gyrovector 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.,

\[
P(A_{neg}B'C'D'E') - P(A_{pos}B'C'D'E') = (A_{neg}E' + A_{neg}B') - (A_{pos}E' + A_{pos}B'),
\]

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 gyroline 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 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 char-

| Dataset     | \(|E|\) | Arity | \(|R|\) | #Train | #Valid | #Test | #2-ary | #3-ary | #4-ary | #≥ 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
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 copied from original papers. Others are copied from RAM.

| Model   | WikiPeople |     |     |     | JF17K |     |     |     | FB-AUTO |     |     |     |
|---------|------------|-----|-----|-----|-------|-----|-----|-----|---------|-----|-----|-----|
|         | MRR | Hit@10 | Hit@3 | Hit@1 | MRR | Hit@10 | Hit@3 | Hit@1 | MRR | Hit@10 | Hit@3 | Hit@1 |
| RAE     | .253 | .463 | .343 | .118 | .396 | .561 | .433 | .312 | .703 | .854 | .764 | .614 |
| NaLP    | .338 | .466 | .364 | .272 | .386 | .517 | .413 | .386 | .672 | .774 | .712 | .611 |
| tNaLP+  | .339 | .473 | .369 | .269 | .449 | .598 | .484 | .370 | -   | -    | -    | -    |
| HINGE   | .333 | .477 | .361 | .259 | .473 | .618 | .490 | .397 | .678 | .774 | .772 | .630 |
| NeuInfe | .350 | .467 | .381 | .282 | .451 | .604 | .484 | .373 | .737 | .805 | .755 | .700 |
| HyPE    | .292 | .502 | .375 | .162 | .507 | .669 | .550 | .421 | .804 | .856 | .824 | .774 |
| S2S     | .372 | .533 | .439 | .277 | .528 | .690 | .570 | .457 | -   | -    | -    | -    |
| RAM     | .380 | .539 | .445 | .279 | .539 | .690 | .573 | .463 | .830 | .876 | .851 | .803 |

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

| Model   | JF17K | FB-AUTO |
|---------|-------|---------|
|         | MRR   | Hit@10 | Hit@1 |
| RAM     | .539  | .690   | .463  |
| PerimeterE | .498  | .674   | .401  |
| PolygonE(Eu) | .548  | .695   | .474  |
| PolygonE | .565  | .708   | .485  |
|         | MRR   | Hit@10 | Hit@1 |
| RAM     | .830  | .876   | .803  |
| PerimeterE | .824  | .884   | .792  |
| PolygonE(Eu) | .832  | .896   | .807  |
| PolygonE | .858  | .921   | .826  |

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

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 in a hierarchy structure. And these high-hierarchical enti-
Table 5: Binary relational KBC results on benchmark datasets. Best results are in boldface and second best are underlined.

| 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 0.882 0.756 |
| 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   | 0.167 | 0.313 0.562 | 0.132 | 0.495 | 0.943 | 0.113 | 0.463 | 0.749 | 0.297 |
| S2S    | 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 |
| Murp   | 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 |
| ATTH   | 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 6: Influence of translocations on FB-AUTO. T means translation, R stands for rotation.

| 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)| **.888** | **.921** | **.871** | **.826** |

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.
| Operation                  | Symbol | Formalization                                                                 | Description                                                                 |
|---------------------------|--------|-------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Möbius addition           | ⊕      | $x \oplus_c y = \frac{(1+2c(x,y)+\|y\|^2)x+(1-c\|x\|^2)y}{1+2c(x,y)+c^2\|x\|^2\|y\|^2}$ | Vector Addition                                                            |
| Exponential map           | exp    | $\exp_{x}(\cdot) = x \oplus_c (\tanh \left( \frac{\sqrt{c} \| \cdot \| \| y \| \}}{2} \right) \frac{\cdot}{\sqrt{c} \| y \|})$ | Map from Euclidean to Hyperbolic space                                     |
| Logarithmic map           | log    | $\log_{x}(\cdot) = \frac{2}{\sqrt{c}} \gamma \cdot \tanh^{-1} \left( \frac{\sqrt{c} \| -x \oplus_c \cdot \| \}}{2} \right) \frac{-x \oplus_c \cdot}{\| -x \oplus_c \cdot \|}$ | Map from Hyperbolic to Euclidean space                                    |
| Matrix-vector product     | ⊗      | $M \otimes_c x = \exp_{M} (M \log_{x}(x))$                                | Matrix-vector multiplication                                               |
| Möbius half               | ⊕      | $\frac{1}{2} \oplus_c x = \frac{2x_0 - x \cdot x_0}{1+\|x\|^2}$, $\gamma_x = \sqrt{\frac{1}{1-\|x\|^2}}$ | $\gamma_x$: gamma factor in relativity theory                              |
| Gyromidpoint              | $O$    | $O(e_1, \ldots, e_n) = \frac{1}{2} \oplus_c \frac{\sum_{i=1}^{n} 2c^{\gamma_i} e_i}{\sum_{i=1}^{n} (2c^{\gamma_i} - 1)}$ | Centroid for gyro-polygon                                                 |
| Distance function         | $d_{\oplus}(\cdot, \cdot)$ | $d_{\oplus}(x, y) = \frac{2}{\sqrt{c}} \tanh^{-1} \left( \frac{\sqrt{c} \| -x \oplus_c y \| \}}{2} \right)$. | Distance between gyro-vectors                                               |

Table 7: Summary of Operations in gyro-vector space

### Appendices

#### A Gyrovector 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 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 $\alpha, \beta$ in the scoring function. The optimal settings are reported in Table 8.

| Operation                  | Symbol | Formalization                                                                 | Description                                                                 |
|---------------------------|--------|-------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| Möbius addition           | ⊕      | $x \oplus_c y = \frac{(1+2c(x,y)+\|y\|^2)x+(1-c\|x\|^2)y}{1+2c(x,y)+c^2\|x\|^2\|y\|^2}$ | Vector Addition                                                            |
| Exponential map           | exp    | $\exp_{x}(\cdot) = x \oplus_c (\tanh \left( \frac{\sqrt{c} \| \cdot \| \| y \| \}}{2} \right) \frac{\cdot}{\sqrt{c} \| y \|})$ | Map from Euclidean to Hyperbolic space                                     |
| Logarithmic map           | log    | $\log_{x}(\cdot) = \frac{2}{\sqrt{c}} \gamma \cdot \tanh^{-1} \left( \frac{\sqrt{c} \| -x \oplus_c \cdot \| \}}{2} \right) \frac{-x \oplus_c \cdot}{\| -x \oplus_c \cdot \|}$ | Map from Hyperbolic to Euclidean space                                    |
| Matrix-vector product     | ⊗      | $M \otimes_c x = \exp_{M} (M \log_{x}(x))$                                | Matrix-vector multiplication                                               |
| Möbius half               | ⊕      | $\frac{1}{2} \oplus_c x = \frac{2x_0 - x \cdot x_0}{1+\|x\|^2}$, $\gamma_x = \sqrt{\frac{1}{1-\|x\|^2}}$ | $\gamma_x$: gamma factor in relativity theory                              |
| Gyromidpoint              | $O$    | $O(e_1, \ldots, e_n) = \frac{1}{2} \oplus_c \frac{\sum_{i=1}^{n} 2c^{\gamma_i} e_i}{\sum_{i=1}^{n} (2c^{\gamma_i} - 1)}$ | Centroid for gyro-polygon                                                 |
| Distance function         | $d_{\oplus}(\cdot, \cdot)$ | $d_{\oplus}(x, y) = \frac{2}{\sqrt{c}} \tanh^{-1} \left( \frac{\sqrt{c} \| -x \oplus_c y \| \}}{2} \right)$. | Distance between gyro-vectors                                               |

Table 7: Summary of Operations in gyro-vector space

#### Dimensions

As we can obtain from Figure 7, increasing dimension from 20 to 50 or to 100 brings noticeable performance gain. PolygonE can achieve good performance with very low dimension, it surpasses the baseline RAM with 20 dims. To compromise between time and performance, the dimension is set to 100 in main context. Actually, with 200 dims or 500 dims, PolygonE may achieve even better results than reported.

$\alpha, \beta$ 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 $s_2 = -\alpha \cdot d^2(e_1, O) + b_1 + b_2$ is the score for PolygonE.

Figure 7: MRR of PolygonE with different dimensions.

d$(e_1, e_2) = 2 \cdot d(e_1, O)$, $\alpha = 4 \rightarrow s_1 = s_2$, then PolygonE generalizes to Murp.

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

Figure 8: MRR of PolygonE with different $\beta$ in scoring function.
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. In Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; and Turlach, B., eds., Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing, EMNLP 2012, Edinburgh, Scotland, October 30-November 2, 2012, 1471–1479. Association for Computational Linguistics.

Bordes, A.; Usunier, N.; Garcia-Duran, 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, 6141–6151. 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 McIraith, 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 Sparser 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.; Bégineul, 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.

Guo, J.; Wu, X.; Wang, Y.; and Cheng, X. 2020. NeulInfer: 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.

Guo, J.; 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

International Joint Conference on Artificial Intelligence, IJCAI 2020, 2191–2197. ijcai.org.
