Arbitrary Time Information Modeling via Polynomial Approximation for Temporal Knowledge Graph Embedding

Zhiyu Fang, Jingyan Qin*, Xiaobin Zhu, Chun Yang, Xu-Cheng Yin
School of Computer & Communication Engineering
University of Science and Technology Beijing, Beijing, China
{mr.fangzy, qinjingyanking}@foxmail.com, {zhuxiaobin, chunyang, xuchengyin}@ustb.edu.cn

Abstract

Distinguished from traditional knowledge graphs (KGs), temporal knowledge graphs (TKGs) must explore and reason over temporally evolving facts adequately. However, existing TKG approaches still face two main challenges, i.e., the limited capability to model arbitrary timestamps continuously and the lack of rich inference patterns under temporal constraints. In this paper, we propose an innovative TKGE method (PTBox) via polynomial decomposition-based temporal representation and box embedding-based entity representation to tackle the above-mentioned problems. Specifically, we decompose time information by polynomials and then enhance the model's capability to represent arbitrary timestamps flexibly by incorporating the learnable temporal basis tensor. In addition, we model every entity as a hyperrectangle box and define each relation as a transformation on the head and tail entity boxes. The entity boxes can capture complex geometric structures and learn robust representations, improving the model's inductive capability for rich inference patterns. Theoretically, our PTBox can encode arbitrary time information or even unseen timestamps while capturing rich inference patterns and higher-arity relations of the knowledge base. Extensive experiments on real-world datasets demonstrate the effectiveness of our method.

Keywords: temporal knowledge graph, polynomial decomposition of time, probabilistic box embedding

1. Introduction

Knowledge Graphs (KGs) are widely used in question answering, information retrieval, and recommender systems by representing human-summarized knowledge via multi-relational graphs (Hu et al., 2022). KGs usually can be viewed as a collection of facts in triple form (h, r, t), which represent head entity h is related to tail entity t by relation r. However, many facts in the real-world are time-sensitive, making the triples of (static) KGs cannot describe the dynamic evolution of facts over time. For example, *the president of the USA is Barack Obama only for the period 2009-2017 and is Donald Trump only for the period 2017-2021.* Therefore, Temporal Knowledge Graphs (TKGs), which introduce the timestamp to expand the fact into a quadruple form (h, r, t, τ), have recently drawn growing attention from both academic and industrial communities.

To effectively represent temporal information and construct a complete knowledge graph, Temporal Knowledge Graph Embedding (TKGE) methods usually learn low-dimensional representations of entities and relations under temporal constraints and predict missing triple links. A popular strategy in TKGE directly treats time information as the feature equivalent to entities or relations, transforming various existing KGE methods into TKGE. TComplex (Lacroix et al., 2020) expands the entity-relation third-order tensor of ComplEx (Trouillon et al., 2016) into an entity-relation-time fourth-order tensor via canonical polyadic decomposition. Under the setting of the original KGE model, these KGE-based extension methods simply model time information into entities or relations and cannot effectively model temporal characteristics. To capture rich time information, another strategy of TKGE has been proposed. These methods typically exhibit good temporal generalization due to clever network architectures crafted for integrating time information. HyTE (Dasgupta et al., 2018) proposes hyperplane-based TKG embedding to view the TKG as a collection of KGs embedded within different temporal hyperplanes. ATISE (Xu et al., 2020b) utilizes the theory of additive time series decomposition to represent entities and relations as time series features that can be decomposed into trend, seasonal, and random components. DyERNIE (Han et al., 2020) defines the interaction between velocity vectors and times in the tangent space to describe the dynamic evolution of entities. Although these methods consider the impact of temporal characteristics on entity-relation pairs, they lack the capability to model arbitrary timestamps continuously.

Generally, existing TKGE methods use simple embedding vectors and the parallelogram rule in Euclidean space for feature representation, capturing limited structural information for the knowledge graph. This results in the restricted reasoning capability of the model, especially for inference patterns under temporal constraints. TTransE (Leblay and Chekol, 2018) employs the parallelogram rule in Euclidean space to measure the score of entity-

*Corresponding author

LREC-COLING 2024, pages 1455–1465

relation pairs, which makes it unable to capture symmetry pattern. RotateQVS (Chen et al., 2022) defines the rotation of the entity around the time axis in quaternion vector space and calculates the score via the distance of the complex vectors, causing it cannot capture hierarchical pattern.

To address the above problems, we propose an innovative TKGE method based on polynomial approximation, namely PTBox[1]. Our method comprises two modules: polynomial decomposition-based temporal representation and box embedding-based entity representation. Specifically, we leverage the polynomial approximation theory to decompose the embedding of any given time point into a product of a coefficient vector and a learnable feature tensor. In this way, our approach allows easily for representation of arbitrary time points, even those that have never been encountered before. Due to the inherent properties of box-type geometric embedding, it can represent varied relations with respect to transitivity and is closed under intersection. Hence, we further represent the head and tail entities using box embeddings to enhance the rigid inference ability of PTBox. Empirically, we conduct detailed experimental evaluations over two popular TKGE benchmarks and prove our method can capture time-evolving information. Moreover, we analyze the learned box embeddings and show the abilities of our PTBox for modeling various relation patterns, including temporal evolution.

In summary, our main contributions are three-fold:

- We propose an innovative temporal knowledge graph embedding method, namely PTBox. Experimental results verify the state-of-the-art performance of our method on two publicly available datasets.
- Proposing an interpretable time representation method that decomposes time information by polynomial approximation theory to flexibly represent arbitrary timestamp.
- Proposing a box-embedding-based entity representation method that effectively represents calibrated probability distributions and learns rigid inference patterns.

2. Related Work

2.1. Knowledge Graph Embedding

Real-world knowledge graphs are usually incomplete, necessitating completion techniques to enable their effective application in downstream tasks. Knowledge graph embedding (KGE), as the popular static knowledge graph completion technique, employs embeddings to represent entities and relations. By learning scores for all possible facts, KGE methods predict missing links, e.g. $(the\,Beatles, genre, ?)$. As an active research area, numerous methods have been proposed to address the challenges of representing and completing knowledge graphs. These KGE methods can be broadly categorized into several key approaches: translation-based methods, tensor-based methods, and neural network-based methods.

Translation-based methods define scoring functions based on the translation between entities and relations. The classical model TransE (Bordes et al., 2013) utilizes the parallelogram rule in vector space to define the scoring function, assuming that adding the head entity vector and the relation vector should be as close as possible to the tail entity vector. Subsequently, several improved models in the Trans-series have been proposed, such as TransH (Wang et al., 2014), TransN (Wang and Cheng, 2018), TransC (Lv et al., 2018). To address the contradiction between the optimization and regularization in the TransE, TorusE (Ebisu and Ichise, 2018) introduces the Lie Group to learn embedding representations in a torus space. To model rich inference patterns (e.g., symmetry/antisymmetry, inversion, and composition), RotatE (Sun et al., 2019) defines each relation as a rotation in the complex vector space from the head entity to the target entity.

Tensor-based models treat the knowledge graph as a multi-dimensional tensor and aim to factorize this tensor to learn entity and relation embeddings. RESCAL (Nickel et al., 2011) factorizes the tensor into low-rank matrices, capturing the interactions between entities and relations. DistMult (Yang et al., 2015) restricts the representation matrix of relations to diagonal matrices, greatly reducing model complexity. ComplEx (Trouillon et al., 2016) extends the representation of entities and relations to the complex vector space, providing better modeling of asymmetric relations. SimplE (Kazemi and Poole, 2018), based on the canonical polyadic decomposition, represents each entity and relation with two vectors, thereby increasing the correlation between head and tail entities. Additionally, some methods model relations in hypercomplex spaces to learn representations with more geometric features (Zhang et al., 2019),(Nguyen et al., 2022).

Neural network-based methods apply deep learning techniques to model entity and relation representations, aiming to obtain embeddings with strong generalization ability and robustness. Among them, ConvE (Dettmers et al., 2018) is based on convolutional neural networks, R-GCN (Schlichtkrull et al., 2018) and M2GNN (Wang et al., 2021) are based on graph neural networks, and A2N (Bansal et al., 2019) leverages attention mechanisms. Additionally, there are methods like MuRP

[1] We release our code at https://github.com/seeyourmind/PTBox

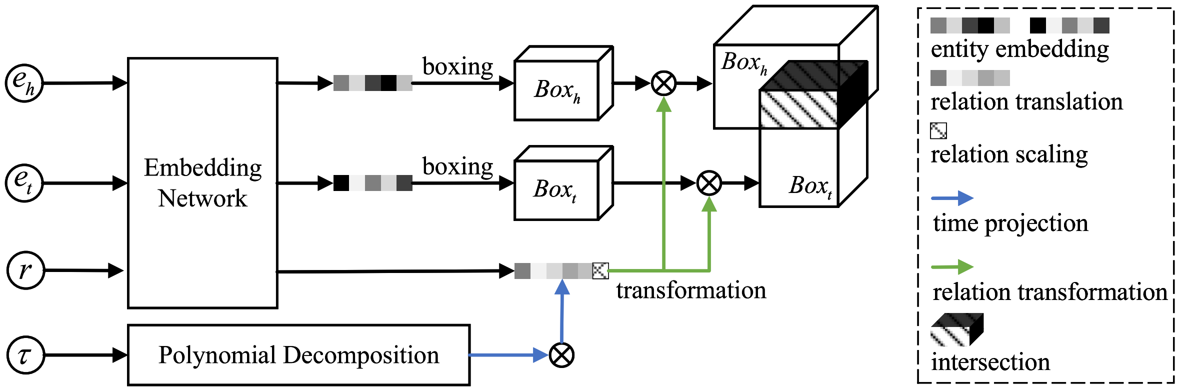

Figure 1: the overall of our proposed PTBox. e_h, e_t, r and τ denote head entity, tail entity, relation and timestamp for one quadruple fact (h, r, t, τ), respectively. $\otimes$ denotes Hadamard product. Our model obtains the time representation via the polynomial decomposition mechanism and represent the probability of the fact being established by the intersection between entity boxes.

(Balazevic et al., 2019) and ROTH (Chami et al., 2020) that model entity and relation representations in non-Euclidean spaces. Although KGE methods have been widely applied in recent years, they face limitations in capturing the dynamic evolution of facts in real-world scenarios. To address this problem, research focuses on addressing challenges such as handling dynamic knowledge graphs and incorporating temporal information.

2.2. Temporal Knowledge Graph Embedding

Temporal Knowledge Graph Embedding (TKGE) methods aim to incorporate temporal information into the representation learning process of knowledge graphs. This incorporation allows modeling of temporal dynamics and the evolution of facts. These methods extend traditional KGE approaches by introducing temporal factors, such as timestamps or time intervals, associated with triples in the knowledge graph. TTransE (Leblay and Chekol, 2018) extends TransE by considering time embedding as an equivalent vector to entities and relations. TA-DistMult (García-Durán et al., 2018) extends DistMult by training a recursive neural network with sequences of tokens representing time predicates and digits in timestamps. DE-SimplE (Goel et al., 2020) extends SimplE by incorporating a diachronic entity embedding function to provide representations for entities at any given timestamp. In addition, TComplex (Lacroix et al., 2020) extends ComplEx, BoxTE (Messner et al., 2022) extends BoxE (Abboud et al., 2020), ChronoR (Sadeghian et al., 2021) and RotateQVS (Chen et al., 2022) extend RotatE.

On the other hand, some methods focus on incorporating time information by leveraging deep learning to craft customized network architectures or optimization functions. HyTE (Dasgupta et al., 2018) represents timestamps as mutually independent hyperplanes, where entities and relations satisfy the TransE assumption. ATiSE (Xu et al., 2020b) leverages the additive time series decomposition to treat entities and relations as temporal data, decomposing them into trend, seasonal, and random components. TeRo (Xu et al., 2020a) combines the RotatE and TransE models by representing the temporal evolution of entity embeddings as a rotation in the complex vector space, starting from the initial time to the current time. DyERNIE (Han et al., 2020) models entities on a mixed curvature manifold and defines the tangent vector of a given entity as the velocity of the entity's evolution over time, enabling the description of dynamic evolutionary processes of entities. By incorporating temporal information, TKGE methods enhance knowledge graph embedding to capture temporal dynamics, and support downstream applications in evolving real-world scenarios. Although both mainstream categories of TKGE methods have their respective advantages, there is currently limited work that combines the strengths of these two approaches, namely having interpretable time representations while also supporting rich inference patterns. Therefore, this paper proposes a novel TKGE method via entity boxes and polynomial decomposition of time that aims to bridge this gap.

3. Methodology

In this section, we introduce the proposed PTBox method. The architecture is shown in Figure 1. We first describe the employed notations and definitions in Section 3.1. Then, we present the model framework and the two modules of our method in Section 3.2 and 3.3, respectively. In addition, we discuss the parameter learning strategy in Section 3.4, and model properties in Section 3.5.

3.1. Problem Formulation

Temporal knowledge graphs represent events or facts using quadruples (h, r, t, τ), where $h \in \mathcal{E}$ and $t \in \mathcal{E}$ represent the head and tail entities, $r \in \mathcal{R}$ represents the relations, $\tau \in \mathcal{T}$ represents the timestamps. Then, a TKG can be formulated as $\mathcal{G} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T}$. For example, *the quadruple (JosephineTewson, wasBornIn, Hampstead, 1931-02-26) describes the fact that Josephine Tewson was born in Hampstead on February 26, 1931.* TKGE methods aim to complete knowledge graphs by leveraging link prediction task, which utilize a scoring function to predict missing head or tail entities in $\mathcal{G}$ within a specified temporal context. Typically, the score function is learned with the formula as $f : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T} \to \mathbb{R}$, that assigns a score $s = f(h, r, t, \tau)$ to each quadruple, indicating the prediction that a particular quadruple corresponds to a true fact. Followed closely, a non-linearity, such as the logistic or sigmoid function, is often used to convert the score to a predicted probability $p = \sigma(s) \in [0, 1]$ of the quadruple being true.

3.2. Polynomial Decomposition based Temporal Representation

The incorporation of temporal information represents the primary difference between TKG and traditional KG methods. As a result, effectively modeling temporal embeddings becomes a critical task for TKGE. Although existing TKGE methods have successfully integrated temporal information with entity-relation triplets, a limited number of approaches directly mathematically model continuous time information. Therefore, we model the timestamp via polynomial decomposition-based representation (PTR) to learn interpretable representations of continuous time.

According to Weierstrass approximation theorem (Pinkus, 2000), a continuous function defined on a closed interval can be uniformly approximated by a polynomial function. Namely,

$$f \in C[a, b], \forall \epsilon > 0, \exists P_n \Rightarrow \forall x \in [a, b], |f - P_n| < \epsilon \tag{1}$$

where P_n is the polynomial function used to uniformly approximate the continuous function f. Based on Stone-Weierstrass theorem (Cotter, 1990), topological space $\mathbb{R}$ is a Hausdroff space, and the Weierstrass approximation theorem is satisfied in this space. Accordingly, we assume that time information can be expressed as a nonlinear function on the closed interval $[0, 1]$, and design a multi-layer perceptron with a sigmoid output layer to learn this function. Then, we can leverage the Bernstein polynomial to represent P_n, which can be formulated as:

$$P_n(f_\tau, x) = \sum_{k=0}^{n} f(\frac{k}{n}) \binom{n}{k} x^k (1 - x)^{n-k} \tag{2}$$

where $f_\tau(x)$ denotes the function describing time information. f_τ scales the input timestamps to $[0, 1]$. Given n, we convert $P_n(f_\tau, x)$ into the matrix form, which can be formulated as:

$$\mathcal{P}_\tau = P_n(f_\tau, x) = \boldsymbol{\alpha_\tau} \cdot \boldsymbol{X}, \tag{3}$$

where $\boldsymbol{\alpha_\tau}$ denotes the coefficient matrix and $\boldsymbol{X}$ denotes the polynomial matrix. Note that $\boldsymbol{X} \in \mathbb{R}^{k \times d}$ is the temporal basis tensor, which learns the basic meta-features of time information. Then, based on Equation3, we can easily model the temporal representation of any given timestamp. Moreover, according to the representations we can further dynamically model entities and relations under temporal constraints.

3.3. Box Embedding based Entity Representation

Temporal knowledge graphs inherently contain rich geometric structural information. Geometric embedding methods possess the natural ability to represent transitive asymmetric relations via containment. Among these methods, box embeddings represent objects as n-dimensional hyperrectangles and exhibit closure under intersection. This characteristic makes box embeddings well-suited for capturing complex relationships within knowledge graphs. Hence, we model the entities via box embedding based entity representations (BER) to enhance the representation and reasoning capabilities of our PTBox model.

In our PTBox, every entity $e_i \in \mathcal{E}$ is represented by a n-dimensional axis-aligned hyperrectangle (namely box) $Box(e_i) \subseteq \mathbb{R}^d$, which can be viewed as a lattice, a special poset. Each box is represented by a pair of vectors, which correspond to the maximum coordinates e_i^M and minimum coordinates e_i^m of the box, respectively. Considering the entities in the knowledge graph as a non-strict partial order set, we can then define the relationship between e_i and e_j by inclusion of boxes as follows:

$$\begin{aligned} e_i \vee e_j &= \prod_k [\min(e_k^{m,i}, e_k^{m,j}), \max(e_k^{M,i}, e_k^{M,j})], \\ e_i \wedge e_j &= \begin{cases} \perp, & \text{if } e_i, e_j \text{ disjoint} \\ \prod_k [\max(e_k^{m,i}, e_k^{m,j}), \min(e_k^{M,i}, e_k^{M,j})], & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

where $\vee$, $\wedge$, and $\perp$ denote partial order relations, $\vee$ is the smallest enclosing box, $\wedge$ is the intersecting box, and $\perp$ is the empty box. $\min(\cdot)$ and $\max(\cdot)$ are functions used to calculate the minimum and maximum values, respectively. Further, following the viewpoint proposed by Vilnis et al. (Vilnis et al., 2018), we can interpret the volume of a box as a non-normalized probability. Therefore, utilizing inclusion-exclusion with set intersection over $Box(\cdot)$, the joint probability and conditional probability of relationships in the knowledge graph can

Inference Pattern	Setting
Symmetry: $r_1(e_1, e_2\|\tau) \Rightarrow r_1(e_2, e_1\|\tau)$	$P_{r_1}(e_1\|e_2) = P_{r_1}(e_2\|e_1) \neq 0$
Antisymmetry: $r_1(e_1, e_2\|\tau) \Rightarrow \neg r_1(e_2, e_1\|\tau)$	$P_{r_1}(e_1\|e_2) \neq 0, P_{r_1}(e_2\|e_1) = 0$
Inversion: $r_1(e_1, e_2\|\tau) \Leftrightarrow r_2(e_2, e_1\|\tau)$	$P_{r_1}(e_1\|e_2) = P_{r_2}(e_2\|e_1) \neq 0$
Composition: $r_1(e_1, e_2\|\tau) \wedge r_2(e_2, e_3\|\tau) \Rightarrow r_3(e_1, e_3\|\tau)$	$P_{r_3}(e_1, e_2, e_3) \neq 0$
Hierarchy: $r_1(e_1, e_2\|\tau) \Rightarrow r_2(e_1, e_2\|\tau)$	$P_{r_1,r_2}(e_1\|e_2) \geq P_{r_1}(e_1\|e_2)P_{r_2}(e_1\|e_2) \neq 0$
Intersection: $r_1(e_1, e_2\|\tau) \wedge r_2(e_1, e_2\|\tau) \Rightarrow r_3(e_1, e_2\|\tau)$	$P_{r_3}(e_1\|e_2) \geq P_{r_1,r_2}(e_1\|e_2) \neq 0$
Mutual exclusion: $r_1(e_1, e_2\|\tau) \wedge r_2(e_1, e_2\|\tau) \Rightarrow \perp$	$P(\mathcal{B}(e_1^{r_1}) \cap \mathcal{B}(e_2^{r_1}), \mathcal{B}(e_1^{r_2}) \cap \mathcal{B}(e_2^{r_2})) = 0$

Table 1: Inference patterns/generalized inference patterns captured by our PTBox with fixed timestamp τ.

be easily calculated.

$$\begin{aligned} P(e_i, e_j, e_k) =& Vol(Box(e_i) \cap Box(e_j) \cap Box(e_k)), \\ P(e_i|e_j) =& \frac{Vol(Box(e_i) \cap Box(e_j))}{Vol(Box(e_j))}, \end{aligned} \tag{5}$$

where P denotes the probability function, $Vol(\cdot)$ denotes the volume function of the box, and $\cap$ denotes the intersection operator between boxes.

However, the parameter settings of box embeddings result in equivalent probability distributions, rendering conventional gradient-based deep learning optimization algorithms impractical for learning (Vilnis et al., 2018). To mitigate this, we employ Gumbel boxes proposed by Dasgupta et al. (Dasgupta et al., 2020; Chen et al., 2021) to model our box embeddings. The maximum and minimum coordinates of Gumbel boxes follow the Gumbel distribution, then the boxes can be formulated as:

$$\begin{aligned} Box(e) &= \prod_{i=1}^{d} [e_i^m, e_i^M], \\ e_i^m &\sim \text{MaxGumbel}(\mu_i^m, \beta), \\ e_i^M &\sim \text{MinGumbel}(\mu_i^M, \beta), \end{aligned} \tag{6}$$

where μ is a location parameter and β is a scale parameter. The mean and variance of Gumbel distribution are $\mu + \gamma\beta$ and $\frac{\pi^2}{6}\beta^2$, where γ is the Euler–Mascheroni constant. Gumbel distribution as generalized extreme value distribution is min and max stable, keeping the Gumbel boxes closed under intersection. Hence, the approximation of volume in Gumbel boxes can be formulated as:

$$\mathbb{E}[Vol(Box(e))] \approx \prod_{i=1}^{d} \beta \log\left(1 + \exp\left(\frac{\mu_i^M - \mu_i^m}{\beta} - 2\gamma\right)\right). \tag{7}$$

3.4. Modeling and Evaluation of Quadruples

As mentioned previously, given a quadruple (h, r, t, τ), our method models head and tail entities as two Gumbel boxes $Box(e_h)$, $Box(e_t)$. Timestamp τ is modeled as a temporal projection $\mathcal{P}_\tau$. Then, the evolutionary dynamics of entities and relations over time can be formulated as:

$$\begin{aligned} e'_h =& \mathcal{P}_\tau(e_h; W) = Box(e_h) + (W^T Box(e_h))W, \\ e'_t =& \mathcal{P}_\tau(e_t; W) = Box(e_t) + (W^T Box(e_t))W, \\ r'_t =& \mathcal{P}_\tau(r_t; W) = r_t + (W^T r_t)W, \end{aligned} \tag{8}$$

where W denotes the weight of temporal projection, e'_h and e'_t denote the evolutionary representations of head and tail entities, r'_t denotes the evolutionary representation of relation. Note that the mapped entity representations remain Gumbel boxes.

Furthermore, we consider each relation r as an affine transformation $T_r \subseteq \mathbb{R}^{2\times d}$ acting on entity boxes, where $T_r[0]$ represents the translation and $T_r[1]$ represents the scaling. Given a entity e, the relation transformation can be formulated as:

$$\begin{aligned} e^t =& f_r^t(e|T_r) = Box(e) + T_r[0], \\ e^s =& f_r^s(e|T_r) = Box(e) \odot T_r[1], \end{aligned} \tag{9}$$

where e^t denotes the translation of e, e^s denotes the scaling of e, and $\odot$ is the Hadamard product. To simplify the notations, we use $f_r^t(\cdot)$ and $f_r^s(\cdot)$ to denote the two transformation operations of relation r. The composition of these functions satisfies the properties of an Abelian group, allowing them to act on Gumbel boxes while preserving the relationships and structure of the boxes. Consequently, given the quadruple (h, r, t, τ), we define a scoring function as the volume intersection between two new boxes formed by the evolved boxes of h and t under the transformation of r at τ. This can be formulated as:

$$\mathcal{S}(h, r, t, \tau) = \frac{\mathbb{E}[Vol(f_r^s \circ f_r^t(\mathcal{P}_\tau(e_h; W)) \cap f_r^s \circ f_r^t(\mathcal{P}_\tau(e_t; W)))]}{\mathbb{E}[Vol(f_r^s \circ f_r^t(\mathcal{P}_\tau(e_t; W)))]}. \tag{10}$$

3.5. Analysis of Model Properties

We analyze the representation power and inductive capacity of PTBox. The conclusion of our analysis indicates that PTBox is locally identifiable and can capture rich inference patterns and higher-arity relations. We additionally analyze the complexity of PTBox, and prove that it runs in time $O(d)$ and space $O((|E| + |R| + K)d)$, where $|E|$ and $|R|$ are the maximal quantity.

Dataset	YAGO11k	WikiData
Entities	10,623	500
Relations	10	24
Time Span	-453∼2844	1479∼2018
Train	16,408	32,497
Validation	2,050	4,062
Test	2,051	4,062

Table 2: Statistics of two experimented datasets.

Local Identifiability. The local identifiability of a model is used to measure whether its parameters are sensitive to local features. Assuming a set of parameters Ω is local identifiable if, for all $\theta \in \Omega$, there exists $N(\theta)$, a neighborhood of θ, such that for all $\theta' \in N(\theta), L(x|\theta') \neq L(x|\theta)$. According to Vilnis et al.'s observations (Vilnis et al., 2018), the parameter space of probability box embeddings has large degrees of freedom, which leads to the lack of local identifiability. Although this property implies that the model's parameters are not overly influenced by local variations or noise in the data, the lack of local identifiability poses significant challenges in the training and optimization of the model. Our approach tackles this problem by employing the Gumbel distribution to represent box embeddings. To effectively mitigate the aforementioned problems with optimization, we retain the uncertainty over the box intersections and make all parameters contribute to the data likelihood in an appropriate manner. Moreover, computing the volume using conditional probabilities in Equation5 can further alleviate the unboundedness problem in the base measure space (Dasgupta et al., 2020).

Inference Patterns. We study the inductive capacity of PTBox in terms of common inference patterns appearing in the TKGE literature, and details as shown in Table1. Inference patterns are important for downstream tasks of knowledge graphs, and jointly capturing multiple inference patterns is meaningful but challenging. Our PTBox captures all generalized inference patterns given in Table1 through box configurations. For example, PTBox inherently supports probability and intersection rules. Symmetry can be captured by ensuring the conditional probabilities $P_{r_1}(e_1|e_2)$ and $P_{r_1}(e_2|e_1)$ exist and are equal. Composition can be captured by ensuring the joint probability $P_{r_3}(e_1, e_2, e_3)$ exists. Mutual exclusion is captured by disjointness between the intersection boxes under relations r_1 and r_2, respectively. Compared to earlier methods, TransE (Bordes et al., 2013) fails to capture symmetry and composition, RotatE (Sun et al., 2019) fails to capture hierarchy, and ComplEx (Trouillon et al., 2016) fails to capture composition and intersection. Clearly, our method captures more diverse inference patterns and exhibits stronger inductive capability. In addition, based on the configuration of PTBox, we compute the evolved representations of relations at different timestamps, allowing us to capture cross-time inference patterns, as mentioned in (Messner et al., 2022).

Runtime and Space Complexity. In terms of runtime complexity, for any quadruple (h, r, t, τ), we firstly compute d-dimensional temporal embedding through polynomial decomposition. Then, we multiply relations to obtain two d-dimensional relation embeddings, and finally compute the evolved entity embeddings using multiplication and addition operations. The volume function runs in $O(d)$ for every box. In terms of space complexity, PTBox stores two d-dimensional vectors for each entity box, two d-dimensional vectors for each relation, and a $K \times d$ matrix for time. Hence, for a KG with $|E|$ entities and $|R|$ relations, PTBox requires $(|E| + |R| + K)d$ parameters.

4. Experiments

We evaluate the performance of our proposed PTBox model on two popular TKG benchmarks.

4.1. Experimental Setup

Datasets. We use two well-known datasets for evaluation, namely, YAGO11k, and WikiData. YAGO11k and WikiData are temporal subgraphs extracted from YAGO3 and Wikipedia, respectively. As a subset of YAGO3, YAGO11k incorporates information from multiple sources, including Wikipedia, WordNet, and GeoNames, ensuring its richness and reliability. In this paper, the WikiData is Wikipedia12k proposed by HyTE (Dasgupta et al., 2018). Similar to YAGO11k, Wikipedia12k contains the facts involving time intervals. The detailed statistics of the datasets are presented in Table 2.

Evaluation Metrics. We utilize the link prediction task to evaluate the effectiveness of the PTBox model. We employ classic evaluation metrics, which include mean rank (MR), mean reciprocal rank (MRR) and hits at 1/3/10 (Hits@1/3/10). These metrics all represent the rankings of missing ground-truth entities in the prediction results. For each query, we report the mean results of both the subject and object entity prediction tasks.

Implementation Details. We implemented our PTBox model in PyTorch and trained the model on a GPU (RTX 3090). We configured the parameters based on the MRR and Hits@10 performance achieved by the model on the validation set. For polynomial decomposition based temporal representation, the order of temporal polynomial k is set to 20. For box embedding based entity representation, the embedding dimension d is set to 128, the distribution of box embeddings follows $Gumbel(0.01, 1)$ and $Gumbel(-0.1, -0.001)$. We

Model	YAGO11k			WikiData		
	MRR	Hits@3	Hits@10	MRR	Hits@3	Hits@10
TransE	0.100	0.138	0.244	0.178	0.192	0.339
DistMult	0.158	0.161	0.268	0.222	0.238	0.460
RotatE	0.167	0.167	0.305	0.116	0.236	0.461
QuatE	0.164	0.148	0.270	0.125	0.243	0.416
TTransE	0.108	0.150	0.251	0.172	0.184	0.329
HyTE	0.105	0.143	0.272	0.180	0.197	0.333
TA-DistMult	0.161	0.171	0.292	0.218	0.232	0.447
ATiSE	0.170	0.171	0.288	0.280	0.317	0.481
TeRo	0.187	0.197	0.319	**0.299**	0.329	0.507
RotateQVS	**0.189**	0.199	0.323	-	-	-
PTBox	0.162	**0.222**	**0.347**	0.290	**0.331**	**0.527**

Table 3: Link prediction results on YAGO11k, WikiData for our proposed and baseline methods. The best results are marked in bold.

train our model by Adam optimizer, and set the learning rate as 0.0001 for all datasets.

4.2. Results of TKGE on Link Prediction

In this section, we compare the performance of our proposed PTBox with that of static and dynamic methods based on the TKG link prediction task.

Baseline Models. We compare our approach with several state-of-the-art (SOTA) approaches, including static KGE methods and dynamic TKGE methods. Among them, TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), RotatE (Sun et al., 2019), and QuatE (Zhang et al., 2019) are static KGE methods, which focus on modeling triples of static facts. TTransE (Leblay and Chekol, 2018), HyTE (Dasgupta et al., 2018), TA-DistMult (García-Durán et al., 2018), ATiSE (Xu et al., 2020b), TeRo (Xu et al., 2020a), and RotateQVS (Chen et al., 2022) are dynamic TKGE methods, which focus on modeling quadruples of temporal facts.

Experimental Results. We present the performance of different models on temporal datasets in Table 3. From the results compared to the baseline models, our method consistently achieves performance improvements across all datasets. We observe that PTBox outperforms RotateQVS in terms of Hits@3 and Hits@10 on YAGO11k. PTBox also outperforms TeRo in terms of Hits@3 and Hits@10, and is competitive with TeRo in terms of MRR on WikiData. Specifically, our method achieves an MRR of 16.2%, Hits@3 of 22.2%, and Hits@10 of 34.7% on the YAGO11k, and an MRR of 29.0%, Hits@3 of 33.1%, and Hits@10 of 52.7% on the WikiData, respectively.

Table 3 lists the link prediction results on On YAGO11k and WikiData where time annotations in facts are time intervals. Compared with static KGE methods, our PTBox outperforms RotatE by 5.5% regarding Hits@3, and by 4.2% regarding Hits@10 on YAGO11k, respectively. Our PTBox also outperforms DistMult by 6.8% regarding MRR, outperforms QuatE by 8.8% regarding Hits@3, and outperforms RotatE by 6.6% regarding Hits@10 on WikiData, respectively. It means that temporal information can effectively enhance the performance of knowledge completion and reasoning by supplementing the facts. Compared with temporal KGE methods, our PTBox outperforms RotateQVS by 2.3% regarding Hits@3, and by 2.4% regarding Hits@10 on YAGO11k, respectively. It means that our proposed strategy of modeling time through polynomial decomposition can effectively capture temporal information, while providing good embedding representations for temporal knowledge graphs. On the other hand, our PTBox underperforms on the MRR but outperforms on the Hits@3 and Hits@10 compared to both RotateQVS and TeRo. This indicates that our model has limitations in top-1 accuracy, but has higher recall compared to theirs. In addition, both of these methods fix the representation of timestamps in the dataset during the training phase, and they cannot model unseen timestamps as flexibly as our PTBox does.

Model	YAGO11k		WikiData	
	MR	Hits@1	MR	Hits@1
TransE	1.70	0.784	1.35	0.884
TransH	1.53	0.761	1.40	0.881
HolE	2.57	0.693	2.23	0.840
t-TransE	1.66	0.755	1.97	0.742
HyTE	1.23	0.812	1.13	0.926
PTBox	**1.12**	**0.896**	**1.12**	**0.934**

Table 4: Relation prediction results on YAGO11k and WikiData for our proposed and baseline methods. The best results are marked in bold.

PTR	BER	YAGO11k MRR	YAGO11k Hits@3	YAGO11k Hits@10	WikiData MRR	WikiData Hits@3	WikiData Hits@10
		0.105	0.143	0.272	0.180	0.197	0.333
✓		0.137	0.174	0.313	0.259	0.278	0.478
	✓	0.127	0.180	0.280	0.253	0.281	0.426
✓	✓	**0.162**	**0.222**	**0.347**	**0.290**	**0.331**	**0.527**

Table 5: Ablation study on YAGO11k and WikiData for our proposed two strategies under link prediction task. The best results are marked in bold.

Mode	YAGO11k MRR	YAGO11k Hits@10	WikiData MRR	WikiData Hits@10
$\mathcal{P}_\tau(e)$	0.127	0.289	0.267	0.485
$\mathcal{P}_\tau(r)$	**0.162**	**0.347**	**0.290**	**0.527**
$\mathcal{P}_\tau(e,r)$	0.135	0.311	0.277	0.503

Table 6: Ablation study on YAGO11k and WikiData for examining different evolutionary patterns of knowledge graph triples over time. The best results are marked in bold.

4.3. Results of TKGE on Relation Prediction

To further analyze the performance of our model, we conduct relation prediction experiments to examine whether temporal information is beneficial in resolving ambiguities among relations. Detailed results are shown in Table 4.

Based on the experimental results, the three static methods (TransE, TransH, and HolE (Nickel et al., 2016)) demonstrate inferior performance compared to the two dynamic methods (HyTE and our PTBox) on both the YAGO11k and WikiData. It means that time information surely contributes to relation disambiguation. Although the t-TransE (Jiang et al., 2016) is a temporal model, it does not directly model temporal information. Instead, it achieves implicit temporal fusion through relation ordering. As a result, compared to the other two methods, it does not fully leverage temporal information to resolve ambiguities among relations. Different from HyTE can only model times present in the training set, our approach utilizes the Weierstrass approximation theorem (as outlined in Section 3.2) to learn a coefficient matrix for any given time, and then we can easily model the temporal representation of any timestamp by combining with the temporal basis tensor (as outlined in Equation.4). It means that our proposed polynomial based temporal representation offers more flexibility in modeling time, which enhances the performance of relation prediction. Consequently, our method has achieved SOTA performance on both datasets. Specifically, compared with HyTE, we can increase Hits@1 from 81.2% to 89.6% on YAGO11k, and from 92.6% to 93.4% on WikiData, respectively.

4.4. Ablation Study

In this section, we conduct ablation study to evaluate the effectiveness and necessity of our proposed components on two datasets. As mentioned in Section 3.2 and 3.3, we propose two strategies to model time and entities, namely PTR and BER. Therefore, we compare the impact of different strategies on performance across all datasets, and the results are reported in Table 5. Moreover, we compare different evolutionary patterns of the knowledge graph over time, as shown in Table 6.

PTR and BER. As shown in Table 5, we employ HyTE as the baseline and report the experimental results of PTR and BER when activated independently and jointly. The experimental results demonstrate that both the proposed PTR and BER contribute to enhancing the performance of KGE on link prediction tasks when compared to the baseline model. In particular, their joint application leads to significant improvements across all datasets. According to the results in the second row, our PTR strategy can boost the performance of link prediction by modeling temporal information more effectively than baseline model. According to the results in the third row, our BER strategy also benefits in improving model performance. This shows that leveraging geometric modeling and intersection operation can cover rich reasoning patterns. Comparing the second and third rows of Table 5, we can observe that the PTR strategy improves performance slightly better than BER. This means that effective modeling of time information is more important in the TKGE methods.

Different Evolutionary Patterns. To explore the impact of different evolutionary patterns on the model, we summarize the experimental results of three different patterns in Table 6. Among them, $\mathcal{P}_\tau(e)$ represents time acting on entities, $\mathcal{P}_\tau(r)$ represents time acting on relations, and $\mathcal{P}_\tau(e,r)$ represents time simultaneously acting on both entities and relations. Compared to entities, the types of relations are fewer and their representations in the feature space are relatively sparse. Therefore, the

Test quadruples	HyTE	Ours
Katie_Holmes, ?, Tom_Cruise, [2006, 2012]	**isMarriedTo**, hasWonPrize	**isMarriedTo**, created
Tricia_Devereaux, ?, Illinois, [1975, 1975]	diedIn, **wasBornIn**	**wasBornIn**, diedIn
Jeremy_Lloyd, ?, London, [2014, 2014]	wasBornIn, **diedIn**	**diedIn**, wasBornIn
Will_Haining, ?, Fleetwood_Town_F.C., [2011, -]	isMarriedTo, **playsFor**	**playsFor**, isMarriedTo
Bob_Hope, ?, Toluca_Lake,_Los_Angeles, [2003, 2003]	isMarriedTo, hasWonPrize	isMarriedTo, **diedIn**

Table 7: Case study of qualitative analysis on relation prediction. The order of prediction is in descending order. Correct one is in **bold**.

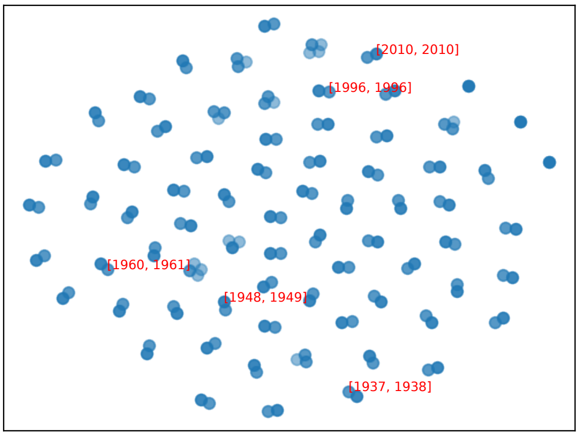

Figure 2: Visualization of polynomial decomposition based temporal representations on YAGO11k.

evolution of relations over time can produce more significant differences. We believe that this view is also consistent with real-world scenarios, where individuals maintain independence and the changes over time are the relationships between individuals.

4.5. Qualitative Analysis

We conducted two qualitative analysis experiments to intuitively demonstrate the performance of our model, including the case study of relation prediction with results shown in Table 7 and the visualization of temporal representations with results illustrated in Figure 2.

Table 7 presents a comparative analysis of predictions made on some samples from the YAGO11k test set by our method and HyTE. Overall, our approach demonstrates superior accuracy in relation prediction compared to HyTE. This is particularly evident in the case of the easily confusable relations $diedIn$ and $wasBornIn$, where our method consistently predicts the ground truth accurately. Furthermore, for the fact *(Bob Hope, diedIn, Toluca Lake Los Angeles, [2003, 2003])*, unlike HyTE, which fails to deliver the correct result, our method achieves a hit within the top 2 predictions.

Evaluation metric for continuous modelling of time is challenging to quantify, but we can employ the same strategy as HyTE, visualizing the distribution of temporal embeddings in the vector space. From Figure 2, we can observe that timestamps with close time intervals are close to each other in vector space, and vice versa. This indicates that our proposed polynomial decomposition based temporal representation is capable of effectively modeling temporal information in a continuous manner.

5. Conclusion

In this paper, we propose an innovative TKGE method based on polynomial approximation for modeling arbitrary time information, namely PTBox. Our main contributions lies in polynomial decomposition-based temporal representation and box embedding-based entity representation. To enhance the performance of the TKGE, we focus on improving the capabilities to continuously model arbitrary time information and infer under temporal constraints. Our method decomposes timestamp by polynomial approximation theory to flexibly represent time information. Furthermore, to capture complex geometric structures and learn rich inference patterns, we model entities by box embeddings and define each relation as a transformation on the head and tail entity boxes. Theoretically, our proposed PTBox can encode arbitrary time information or even unseen timestamps, while capturing higher-arity relations of the knowledge base. Extensive experiments on real-world datasets demonstrate the effectiveness of our method.

6. Acknowledgements

This research was supported by the National Science and Technology Major Project (2022ZD0119204), National Key Research and Development Program of China (2020AAA0109701), National Science Fund for Distinguished Young Scholars (62125601).

7. References

Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. Boxe: A box embedding model for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 9649–9661.

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019. Multi-relational poincaré graph embeddings. In *Advances in Neural Information Processing Systems*, pages 4465–4475.

Trapit Bansal, Da-Cheng Juan, Sujith Ravi, and Andrew McCallum. 2019. A2N: attending to neighbors for knowledge graph inference. In *Proceedings of the Conference of the Association for Computational Linguistics*, pages 4387–4392.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914.

Kai Chen, Ye Wang, Yitong Li, and Aiping Li. 2022. Rotateqvs: Representing temporal information as rotations in quaternion vector space for temporal knowledge graph completion. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 5843–5857.

Xuelu Chen, Michael Boratko, Muhao Chen, Shib Sankar Dasgupta, Xiang Lorraine Li, and Andrew Mccallum. 2021. Probabilistic box embeddings for uncertain knowledge graph reasoning. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 882–893.

N.E. Cotter. 1990. The stone-weierstrass theorem and its application to neural networks. *IEEE Transactions on Neural Networks*, 1(4):290–295.

Shib Sankar Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Li, and Andrew McCallum. 2020. Improving local identifiability in probabilistic box embeddings. In *Advances in Neural Information Processing Systems*, pages 182–192.

Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha P. Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2001–2011.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1811–1818.

Takuma Ebisu and Ryutaro Ichise. 2018. Toruse: Knowledge graph embedding on a lie group. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1819–1826.

Alberto García-Durán, Sebastijan Dumancic, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 4816–4821.

Rishab Goel, Seyed Mehran Kazemi, Marcus A. Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3988–3995.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020. Dyernie: Dynamic evolution of riemannian manifold embeddings for temporal knowledge graph completion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 7301–7316.

Shiyu Hu, Xin Zhao, Lianghua Huang, and Kaiqi Huang. 2022. Global instance tracking: Locating target more like humans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):576–592.

Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. Encoding temporal information for time-aware link prediction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2350–2354.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems*, pages 4289–4300.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. In *Proceedings of the International Conference on Learning Representations*.

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion of the International World Wide Web Conferences*, pages 1771–1776.

Xin Lv, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Differentiating concepts and instances for knowledge graph embedding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1971–1979.

Johannes Messner, Ralph Abboud, and İsmail İlkan Ceylan. 2022. Temporal knowledge graph completion using box embeddings. In *Proceedings* *of the AAAI Conference on Artificial Intelligence*, pages 7779–7787.

Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, and Dinh Q. Phung. 2022. Quatre: Relation-aware quaternions for knowledge graph embeddings. In *Companion of the International World Wide Web Conferences*, pages 189–192.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1955–1961.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the International Conference on Machine Learning*, pages 809–816.

Allan Pinkus. 2000. Weierstrass and approximation theory. *Journal of Approximation Theory*, 107(1):1–66.

Ali Sadeghian, Mohammadreza Armandpour, Anthony Colas, and Daisy Zhe Wang. 2021. Chronor: Rotation based temporal knowledge graph embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6471–6479.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proceedings of the European Semantic Web Conference*, volume 10843, pages 593–607.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the International Conference on Learning Representations*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the International Conference on Machine Learning*, volume 48, pages 2071–2080.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic embedding of knowledge graphs with box lattice measures. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 263–272.

Chun-Chih Wang and Pu-Jen Cheng. 2018. Translating representations of knowledge graphs with neighbors. In *Proceedings of the International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 917–920.

Shen Wang, Xiaokai Wei, Cícero Nogueira dos Santos, Zhiguo Wang, Ramesh Nallapati, Andrew O. Arnold, Bing Xiang, Philip S. Yu, and Isabel F. Cruz. 2021. Mixed-curvature multi-relational graph neural network for knowledge graph completion. In *Proceedings of the International World Wide Web Conference*, pages 1761–1771.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1112–1119.

Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2020a. Tero: A time-aware knowledge graph embedding via temporal rotation. In *Proceedings of the International Conference on Computational Linguistics*, pages 1583–1593.

Chenjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Yazdi, and Jens Lehmann. 2020b. Temporal knowledge graph completion based on time series gaussian embedding. In *Proceedings of the International Semantic Web Conference*, pages 654–671.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations*.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems*, pages 2731–2741.