Improving Temporal Knowledge Graph Completion via Tensor Decomposition with Relation-Time Context and Multi-Time Perspective

Nam Le[1,2][a], Thanh Le[1,2][b] and Bac Le[1,2][c]
[1]*Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam*
[2]*Vietnam National University, Ho Chi Minh City, Vietnam*
{lnnam, lnthanh, lhbac}@fit.hcmus.edu.vn

Keywords: Knowledge Graph Completion, Temporal Knowledge Graph, Tensor Decomposition, Relation-Time Context, Fusion Feature Embedding.

Abstract: Knowledge graphs have progressively incorporated temporal dimensions to effectively mirror the dynamism of real-world data, proving instrumental in applications ranging from question answering to event prediction. While the ubiquity of data incompleteness and well-established challenges of traditional knowledge graph embedding techniques remain acknowledged, this paper propels the frontier of this research area. We introduce **M**ulti-Time **P**erspective Relation-Time Context **ComplEx** Embedding (MPComplEx), a tensor decomposition-based completion temporal knowledge graph model that not only assimilates temporal and relational interactions specific to timestamps but also integrates advanced time perspective features from the recent TPComplEx models. Our experimental evaluations illustrate dramatic enhancements over conventional models, achieving state-of-the-art performance on benchmark datasets with notable increments: 4.30%/4.79% on ICEWS-14, 11.70%/11.48% on ICEWS-05-15, 21.50%/31.20% on YAGO15k, and 26.90%/66.09% on GDELT in term of absolute/relative performance gains on mean reciprocal rank (MMR).

1 INTRODUCTION

In the contemporary era of information proliferation, many applications are swiftly emerging, leveraging the robust framework of Knowledge Graphs (KGs). These applications range from recommendation systems (Chen et al., 2022) to temporal question answering (Mavromatis et al., 2022). KGs act as repositories of real-world knowledge, encapsulating this information in the structured form of tuples *(subject, relation, object)*. To continue mining how events are involved in the timeline, Temporal Knowledge Graphs (TKGs), which extend KG, are constructed by introducing a temporal dimension to the representation of evolution knowledge. They facilitate the meticulous tracking of the evolution of events, encoding events as quadruples *(subject, relation, object, timestamp)*, with timestamp denoted by either time point or time interval. For example, a depicted TKG in Fig.1 illustrates Albert Einstein's tenure at ETH Zurich from 1912 to 1914.

Temporal Knowledge Graph completion (TKGC) is a reasoning task that aims to make the prediction for

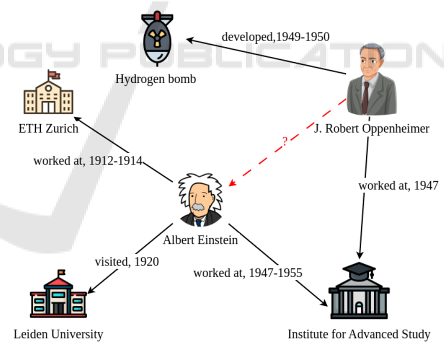

Figure 1: An example of a TKG is illustrated, with the task of predicting a missing event depicted as a dashed line.

the missing events that have a high probability of occurring, e.g., *(Albert Einstein, collaborated, J. Robert Oppenheimer, 1947-1955)* shown in Fig. 1. Recently, literature on this research field has divided into two main categories: (1) extrapolation-based model, which aims to predict future events based on historical fact records such as RE-GCN (Li et al., 2021) and DaeMon (Dong et al., 2023). (2) the remaining interpolation-based model is TKG embedding models (TKGE), which aim to predict missing events based on evaluating the plausibility of potential events via a scoring function with embedding vectors of enti-

[a] https://orcid.org/0000-0002-2273-5089
[b] https://orcid.org/0000-0002-2180-4222
[c] https://orcid.org/0000-0002-4306-6945

Le, N., Le, T. and Le, B.
Improving Temporal Knowledge Graph Completion via Tensor Decomposition with Relation-Time Context and Multi-Time Perspective.
DOI: 10.5220/0013130500003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 3*, pages 326-333
ISBN: 978-989-758-737-5; ISSN: 2184-433X

ties, relations and their associated timestamps, including TTransE (Leblay and Chekol, 2018), HyTE (Dasgupta et al., 2018), and TA-DistMult (García-Durán et al., 2018). This approach focuses on events that have no constraint on the occurring order. This work focuses on designing a TKGE model to tackle issues of existing interpolation-based models.

The recent development in interpolation-based models has significantly improved performance for this prediction task. Almost state-of-the-art models are based on tensor decomposition framework (Trouillon et al., 2016) such as TComplEx (Lacroix et al., 2019), TNTComplEx (Lacroix et al., 2019), TPComplEx (Yang et al., 2024) and MvTuckER (Wang et al., 2024). However, these models still face several challenges: (1) The flexibility of these models is limited; (2) Not utilizing temporal information to improve the quality of learning embedding during model training; (3) The connection between relations and the timestamps attached to them has not been fully exploited. To our best knowledge, TPComplEx (Yang et al., 2024) is the first model to incorporate temporal information into the learning model. Still, it does not consider the contribution of this information to the score function. Furthermore, temporal embedding and its additional are not linked with relation embedding.

To address the challenges outlined above, we introduce the **M**ulti-Time **P**erspective Relation-Time Context **ComplEx** Embedding (MPComplEx), a novel model specifically designed to enhance the flexibility and performance of TKGC tasks within the tensor decomposition framework. The significance of this model lies in its ability to capture the dynamic nature of relations in knowledge graphs over different periods, thereby improving the accuracy of temporal knowledge graph completion tasks. We mainly introduce adjustable weights for entities and additional time embeddings to obtain a more flexible scoring function, thereby increasing the model's flexibility in learning from any dataset and adapting to various tensor decomposition models. Furthermore, the connection between temporal and relation embedding is investigated and modeled as relation-time embedding and incorporated with relation embedding via weighted combination action, thus allowing the time evolution information to join the decision process. Compared to TPComplEx, our proposed model has better generalization with the adjustable contribution of time information for head and tail entities. Moreover, the new fusion representation for relation helps enhance the embedded representation quality. The main contributions of our work are summarized as follows:

- We propose a novel temporal knowledge graph completion model, MPComplEx, based on tensor decomposition frameworks with weighted feature combination strategy and incorporating the feature of connection between relation and its associated timestamp.
- In the proposed model, we introduce weights associated with head and tail entities for controlling their behaviors. Besides that, the participants of additional temporal embeddings are also controlled similarly to capture multiple time perspectives. Furthermore, the correlation of relation - timestamp is modeled via a dot product and is combined and weighted with relational embedding to enhance the quality of learned embedding.
- Our experimental results on standard benchmark datasets of TKGs show significant improvements over conventional models across all link prediction metrics.

The remainder of our paper is organized as follows: Section 2 introduces related works, focusing mainly on TKGC models based on tensor decomposition. Section 3 details our proposed model. Section 4 discusses the experimental setup, primary findings, and ablation studies. Finally, Section 5 summarizes our findings and outlines potential exciting directions for future research.

2 RELATED WORK

Conventional KGE models, such as TransE (Bordes et al., 2013) and ComplEx (Trouillon et al., 2016), try to forecast connections by acquiring embeddings for entities and predicates, hence evaluating the credibility of facts. These models have developed to effectively process intricate relational patterns, with recent innovations such as RotatE (Sun et al., 2019) specifically designed to handle various relational patterns such as symmetric and anti-symmetric relation patterns.

Building upon this paradigm, TKGE models integrate temporal events to capture evolving relationships. For instance, TTransE (Leblay and Chekol, 2018) incorporates relations and timestamps into a unified space, enhancing the original TransE model. HyTE (Dasgupta et al., 2018) applies a mapping function to each timestamp, translating entities and relations by adjusting them to a hyperplane. The TeRo (Xu et al., 2020) model enhances entity embeddings by incorporating timestamps to represent temporal progression. It utilizes relational rotations to capture temporal dynamics. Recent models such as

TA-DistMult (García-Durán et al., 2018) and TComplEx (Lacroix et al., 2019) extended based on previous models based on tensor decomposition for static data with more advantages to capture the time evolution. TA-DistMult decomposes timestamps into individual tokens and incorporates them into relation representations using Recurrent Neural Networks. Also, TComplEx enhances the ComplEx model by utilizing complex-valued vectors to handle relations that are not symmetric. TimePlex (Jain et al., 2020b) leverages the recurring nature of events to facilitate dynamic relational interactions. ChronoR (Sadeghian et al., 2021) extends the RotatE model by connecting timestamps with relations and seeing their combination as a rotational transformation. TPCompEx (Yang et al., 2024) introduces modules incorporating distinct temporal embeddings into things, considering various time perspectives. However, dealing with events with the same relation and co-occur takes work.

3 THE PROPOSED MODEL

Given temporal knowledge graph $\mathcal{G} = \{\mathcal{Q}, \mathcal{E}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{E}$, $\mathcal{R}$, $\mathcal{T}$, and $\mathcal{Q}$ respectively represent the sets of entities, relations, timestamps, and quadruplets. A quadruplet is denoted as (s,r,o,t), where $r \in \mathcal{R}$ represents the connection between a subject (head entity) $s \in \mathcal{E}$ and an object (tail entity) $o \in \mathcal{E}$ at a specific timestamp $t \in \mathcal{T}$. In any TKG, the number of relations is much less than that of entities, so entity prediction is more challenging than relation prediction. Therefore, TKC tasks often focus on predicting missing entities in a given data set like $(s,r,?,t)$ where ? donates the missing element.

3.1 Baseline Models: TComplEx and TPComplEx

This section examines two decomposition models, TComplEx and TPComplex, for the TKC problem. In order to address the TKC problem, TComplEx (Lacroix et al., 2019) is an augmented iteration of the ComplEx decomposition model. This approach uses complex vector embedding and Hermitian products to compute scores for a collection of four facts. Its score function can be formulated as follows:

$$\phi(s,r,o,t) = \mathrm{Re}\left(\left\langle C_s, C_r, \overline{C_o}, C_t \right\rangle\right), \tag{1}$$

where $\phi()$ denotes the scoring function, Re(.) returns the real vector component for input embedding; $C_s, C_r, C_o, C_t \in \mathbb{R}^{2\times d}$ denotes the complex embedding with embedding rank d for subject, relation, object, and timestamp, respectively. Recently, TPComplEx (Yang et al., 2024) investigated that TComplEx can not handle inversion relation on timestamp, i.e., its temporal complex embedding will degenerate to the real or imaginary part if this relation type exists. Introducing additional temporal complex embeddings shows the potential in modeling relations with the property of simultaneousness. Its score function can be formulated as follows:

$$\phi(s,r,o,t) = \mathrm{Re}\left(\left\langle C_s + C_{t_2}, C_r, \overline{C_o + C_{t_3}}, C_{t_1} \right\rangle\right), \tag{2}$$

where C_{t_1} is the temporal embedding, and C_{t_2}, C_{t_3} are additional temporal embedding for subject and object, respectively.

Based on these theoretical considerations, TPComplEx possesses the capability to represent relational patterns that adhere to the property of simultaneousness (see (Yang et al., 2024, Definition 3)). Often, the limitations obtained for temporal embedding are excessively stringent. In addition, according to our observations, TPComplEx fails to consider the relationship between a given relation and the time component, which might significantly impact the ability to reason about missing links in graph data. In the following section, we present a detailed and systematic guide to implementing our proposed strategy, which offers more significant potential in integrating diverse perspectives of time and the underlying connections between relation and its associated timestamps.

3.2 Fusing Relation-Time Context and Time Properties

Following the methodology of TPComplEx, we incorporate additional temporal embeddings into TComplEx with control variables $\alpha_1, \alpha_2 \in \mathbb{R}^+ \cup \{0\}$ and $\beta_1, \beta_2 \in \mathbb{R}^+ \cup \{0\}$ for subject and object embedding, respectively. Consequently, the score function becomes:

$$\begin{aligned}\phi_1(s,r,o,t) = \phi_{\text{base}}(s,r,o,t) + G_r^{t_2}(C_{t_2}) \\ + G_r^{t_3}(C_{t_3}) + G_r^{t_4}(C_{t_4}),\end{aligned} \tag{3}$$

where $\phi_{\text{base}}(s,r,o,t) = \mathrm{Re}\left(\left\langle \alpha_1 C_s, C_r, \beta_1 \overline{C_o}, C_{t_1} \right\rangle\right)$, C_{t_1} represents the temporal embedding, while $C_{t_2}, C_{t_3}, C_{t_4}$ denote additional temporal embeddings. The weights $\{\alpha_1, \beta_1,\}$ are associated with the subject and object embeddings, respectively. And $\{\alpha_2, \beta_2\}$ correspond to the additional temporal embeddings C_{t_2}, and C_{t_3}, respectively. For C_{t_4} embedding, we will handle it later to simplify the model structure.

For subject and object aggregation (see (Yang et al., 2024, Definition 4)), we define two first addi-

Table 1: Model parameters and embedding ranks for our baselines and proposed models.

Models	Parameters	ICEWS14	ICEWS05-15	YAGO15k	GDELT
ComplEx	$2d(\|\mathcal{E}\|+2\|\mathcal{R}\|)$	1820	860	1960	3820
TComplEx	$2d(\|\mathcal{E}\|+\|T\|+2\|\mathcal{R}\|)$	1740	1360	1940	2270
TPComplEx	$2d(\|\mathcal{E}\|+3\|\mathcal{T}\|+2\|\mathcal{R}\|)$	1594	886	1892	1256
MPComplEx	$2d(\|\mathcal{E}\|+3\|\mathcal{T}\|+2\|\mathcal{R}\|)$	1500	800	1500	1200

tional embeddings as:

$$G_r^{t_2}(C_{t_2}) = \operatorname{Re}\left(\left\langle C_r, \overline{C_o}, C_{t_1}, \alpha_2 C_{t_2}\right\rangle\right), \tag{4}$$

$$G_r^{t_3}(C_{t_3}) = \operatorname{Re}\left(\left\langle C_s, C_r, C_{t_1}, \beta_2 \overline{C_{t_3}}\right\rangle\right). \tag{5}$$

And for modelling the associated relation and timestamp (see (Yang et al., 2024, Definition 5)), we can formulate $G_r^{t_4}(C_{t_4}) = \operatorname{Re}(\langle C_r, C_{t_1}, C_{t_4}\rangle)$. To simplify the model structure, the additional embedding C_{t_4} can be defined as $C_{t_4} = \left\langle \alpha_2 C_{t_2}, \beta_2 \overline{C_{t_3}}\right\rangle$. Thus, we have:

$$G_r^{t_4}(C_{t_4}) = \operatorname{Re}\left(\left\langle C_r, C_{t_1}, \alpha_2 C_{t_2}, \beta_2 \overline{C_{t_3}}\right\rangle\right). \tag{6}$$

To construct the score function, we put Eq. 4, 5 and 6 into the Eq. 3. Thus, we obtain a new score function, which is defined as:

$$\begin{aligned} \phi_1(s,r,o,t) = \phi_{\text{base}}(s,r,o,t) & \\ + \operatorname{Re}\left(\left\langle C_r, \beta_1 \overline{C_o}, C_{t_1}, \alpha_2 C_{t_2}\right\rangle\right) & \\ + \operatorname{Re}\left(\left\langle \alpha_1 C_s, C_r, C_{t_1}, \beta_2 \overline{C_{t_3}}\right\rangle\right) & \\ + \operatorname{Re}\left(\left\langle C_r, C_{t_1}, \alpha_2 C_{t_2}, \beta_2 \overline{C_{t_3}}\right\rangle\right) & \end{aligned} \tag{7}$$

Simplifying the above expression, we have:

$$\phi_1(s,r,o,t) = \operatorname{Re}\left(\left\langle C_{st_2}, C_r, \overline{C_{ot_3}}, C_{t_1}\right\rangle\right), \tag{8}$$

where $C_{st_2} = \alpha_1 C_s + \alpha_2 C_{t_2}$, $C_{ot_3} = \beta_1 C_o + \beta_2 C_{t_3}$. This version is called eTPComplEx (*extended TPComplEx*). Clearly, when $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = 1$, Eq. 8 becomes the score function of TPComplEx.

To capture the interaction between relation embedding and temporal embedding, we design a new embedding that can model the interaction between them. This embedding, referred to as the relation-time context embedding, is denoted by $C_{rt} = \langle C_r, C_t\rangle$. Utilizing this embedding, we then construct a score function with previous defined additional temporal embedding C_{t_2}, C_{t_3} and C_{t_4} as follows:

$$\begin{aligned} \phi_2(s,r,o,t) = \phi_{\text{base}}(s,r,o,t) + G_r^{t_2}(\alpha_2 C_{t_2}) & \\ + G_r^{t_3}(\beta_2 C_{t_3}) + G_r^{t_4}(C_{t_4}). & \end{aligned} \tag{9}$$

By expanding the above score function similar to Eq. 7, we have the score function when using relation-time context embedding as follows:

$$\phi_2(s,r,o,t) = \operatorname{Re}\left(\left\langle C_{st_2}, C_{rt}, \overline{C_{ot_3}}, C_{t_1}\right\rangle\right), \tag{10}$$

This version is called cTPComplEx (*relation-time context TPComplEx*).

By combining the weighted score functions from Eq. 10 and Eq. 8, we derive a more generalized score function, which is formulated as follows:

$$\begin{aligned} \phi(s,r,o,t) &= \phi_1(s,r,o,t) + (1-\gamma)\phi_2(s,r,o,t) \\ &= \operatorname{Re}\left(\left\langle C_{st_2}, C_{rc}, \overline{C_{ot_3}}, C_{t_1}\right\rangle\right), \end{aligned} \tag{11}$$

where the combined embedding C_{rc} is defined as $C_{rc} = \gamma C_r + (1-\gamma) C_{rt}$. Here, γ is the weight factor that balances the percentage of features of the relation derived from C_r and the relation-time context embedding C_{rt}. This version is named MPComplEx.

Clearly, when $\alpha_1 = \alpha_2 = 1, \beta_1 = \beta_2 = 0, \gamma = 1$, the score function in Eq. 11 becomes the score function of TComplEx. Moreover, in the case of $\alpha_1 = \alpha_2 = \beta_1 = \beta_2 = 1, \gamma = 1$, the score function of MPComplEx becomes the score function of TPComplEx.

3.3 Optimization

Following the methodologies outlined in (Lacroix et al., 2019; Yang et al., 2024), we compute the instantaneous multi-class loss for each training quadruple (s,r,o,t) as follows:

$$\mathcal{L} = -\phi(s,r,o,t) + \log\left[\sum_{\substack{o' \neq o \\ o' \in \mathcal{E}}} \exp\left(\phi(s,r,o',t)\right)\right], \tag{12}$$

where $\phi(.)$ represents the score function. In addition, we include a regularization term, $\mathcal{L}_{reg}$. Therefore, the final loss function used for training is given by:

$$\mathcal{L}_{\text{total}} = \mathcal{L} + \mathcal{L}_{reg}. \tag{13}$$

Similarly to TPComplEx (Yang et al., 2024), our model incorporates entities with temporal bias into the regularization process and adopts N3 regularization as defined by (Lacroix et al., 2019). The regularization function is expressed as:

$$\begin{aligned} \mathcal{L}_{reg} = \lambda_1 \left(\|C_{st_2}\|_3^3 + \|C_{rc}\|_3^3 + \|C_{ot_3}\|_3^3\right) & \\ + \lambda_2 \|C_t\|_3^3, & \end{aligned} \tag{14}$$

where λ_1 and λ_2 are the regularization weights for the entity-relation and temporal embeddings, respectively.

Table 2: Statistic information of four standard benchmark datasets. The first three columns present the number of entities, relations, and timestamps, and remain columns present the number of quadruples for each dataset.

Dataset	#Entities	#Relations	#Timestamps	#Train	#Validation	#Test
ICEWS14	7128	230	365	72,826	8941	8963
ICEWS05-15	10,488	251	4017	386,962	46,275	46,092
YAGO15k	15,403	34	198	110,441	13,815	13,800
GDELT	500	20	366	2,735,685	341,961	341,961

3.4 Computational Complexity

Table 1 presents the computational complexity of embedding models through the number of parameters required for training and the embedding ranks or dimensions. It demonstrates that our models maintain the same parameter count as other tensor decomposition-based TKGC models, such as TComplEx and TPComplEx.

4 EXPERIMENTS AND RESULTS

4.1 Experiment Setup

4.1.1 Standard Benchmark Datasets

During the experiment process, four standard benchmark datasets of TKGs are used, namely ICEWS14, ICEWS05-15, YAGO15k (García-Durán et al., 2018), and GDELT (Trivedi et al., 2017). Table 2 summarizes the details of the four datasets.

4.1.2 Baselines

We evaluate our proposed models by comparing them to previous well-performed TKGE models that are considered to be at the forefront of the field, include: TTransE (Leblay and Chekol, 2018); TComplEx, TNTComplEx (Lacroix et al., 2019); TimePlex (Jain et al., 2020a); ChronoR (Sadeghian et al., 2021); TeLM (Xu et al., 2021); BTDG (Lai et al., 2022); TBDRI (Yu et al., 2023); SANe (Li et al., 2024); MTComplEx (Zhang et al., 2024); TPComplEx (Yang et al., 2024); MvTuckER (Wang et al., 2024). Our model is based on TPComplEx, are designed to improve performance while maintaining the same number of embedding parameters.

4.1.3 Metrics and Implementation Details

To evaluate our proposed model, after ranking all the candidates according to their scores calculated by the scoring function, we employ two metrics that are used widely in temporal knowledge graph research including Mean Reciprocal Rank (MRR), and Hit@k. The higher MRR and Hits@n indicate better performance. Our models are based on TComplEx (Lacroix et al., 2019) and TPComplEx (Yang et al., 2024), utilizing the PyTorch library (Paszke et al., 2019) and running on NVIDIA GeForce RTX 3070 8Gb VRAM. Following the methodologies of TPComplEx, we tune our models using grid search to optimize hyperparameters based on validation dataset performance. Control variables for entity embeddings, such as $\alpha_1, \alpha_2, \beta_1, \beta_2$, are adjusted within $\{1, 1.5, 2, 2.5\}$, while those for relation embedding, γ, range from $\{0, 0.25, 0.5, 0.75, 0.85, 0.95\}$. Regularization rates λ_1 and λ_2 are set within $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$. During training, we maintain a consistent batch size of 1000 and employ Adagrad (Duchi et al., 2011) to optimize our model with a fixed learning rate of 0.1 across all datasets. Our source codes is available at https://github.com/lnhutnam/MPComplEx.

4.2 Comparative Study

To evaluate the capabilities of MPComplEx, we conducted a comparative assessment against the current state-of-the-art TKGC models, with results presented in Table 3. The performance metrics for all baseline models were directly sourced from the original paper on TPComplEx. Our findings indicate that the proposed model consistently outperforms all baseline models across the four evaluation datasets. Specifically, compared to the top-performing model, TPComplEx, our proposed approach demonstrates significant improvements across all metrics. The differences between our MPComplEx model and TPComplEx are quantified through absolute performance gains (APG) and relative performance gains (RPG). In terms of APG, our model achieves improvements of 4.3%, 11.7%, 21.5%, and 26.9% on MRR and 6.6%, 16.3%, 24.8%, and 30.1% in Hit@1 for the ICEWS14, ICEWS05-15, YAGO15k, and GDELT datasets, respectively. Moreover, with RPG, our model achieves 4.79%, 11.48%, 31.20%, and 66.09% on MRR and 7.63%, 17.38%, 38.10%, and 91.49% on Hit@1 on these datasets.

Table 3: Experiment results on the ICEWS14, ICEWS05-15, YAGO15k, and GDELT datasets. The highest score is highlighted in **bold**, and the second-best score is underlined. The absolute performance gains (APG) and the relative performance gain (RPG) indicate the performance improvement of our model compared with the best-performing baseline TPComplEx. APG and RPG are calculated by $APG = R_{ours} - R_{baseline}$ and $RPG = (R_{ours} - R_{baseline})/R_{baseline}$ where R_{ours} and $R_{baseline}$ are the results of our model and baseline TPComplEx, respectively.

Model	MRR ↑	Hit@1 ↑	Hit@10 ↑	MRR ↑	Hit@1 ↑	Hit@10 ↑
	ICEWS14			ICEWS05-15		
TTransE (Leblay and Chekol, 2018)	0.255	0.074	0.601	0.271	0.084	0.616
TComplEx (Lacroix et al., 2019)	0.610	0.530	0.770	0.660	0.590	0.800
ChronoR (Sadeghian et al., 2021)	0.625	0.547	0.773	0.675	0.596	0.820
TeLM (Xu et al., 2021)	0.625	0.545	0.774	0.678	0.599	0.823
BTDG (Lai et al., 2022)	0.601	0.516	0.753	0.627	0.534	0.798
TBDRI (Yu et al., 2023)	0.652	0.552	0.785	0.709	0.646	0.821
SANe (Li et al., 2024)	0.638	0.558	0.782	0.683	0.605	0.823
MTComplEx (Zhang et al., 2024)	0.629	0.548	0.782	0.675	0.592	0.822
TPComplEx (Yang et al., 2024)	<u>0.898</u>	<u>0.865</u>	<u>0.954</u>	<u>0.845</u>	<u>0.794</u>	<u>0.934</u>
MvTuckER (Wang et al., 2024)	0.654	0.577	0.797	0.698	0.618	0.841
MPComplEx (Ours)	**0.941**	**0.931**	**0.957**	**0.962**	**0.957**	**0.974**
APG (%) ↑	**4.30**	**6.60**	**0.30**	**11.70**	**16.30**	**4.00**
RPG (%) ↑	**4.79**	**7.63**	**0.31**	**11.48**	**17.38**	**2.78**
	YAGO15k			GDELT		
TTransE (Leblay and Chekol, 2018)	0.321	0.230	0.510	0.115	0.000	0.318
TComplEx (Lacroix et al., 2019)	0.360	0.280	0.540	0.298	0.213	0.464
ChronoR (Sadeghian et al., 2021)	0.366	0.292	0.538	-	-	-
TBDRI (Yu et al., 2023)	0.368	0.301	0.554	0.269	0.164	0.441
SANe (Li et al., 2024)	-	-	-	0.301	0.212	0.476
TPComplEx (Yang et al., 2024)	<u>0.689</u>	<u>0.651</u>	<u>0.762</u>	0.407	0.329	0.559
MvTuckER (Wang et al., 2024)	-	-	-	<u>0.549</u>	<u>0.477</u>	<u>0.682</u>
MPComplEx (Ours)	**0.904**	**0.899**	**0.914**	**0.676**	**0.630**	**0.762**
APG (%) ↑	**21.50**	**24.80**	**15.20**	**26.90**	**30.1**	**20.30**
RPG (%) ↑	**31.20**	**38.10**	**19.95**	**66.09**	**91.49**	**36.31**

These experimental results demonstrate the effectiveness of incorporating weighted combinations of additional temporal embeddings with their corresponding subject and object embeddings, which significantly enhances the flexibility of our model compared to TPComplEx. Furthermore, integrating weighted time-relational features into relation embedding has allowed the proposed model to substantially improve over TPComplEx, particularly in cases where the facts recorded by each timestamp are few or unique, as observed in the YAGO15k dataset. Additionally, for datasets with fewer entities and relations but a large number of facts, such as GDELT, our model exhibits scalability with large-scale data, requiring fewer hyperparameters than previous tensor decomposition models like ComplEx, TComplEx or TPComplEx while still delivering notable performance across various evaluation metrics.

4.3 Ablation Study

4.3.1 Analysis of the Effects of Relation-Time Context Features

To assess the effectiveness of relation-time features, we varied the proportion of these features by adjusting the hyperparameter γ. A value of $\gamma = 1$ corresponds to using only the original relation embedding, while $\gamma = 0$ indicates exclusive use of the relation-time feature embedding. The results, shown in Table 4, show that relation-time features make the proposed model better by an average of 0.76% across four datasets compared to the model that does not use these features. However, in the GDELT dataset, these features do not significantly impact performance. This outcome highlights the challenges in optimally tuning control variables for original relations and relation-time features.

Table 4: The influence of relation-time context features on the datasets.

Case study	MRR ↑	Hit@1 ↑	Hit@10 ↑	MRR ↑	Hit@1 ↑	Hit@10 ↑
	ICEWS14			ICEWS05-15		
Only relation-time	0.923	0.912	0.940	0.930	0.920	0.948
W/o relation-time	0.920	0.910	0.940	0.946	0.939	0.963
Fusion features	0.941	0.931	0.957	0.962	0.957	0.974
	YAGO15k			GDELT		
Only relation-time	0.794	0.774	0.834	0.719	0.680	0.793
W/o relation-time	0.774	0.752	0.815	0.719	0.680	0.793
Fusion features	0.904	0.899	0.914	0.676	0.630	0.762

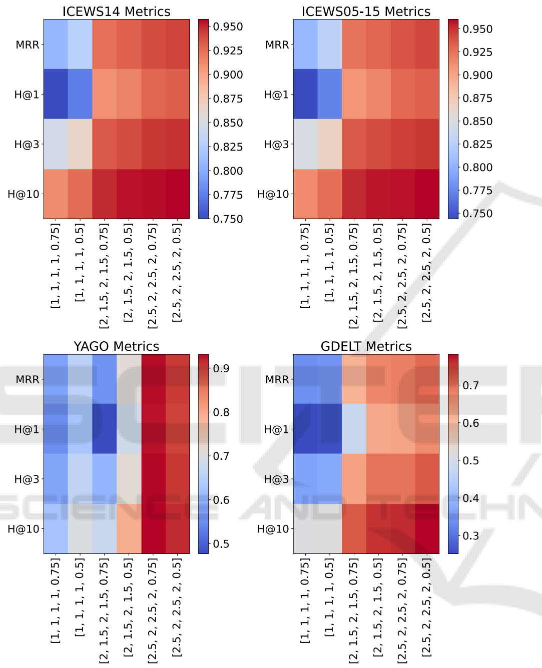

Figure 2: Visualization of the effect of $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma$ for MPComplEx on the ICEWS14, ICEWS05-15, YAGO15k, and GDELT datasets.

4.3.2 Analysis of the Effects of Weight Combinations

The proposed model incorporates weights for combining entity embeddings with additional temporal embeddings, specifically $\alpha_1, \alpha_2, \beta_1, \beta_2$, and relation embeddings with relation-time embeddings, denoted by γ. To evaluate the impact of these weights on model performance, we conduct experiments with different weight combinations across four datasets: ICEWS14, ICEWS05-15, YAGO15k, and GDELT. The results are presented in Fig. 2 which each hyper-parameter combination has order $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma$. For the ICEWS14, ICEWS05-15, and GDELT datasets, the MPComplEx model consistently improves across all evaluation metrics as the weight values increase, with optimal performance observed when the relation embedding weight is set to $\gamma = 0.5$. In contrast, for the YAGO15k dataset, smaller weight values yield sub-optimal results. However, when the weights are set to 2.5 or higher, coupled with a relation embedding weight of $\gamma = 0.75$, the model exhibits significant performance gains, achieving its highest performance.

5 CONCLUSIONS

This paper introduces the Multi-Time Perspective Relation-Time Context ComplEx Embedding model (MPComplEx), which addresses several key challenges of interpolation models based on tensor decomposition. By incorporating additional flexible temporal embeddings with adjustable weights, our model enhances flexibility and improves the ability to capture various time perspective properties while maintaining computational efficiency with a fixed number of parameters compared to baseline models. Especially, the correlation between relation embeddings and their corresponding timestamps is modeled and integrated with weighted contributions into the scoring function, thereby enhancing the quality of relation embeddings through temporal information and improving prediction results. Looking ahead, further exploration of the model's potential in managing cross-temporal patterns and addressing the challenge of extrapolation presents promising directions for future research.

ACKNOWLEDGEMENTS

This research is supported by research funding from Faculty of Information Technology, University of Science, Vietnam National University - Ho Chi Minh City.

REFERENCES

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Chen, W., Wan, H., Guo, S., Huang, H., Zheng, S., Li, J., Lin, S., and Lin, Y. (2022). Building and exploiting spatial–temporal knowledge graph for next poi recommendation. *Knowledge-Based Systems*, 258:109951.

Dasgupta, S. S., Ray, S. N., and Talukdar, P. P. (2018). Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Conference on Empirical Methods in Natural Language Processing*.

Dong, H., Ning, Z., Wang, P., Qiao, Z., Wang, P., Zhou, Y., and Fu, Y. (2023). Adaptive path-memory network for temporal knowledge graph reasoning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 2086–2094.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

García-Durán, A., Dumancic, S., and Niepert, M. (2018). Learning sequence encoders for temporal knowledge graph completion. In *Conference on Empirical Methods in Natural Language Processing*.

Jain, P., Rathi, S., Chakrabarti, S., et al. (2020a). Temporal knowledge base completion: New algorithms and evaluation protocols. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3733–3747.

Jain, P., Rathi, S., Mausam, and Chakrabarti, S. (2020b). Temporal Knowledge Base Completion: New Algorithms and Evaluation Protocols. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3733–3747. Association for Computational Linguistics.

Lacroix, T., Obozinski, G., and Usunier, N. (2019). Tensor decompositions for temporal knowledge base completion. In *International Conference on Learning Representations*.

Lai, Y., Chen, C., Zheng, Z., and Zhang, Y. (2022). Block term decomposition with distinct time granularities for temporal knowledge graph completion. *Expert Systems with Applications*, 201:117036.

Leblay, J. and Chekol, M. W. (2018). Deriving validity time in knowledge graph. *Companion Proceedings of the The Web Conference 2018*.

Li, Y., Zhang, X., Zhang, B., Huang, F., Chen, X., Lu, M., and Ma, S. (2024). Sane: Space adaptation network for temporal knowledge graph completion. *Information Sciences*, 667:120430.

Li, Z., Jin, X., Li, W., Guan, S., Guo, J., Shen, H., Wang, Y., and Cheng, X. (2021). 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.

Mavromatis, C., Subramanyam, P. L., Ioannidis, V. N., Adeshina, A., Howard, P. R., Grinberg, T., Hakim, N., and Karypis, G. (2022). Tempoqr: temporal question reasoning over knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, pages 5825–5833.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Sadeghian, A., Armandpour, M., Colas, A., and Wang, D. Z. (2021). Chronor: Rotation based temporal knowledge graph embedding. In *Proceedings of the AAAI conference on artificial intelligence*, pages 6471–6479.

Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. (2019). Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Trivedi, R., Dai, H., Wang, Y., and Song, L. (2017). Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *international conference on machine learning*, pages 3462–3471. PMLR.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016). Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR.

Wang, H., Yang, J., Yang, L. T., Gao, Y., Ding, J., Zhou, X., and Liu, H. (2024). Mvtucker: Multi-view knowledge graphs representation learning based on tensor tucker model. *Information Fusion*, 106:102249.

Xu, C., Chen, Y.-Y., Nayyeri, M., and Lehmann, J. (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*, pages 2569–2578.

Xu, C., Nayyeri, M., Alkhoury, F., Shariat Yazdi, H., and Lehmann, J. (2020). TeRo: A time-aware knowledge graph embedding via temporal rotation. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics.

Yang, J., Ying, X., Shi, Y., and Xing, B. (2024). Tensor decompositions for temporal knowledge graph completion with time perspective. *Expert Systems with Applications*, 237:121267.

Yu, M., Guo, J., Yu, J., Xu, T., Zhao, M., Liu, H., Li, X., and Yu, R. (2023). Tbdri: block decomposition based on relational interaction for temporal knowledge graph completion. *Applied Intelligence*, 53(5):5072–5084.

Zhang, F., Chen, H., Shi, Y., Cheng, J., and Lin, J. (2024). Joint framework for tensor decomposition-based temporal knowledge graph completion. *Information Sciences*, 654:119853.