Hyperbolic Graph Neural Network for Temporal Knowledge Graph Completion

Yancong Li[1], **Xiaoming Zhang**[1,2*], **Ying Cui**[1], **Shuai Ma**[2]
[1]School of Cyber Science and Technology, Beihang University, Beijing, China
[2]State Key Laboratory of Software Development Environment, Beihang University, Beijing, China
{liyancong, yolixs*, cuiying_0719, mashuai}@buaa.edu.cn

Abstract

Temporal Knowledge Graphs (TKGs) represent a crucial source of structured temporal information and exhibit significant utility in various real-world applications. However, TKGs are susceptible to incompleteness, necessitating Temporal Knowledge Graph Completion (TKGC) to predict missing facts. Existing models have encountered limitations in effectively capturing the intricate temporal dynamics and hierarchical relations within TKGs. To address these challenges, HyGNet is proposed, leveraging hyperbolic geometry to effectively model temporal knowledge graphs. The model comprises two components: the Hyperbolic Gated Graph Neural Network (HGGNN) and the Hyperbolic Convolutional Neural Network (HCNN). HGGNN aggregates neighborhood information in hyperbolic space, effectively capturing the contextual information and dependencies between entities. HCNN interacts with embeddings in hyperbolic space, effectively modeling the complex interactions between entities, relations, and timestamps. Additionally, a consistency loss is introduced to ensure smooth transitions in temporal embeddings. The extensive experimental results conducted on four benchmark datasets for TKGC highlight the effectiveness of HyGNet. It achieves state-of-the-art performance in comparison to previous models, showcasing its potential for real-world applications that involve temporal reasoning and knowledge prediction.

Keywords: Knowledge Graph, Temporal Knowledge Graph, Temporal Knowledge Graph Completion, Hyperbolic Geometry, Poincaré ball, Graph Neural Networks

1. Introduction

Knowledge Graphs (KGs) (Hogan et al., 2021) are structured representations of knowledge that capture the relations between entities. As a type of ontology or semantic network, KGs consist of nodes representing individual entities and edges representing the relations among them. However, due to the vast and constantly expanding amount of data that can be included in a KG, it is often difficult to ensure completeness. As a result, research into knowledge graph completion (KGC) has become a topic of great interest in recent years (Bordes et al., 2013; Di et al., 2021; Trouillon et al., 2016).

One aspect that is often overlooked by existing methods is that the accuracy of most real-world facts depends on the time at which they are considered. To address this issue, temporal knowledge graphs (TKGs) (Boschee et al., 2015; Mahdisoltani et al., 2014; Erxleben et al., 2014) have been introduced to capture the temporal dependencies of real-world facts. Similar to static KGs, TKGs also suffer from incompleteness, which makes the task of temporal knowledge graph completion (TKGC) (Cai et al., 2023) particularly challenging. TKGC aims to infer missing facts and relations in a temporal context, taking into account the changing nature of the underlying data over time.

Facts in TKGs often exhibit a hierarchical structure (Zhang et al., 2020), where some facts are more general and abstract while others are more specific and concrete. For example, consider a TKG representing the history of a particular company. At a higher level, general facts like founding dates and locations of the company may be encountered, while at a lower level, more specific facts like names of individual employees and their job titles might exist. The current methods for completion of data in Euclidean space have limitations and distortions when representing large-scale hierarchical data. These limitations arise due to the inability of Euclidean metrics to capture the complex relationships present in the data (Sala et al., 2018; Sarkar, 2011). Hierarchical structures can be better embedded in hyperbolic space. Therefore, hyperbolic geometry is robust in capturing hierarchical patterns. This is due to the property of hyperbolic space where it has exponentially more space available in the vicinity of each point compared to Euclidean space (Nickel and Kiela, 2017). Consequently, this makes hyperbolic geometry an attractive choice for modeling hierarchical data.

The application of hyperbolic geometry to TKGs poses several challenges. One of the main challenges in applying hyperbolic geometry to TKGC is the development of effective algorithms for embedding temporal information in hyperbolic space. This requires consideration of the unique properties of hyperbolic space, such as its non-Euclidean curva-

*Corresponding author

LREC-COLING 2024, pages 8474–8486
20-25 May, 2024.

ture and negative curvature, which can impact the efficacy of existing embedding methods. Another challenge is the necessity to develop techniques for modeling intricate temporal relations among events occurring at different times, which may entail capturing complex patterns.

To model the hierarchical patterns and temporal properties present in TKGs, we introduce a novel approach that embeds these TKGs into hyperbolic space. Our proposed model consists of a gated hyperbolic graph neural network that aggregates information from neighboring entities within the TKG, as well as a hyperbolic convolutional neural network that captures heterogeneous interactions between entities, relations, and timestamps.

The contributions can be summarized as follows:

- A hyperbolic graph neural network is introduced, enabling the embedding of TKGs into hyperbolic space, which accommodates hierarchical patterns and temporal properties.
- The model consists of a gated hyperbolic graph neural network for neighborhood information aggregation and a hyperbolic convolutional neural network for modeling heterogeneous interactions.
- By leveraging the intrinsic curvature properties of hyperbolic geometry, the model achieves state-of-the-art performance on four benchmark datasets.

2. Related Work

2.1. Static Knowledge Graph Completion

Static knowledge graph completion (KGC) models aim to infer missing facts from static KGs. These time-agnostic models assume invariable semantics for entities and relations.

TransE (Bordes et al., 2013) is a canonical model that treats relations as translations within entity embeddings. TransE has sparked numerous subsequent extensions and variations in research endeavors, each exploring distinct methods for capturing the intricate structures inherent in KGs (Wang et al., 2014; Lin et al., 2015; Ji et al., 2015). Besides translation-based models, there are also other types of models for static KGC, such as the semantic matching models(Nickel et al., 2011; Yang et al., 2015; Trouillon et al., 2016), the tensor factorization models(Balažević et al., 2019; Di et al., 2021), and the neural network models(Vashishth et al., 2020; Rosso et al., 2020).

2.2. Temporal Knowledge Graph Completion

Recent research has focused on incorporating temporal information into KGC models to enhance their performance. The main objective of these models is to infer missing facts from a TKG that exhibits time-dependent characteristics.

TeRo (Xu et al., 2020a) models the temporal evolution of entity embeddings through rotation in the complex vector space, spanning from the initial time to the current time. TeLM (Xu et al., 2021) employs multivector embeddings along with a linear temporal regularizer for 4th-order tensor factorization of a TKG. SANe (Li et al., 2022) maps facts with different timestamps into distinct latent spaces and explores the overlap in these spaces, thereby capturing both the time-variability and time-stability within TKGs.

2.3. Hyperbolic Geometry-Based Approaches

Hyperbolic spaces, known for their properties of exponential growth, are considered highly effective for processing data characterized by hierarchical or power-law distributions. In the realm of hyperbolic graph neural networks, significant progress has been reported, notably in the works reviewed by (Yang et al., 2022) and proposed by (Liu et al., 2019). Yet, these advancements are mainly tailored to undirected and static graphs, which limits their effectiveness for TKGC tasks.

The introduction of hyperbolic geometry to TKGs adds a new level of complexity, particularly in modeling the evolution of entities and their relations over time. The contributions made by (Dasgupta et al., 2018) and (Montella et al., 2021) represent important steps in incorporating temporal aspects into KGs. However, they face challenges in aggregating neighborhood information for entities, a key factor in capturing the dynamics within TKGs fully.

In contrast to the works of (Yang et al., 2022) and (Liu et al., 2019), which excel in the context of undirected and static graphs but struggle with applicability to TKGs, our model incorporates hyperbolic geometry to address the complexity of temporal and hierarchical data in TKGC tasks. Moreover, our approach introduces a time-aware gating mechanism in hyperbolic spaces, in contrast to the methods of (Dasgupta et al., 2018) and (Montella et al., 2021), which lack in aggregating entity neighborhood information. This mechanism not only aids in modeling temporal relations but also improves the aggregation of entities' adjacency information.

3. Preliminary

Hyperbolic geometry is characterized by a space of constant negative curvature (Cannon et al., 1997). In contrast to Euclidean space, which has zero curvature, hyperbolic geometry exhibits an exponential increase in space with radius, rather than a polynomial increase. The Poincaré ball model (Birman and Ungar, 2001) in hyperbolic geometry is selected due to its compatibility with gradient-based optimization and its intuitive visualization of hyperbolic embedding.

The Poincaré ball is formally defined as an open d-dimensional ball with negative curvature $-c$ ($c > 0$), denoted as $\mathbb{B}^{d,c} = \{\mathbf{x} \in \mathbb{R}^d : c\|\mathbf{x}\|^2 < 1\}$, where $\|\cdot\|_2$ represents the Euclidean norm. In the Poincaré ball model, each point $\mathbf{x}$ within $\mathbb{B}^{d,c}$ has a continuous bijection between its neighborhood and $\mathbb{R}^d$, with a continuous inverse mapping.

For $\mathbf{x} \in \mathbb{B}^{d,c}$, the local first-order approximation of the Poincaré ball at $\mathbf{x}$ can be characterized by the tangent space $\mathcal{T}\mathbf{x}\mathbb{B}^{d,c}$, which is isomorphic to $\mathbb{R}^d$ in Euclidean geometry (Wilson et al., 2014). In other words, $\mathcal{T}\mathbf{x}\mathbb{B}^{d,c} = \mathbb{R}^d$. The exponential map serves as a mapping from the tangent space $\mathcal{T}\mathbf{x}\mathbb{B}^{d,c}$ to $\mathbb{B}^{d,c}$, while the logarithmic map performs the reverse, mapping points from $\mathbb{B}^{d,c}$ to the tangent space $\mathcal{T}\mathbf{x}\mathbb{B}^{d,c}$.

For $\mathbf{x} \in \mathbb{B}^{d,c}$, the exponential map $\exp_{\mathbf{x}}^c$ and the logarithmic map $\log_{\mathbf{x}}^c$ are defined as follows for $\mathbf{v} \neq \mathbf{0}$ and $\mathbf{y} \neq \mathbf{x}$:

$$\exp_{\mathbf{x}}^c(\mathbf{v}) = \mathbf{x} \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda_{\mathbf{x}}^c\|\mathbf{v}\|}{2}\right)\frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}\right), \tag{1}$$

$$\log_{\mathbf{x}}^c(\mathbf{y}) = \frac{2}{\sqrt{c}\lambda_{\mathbf{x}}^c}\tanh^{-1}\left(\sqrt{c}\|-\mathbf{x}\oplus_c\mathbf{y}\|\right)\frac{-\mathbf{x}\oplus_c\mathbf{y}}{\|-\mathbf{x}\oplus_c\mathbf{y}\|} \tag{2}$$

where $\oplus_c$ represents the Möbius addition for any $\mathbf{x}, \mathbf{y} \in \mathbb{B}^{d,c}$:

$$\mathbf{x}\oplus_c\mathbf{y} = \frac{\left(1+2c\langle\mathbf{x},\mathbf{y}\rangle + c\|\mathbf{y}\|^2\right)\mathbf{x} + \left(1-c\|\mathbf{x}\|^2\right)\mathbf{y}}{1+2c\langle\mathbf{x},\mathbf{y}\rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}. \tag{3}$$

Here, $\lambda_{\mathbf{x}}^c$ denotes the hyperbolic metric at point $\mathbf{x}$.

Without loss of generality, the exponential map and logarithmic map in the Poincaré ball model are defined with the origin $\mathbf{0} \in \mathbb{B}^{d,c}$ as a fixed point. Therefore, Equations 1 and 2 can be reformulated as:

$$\exp_{\mathbf{0}}^c(\mathbf{v}) = \tanh\left(\sqrt{c}\|\mathbf{v}\|\right)\frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}, \tag{4}$$

$$\log_{\mathbf{0}}^c(\mathbf{y}) = \tanh^{-1}\left(\sqrt{c}\|\mathbf{y}\|\right)\frac{\mathbf{y}}{\sqrt{c}\|\mathbf{y}\|}. \tag{5}$$

In the Poincaré ball model, matrix multiplication can be performed to project a point $\mathbf{x} \in \mathbb{B}^{d,c}$ onto the tangent space $\mathcal{T}_{\mathbf{0}}\mathbb{B}^{d,c}$ at $\mathbf{0}$ using the logarithmic map. This projected point is then multiplied by the weight matrix $\mathbf{W} \in \mathbb{R}^{d\times k}$, and subsequently re-projected back onto the Poincaré ball using the exponential map:

$$\mathbf{W}\otimes_c\mathbf{x} = \exp_{\mathbf{0}}^c\left(\mathbf{W}\log_{\mathbf{0}}^c(\mathbf{x})\right). \tag{6}$$

Finally, in the Poincaré ball model, induced distance (Faraki et al., 2018) between two points $\mathbf{x}, \mathbf{y} \in \mathbb{B}^{d,c}$ is defined as:

$$d_{\mathbb{B}}^c(\mathbf{x},\mathbf{y}) = \frac{2}{\sqrt{c}}\tanh^{-1}\left(\sqrt{c}\|-\mathbf{x}\oplus_c\mathbf{y}\|\right). \tag{7}$$

4. Methodology

A temporal knowledge graph (TKG) can be represented by a set of quadruples $\mathcal{G} = \{(h,r,t,\tau)\,|\,h,t \in \mathcal{E}, r \in \mathcal{R}, \tau \in \mathcal{T}\}$. Here, $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ represent sets of entities, relations, and timestamps, respectively. Each quadruple corresponds to a time-dependent fact that establishes a connection between a head entity h and a tail entity t through the relation r at the timestamp τ. The objective of TKGC is to predict the missing tail entity t or head entity h in response to a query $(h, r, ?, \tau)$ or $(?, r, t, \tau)$, respectively, by leveraging the available temporal facts. Specifically, TKGC focuses solely on predicting missing facts at observed timestamps, which is known as the interpolation task (Jin et al., 2020).

To address the challenges of TKGC, we propose a **Hy**perbolic **G**raph **N**eural N**et**work (HyGNet) as shown in Figure 1. The HyGNet model consists of two key sub-modules. The first sub-module is a hyperbolic gated graph neural network (HGGNN) that aggregates the neighborhood information of entities in the hyperbolic space. The HGGNN captures the contextual information and dependencies among entities, enabling a more comprehensive understanding of their interactions and dynamics within the TKG. The second sub-module is a hyperbolic convolutional neural network (HCNN). The HCNN integrates the embeddings in a hyperbolic space, allowing for a more effective modeling of the complex interactions between entities, relations, and timestamps.

Specifically, an entity aggregates information from its neighboring entities and relations in hyperbolic space, with time-dependent gating units controlling the weights during aggregation. The aggregation takes place in the tangent space, resulting in the embedded representation of the entity. Subsequently, the embeddings of the head entity, relation, and time are stacked and subjected

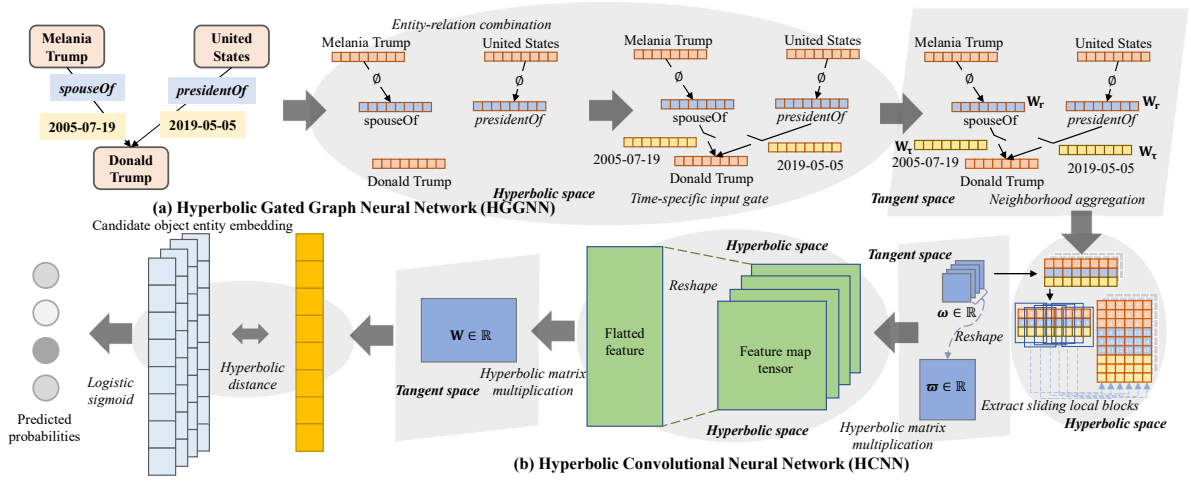

Figure 1: Overview of HyGNet. HGGNN aggregates neighborhood information within the hyperbolic geometry framework, considering temporal dependencies. HCNN defines convolution operations in hyperbolic space to capture intricate interactions.

to convolutional operations. Hyperbolic convolution operations are transformed into principled hyperbolic matrix multiplication. Through hyperbolic linear transformations, the feature tensor is manipulated to obtain the representation of the predicted entity.

4.1. Hyperbolic Gated Graph Neural Network

The Hyperbolic Gated Graph Neural Network (HGGNN) serves as an extension of inductive graph neural networks within the hyperbolic geometry framework. HGGNN benefits from both the expressive capabilities of graph neural networks and the representational power of hyperbolic embeddings. HGGNN employs hyperbolic geometry to aggregate the neighborhood information of entities within hyperbolic space. It leverages the temporal information to determine the extent of directed edge information flow.

Given an embedding $\mathbf{h} \in \mathbb{B}^{d,c}$ of an entity and an embedding $\mathbf{r} \in \mathbb{B}^{d,c}$ of a relation within the hyperbolic space, the integration of relation embeddings into the aggregation process of entity neighborhoods utilizes the entity-relation combination operation commonly employed in knowledge graph embedding methodologies (Vashishth et al., 2019). This operation is formulated as:

$$\hat{\mathbf{h}} = \phi\left(\mathbf{h}, \mathbf{r}\right), \tag{8}$$

where d represents the embedding dimension and c denotes trainable curvature in hyperbolic space. Here, $\phi : \mathbb{B}^{d,c} \times \mathbb{B}^{d,c} \rightarrow \mathbb{B}^{d,c}$ is a composition operator. In the context of HGGNN, $\phi\left(\cdot\right)$ is constrained to non-parametric Möbius addition, i.e., $\hat{\mathbf{h}} = \mathbf{h} \oplus_c -\mathbf{r}$. However, it should be noted that the versatility of $\phi\left(\cdot\right)$ can extend to parametric operations such as hyperbolic matrix multiplication, though such analysis is deferred to future investigations.

Before performing neighborhood aggregation, a gating mechanism is devised to modulate the extent to which $\hat{\mathbf{h}}$ can be propagated. This gating mechanism is influenced by temporal information, determining the degree of passage for $\hat{\mathbf{h}}$.

Consequently, based on a temporal embedding $\boldsymbol{\tau} \in \mathbb{B}^{d,c}$, which determines the information passage along edges, an input gate is formulated as follows:

$$\begin{aligned} i_\tau = \sigma\Big(\log_{\mathbf{0}}^c \Big(\exp_{\mathbf{0}}^c(\mathbf{W}_{\lambda(\hat{h})} \log_{\mathbf{0}}^c(\hat{\mathbf{h}})) \\ \oplus_c \exp_{\mathbf{0}}^c(\mathbf{W}_{\lambda(\tau)} \log_{\mathbf{0}}^c(\boldsymbol{\tau})) \Big) \Big), \end{aligned} \tag{9}$$

where $\mathbf{W}_{\lambda(\hat{h})}, \mathbf{W}_{\lambda(\tau)} \in \mathbb{R}^{d \times d}$.

Following the processing by the input gate i_τ, the adjusted $\mathbf{h}_{i_\tau}$ for subsequent neighborhood aggregation is derived:

$$\mathbf{h}_{i_\tau} = \exp_{\mathbf{0}}^c \left(\log_{\mathbf{0}}^c(\hat{\mathbf{h}}) * i_\tau \right), \tag{10}$$

where $*$ denotes element-wise multiplication, resulting in $\mathbf{h}_{i_\tau} \in \mathbb{B}^{d,c}$.

Furthermore, information related to directed relations is allowed to propagate in both the original and reverse directions. Therefore, $\mathcal{G}$ is extended to encompass corresponding reverse relations and self-loops of entities, yielding:

$$\begin{aligned} \mathcal{G}' = \{(h, r, t, \tau)\} \cup \{(t, r^{-1}, h, \tau)\} \\ \cup \{(h, r_{loop}, h, \tau)\} \cup \{t, r_{loop}, t, \tau\}, \end{aligned} \tag{11}$$

where $r^{-1} \in \mathcal{R}$ denotes the inverse relation, and r_{loop} signifies self-loop relation. To distinguish be-

tween original, inverse, and self-loop relations, separate weight matrices are defined for them. Hence, the $\mathbf{W}_{\lambda(\hat{h})}$ and $\mathbf{W}_{\lambda(\tau)}$ in Equation 9 are set with relation-specific parameters based on their directions. Taking $\mathbf{W}_{\lambda(\hat{h})}$ as an example, let $\lambda(\hat{h}) = dir(r)$, which is defined as:

$$\mathbf{W}_{\lambda(\hat{h})} = \begin{cases} \mathbf{W}_o, \ \mathcal{G} = \{(h, r, t, \tau)\} \\ \mathbf{W}_i, \ \mathcal{G} = \{(t, r^{-1}, h, \tau)\} \\ \mathbf{W}_s, \ \{(h, r_{loop}, h, \tau)\} \cup \{t, r_{loop}, t, \tau\} \end{cases} \tag{12}$$

A similar procedure is adopted for $\mathbf{W}_{\lambda(\tau)}$ to capture direction-specific parameters.

Aggregation stands as a pivotal step in HGGNN, as it captures the structural and feature information of neighboring entities. Graph neural networks involve a sequence of fundamental operations, including linear mappings and element-wise non-linear message passing over a set of entities in a given space. After processing $\mathbf{h}_{i_\tau}$ as described in Equation 10, a hyperbolic aggregation is performed as follows:

$$\mathbf{h}_v^{l+1} = f\left(\exp_{\mathbf{0}}^c\left(\sum_{u \in \mathcal{N}(v)} \log_{\mathbf{0}}^c\left(\mathbf{h}_{i_\tau}^l\right)\right)\right). \tag{13}$$

In layer l, the logarithmic projection $\log_{\mathbf{0}}^c$ is employed to map $\mathbf{h}_{i_\tau}$ onto the tangent space for aggregation, and the exponential projection $\exp_{\mathbf{0}}^c$ is utilized to project the aggregated tangent vectors back onto the hyperbolic space. Here, f represents a hyperbolic non-linear activation function satisfying $f : \mathbb{B}^{d,c} \to \mathbb{B}^{d,c}$, and takes the form $\exp_{\mathbf{0}}^c(\sigma \log_{\mathbf{0}}^c(\cdot))$, where σ is a Euclidean non-linear activation. The choice of performing aggregation in the tangent space is rooted in the fact that the tangent space of a point on the hyperbolic space is always Euclidean or a subset of Euclidean space.

Following the entity embedding updates defined in Equation 13, transformations are applied to the relation embeddings $\mathbf{r}^l$ and time embeddings $\boldsymbol{\tau}^l$ at layer l as follows:

$$\mathbf{r}^{l+1} = \exp_{\mathbf{0}}^c\left(\mathbf{W}_r^l \log_{\mathbf{0}}^c\left(\mathbf{r}^l\right)\right), \tag{14}$$

$$\boldsymbol{\tau}^{l+1} = \exp_{\mathbf{0}}^c\left(\mathbf{W}_\tau^l \log_{\mathbf{0}}^c\left(\boldsymbol{\tau}^l\right)\right). \tag{15}$$

Here, $\mathbf{W}_r, \mathbf{W}_\tau \in \mathbb{R}^{d \times d}$ represent learnable transformation matrices.

4.2. Hyperbolic Convolutional Neural Network

Convolutional Neural Networks (CNNs) are widely employed in various fields for their ability to capture intricate spatial features within data. In the context of TKGC, to comprehensively capture the intricate interactions between entities, relations, and timestamps, we propose the Hyperbolic Convolutional Neural Network (HCNN). The essence of HCNN lies in its definition of convolution operations in hyperbolic space, which is an aspect often missing in existing TKGC methodologies.

Considering the absence of a direct convolution definition in hyperbolic geometry, the convolution operation is transformed into well-defined hyperbolic matrix multiplication. To elaborate, the key idea is to map the convolution kernel and corresponding feature mappings within sliding local blocks into matrices, thereby converting hyperbolic convolution into matrix multiplication.

Upon processing through the HGGNN, the embedded representations of entities, relations, and timestamps, denoted as $\mathbf{h} \in \mathbb{B}^{d,c}$, $\mathbf{r} \in \mathbb{B}^{d,c}$, and $\boldsymbol{\tau} \in \mathbb{B}^{d,c}$, respectively, are concatenated into a 2D matrix, represented as $[\mathbf{h}, \mathbf{r}, \boldsymbol{\tau}] \in \mathbb{B}^{3 \times d,c}$. Given the matching dimensions of $\mathbf{h}$, $\mathbf{r}$, and $\boldsymbol{\tau}$, alongside the fact that all operations are conducted in a hyperbolic space with a learnable curvature c, this implies that all entities, relations, and timestamps are projected into the same hyperbolic space. Consequently, the stacking, reshaping, and subsequent processing of embeddings are feasible within this unified hyperbolic space.

To convert the hyperbolic convolution operation into principled matrix multiplication within hyperbolic geometry, a function $\varphi(\cdot)$ is defined. It extracts sliding local blocks from $[\mathbf{h}, \mathbf{r}, \boldsymbol{\tau}] \in \mathbb{B}^{3 \times d,c}$ based on learnable filters $\boldsymbol{\omega} \in \mathbb{R}^{n \times 3 \times k}$, where n denotes the number of filters, and k indicates the filter size. The extraction process is illustrated in Figure 1. Prior to extracting the sliding local blocks, zero padding is applied to $[\mathbf{h}, \mathbf{r}, \boldsymbol{\tau}] \in \mathbb{B}^{3 \times d,c}$. Consequently, the extracted feature matrix is represented as $\mathbf{x} \in \mathbb{B}^{d \times 3k,c}$ (with a fixed stride of 1). A set of learnable filters $\boldsymbol{\omega} \in \mathbb{R}^{n \times 3 \times k}$ is introduced, which is employed to extract sliding local blocks from the concatenated embeddings $[\mathbf{h}, \mathbf{r}, \boldsymbol{\tau}] \in \mathbb{B}^{3 \times d,c}$, where n represents the number of filters, and k indicates the filter size. The extraction process is illustrated in Figure 1. Similarly, the filter $\boldsymbol{\omega} \in \mathbb{R}^{n \times 3 \times k}$ is reshaped into $\boldsymbol{\varpi} \in \mathbb{R}^{n \times 3k}$ to facilitate matrix multiplication. In this case, the "convolution" operation is transformed into the following operation:

$$\mathbf{x} \circledast^c \boldsymbol{\varpi} := \exp_{\mathbf{0}}^c\left(\log_{\mathbf{0}}^c\left(\mathbf{x}\right) \boldsymbol{\varpi}^\top\right) \oplus_c \mathbf{b}_1, \tag{16}$$

where $\mathbf{b}_1 \in \mathbb{B}^{nd}$ is a bias vector.

Subsequently, the result obtained from $\mathbf{x} \circledast^c \boldsymbol{\varpi}$ is folded and reshaped into a tensor $\hat{\mathbf{x}} \in \mathbb{B}^{n \times d,c}$. Since matrix multiplication is well-defined in hyperbolic geometry, the "convolution" in Equation 16 is feasible. To enhance the expressiveness of the model, a element-wise nonlinearity $\sigma^{\otimes_c}(\cdot)$ is applied to $\hat{\mathbf{x}}$. To ensure that the features after the pointwise non-linearity still fall within the Poincaré

ball, $\sigma^{\otimes_c}(\cdot)$ should guarantee $\|\sigma^{\otimes_c}(\hat{\mathbf{x}}) \leq \hat{\mathbf{x}}\|$, satisfying $\sigma^{\otimes_c} : \mathbb{B} \to \mathbb{B}$. Consequently, the non-linear activation functions Rectified Linear Unit (ReLU) (Glorot et al., 2011) and Leaky Rectified Linear Unit (Leaky ReLU) (Maas et al., 2013) are applicable. To sum up, the operation of hyperbolic convolutional neural network operates as follows:

$$\tilde{\mathbf{x}} = \sigma^{\otimes_c}\left(\mathbf{x} \circledast^c \boldsymbol{\varpi}\right), \tag{17}$$

where $\tilde{x}$ represents a feature map tensor, and $\tilde{x} \in \mathbb{B}^{n \times d,c}$.

Next, a linear transformation is applied to $\tilde{\mathbf{x}}$. First, $\tilde{\mathbf{x}}$ is flattened and reshaped into a vector, denoted as $vec(\tilde{\mathbf{x}}) \in \mathbb{B}^{nd,c}$. A matrix $\mathbf{W} \in \mathbb{R}^{d \times nd}$ is defined to perform hyperbolic matrix multiplication on the flattened vector. Therefore, we have:

$$\begin{aligned} \mathbf{v} &= \mathbf{W} \otimes_c vec\left(\tilde{\mathbf{x}}\right) \oplus_c \mathbf{b}_2 \\ &= \exp_{\mathbf{0}}^c\left(\mathbf{W} \log_{\mathbf{0}}^c\left(vec\left(\tilde{\mathbf{x}}\right)\right)\right) \oplus_c \mathbf{b}_2, \end{aligned} \tag{18}$$

where $\mathbf{v} \in \mathbb{B}^{d,c}$ represents the transformed feature vector, $\otimes_c$ denotes hyperbolic matrix multiplication, $\oplus_c$ signifies the Möbius addition, and $\mathbf{b}_2 \in \mathbb{B}^{d,c}$ is a bias vector.

To predict the query $q = (h, r, ?, \tau)$, the hyperbolic induced distance between $\mathbf{v} \in \mathbb{B}^{d,c}$ and the embedded candidate tail entity $\mathbf{t} \in \mathbb{B}^{d,c}$, after processing by HGGNN, is computed as follows:

$$d_{\mathbb{B}}^c\left(\mathbf{v}, \mathbf{t}\right) = \frac{2}{\sqrt{c}} \tanh^{-1}\left(\sqrt{c}\|-\mathbf{v} \oplus_c \mathbf{t}\|\right). \tag{19}$$

The distance between $\mathbf{v}$ and the embedding of the correct tail entity is expected to be minimized. The smaller the hyperbolic induced distance between $\mathbf{v}$ and $\mathbf{t}$, the more likely the quadruple is positive, and vice versa.

Therefore, the formal definition of the scoring function is as follows:

$$\begin{aligned} \psi\left(h, r, t, \tau\right) = -d_{\mathbb{B}}^c\Big(\mathbf{W} \otimes_c vec\Big(\sigma^{\otimes_c}((\varphi([\mathbf{h}, \mathbf{r}, \boldsymbol{\tau}]) \\ \circledast^c \boldsymbol{\varpi}) \oplus_c \mathbf{b}_1)\Big) \oplus_c \mathbf{b}_2, \mathbf{t}\Big). \end{aligned} \tag{20}$$

4.3. Training and Optimization

During the training process, the score $\psi\left(h, r, t, \tau\right)$ is transformed via the logistic sigmoid function $\sigma\left(\cdot\right)$, resulting in $p(h, r, t, \tau) = \sigma\left(\psi\left(h, r, t, \tau\right)\right)$. This transformation yields $p(h, r, t, \tau)$, which represents the predicted probability that the candidate tail entity t corresponds to the query $(h, r, ?, \tau)$. The training objective aims to minimize the negative log-likelihood loss:

$$\begin{aligned} \mathcal{L}_t = -\frac{1}{|\mathscr{G}|} \sum_{(h,r,t,\tau) \in \mathscr{G}} \Big(\log p(h, r, t, \tau) \\ + |S_t| \sum_{\tilde{t} \in S_t} \log(1 - p(h, r, \tilde{t}, \tau))\Big) \end{aligned} \tag{21}$$

where S_t represents the negative samples by replacing the tail entity of (h, r, t, τ) as $\tilde{t}$.

To ensure temporal stability in the embeddings of adjacent timestamps and to mitigate abrupt changes, a regularization constraint is applied. This constraint enforces a smooth transition between embeddings corresponding to temporally adjacent entities, promoting consistency and coherence in the representation of temporal information. The objective behind this regularization is to enhance the model's ability to capture gradual variations in temporal dynamics, aligning with the underlying assumption of smooth temporal transitions in TKGs.

Specifically, for each timestamp $\tau \in \mathcal{T}$, we derive embeddings denoted by $\boldsymbol{\tau} \in \mathbb{B}^{d,c}$. The set O is then constituted as $\{\boldsymbol{\tau} | \tau \in \mathcal{T}\}$ and sorted based on timestamps. The consistency loss as follows:

$$\mathcal{L}_c = \frac{1}{|O|} \sum_{i=1}^{|O|-1} d_{\mathbb{B}}^c\left(O_{i+1} - O_i\right), \tag{22}$$

where $d_{\mathbb{B}}^c(\cdot, \cdot)$ computes the hyperbolic induced distance, and O_i represents the i-th element in O.

The overall loss function for training HyGNet is defined as follows:

$$\mathcal{L} = \mathcal{L}_t + \lambda \mathcal{L}_c, \tag{23}$$

where λ represents a hyperparameter that controls the relative importance of the two losses.

5. Experiments

5.1. Experimental setup

Datasets Four publicly available benchmark datasets are employed to evaluate the proposed model, including ICEWS14 (Garcia-Duran et al., 2018), ICEWS05-15 (Garcia-Duran et al., 2018), YAGO11k (Dasgupta et al., 2018), and Wikidata12k (Dasgupta et al., 2018). ICEWS14 and ICEWS05-15 consist of discrete time-annotated sociopolitical events, such as (*Barack Obama, Make a visit, South Korea, 2014-03-15*). On the other hand, facts in both YAGO11k and Wikidata12k are accompanied by time annotations, with each fact represented as a time interval. To conform to the processing requirements, the facts with time intervals are discretized into multiple quadruplets, each associated with a single timestamp. Table 1 summarizes the statistics of these four benchmark datasets.

Baselines HyGNet is compared with both static and temporal KGC models. Static KGC models, which do not consider temporal information, include TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), RotatE (Sun et al., 2019), ComplEx-N3 (Lacroix et al., 2018) and QuatE^2 (Zhang et al.,

Table 1: Statistics of TKGC benchmark datasets. The time span is measured in years.

Datasets	#Entities	#Relations	Time span	#Train	#Valid	#Test
ICEWS14	6,869	230	2014	72,826	8,941	8,963
ICEWS05-15	10,094	251	2005-2015	386,962	46,275	46,092
YAGO11k	10,623	10	-453-2844	16,408	2,050	2,051
Wikidata12k	12,554	24	1709-2018	32,497	4,062	4,062

2019). In contrast, temporal KGC models are designed to process temporal information and capture evolving patterns in TKGs, such as TTransE (Leblay and Chekol, 2018), HyTE (Dasgupta et al., 2018), ATiSE (Xu et al., 2020b), TeRo (Xu et al., 2020a), TimePlex (Jain et al., 2020), TComplEx (Lacroix et al., 2020), TeLM (Xu et al., 2021), SANe (Li et al., 2022), and TGeomE+ (Xu et al., 2023).

Evaluation Metrics For each quadruple (h, r, t, τ), simultaneous optimization of the model is achieved by leveraging two queries, namely $(h, r, ?, \tau)$ and $(?, r, t, \tau)$. In practical terms, a reciprocal relation (t, r^{-1}, h, τ) is introduced for each quadruple (h, r, t, τ). Consequently, the query $(?, r, t, \tau)$ is substituted with $(t, r^{-1}, ?, \tau)$. These operations do not result in a loss of generality (Jain et al., 2020; Xu et al., 2021).

Evaluation metrics such as MRR (Mean Reciprocal Rank) and Hits@N are utilized for assessing the performance of the model. MRR is calculated as the average of the reciprocal values of all computed ranks. Hits@N represents the percentage of times the true entity candidate appears among the top N ranked candidates, where $\mathrm{N} \in \{1, 3, 10\}$. Among these metrics, MRR holds significance as an evaluation index that is less susceptible to outliers (Garcia-Duran et al., 2018). Higher values of MRR and Hits@N indicate superior performance of the model. All evaluations are conducted under the widely adopted time-wise filtering setting, as employed in previous studies (Xu et al., 2020a).

Implementation details Hyperparameters are determined based on the MRR performance on the validation set. All embeddings are initialized near the origin in the tangent space of the hyperbolic space and then recovered using the exponential mapping. The Adam optimizer is employed for optimization, and the learning rate is chosen from $\{0.05, 0.002, 0.001, 0.0005\}$. During training, 256 mini-batches are created for each epoch. The number of training epochs is set to 1000. λ is fixed at 0.0001. The embedding dimension is set to $d = 300$ for ICEWS05-15, while for other datasets, it is set to $d = 200$. The number of convolutional filters is fixed at 128, and the kernel size is chosen from $k \in \{3, 5, 7, 9\}$. The code will be published on https://github.com/codeofpaper/HyGNet.

5.2. Main Results

The experimental results are summarized in Tables 2 and 3. Most TKGC models outperform static KGC models, highlighting the importance of considering temporal information.

The proposed HyGNet exhibits remarkable performance across all datasets, achieving state-of-the-art results. The model exhibits even more remarkable performance on the YAGO11k and Wikidata12k datasets. Notably, these datasets, which encompass encyclopedic knowledge, are characterized by hierarchical and structured knowledge, in contrast to event-based ICEWS datasets. This observation suggests that HyGNet is particularly well-suited to model hierarchical knowledge structures.

5.3. Ablation Study

To verify the effectiveness of various components of the proposed model, four variants of HyGNet are explored. The results of are presented in Table 4, where symbol ✓denotes the presence of specific components in the experiment, and symbol ✗signifies their absence. Several key observations can be made: (1) The performance drop in V1 when HGGNN is removed indicates the importance of aggregating neighborhood information in hyperbolic space. (2) V1 outperforms its corresponding Euclidean version, V3, and HyGNet outperforms its Euclidean counterpart, V4, underscoring the effectiveness of leveraging hyperbolic geometry for modeling TKGs. (3) The slight performance decrease in V4 when consistency loss is removed suggests that considering smoothness in temporal information contributes to performance improvement. (4) HyGNet achieves the best performance, indicating the effectiveness of all its components.

5.4. Analysis

Generalizing to Unseen Timestamps. In practical applications, TKG data may contain timestamps that have not been previously encountered. Validating the model's performance on unseen timestamps allows for an assessment of its robustness and generalization capability when confronted with novel data. Consistent with Goel et al. (Goel et al., 2020), we resample ICEWS14 by removing the

Table 2: Link prediction results on ICEWS14 and ICEWS05-15. Results marked with $*$, †, and $\diamond$ are taken from the studies by (Garcia-Duran et al., 2018), (Xu et al., 2020a), and (Xu et al., 2021), respectively. Dashes indicate unobtainable results, and all other results are from the original papers.

Datasets	ICEWS14				ICEWS05-15			
Metrics	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1	MRR
TransE*	.637	–	.094	.280	.663	–	.090	.294
DistMult*	.672	–	.323	.439	.691	–	.337	.456
RotatE†	.690	.478	.291	.418	.595	.355	.164	.304
ComplEx-N3†	.716	.527	.347	.467	.729	.535	.362	.481
QuatE^{2}†	.712	.530	.353	.471	.727	.529	.370	.482
TTransE†	.601	–	.074	.255	.616	–	.084	.271
HyTE†	.655	.416	.108	.297	.681	.445	.116	.316
ATiSE	.757	.632	.423	.545	.803	.623	.394	.533
TeRo	.732	.621	.468	.562	.795	.668	.469	.586
TimePlex	.771	–	.515	.604	.818	–	.545	.640
TComplEx$^{\diamond}$	.770	.660	.530	.610	.800	.710	.590	.660
TeLM$^{\diamond}$	.774	.673	.545	.625	.823	.728	.599	.678
SANe	.782	.688	.558	.638	.823	.734	.605	.683
TGeomE+	.780	.678	.545	.628	.831	.735	.603	.684
HyGNet	**.783**	**.691**	**.568**	**.645**	**.837**	**.747**	**.612**	**.693**

Table 3: Comparison of link prediction results on YAGO11k and Wikidata12k. Results marked with $*$, †, and $\diamond$ are taken from (Xu et al., 2020b), (Xu et al., 2020a), and (Xu et al., 2021), respectively. Dashes indicate unobtainable results, and all other results are from the original papers.

Datasets	YAGO11k				Wikidata12k			
Metrics	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1	MRR
TransE*	.244	.138	.015	.100	.339	.192	.100	.178
DistMult*	.268	.161	.107	.158	.460	.238	.119	.222
RotatE*	.305	.167	.103	.167	.461	.236	.116	.221
ComplEx-N3*	.282	.154	.106	.167	.436	.253	.123	.233
QuatE^{2*}	.270	.148	.107	.164	.416	.243	.125	.230
TTransE†	.251	.150	.020	.108	.329	.184	.096	.172
HyTE†	.272	.143	.015	.105	.333	.197	.098	.180
ATiSE	.301	.189	.126	.185	.462	.288	.148	.252
TeRo†	.319	.197	.121	.187	.507	.329	.198	.299
TimePlex	.367	–	.169	.236	.532	–	.228	.334
TComplEx$^{\diamond}$	.307	.183	.127	.185	.539	.357	.233	.331
TeLM$^{\diamond}$	.321	.194	.129	.191	.542	.360	.231	.332
SANe	.401	.266	.180	.250	.640	.483	.331	.432
TGeomE+	.327	.198	.130	.195	.546	.361	.232	.333
HyGNet	**.442**	**.295**	**.195**	**.276**	**.671**	**.503**	**.341**	**.450**

quadruples of the 5^{th}, 15^{th}, and 25^{th} days of each month from the training set, and randomly split the removed quadruples into the validation set and test set. The results obtained in Table 5 indicate that HyGNet outperforms SANe with a 6.8% improvement in MRR, demonstrating the effectiveness of HyGNet in generalizing to unseen timestamps.

Performance Evaluation on Various Relations. To assess the efficacy of HyGNet in hierarchical modeling, evaluations are conducted on several relations within YAGO11k, specifically *playsFor*, *hasWonPrize*, *graduatedFrom*, and *isAffiliatedTo*. These relations inherently exhibit hierarchical characteristics, such as a university graduating numerous students in a particular year. HyGNet is expected to demonstrate superior performance on

Table 4: Comparison of results for different variations of the model on ICEWS14.

Variant Models	HGGNN	HCNN	HGGNN$_{euc}$	HCNN$_{euc}$	Consistency Loss	Hits@10	Hits@3	Hits@1	MRR
V1	✗	✓	✗	✗	✓	.780	.686	.563	.640
V2	✗	✗	✓	✓	✓	.748	.656	.534	.610
V3	✗	✗	✗	✓	✓	.725	.636	.517	.590
V4	✓	✓	✗	✗	✗	.782	.683	.558	.636
HyGNet	✓	✓	✗	✗	✓	**.783**	**.691**	**.568**	**.645**

Table 5: Evaluation of generalization performance on ICEWS14 dataset for queries with unseen timestamps.

Metrics	Hits@10	Hits@3	Hits@1	MRR
DistMult	.620	.462	.302	.410
DE-SimplE	.624	.492	.333	.434
TComplEx	.625	.492	.348	.443
SANe	.709	.569	.394	.503
HyGNet	**.732**	**.605**	**.431**	**.537**

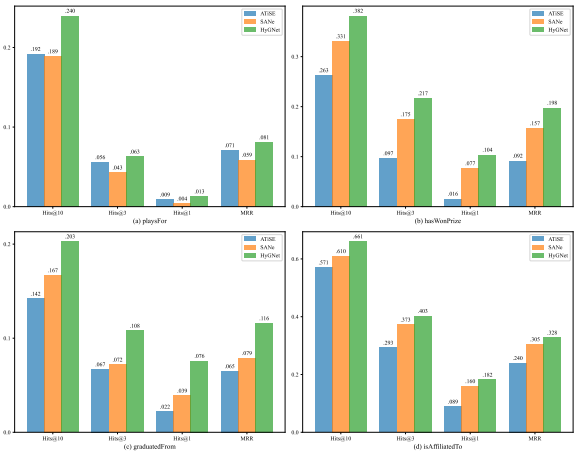

Figure 2: Results from several relations in YAGO11k achieved by ATiSE (Xu et al., 2020b), SANe (Li et al., 2022), and HyGNet.

such hierarchical relations. The outcomes presented in Figure 2 indicate that HyGNet outperforms other methods across all metrics. This validates the effectiveness of HyGNet in modeling inherently hierarchical knowledge.

Complexity and Parameter Count Analysis. To evaluate the computational efficiency of HyGNet, it is compared with existing baseline models in terms of space complexity and parameter count.

As shown in Table 6, the space complexity of HyGNet aligns with baseline models, where N_e, N_r, N_τ, and d represent the number of entities, relations, timestamps, and the embedding dimension,

Table 6: Comparison of space complexity and parameter count

Model	Space Complexity	Parameter Count
TeRo	$\mathcal{O}\left(N_e d + N_r d\right)$	8,231,500
ATiSE	$\mathcal{O}\left(N_e d + N_r d\right)$	18,980,089
SANe	$\mathcal{O}\left(N_e d + N_r d + N_\tau d\right)$	21,670,219
HyGNet	$\mathcal{O}\left(N_e d + N_r d + N_\tau d\right)$	7,047,370

respectively. This similarity stems from the fact that HyGNet does not introduce additional trainable parameters compared to its Euclidean version, thus maintaining equivalent space complexity with its Euclidean counterpart. Furthermore, on the ICEWS14 dataset, HyGNet's parameter count is comparable to TeRo (Xu et al., 2020a) and ATiSE (Xu et al., 2020b) but significantly lower than SANe (Li et al., 2022). In terms of time complexity, due to the introduction of hyperbolic geometry, running one epoch of HyGNet on ICEWS14 takes approximately 5 minutes, and completing 1000 epochs requires around 3.5 days. The training duration is considered acceptable, yet we are committed to reducing time complexity in future iterations.

6. Conclusion

This paper proposes HyGNet, a novel model that leverages hyperbolic geometry to capture temporal dependencies and spatial relations in TKGs. By aggregating entity neighborhood information and facilitating heterogeneous interactions among entity, relation, and timestamp embeddings in hyperbolic space, HyGNet offers a comprehensive framework for TKGC. Extensive experiments on various benchmark datasets demonstrate HyGNet's clear superiority over existing state-of-the-art methods. The model's remarkable performance improvements underscore its ability to effectively model temporal information and intricate hierarchical relations in TKGs, highlighting the significant potential of hyperbolic geometry in this domain.

Acknowledgments

The authors would like to thank the anonymous reviewers for the insightful comments. This work was supported in part by the National Natural Science Foundation of China (No. 62272025 and No. U22B2021), and in part by Fund of the State Key Laboratory of Software Development Environment.

Bibliographical References

Luyi Bai, Xiangnan Ma, Xiangxi Meng, Xin Ren, and Yujing Ke. 2023. Roan: A relation-oriented attention network for temporal knowledge graph completion. *Engineering Applications of Artificial Intelligence*, 123:106308.

Ivana Balažević, Carl Allen, and Timothy Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Giacomo Balloccu, Ludovico Boratto, Gianni Fenu, and Mirko Marras. 2022. Post processing recommender systems with knowledge graphs for recency, popularity, and diversity of explanations. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Graciela S Birman and Abraham A Ungar. 2001. The hyperbolic derivative in the poincaré ball model of hyperbolic geometry. *Journal of mathematical analysis and applications*, 254(1):321–333.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. Icews coded event data. *Harvard Dataverse*, 12.

BSI. 1973a. *Natural Fibre Twines*, 3rd edition. British Standards Institution, London. BS 2570.

BSI. 1973b. Natural fibre twines. BS 2570, British Standards Institution, London. 3rd. edn.

Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. 2023. Temporal knowledge graph completion: A survey.

James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. 1997. Hyperbolic geometry. *Flavors of geometry*, 31:59–115.

A. Castor and L. E. Pollux. 1992. The use of user modelling to guide inference and learning. *Applied Intelligence*, 2(1):37–53.

Ines Chami, Rex Ying, Christopher Re, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*.

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 60th Annual Meeting of the Association for Computational Linguistics*.

J.L. Chercheur. 1994. *Case-Based Reasoning*, 2nd edition. Morgan Kaufman Publishers, San Mateo, CA.

N. Chomsky. 1973. Conditions on transformations. In *A festschrift for Morris Halle*, New York. Holt, Rinehart & Winston.

Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing*.

Shimin Di, Quanming Yao, and Lei Chen. 2021. Searching to sparsify tensor decomposition for n-ary relational data. In *Proceedings of the Web Conference*.

Yang Ding, Jing Yu, Bang Liu, Yue Hu, Mingxin Cui, and Qi Wu. 2022. Mukea: Multimodal knowledge extraction and accumulation for knowledge-based visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Takuma Ebisu and Ryutaro Ichise. 2019. Generalized translation-based embedding of knowledge graph. *IEEE Transactions on Knowledge and Data Engineering*, 32(5):941–951.

Umberto Eco. 1990. *The Limits of Interpretation*. Indian University Press.

Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. 2014. Introducing wikidata to the linked data web. In *International semantic web conference*.

Masoud Faraki, Mehrtash T. Harandi, and Fatih Porikli. 2018. A comprehensive look at coding techniques on riemannian manifolds. *IEEE*

Transactions on Neural Networks and Learning Systems, 29(11):5701–5712.

Alberto Garcia-Duran, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*.

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2021. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International Conference on Learning Representations*.

Paul Gerhard Hoel. 1971a. *Elementary Statistics*, 3rd edition. Wiley series in probability and mathematical statistics. Wiley, New York, Chichester. ISBN 0 471 40300.

Paul Gerhard Hoel. 1971b. *Elementary Statistics*, 3rd edition, Wiley series in probability and mathematical statistics, pages 19–33. Wiley, New York, Chichester. ISBN 0 471 40300.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (CSUR)*, 54(4):1–37.

Prachi Jain, Sushant Rathi, Soumen Chakrabarti, et al. 2020. Temporal knowledge base completion: New algorithms and evaluation protocols. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Otto Jespersen. 1922. *Language: Its Nature, Development, and Origin*. Allen and Unwin.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*.

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion Proceedings of the The Web Conference 2018*.

Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*.

Yancong Li, Xiaoming Zhang, Bo Zhang, and Haiying Ren. 2022. Each snapshot to each space: Space adaptation for temporal knowledge graph completion. In *International Semantic Web Conference*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*.

Qi Liu, Maximilian Nickel, and Douwe Kiela. 2019. Hyperbolic graph neural networks. *Advances in neural information processing systems*, 32.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. 2014. Yago3: A knowledge base from multilingual wikipedias. In *7th biennial conference on innovative data systems research*.

Johannes Messner, Ralph Abboud, and Ismail Ilkan Ceylan. 2022. Temporal knowledge graph completion using box embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Sebastien Montella, Lina M Rojas Barahona, and Johannes Heinecke. 2021. Hyperbolic temporal knowledge graph embeddings with relational and time curvatures. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*.

Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. 2020. Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In *Proceedings of The Web Conference*.

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*.

Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. 2018. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*.

Rik Sarkar. 2011. Low distortion delaunay embedding of trees in hyperbolic plane. In *International symposium on graph drawing*.

Charles Joseph Singer, E. J. Holmyard, and A. R. Hall, editors. 1954–58. *A history of technology*. Oxford University Press, London. 5 vol.

Jannik Strötgen and Michael Gertz. 2012. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 3746–3753, Istanbul, Turkey. European Language Resource Association (ELRA).

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

S. Superman, B. Batman, C. Catwoman, and S. Spiderman. 2000. *Superheroes experiences with books*, 20th edition. The Phantom Editors Associates, Gotham City.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. 2020. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proceedings of the AAAI conference on artificial intelligence*.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2019. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*.

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*.

Richard C Wilson, Edwin R Hancock, Elżbieta Pekalska, and Robert PW Duin. 2014. Spherical and hyperbolic embeddings of data. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2255–2269.

Chengjin Xu, Yung-Yu Chen, Mojtaba Nayyeri, and Jens Lehmann. 2021. Temporal knowledge graph completion using a linear temporal regularizer and multivector embeddings. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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 28th International Conference on Computational Linguistics*.

Chengjin Xu, Mojtaba Nayyeri, Yung-Yu Chen, and Jens Lehmann. 2023. Geometric algebra based embeddings for static and temporal knowledge graph completion. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4838–4851.

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

Bishan Yang, Scott 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*.

Menglin Yang, Min Zhou, Zhihao Li, Jiahong Liu, Lujia Pan, Hui Xiong, and Irwin King. 2022. Hyperbolic graph neural networks: A review

of methods and applications. *arXiv preprint arXiv:2202.13852*.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*.

Taolin Zhang, Chengyu Wang, Nan Hu, Minghui Qiu, Chengguang Tang, Xiaofeng He, and Jun Huang. 2022. Dkplm: Decomposable knowledge-enhanced pre-trained language model for natural language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Appendix A. Criteria for Relation Selection in Performance Evaluation

This section explains how we chose certain relations to assess the performance of the HyGNet model, focusing on hierarchical relations within TKGC. We use the YAGO11k dataset for this evaluation, reflecting the complexity of real-world encyclopedic knowledge. This dataset includes various relations: wasBornIn, diedIn, worksAt, playsFor, hasWonPrize, isMarriedTo, owns, graduatedFrom, isAffiliatedTo, and created.

In the analysis, we aimed to validate whether HyGNet can effectively model relations with inherent hierarchy, and for this purpose, we designed an algorithm to compute a hierarchical score. The hierarchical score calculation involved computing values for both head to tail and tail to head relations and weighting their summation. For the relations wasBornIn, diedIn, worksAt, playsFor, hasWonPrize, isMarriedTo, owns, graduatedFrom, isAffiliatedTo, and created, the hierarchical values were 4, 29, 33, 171, 18, 32, 160, 3, 165, and 33, respectively. A higher value indicates a greater degree of hierarchy for the respective relation. Therefore, we selected the top 4 relations based on their hierarchical values in descending order, namely playsFor, hasWonPrize, graduatedFrom, and isAffiliatedTo. This choice allows us to focus on relations that pose a greater challenge due to their hierarchical nature.

The results in Figure 2 show that HyGNet performs well on these selected relations, especially on those with higher hierarchical scores, compared to baseline models. This demonstrates HyGNet's strength in modeling and understanding hierarchical relations in TKGs, making it useful for complex TKGC tasks.

Table 7: Comparison of link prediction results on GDELT.

Metrics	Hits@10	Hits@3	Hits@1	MRR
TransE	.312	.158	.000	.113
DistMult	.348	.208	.117	.196
TTransE	.318	.160	.000	.115
HyTE	.326	.165	.000	.118
SANe	.476	.326	.212	.301
HyGNet	**.509**	**.358**	**.237**	**.329**

Appendix B. Experimental Results on the GDELT Dataset

To further validate the robust performance of HyGNet on large-scale temporal knowledge graphs, we conducted additional experiments on a more extensive and diverse dataset, the Global Database of Events, Language, and Tone (GDELT) (Leetaru and Schrodt, 2013). GDELT is a subset of a larger global event, language, and tone database, capturing a temporal knowledge graph with records dating back to 1979, reflecting human behavior.

Our experiments focused on facts with timestamps ranging from April 1, 2015, to March 31, 2016, within this vast dataset. Specifically, it includes facts related to 500 of the most common entities and 20 of the most common relations, providing a valuable testing platform for evaluating model performance in diverse and challenging environments. The training set of GDELT comprises 2,735,685 quadruples, nearly 38 times larger than the ICEWS14 dataset.

The experimental results shown in Table 7 indicate HyGNet outperforms baseline models on GDELT, achieving state-of-the-art results. This achievement is significant, especially considering the complexity and scale of the GDELT dataset.

GDELT exhibits higher density and significant temporal dynamics compared to the two ICEWS datasets, including 2.7 million training facts, 500 entities, and 20 relations. Some facts persist across multiple consecutive timestamps, while others are transient and sparse. This inherent complexity emphasizes the challenging nature of the dataset, requiring powerful reasoning capabilities.