TeRDy: Temporal Relation Dynamics through Frequency Decomposition for Temporal Knowledge Graph Completion

Ziyang Liu, Chaokun Wang
School of Software, BNRist, Tsinghua University, Beijing 100084, China
liu-zy21@mails.tsinghua.edu.cn, chaokun@tsinghua.edu.cn

Abstract

Temporal knowledge graph completion aims to predict missing facts in a knowledge graph by leveraging temporal information. Existing methods often struggle to capture both the long-term changes and short-term variability of relations, which are crucial for accurate prediction. In this paper, we propose a novel method called TeRDy for temporal knowledge graph completion. TeRDy captures temporal relational dynamics by utilizing time-invariant embeddings, along with long-term temporally dynamic embeddings (e.g., enduring political alliances) and short-term temporally dynamic embeddings (e.g., transient political events). These two types of embeddings are derived from low- and high-frequency components via frequency decomposition. Also, we design temporal smoothing and temporal gradient to seamlessly incorporate timestamp embeddings into relation embeddings. Extensive experiments on benchmark datasets demonstrate that TeRDy outperforms state-of-the-art temporal knowledge graph embedding methods.

1 Introduction

Temporal knowledge graphs represent dynamic relationships between entities over time, capturing the evolution of facts and events (Liang et al., 2023; Ying et al., 2024; Wang et al., 2025). These graphs are crucial for applications such as event prediction (Tang et al., 2024; Chen et al., 2024), recommender systems (Zhao et al., 2022; Hu et al., 2024), and decision support (Liu et al., 2022; Wang et al., 2024), where understanding the temporal evolution of relationships is essential. The challenge in the temporal knowledge graph completion (TKGC) lies in predicting missing facts by leveraging the temporal information contained in the graph (Cai et al., 2023; Bai et al., 2023). Given the evolving nature of real-world data, accurately modeling the dynamic relationships between entities is critical for improving prediction accuracy. Despite substantial progress, existing methods (Chen et al., 2022; Ying et al., 2024) still struggle to capture both the long-term changes and short-term variability of relations, which are crucial for making accurate prediction in such dynamic environments.

Figure 1: Three types of relations in the temporal knowledge graph. The NBA player Stephen Curry has: (i) Time-invariant relations, which remain constant over time, such as his connections to family and being an NBA player; (ii) Long-term temporally dynamic relations, which stay temporal stability over short periods but can change, like his teammates after trade deadlines; (iii) Short-term temporally dynamic relations, which vary over short periods, such as his opponents in each game.

Recent advancements in the TKGC have largely focused on embedding-based methods (Lacroix et al., 2020; Ying et al., 2024), where entities, relations, and timestamps are mapped to vector representations, i.e., embeddings. These embeddings aim to capture both the structural and temporal aspects of the knowledge graph. While existing methods achieve promising results, they fail to distinguish between the long-term changes and short-term variations in time-dependent relations. For instance, consider the case of the NBA player Stephen Curry, as shown in Fig. 1. His relations can be categorized into three types: (i) Time-invariant relations, which remain constant over time, such as his connection to his family and his status as an NBA player; (ii) Long-term temporally dynamic

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9611–9622

(LTTD) relations, which remain temporal stability over short periods but may change over long periods, e.g., his teammates may vary after trade deadlines; (iii) Short-term temporally dynamic (STTD) relations, which exhibit significant fluctuations over short periods, such as his opponents in each game. These different types of relations require distinct treatment to effectively model their temporal dynamics, but existing methods fail to capture them adequately.

To address this gap, we propose a novel method called TeRDy (**Te**mporal **R**elation **D**ynamics through frequen**y** decomposition), for TKGC. By applying low-pass and high-pass filtering, TeRDy decomposes the relation embedding in the frequency domain into low- and high-frequency components, where the low-frequency component captures LTTD relations and the high-frequency component captures STTD relations. To incorporate timestamp embeddings into relation embeddings, we designs temporal smoothing for LTTD relation embeddings and temporal gradient for LTTD relation embeddings, respectively. The frequency decomposition enables the more comprehensive modeling of temporal relational dynamics in the knowledge graph, leading to accurate prediction of missing facts. For example, TeRDy improves the mean reciprocal rank by 3.4% compared to the strong baseline on the dataset of GDELT.

Our contributions are as follows:

- We investigate the temporal relation dynamics in the temporal knowledge graph and propose three types of relation embeddings: time-invariant, LTTD, and STTD (Section 4.1).
- We design a new method called TeRDy based on fast Fourier transform and its inverse to convert relation embeddings between the temporal and frequency domains (Section 4.2). We design temporal smoothing and gradient to deal with LTTD and STTD relations, respectively (Section 4.3).
- We validate TeRDy through extensive experiments, demonstrating that it significantly outperforms state-of-the-art methods in the TKGC task (Section 5).

2 Related Work

2.1 Transformation Functions-based Methods

Transformation functions-based TKGC methods encode temporal information through transformations applied to entity and relation embeddings. BoxTE (Messner et al., 2022) extends the BoxE model (Abboud et al., 2020) by incorporating a relation-specific transfer matrix to capture temporal evolution, associating timestamps with relations and enabling richer inference patterns. Dai et al. (Dai et al., 2024) further enhance BoxTE by introducing generative adversarial networks to generate plausible quadruplets. The discriminator distinguishes between real and generated quadruplets, improving the quality of temporal embeddings. To address the vanishing gradient problem in discrete data, they employ Wasserstein distance and GumbelSoftmax relaxation. Special transformation functions, like diachronic embedding functions, efficiently encode timestamps by associating them with entities and relations. Goel et al. (Goel et al., 2020) propose a general diachronic embedding function (e.g., DE-SimplE and DE-DistMult), which is model-agnostic and can generate entity representations at any timestamp, improving the accuracy of TKGC.

2.2 Complex Embedding Functions-based Methods

Complex embedding functions-based TKGC methods embed knowledge graphs into complex spaces to capture relational patterns like symmetry, antisymmetry, and inversion, while also integrating the time dimension to model the evolution of entities and relations. ChronoR(Sadeghian et al., 2021) extends RotatE(Sun et al., 2019) by incorporating timestamps into relations, modeling each relation-timestamp pair as a rotation in complex space to capture temporal evolution. TComplEx and TNTComplEx (Lacroix et al., 2020) extend ComplEx to the temporal domain by representing TKGs as higher-order tensors. TNTComplEx further separates temporal and static components, allowing it to model both dynamic and static relations. RotateQVS (Chen et al., 2022) uses quaternion space, where temporal information serves as the rotation axis to model entity evolution through quaternion embeddings. TCompoundE (Ying et al., 2024) models temporal dynamics via relation- and time-specific translation and scaling transformations.

2.3 Neural Modeling Methods

RESCAL (Nickel et al., 2011) is a tensor-based approach to relational learning with fast and efficient inference. ConT (Ma et al., 2019) extends RESCAL by introducing high-dimensional time embeddings. DyERNIE (Han et al., 2020) models

temporal evolution on Riemannian manifolds using velocity vectors. ATiSE (Xu et al., 2019) captures temporal patterns via additive frequency components, but at higher computational cost. The dual memory models proposed in (Tresp et al., 2017) integrate episodic and semantic memory to distinguish between time-aware and static representations. The work in (Han et al., 2021b) offers a unified evaluation framework for TKGC, highlighting the importance of fair comparison across methods. TANGO (Han et al., 2021a) employs neural ordinary differential equations to model continuous-time evolution in temporal knowledge graphs. Graph neural network (GNN) models such as TARGCN (Ding et al., 2023a) use neighborhood aggregation but are computationally intensive. Hybrid models (Jin et al., 2020a; Li et al., 2021a; Liang et al., 2023; Feng et al., 2025) combine GNNs with scoring functions to balance efficiency and expressiveness.

3 Preliminaries

3.1 Problem Formulation

Given a temporal knowledge graph $\mathcal{G}$, we denote the set of entities as $\mathcal{E}$, the set of relations as $\mathcal{R}$, and the set of timestamps as $\mathcal{T}$. A fact in the temporal knowledge graph is represented as a quadruple (s, r, o, τ), where subject s and object o are entities from $\mathcal{E}$, r is a relation from $\mathcal{R}$, and τ is a timestamp from $\mathcal{T}$. During training, we utilize a score function $\phi(s, r, o, \tau)$ to model the relationships among entities, relations, and timestamps. The objective of TKGC is to predict missing facts in the graph by leveraging the known facts (Gao et al., 2023; Niu and Li, 2023). Specifically, we aim to predict the missing element in a quadruple, either $(s, r, ?, \tau)$ or $(?, r, o, \tau)$, by feeding the incomplete quadruple and candidate entities into the score function. The entity that achieves the highest score is then selected as the predicted entity to complete the quadruple.

3.2 Compound Geometric Operations

The geometric operations of translation, rotation, and scaling form the foundation for complex transformations, which are used in models like TComplEx (Lacroix et al., 2020) and TCompoundE (Ying et al., 2024). Translation shifts an object by a fixed amount along the coordinate axes. In 2D space, it is represented by the following matrix:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where t_x and t_y represent the translation offsets along the x and y axes, respectively. Applying this matrix to a vector moves it by these amounts. Rotation in 2D space turns an object around a fixed point by angle θ, using the rotation matrix:

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

This operation alters the orientation of the vector without changing its shape or size. Scaling enlarges or shrinks an object proportionally along the axes. In 2D space, the scaling matrix is defined as:

$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where s_x and s_y are the scaling factors along the x and y axes, respectively.

TCompoundE (Ying et al., 2024) models temporal knowledge graphs via compound geometric operations that capture the interaction between relations and timestamps. It introduces two operations: a time-specific and a relation-specific transformation. The time-specific operation combines relation and timestamp embeddings using translation and scaling:

$$\mathbf{S}_{r,\tau} = (\mathbf{S}_r + \mathbf{T}_\tau) \odot \mathbf{S}_\tau, \tag{4}$$

where $\mathbf{S}_r$ is the relation embedding, $\mathbf{T}_\tau$ is the timestamp embedding (for translation), and $\mathbf{S}_\tau$ is the timestamp embedding (for scaling).

The relation-specific operation applies transformation to the subject embedding using:

$$\mathbf{e}_s^{r,\tau} = (\mathbf{e}_s + \mathbf{T}_{r,\tau}) \odot \mathbf{S}_{r,\tau}, \tag{5}$$

where $\mathbf{e}_s$ is the subject embedding and $\mathbf{T}_{r,\tau} = \mathbf{T}_r$. This combined translation-scaling mechanism allows TCompoundE to capture temporal evolution in subject–relation interactions effectively.

4 Methodology

4.1 Overview

TeRDy captures the temporal relation dynamics in $\mathcal{G}$ by using time-invariant, LTTD, and STTD relation embeddings. As shown in Fig. 2, on the whole, TeRDy translates the subject embedding

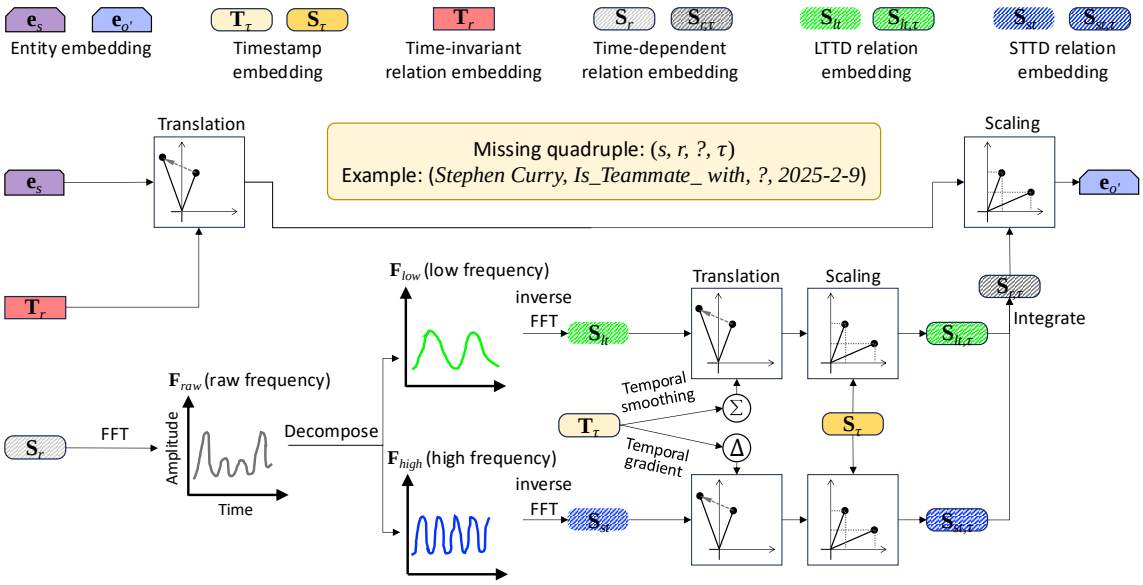


Figure 2: Illustration of TeRDy. It uses time-invariant relation embedding $\mathbf{T}_r$, long-term temporally dynamic (LTTD) relation embedding $\mathbf{S}_{lt,\tau}$, and short-term temporally dynamic (STTD) relation embedding $\mathbf{S}_{st,\tau}$ to capture the temporal relation dynamics.

$\mathbf{e}_s$ into the predicted object embedding $\mathbf{e}_{o'}$ using the time-invariant relation embedding $\mathbf{T}_r$, and then scales it with the time-dependent relation embedding $\mathbf{S}_{r,\tau}$ (Eq. (5)). The raw relation embedding $\mathbf{S}_r$ is transformed into the frequency domain embedding $\mathbf{F}_{raw}$ via fast Fourier transform (Eq. (6)). $\mathbf{F}_{raw}$ is then decomposed into low-frequency component $\mathbf{F}_{low}$ and high-frequency component $\mathbf{F}_{high}$ (Eq. (9)). Afterward, both components are converted back into the time domain through inverse fast Fourier transform and obtain the initial LTTD relation embedding $\mathbf{S}_{lt}$ and initial STTD relation embedding $\mathbf{S}_{st}$ (Eq. (10)). $\mathbf{S}_{lt}$ undergoes temporal smoothing operations to form $\mathbf{S}_{lt,\tau}$ (Eq. (13)), while $\mathbf{S}_{st}$ undergoes temporal gradient operations to form $\mathbf{S}_{st,\tau}$ (Eq. (14)). Finally, $\mathbf{S}_{lt,\tau}$ and $\mathbf{S}_{st,\tau}$ are integrated to produce the final time-dependent relation embedding $\mathbf{S}_{r,\tau}$ (Eq. (16)).

4.2 Frequency-based Relation Decomposition

The relation embedding $\mathbf{S}_r$ of time domain is transformed into the one $\mathbf{F}_{raw}$ of frequency domain using fast Fourier transform (FFT) (Cooley et al., 1969; Liu et al., 1998):

$$\mathbf{F}_{raw} = \mathcal{F}(\mathbf{S}_r), \tag{6}$$

where $\mathcal{F}(\cdot)$ represents the fast Fourier transformation. Next, a frequency mask is created based on the absolute values of the frequencies. The low-frequency mask $\mathbf{M}_{low}$ is designed to emphasize the low-frequency components:

$$\mathbf{M}_{low} = \exp(-|\mathbf{f}| \cdot \alpha), \tag{7}$$

where $|\mathbf{f}|$ represents the frequencies, and α is a predefined factor controlling how much of the low frequencies are preserved. The high-frequency mask $\mathbf{M}_{high}$ is the complement of $\mathbf{M}_{low}$:

$$\mathbf{M}_{high} = \mathbf{1} - \mathbf{M}_{low}\ , \tag{8}$$

where $\mathbf{1}$ is an all-one matrix with the same dimension as $\mathbf{M}_{low}$. Using the frequency masks, the frequency domain embedding $\mathbf{F}_{raw}$ is split into low-frequency component $\mathbf{F}_{low}$ and high-frequency component $\mathbf{F}_{high}$:

$$\mathbf{F}_{low}{=}\mathbf{F}_{raw}{\cdot}\mathbf{M}_{low},\ \mathbf{F}_{high}{=}\mathbf{F}_{raw}{\cdot}\mathbf{M}_{high}\ . \tag{9}$$

The low- and high-frequency components are converted back to the time domain using inverse FFT, and obtain the initial LTTD relation embedding $\mathbf{S}_{lt}$ and initial STTD relation embedding $\mathbf{S}_{st}$:

$$\mathbf{S}_{lt} = \mathcal{F}^{-1}(\mathbf{F}_{low}), \quad \mathbf{S}_{st} = \mathcal{F}^{-1}(\mathbf{F}_{high}), \tag{10}$$

where $\mathcal{F}^{-1}(\cdot)$ denotes inverse FFT.

Based on Eqs. (6)-(10), the LTTD embedding and STTD embedding are derived from a frequency-domain decomposition. After frequency-domain decomposition, the low-frequency component (corresponding to the LTTD embedding) represents the relatively stable dynamics between consecutive time steps and captures long-term trends;

the high-frequency component (corresponding to the STTD embedding) represents the relatively unstable or rapid fluctuations and captures short-term variations in the data. In essence, the LTTD embedding focuses on capturing the long-term stability over time, and the STTD embedding captures the short-term instability between time points.

4.3 Integration of LTTD & STTD embeddings

Here we describe how the low- and high-frequency components of the time-dependent relation embedding are integrated to capture the temporal dynamics of the knowledge graph. We first define the initial timestamp embedding for translation as $\mathbf{T}_\tau = \begin{bmatrix} \mathbf{T}_\tau^{(1)} & \mathbf{T}_\tau^{(2)} & \cdots & \mathbf{T}_\tau^{(D)} \end{bmatrix}$ where $\mathbf{T}_\tau^{(i)}$ represents the timestamp embedding at the i-th embedding dimension (D is embedding size). The smoothed timestamp embedding, which aggregates information across all embedding dimension, is computed as:

$$\mathbf{T}_\tau^{\text{smooth}} = \frac{1}{D} \sum\nolimits_{i=1}^{D} \mathbf{T}_\tau^{(i)} \ . \tag{11}$$

Next, we calculate the temporal gradient for each entity by computing the difference between adjacent embedding dimensions within the same time step. This gradient embedding captures the internal variation across dimensions at a specific point in time, and is expressed as:

$$\mathbf{T}_\tau^{\text{grad}} = \begin{bmatrix} \mathbf{T}_\tau^{(1)} & \mathbf{T}_\tau^{(2)} - \mathbf{T}_\tau^{(1)} & \cdots & \mathbf{T}_\tau^{(D)} - \mathbf{T}_\tau^{(D-1)} \end{bmatrix} . \tag{12}$$

Given the initial timestamp embedding for scaling, $\mathbf{S}_\tau$, the time-dependent relation embeddings are generated by integrating LTTD ones with STTD ones. The LTTD relation embedding $\mathbf{S}_{lt,\tau}$ is obtained by adding the smoothed timestamp embedding $\mathbf{T}_\tau^{\text{smooth}}$ to $\mathbf{S}_{lt}$, and scaling it by $\mathbf{S}_\tau$:

$$\mathbf{S}_{lt,\tau} = (\mathbf{S}_{lt} + \mathbf{T}_\tau^{\text{smooth}}) \odot \mathbf{S}_\tau \ . \tag{13}$$

Similarly, the STTD relation embedding $\mathbf{S}_{st,\tau}$ is obtained by adding the temporal gradient $\mathbf{T}_\tau^{\text{grad}}$ to $\mathbf{S}_{st}$, and scaling it by $\mathbf{S}_\tau$:

$$\mathbf{S}_{st,\tau} = (\mathbf{S}_{st} + \mathbf{T}_\tau^{\text{grad}}) \odot \mathbf{S}_\tau \ . \tag{14}$$

The final time-dependent relation embedding $\mathbf{S}_{r,\tau}$ is then obtained by integrating $\mathbf{S}_{lt,\tau}$ with $\mathbf{S}_{st,\tau}$:

$$\mathbf{S}_{r,\tau} = \mathbf{S}_{lt,\tau} + \mathbf{S}_{st,\tau} \ . \tag{15}$$

Finally, the predicted object embedding $\mathbf{e}_{o'}$ is computed by translating subject embedding $\mathbf{e}_s$ with $\mathbf{T}_r$, and scaling it with $\mathbf{S}_{r,\tau}$:

$$\mathbf{e}_{o'} = (\mathbf{e}_s + \mathbf{T}_r) \odot \mathbf{S}_{r,\tau} \ . \tag{16}$$

4.4 Loss Function

The total loss function $\mathcal{L}$ of TeRDy is the combination of three parts:

$$\mathcal{L} = \mathcal{L}_u + \lambda_\tau \mathcal{L}_\tau + \lambda_f \mathcal{L}_f , \tag{17}$$

where $\mathcal{L}_u$ is the loss associated with reciprocal learning, $\mathcal{L}_\tau$ is the temporal regularizer, and $\mathcal{L}_f$ is the frequency-domain regularizer. Also, λ_τ and λ_f are the weighting factors that control the relative importance of the temporal and frequency-domain regularizers in $\mathcal{L}$.

Based on the principles from TNTComplEx (Lacroix et al., 2020) and TCompoundE (Ying et al., 2024), we use the reciprocal learning loss $\mathcal{L}_u$ to ensure that both forward and inverse relations are consistent. It is defined as:

$$\begin{aligned} \mathcal{L}_u = & -\log\left(\frac{\exp(\phi(s,r,o,\tau))}{\sum_{o' \in \varepsilon} \exp(\phi(s,r,o',\tau))}\right) \\ & -\log\left(\frac{\exp(\phi(o,r^{-1},s,\tau))}{\sum_{s' \in \varepsilon} \exp(\phi(o,r^{-1},s',\tau))}\right) \\ & +\lambda_u \sum_{i=1}^{k} \left(\|\mathbf{e}_s\|_3^3 + \|\mathbf{T}_r + \mathbf{S}_{r,\tau}\|_3^3 + \|\mathbf{e}_o\|_3^3 \right) , \end{aligned} \tag{18}$$

where r^{-1} represents the inverse relation and λ_u is the regularization parameter that controls the strength of the regularization term for the embeddings of the subject, relation, and object. The first two terms are standard negative log-likelihood terms for both the subject-object and object-subject prediction tasks, enforcing the model to make accurate prediction. The third term is a regularization term applied to the embeddings of the subject, relation, and object, encouraging the embeddings to be well-behaved.

The temporal regularizer loss $\mathcal{L}_\tau$ enforces that neighboring timestamps should have similar embeddings, which is important for capturing temporal dynamics. It is defined as:

$$\mathcal{L}_\tau = \frac{1}{N_\tau - 1} \sum\nolimits_{i=1}^{N_\tau - 1} \|\mathbf{T}_{\tau(i+1)} - \mathbf{T}_{\tau(i)}\|_3^3 , \tag{19}$$

where N_τ is the total number of timestamps and $\mathbf{T}_{\tau(i)}$ represents the embeddings at timestamp i,

Table 1: Statistics of the experimental datasets.

Dataset	GDELT	ICEWS14	ICEWS05-15
#Entities	500	7,128	10,488
#Relations	20	230	251
#Timestamps	366	365	4,017
#Train	2,735,685	72,826	386,962
#Dev	31,961	8,963	46,092
#Test	31,961	8,941	46,275

and the norm encourages the embeddings of consecutive timestamps to be close in the embedding space, ensuring temporal continuity.

Also, we define the novel frequency-domain regularizer loss $\mathcal{L}_f$ as follows:

$$\mathcal{L}_f = -\|\mathbf{F}_{low} - \mathbf{F}_{high}\|_2 + \|\mathbf{F}_{high}\|_2 \,, \quad (20)$$

where the first term encourages the low- and high-frequency components to be distinct, while the second term penalizes excessive intensity in the high-frequency components. $\mathcal{L}_f$ helps TeRDy avoid the overlap of low- and high-frequency components and minimize the impact of noisy, high-frequency variations. (See the results in Section 6.2 for details).

5 Experiments

5.1 Datasets

We evaluate TeRDy and baselines on three benchmark datasets: GDELT, ICEWS14, and ICEWS05-15 (Table 1). GDELT (Leetaru and Schrodt, 2013) is a subset of the Global Database of Events, Language, and Tone, covering daily political and social events between April 1, 2015, and March 31, 2016. GDELT focuses on the 500 most frequent entities and 20 common relations, making it a more concise dataset for event extraction. ICEWS14 and ICEWS05-15 (Garcia-Duran et al., 2018) are derived from the Integrated Crisis Early Warning System (Lautenschlager et al., 2015), which records significant political events. ICEWS14 includes data from the year 2014, while ICEWS05-15 spans events from 2005 to 2015. Both datasets capture a wide range of global political events, offering a comprehensive view of temporal dynamics in international relations.

5.2 Baselines

We compare TeRDy with state-of-the-art temporal knowledge graph embedding methods, including TTransE (Leblay and Chekol, 2018), DE-SimplE (Goel et al., 2020), TA-DisMult (García-Durán et al., 2018), ChronoR (Sadeghian et al., 2021), TComplEx (Lacroix et al., 2020), TNTComplEx (Lacroix et al., 2020), BoxTE (Messner et al., 2022), RotateQVS (Chen et al., 2022), TeAST (Li et al., 2023), and TCompoundE (Ying et al., 2024). Among these, TCompoundE demonstrates the highest performance on ICEWS14, ICEWS05-15, and GDELT. As a result, we use TCompoundE as the primary baseline for our evaluation.

5.3 Metrics

We evaluate our model's performance using standard metrics: Mean Reciprocal Rank (MRR) and Hits@k (H@k). For each test quadruple, we calculate the scores for all possible entity substitutions and rank them. For the missing quadruple $(s, r, ?, \tau)$ or $(?, r, o, \tau)$, we predict the missing entity by ranking all candidates. H@k measures the proportion of correct entities within the top-k prediction, where $k = 1, 3, 10$. Higher MRR and H@k values reflect better model performance.

5.4 Experimental Setup

The Adagrad optimizer is used for all datasets, with the following settings - GDELT: The learning rate is set to 0.05, the embedding dimension $D = 6000$, the batch size is 2000, and training runs for 50 epochs. ICEWS14: The learning rate is set to 0.02, the embedding dimension $D = 6000$, the batch size is 4000, and the training runs for 100 epochs. ICEWS05-15: The learning rate is set to 0.008, the embedding dimension $D = 8000$, the batch size is 6000, and the training runs for 100 epochs. The parameter group $(\lambda_u, \lambda_\tau, \lambda_f, \alpha)$ is set as follows: (1e-4, 1e-2, 1, 10) for GDELT, (5e-3, 5e-3, 5e-4, 10) for ICEWS14, and (2e-3, 1e-1, 5e-4, 10) for ICEWS05-15. We implement TeRDy using PyTorch, leveraging the training frameworks of TeAST. All experiments are conducted on a single NVIDIA GeForce RTX 3090 with 24GB of memory. We report the average results on the test set across five runs and the total training time of TeRDy in one run on GDELT, ICEWS14, and ICEWS05-15 is 85.8 minutes, 6.7 minutes, and 56.7 minutes, respectively. Our code and data are available at https://github.com/Young0222/TeRDy.

6 Results and Analysis

6.1 Main Results

We evaluate TeRDy on GDELT, ICEWS14, and ICEWS05-15, comparing it with state-of-the-art methods such as TeAST, and TCompoundE. Our

Table 2: Performance comparison for knowledge graph completion. All results are reported from their respective original papers. The best results are highlighted in **bold**.

Methods	GDELT				ICEWS14				ICEWS05-15			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TTransE	0.115	0.000	0.160	0.318	0.255	0.074	-	0.601	0.271	0.084	-	0.616
DE-SimplE	0.230	0.141	0.248	0.403	0.526	0.418	0.592	0.725	0.513	0.392	0.578	0.748
TA-DisMult	0.206	0.124	0.219	0.365	0.477	0.363	-	0.686	0.474	0.346	-	0.728
ATiSE	-	-	-	-	0.550	0.436	0.629	0.750	0.519	0.378	0.606	0.794
ChronoR	-	-	-	-	0.625	0.547	0.669	0.773	0.675	0.596	0.723	0.820
TComplEx	0.340	0.294	0.361	0.498	0.610	0.530	0.660	0.770	0.660	0.590	0.710	0.800
TNTComplEx	0.349	0.258	0.373	0.502	0.620	0.520	0.660	0.760	0.670	0.590	0.710	0.810
BoxTE	0.352	0.269	0.377	0.511	0.613	0.528	0.664	0.763	0.667	0.582	0.719	0.820
RotateQVS	0.270	0.175	0.293	0.458	0.591	0.507	0.642	0.754	0.633	0.529	0.709	0.813
TeAST	0.371	0.283	0.401	0.544	0.637	0.560	0.682	0.782	0.683	0.604	0.732	0.829
TCompoundE	0.433	0.347	0.469	0.595	0.644	0.561	0.694	0.795	0.692	0.612	0.743	0.837
TeRDy	**0.467**	**0.390**	**0.501**	**0.605**	**0.648**	**0.566**	**0.697**	**0.799**	**0.697**	**0.614**	**0.749**	**0.847**

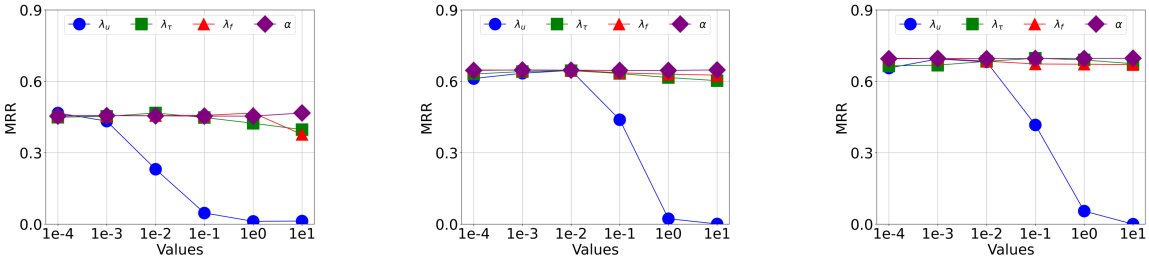
Table 3: Results of different variants of TeRDy. The worst results are highlighted in underline.

Methods	GDELT				ICEWS14				ICEWS05-15			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TeRDy (full model)	0.467	0.390	0.501	0.605	0.648	0.566	0.697	0.799	0.697	0.614	0.749	0.847
Full model w/o $\mathbf{S}_{lt,\tau}$	0.456	0.375	0.491	0.605	0.648	0.566	0.698	0.797	0.692	0.608	0.745	0.844
Full model w/o $\mathbf{S}_{st,\tau}$	0.451	0.370	0.486	0.600	0.640	0.558	0.691	0.793	0.681	0.597	0.733	0.832
Full model w/ $\mathbf{T}_{\tau}^{\text{smooth}}=\mathbf{T}_{\tau}$	0.458	0.377	0.493	0.606	0.646	0.563	0.697	0.797	0.695	0.614	0.746	0.845
Full model w/ $\mathbf{T}_{\tau}^{\text{grad}}=\mathbf{T}_{\tau}$	0.455	0.375	0.491	0.603	0.642	0.556	0.695	0.798	0.691	0.610	0.743	0.845
Full model w/o the 1st term of $\mathcal{L}_f$	0.451	0.372	0.485	0.596	0.649	0.568	0.696	0.799	0.693	0.610	0.742	0.843
Full model w/o the 2nd term of $\mathcal{L}_f$	0.450	0.370	0.484	0.595	0.646	0.565	0.695	0.799	0.692	0.611	0.742	0.842
Full model w/o $\mathcal{L}_f$	0.448	0.368	0.483	0.591	0.647	0.565	0.695	0.799	0.690	0.610	0.739	0.840
Full model w/o $\mathcal{L}_f$ and w/ $\mathbf{S}_{\tau,\tau}=\mathbf{S}_{\tau}$	0.435	0.362	0.475	0.572	0.638	0.558	0.689	0.790	0.688	0.595	0.730	0.829

results show that TeRDy outperforms all baselines in MRR and H@k metrics. In GDELT, TeRDy achieves MRR = 0.467, surpassing TCompoundE (0.433). The same trend is observed in ICEWS14 and ICEWS05-15, with TeRDy achieving MRR = 0.648 and MRR = 0.697, respectively, outperforming TCompoundE at 0.644 and 0.692. For H@k, TeRDy also leads with scores of (e.g., GDELT H@1: 0.390), outperforming TCompoundE (e.g., GDELT H@1: 0.347). The key advantage of TeRDy lies in its ability to capture both long-term and short-term temporal relation dynamics effectively, which other models like TeAST and TCompoundE struggle to model accurately. Also, TeRDy uses frequency-domain decomposition along with temporal smoothing and gradient operations, enabling it to effectively generate and utilize low- and high-frequency components. In addition, frequency-based decomposition ensures that TeRDy captures both long-term and short-term temporal dynamics in the data, regardless of the time granularity of the dataset. The frequency components are inherent in the data, whether the dataset has daily granularity (like GDELT) or monthly granularity (like ICEWS). These frequency components are not directly affected by the time granularity but are rather determined by the underlying temporal patterns in the data. These results demonstrate that TeRDy effectively handles datasets with varying time resolutions.

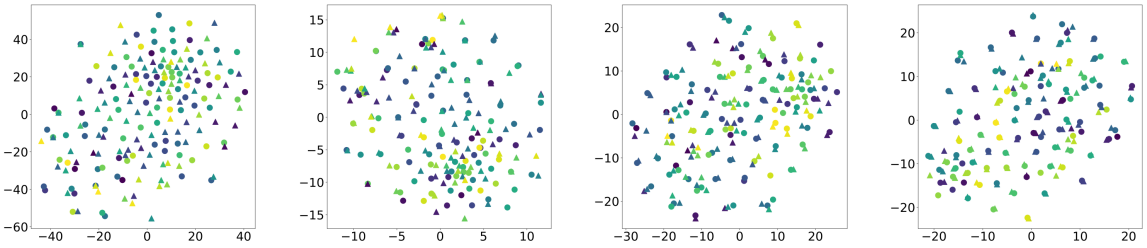
6.2 Results of Ablation Studies

Table 3 presents the results of various TeRDy variants. Modifications to the temporal components (such as removing $\mathbf{S}_{lt,\tau}$ or $\mathbf{S}_{st,\tau}$) result in performance drops. It indicates that both LTTD and STTD relation embeddings are essential for improving top-k prediction accuracy and overall ranking quality. Similarly, we observe that removing $\mathbf{T}_{\tau}^{\text{grad}}$ has a greater impact than removing $\mathbf{T}_{\tau}^{\text{smooth}}$, highlighting the importance of incorporating timestamp embeddings into the STTD relation embeddings in TeRDy. Also, removing or altering $\mathcal{L}_f$ significantly reduces performance, especially in H@1, underscoring the importance of the frequency-domain loss function in the model optimization. In addition, removing frequency decomposition (last row in Table 3) shows a decrease in MRR of 3.2%, 1.0%, and 0.9% across GDELT, ICEWS14, and ICEWS05-15, respectively. It demonstrates that without frequency decomposition, TeRDy struggles to distinguish be-



(a) GDELT (b) ICEWS14 (c) ICEWS05-15

Figure 3: Effects of parameters λ_u, λ_τ, λ_f, and α to model performance on GDELT, ICEWS14, and ICEWS05-15.



(a) Initial state (b) Applying LTTD relation embeddings (c) Applying STTD relation embeddings (d) Applying LTTD and STTD relation embeddings

Figure 4: Visualization of entity embeddings on ICEWS14. The circle and triangle represent the subject entity and object entity, respectively. Circles and triangles of the same color indicate that they belong to the same quadruple.

tween LTTD relations and STTD relations, which leads to less accurate TKGC.

6.3 Results of Sensitivity Experiments

The results of sensitivity experiments are shown in Fig. 3. We can see that across the three datasets, when $\lambda_u \geqslant 0.01$, the performance of TeRDy declines significantly. It indicates that an excessively large λ_u restricts the model's ability to fit the prediction for missing entities in the quadruples. On GDELT, when $\lambda_\tau \geqslant 1$ or $\lambda_f \geqslant 1$, the performance of TeRDy significantly decreases. It suggests that the coefficients for the temporal regularizer and frequency-domain regularizer need to be carefully set for this dataset. In all other cases, the MRR results of TeRDy remain stable with changes in λ_τ, λ_f, and α, demonstrating the model's robustness to these three parameters.

6.4 Results of Embedding Visualization

We applied t-SNE (Van der Maaten and Hinton, 2008) to reduce the dimensionality of entity embeddings for visualization, as shown in Fig. 4. We analyzed entity embeddings under four conditions: (a) initial state, (b) embeddings obtained using LTTD relation embeddings, (c) embeddings obtained using STTD relation embeddings, and (d) embeddings obtained using both LTTD and STTD relation embeddings (i.e., TeRDy). In the figure, nodes that are closer together represent higher similarity in their embeddings. In the analysis, we focus on how often same-color circles and triangles (i.e., subject and object entities in the same temporal quadruple) appear close together in the embedding space. A higher frequency indicates better temporal coherence. In Fig. 4(d), which uses both LTTD and STTD, same-color entities exhibit strong clustering, particularly at points like (0, 22.5), (-4, 20), (-22, -9), (-1, -21), and (10, -13). In contrast, Figs. 4(b) and 4(c), using only LTTD or STTD, show weaker clustering. Fig. 4(a), the initial state, displays no clear structure. These results show that combining LTTD and STTD (i.e., TeRDy) yields more coherent embeddings, with subject–object pairs from the same temporal relation positioned closer together, leading to more accurate modeling of temporal knowledge graph structures.

6.5 Efficiency Analysis

The results of training time and MRR on ICEWS14 are shown in Fig. 5. TeRDy achieves the highest MRR (64.8%) with the shortest training time (400 seconds). On the one hand, although TeAST achieves a relatively high MRR, its training time cost is clearly higher than that of TeRDy. On the

Type	Subject	Relation	Object	Timestamp range	L-only	S-only	TeRDy
LTTD	Citizen (Nigeria)	Make an appeal or request	Government (Nigeria)	Feb–Dec 2014	√ (MRR: 0.25)	× (MRR: 0.24)	√ (MRR:**0.27**)
LTTD	South Africa	Charge with legal action	Men (South Africa)	Jan–Dec 2014	√ (MRR:0.32)	× (MRR:0.32)	√ (MRR:**0.33**)
LTTD	Boko Haram	Use conventional military force	Citizen (Nigeria)	Jan–Dec 2014	√ (MRR:0.35)	× (MRR:0.31)	√ (MRR:**0.37**)
STTD	Abdullah Abdullah	Express intent to cooperate	Ashraf Ghani Ahmadzai	Jul–Aug 2014	× (MRR:0.27)	√ (MRR:0.40)	√ (MRR:**0.41**)
STTD	Citizen (India)	Reject	Separatist (India)	Nov–Dec 2014	× (MRR:0.23)	√ (MRR:0.29)	√ (MRR:**0.31**)
STTD	Islam Karimov	Make a visit	China	Aug-Aug 2014	× (MRR:0.25)	√ (MRR:0.45)	√ (MRR:**0.46**)

Table 4: Case Study Results. L-only uses only low-frequency components to model long-term dynamics; S-only uses only high-frequency components for short-term dynamics. √ and × denote correct and incorrect predictions, respectively. MRR indicates the score for each relation.

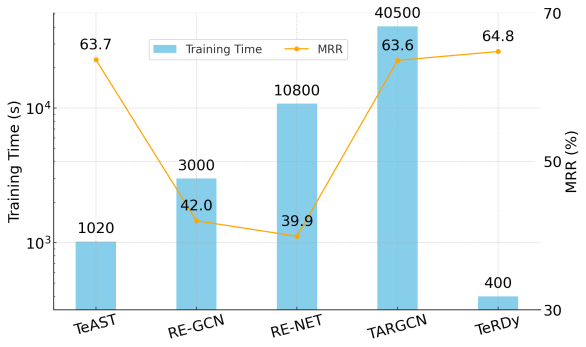


Figure 5: Comparison of training time and MRR on ICEWS14.

other hand, GNN-based methods (RE-GCN (Li et al., 2021b), RE-NET (Jin et al., 2020b), and TARGCN (Ding et al., 2023b)) require significantly longer training times (3,000s, 10,800s, and 40,500s, respectively), mainly due to the inherent inefficiency of GNNs, which rely on message-passing mechanisms to aggregate information from neighboring nodes. These operations are computationally intensive and do not scale well to large knowledge graphs or time-sensitive scenarios.

6.6 Case Study

We perform a case study based on the test data in ICEWS14. We compare our full model TeRDy against two simplified variants: L-only using only the low-frequency component and S-only using only the high-frequency component. We present six real-world cases in Table 4 where STTD cases are defined as relation triples that occur at least three times within a short time span ($\leqslant$ 30 days) and LTTD cases are defined as relation triples that occur at least three times over a long time span ($\geqslant$ 180 days). We have the following analysis: The LTTD relation occurs repeatedly over several months, reflecting a stable and strategic partnership. The L-only variant can capture this long-term trend reasonably well, while the S-only variant often fails to retain this persistent pattern due to its focus on short-term fluctuations. The STTD relation occurs intensively within a short period, i.e., less than one month, likely triggered by a regional conflict or crisis. The S-only variant is able to capture this short-term burst effectively, whereas the L-only variant tends to smooth out such rapid shifts and overlooks their temporal specificity. By integrating both low- and high-frequency components, TeRDy generalizes well across different dynamics, leading to correct predictions in all cases and outperforming both variants in MRR.

7 Conclusion

In this paper, we present TeRDy, a novel method for temporal knowledge graph completion. By applying low-pass and high-pass filtering, TeRDy decomposes the relation embedding in the frequency domain into LTTD and STTD relation embeddings. Then we design temporal smoothing and gradient operations to incorporate timestamp embeddings into both embeddings. Extensive experiments show that TeRDy achieves state-of-the-art performance on benchmark temporal knowledge graph datasets.

Limitations

Like many temporal knowledge graph embedding models, TeRDy cannot generalize to unseen entities or timestamps. Therefore, TeRDy is not well-suited for temporal knowledge graph extrapolation tasks. Also, TeRDy decomposes frequency-domain embeddings into low- and high-frequency components, which may not be suitable for knowledge graphs with simpler or less dynamic relationships.

Acknowledgements

We thank all reviewers for their insightful comments and suggestions to help improve the paper. This work is supported in part by the National Natural Science Foundation of China (No. 62372264). Chaokun Wang is the corresponding author.

References

Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. Boxe: A box embedding model for knowledge base completion. *Advances in Neural Information Processing Systems*, 33:9649–9661.

Luyi Bai, Mingcheng Zhang, Han Zhang, and Heng Zhang. 2023. Ftmf: Few-shot temporal knowledge graph completion based on meta-optimization and fault-tolerant mechanism. *World Wide Web*, 26(3):1243–1270.

Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. 2023. Temporal knowledge graph completion: a survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 6545–6553.

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. *arXiv preprint arXiv:2203.07993*.

Meiqi Chen, Yubo Ma, Kaitao Song, Yixin Cao, Yan Zhang, and Dongsheng Li. 2024. Improving large language models in event relation logical prediction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9451–9478.

James W Cooley, Peter AW Lewis, and Peter D Welch. 1969. The fast fourier transform and its applications. *IEEE Transactions on Education*, 12(1):27–34.

Yuanfei Dai, Wenzhong Guo, and Carsten Eickhoff. 2024. Wasserstein adversarial learning based temporal knowledge graph embedding. *Information Sciences*, 659:120061.

Zifeng Ding, Yunpu Ma, Bailan He, Jingpei Wu, Zhen Han, and Volker Tresp. 2023a. A simple but powerful graph encoder for temporal knowledge graph completion. In *Intelligent systems conference*, pages 729–747. Springer.

Zifeng Ding, Yunpu Ma, Bailan He, Jingpei Wu, Zhen Han, and Volker Tresp. 2023b. A simple but powerful graph encoder for temporal knowledge graph completion. In *Intelligent Systems and Applications - Proceedings of the 2023 Intelligent Systems Conference, IntelliSys 2024, Amsterdam, The Netherlands, September 7-8, 2024, Volume 3*, volume 824 of *Lecture Notes in Networks and Systems*, pages 729–747. Springer.

Siling Feng, Zimin Ye, Qian Liu, and Mengxing Huang. 2025. Rphf-gnn: Recurrent perception of history-future graph neural networks for temporal knowledge graph reasoning. *IEEE Access*.

Yifu Gao, Yongquan He, Zhigang Kan, Yi Han, Linbo Qiao, and Dongsheng Li. 2023. Learning joint structural and temporal contextualized knowledge embeddings for temporal knowledge graph completion. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 417–430.

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*, pages 4816–4821.

Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. *arXiv preprint arXiv:1809.03202*.

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*, volume 34, 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 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7301–7316.

Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. 2021a. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 8352–8364.

Zhen Han, Gengyuan Zhang, Yunpu Ma, and Volker Tresp. 2021b. Time-dependent entity embedding is not all you need: A re-evaluation of temporal knowledge graph completion models under a unified framework. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 8104–8118.

Hengchang Hu, Wei Guo, Xu Liu, Yong Liu, Ruiming Tang, Rui Zhang, and Min-Yen Kan. 2024. User behavior enriched temporal knowledge graphs for sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 266–275.

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020a. Recurrent event network: Autoregressive structure inferenceover temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6669–6683, Online. Association for Computational Linguistics.

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

2020, Online, November 16-20, 2020, pages 6669–6683. Association for Computational Linguistics.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. *arXiv preprint arXiv:2004.04926*.

Jennifer Lautenschlager, Steve Shellman, and Michael Ward. 2015. Icews event aggregations. *(No Title)*.

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion proceedings of the the web conference 2018*, pages 1771–1776.

Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer.

Jiang Li, Xiangdong Su, and Guanglai Gao. 2023. Teast: Temporal knowledge graph embedding via archimedean spiral timeline. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15460–15474.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021a. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 408–417.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021b. Temporal knowledge graph reasoning based on evolutional representation learning. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 408–417. ACM.

Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, and Xinwang Liu. 2023. Learn from relational correlations and periodic events for temporal knowledge graph reasoning. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 1559–1568.

QH Liu, N Nguyen, and XY Tang. 1998. Accurate algorithms for nonuniform fast forward and inverse fourier transforms and their applications. In *IGARSS'98. Sensing and Managing the Environment. 1998 IEEE International Geoscience and Remote Sensing. Symposium Proceedings.(Cat. No. 98CH36174)*, volume 1, pages 288–290. IEEE.

Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. 2022. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4120–4127.

Yunpu Ma, Volker Tresp, and Erik A Daxberger. 2019. Embedding models for episodic knowledge graphs. *Journal of Web Semantics*, 59:100490.

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*, volume 36, pages 7779–7787.

Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel, et al. 2011. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, pages 3104482–3104584.

Guanglin Niu and Bo Li. 2023. Logic and commonsense-guided temporal knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4569–4577.

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*, volume 35, pages 6471–6479.

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*.

Xing Tang, Ling Chen, Hongyu Shi, and Dandan Lyu. 2024. Dhyper: A recurrent dual hypergraph neural network for event prediction in temporal knowledge graphs. *ACM Transactions on Information Systems*, 42(5):1–23.

Volker Tresp, Yunpu Ma, Stephan Baier, and Yinchong Yang. 2017. Embedding learning for declarative memories. In *The Semantic Web: 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28–June 1, 2017, Proceedings, Part I 14*, pages 202–216. Springer.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Jiapu Wang, Sun Kai, Linhao Luo, Wei Wei, Yongli Hu, Alan Wee-Chung Liew, Shirui Pan, and Baocai Yin. 2025. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. *Advances in Neural Information Processing Systems*, 37:8384–8410.

Weiguang Wang, Yingying Feng, Haiyan Zhao, Xin Wang, Ruikai Cai, Wei Cai, and Xia Zhang. 2024. Mdpg: a novel multi-disease diagnosis prediction method based on patient knowledge graphs. *Health Information Science and Systems*, 12(1):15.

Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2019. Temporal knowledge graph embedding model based on additive time series decomposition. *arXiv preprint arXiv:1911.07893*.

Rui Ying, Mengting Hu, Jianfeng Wu, Yalan Xie, Xiaoyi Liu, Zhunheng Wang, Ming Jiang, Hang Gao, Linlin Zhang, and Renhong Cheng. 2024. Simple but effective compound geometric operations for temporal knowledge graph completion. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11074–11086.

Yuyue Zhao, Xiang Wang, Jiawei Chen, Yashen Wang, Wei Tang, Xiangnan He, and Haiyong Xie. 2022. Time-aware path reasoning on knowledge graph for recommendation. *ACM Transactions on Information Systems*, 41(2):1–26.