Tucker Decomposition with Frequency Attention for Temporal Knowledge Graph Completion

Likang Xiao[1,3], **Richong Zhang**[1], **Zijie Chen**[2], **Junfan Chen**[1]
[1]SKLSDE, School of Computer Science and Engineering, Beihang University, Beijing, China
[2]School of Electrical and Computer Engineering, University of Toronto, Toronto, Canada
[3]Shen Yuan Honors College, Beihang University, Beijing, China
{xiaolk, zhangrc, chenjf}@buaa.edu.cn
chenzijie162@gmail.com

Abstract

Temporal Knowledge Graph Completion aims to complete missing entities or relations under temporal constraints. Previous tensor decomposition-based models for TKGC only independently consider the combination of one single relation with one single timestamp, ignoring the global nature of the embedding. We propose a Frequency Attention (FA) model to capture the global temporal dependencies between one relation and the entire timestamp. Specifically, we use Discrete Cosine Transform (DCT) to capture the frequency of the timestamp embedding and further compute the frequency attention weight to scale embedding. Meanwhile, the previous temporal tucker decomposition method uses a simple norm regularization to constrain the core tensor, which limits the optimization performance. Thus, we propose Orthogonal Regularization (OR) variants for the core tensor, which can limit the non-superdiagonal elements of the 3-rd core tensor. Experiments on three standard TKGC datasets demonstrate that our method outperforms the state-of-the-art results on several metrics. The results suggest that the direct-current component is not the best feature for TKG representation learning. Additional analysis shows the effectiveness of our FA and OR models, even with smaller embedding dimensions.

1 Introduction

Knowledge graph (KG) contains a number of structured facts (h, r, t), where a fact expresses a directed relation r from a head entity h to a tail entity t. The complex KGs, such as FreeBase (Berant et al., 2013), DBPedia (Auer et al., 2007), and Wikidata (Vrandečić and Krötzsch, 2014), are collected manually or automatically from structured or unstructured data on the web. Such KGs are successfully applied to several downstream tasks, e.g., Question Answering (Berant et al., 2013) and Recommender System (Wang et al., 2018). However, those works ignore that many facts in the

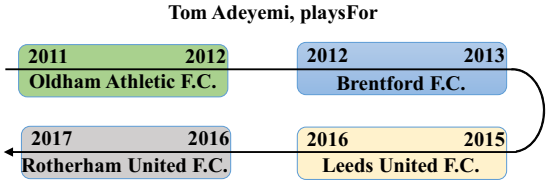


Figure 1: A toy example from the temporal knowledge graph shows an athlete's career. This example illustrates the temporal dependencies of facts that Tom Adeyemi plays for four teams from 2011 to 2017. Toy example scores calculated by TuckER-FA are in the appendix C.

KGs change over time. Temporal facts can be expressed as a quadruplet (h, r, t, τ), with τ being the timestamp. The temporal Knowledge graphs (TKGs), such as ICEWS (Lautenschlager et al., 2015), GDELT (Leetaru and Schrodt, 2013) and YAGO (Mahdisoltani et al., 2015), are then built to handle these facts coupled with timestamps.

One problem that hinders the application of TKGs in downstream tasks is the inevitable incompleteness or knowledge scarcity problem caused by missing entities or relations. Thus, Temporal Knowledge Graph Completion (TKGC) aiming to complete the missing entities or relations over time has become an essential task in the research community. The previous methods for TKGC can be divided into four branches, in which the critical challenge is how to integrate timestamps into KGC modeling. Time-dependent Embedding method (Trivedi et al., 2017; Goel et al., 2020; Dasgupta et al., 2018) considers the temporal information as a transformation or an activation function for entities or relations. Timestamp Embedding method (Han et al., 2021b; Lacroix et al., 2020; Shao et al., 2022) treats timestamps as additional learnable embeddings of the score function. Experimental experience (Han et al., 2021b) suggests that timestamp embeddings generally perform better than time-dependent embeddings. Knowledge Graph Snapshots method (Liao et al., 2021; Li et al.,

Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023), pages 253–265

2021) aggregates multi-relational interactions of cropped subgraph to achieve more precise representations. Historical Context method (Jung et al., 2021; Zhu et al., 2021) model n-hop facts chain or repeat facts to increase the interpretability.

The previous timestamp embedding methods model each quadruplet independently, which only captures the *local* temporal dependencies, ignoring the *global* temporal dependencies between one relation and entire timestamp. As shown in Figure 1, an athlete plays for different teams in different periods. We treat such events as a continuous line of events rather than as separate events. We propose a Frequency Attention (FA) model to address this issue. Specifically, we treat each dimension in the timestamp embedding as a long-term signal and use Discrete Cosine Transform (DCT) to capture the frequency of the signal. Furthermore, we take the frequency and part of the relation embedding as input to calculate attention weights for each timestamp. The proposed frequency attention model can easily apply to exist tensor decomposition methods.

The previous tucker decomposition method is interpreted as a high-dimensional linear classifier to distinguish facts. TuckERTNT (Shao et al., 2022), uses a simple L2 norm as the core tensor regularization. However, this regularization may be overstrict, leading the embedding to change sharply and risk vanishing or explosion. Inspired by orthogonal regularization in (Brock et al., 2019), we propose two variants of orthogonal regularization (OR) for the core tensor, i.e., excluding superdiagonal elements or diagonal elements of each slice matrix of the core tensor. This way, we achieve a balanced core tensor regularization, preventing the embedding norm from vanishing or exploding.

In summary, our work makes the following contributions:

(1) we propose a frequency attention model using DCT to capture the global temporal dependency between relations and entire timestamp.

(2) we introduce two variants of core tensor orthogonal regularization for the tucker decomposition, which can prevent the embedding norm from vanishing or explosion.

(3) Experiment results on three standard datasets show that our model outperforms the SOTA models on several metrics. The additional analysis demonstrates the effectiveness of our frequency attention model and orthogonal regularization.

2 Related Works

2.1 Static KG Embedding

There has been ample research on static knowledge graph embedding. We grouped all mainstream models into four main categories. Tensor decomposition-based models RESCAL (Nickel et al., 2011), Distmult(Yang et al., 2015), ComplEx (Trouillon et al., 2016), and TuckER (Balazevic et al., 2019) compute triplet score in real or complex domain. Distance-based models are built upon euclidean or hyperbolic distance as shown in TransE (Bordes et al., 2013), mainfoldE (Xiao et al., 2016) and RotatE (Sun et al., 2019). DURA (Zhang et al., 2020) figure out that distance-based method can be viewed as decomposition method with a L_2 regularization. Neural-based models use a convolutional network to capture the KG structure information, as shown in ConvE (Dettmers et al., 2018). Other models learn from a variety of experiences from fields. Inspired by reinforcement learning, MultiHopKG (Lin et al., 2018) sample n-hop triplets chain to compute the fact triplet scores and re-rank candidates.

2.2 Temporal KG Embedding

There are two scenarios for integrating temporal information into existing static embedding models, timestamp embedding and time-dependent entity embedding. Time-dependent entity embedding can explicitly model dynamic changes of entities, such as periodicity and trending. A well-known time-dependent entity embedding is diachronic embedding (Goel et al., 2020), which uses the sine function to represent the frequency of entity evolution over different time granularity. (Han et al., 2021b) compares six KG embedding models, and figures out that timestamp embedding can achieve similar or even better performance with significantly fewer parameters. Although timestamp embedding might suffer from the growing number of timestamps, the time granularity can also be controlled within an appropriate range by enlarging. Further analysis in TNTComplEx (Lacroix et al., 2020) points out that time-dependent relation embedding can obtain comparable results to time-dependent entity embedding with smaller computational costs.

From the viewpoint of the subgraph, there are a series of knowledge graph snapshots/subgraphs over time, which contain potential multi-relational interactions. (Liao et al., 2021) adopt probabilistic entity representations based on variational

Bayesian inference to jointly model the entity features and the uncertainty. (Li et al., 2021) employs a multi-layer graph convolutional network on each subgraph to capture the dependencies of adjacent facts. From another contextual perspective, the relevance between the query and its historical context can be used as evidence for reasoning. (Jung et al., 2021) proposes a multi-hop reasoning model using a graph attention layer and finds that temporal displacements are more indicative for inference than timestamps. (Zhu et al., 2021) notice more than 80% of events from 1995 to 2019 in the ICEWS repository are repeated events. In this case, they introduce a copy mechanism to re-rank the candidates.

3 Preliminaries

3.1 Problem Definition

To formally define the problem and describe the solution, we use consistent notations in the rest of the paper. We represent scalars with the lower case letters, e.g., d_r, represent sets with the flower letters, e.g., $\mathcal{E}$, represent vectors with the bold lower case letters, e.g., $\boldsymbol{h}_i$, denoting the i^{th} entity embedding. We use bold upper letters $\boldsymbol{H}$ to denote the embedding matrix and represent the high order tensor with bold flower letters $\boldsymbol{\mathcal{W}}$.

We use $\boldsymbol{a} \odot \boldsymbol{b}$ to denote Hadamard (element-wise) product of two vectors or matrix, $\boldsymbol{a} \otimes \boldsymbol{b}$ to denote the tensor outer product, $[\boldsymbol{A}|\boldsymbol{B}]$ to denote the vector or matrix concatenation operator, $||\cdot||_p$ to denote the p-norm of a vector or tensor.

A temporal knowledge graph $\mathcal{G}$ consists of a set of facts $\{(h_i, r_j, t_k, \tau_l)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T}$, where $\mathcal{E}$ is a finite entity set, $\mathcal{R}$ is a finite relation set and $\mathcal{T}$ is a finite timestamp set. Each quadruplet (h_i, r_j, t_k, τ_l) respectively denotes a relation r_j from the head entity h_i to tail entity t_k at a specific time τ_l. A temporal knowledge graph uses a binary tensor $\mathcal{X} = \{0,1\}^{|\mathcal{E}|\times|\mathcal{R}|\times|\mathcal{E}|\times|\mathcal{T}|}$ to indicate whether the corresponding quadruplets occurs in the KG data set. $|\mathcal{E}|$, $|\mathcal{R}|$ and $|\mathcal{T}|$ denote the number of entities, relations, and timestamps, respectively.

Although knowledge graphs contain large numbers of facts, they are still incomplete due to the complex nature of the real world. TKGC aims to predict the missing entity. We focus on the link prediction problem, aiming to predict the tail entity or head entity through the query $(h_i, r_j, ?, \tau_l)$ or $(?, r_j, t_k, \tau_l)$. The problem reduces to ranking a set of candidate entities to select the most likely entity that makes the partial quadruplet factual. The problem can be formulated as a ranking problem to learn a quadruplet score function $\hat{\mathcal{X}}(\mathcal{E}, \mathcal{R}, \mathcal{E}, \mathcal{T}) \in \mathbb{R}$ to sort all candidate entities.

3.2 Tucker Decomposition for TKG Embedding

Many tensor decomposition methods apply to the KG embeddings, such as bilinear decomposition, canonical decomposition, and tucker decomposition. Among these methods, tucker decomposition, a kind of principal component analysis approach for high-order tensors, is viewed as the general one. In particular, when the super-diagonal elements in the core tensor of Tucker equal 1 and other elements equal 0, tucker decomposition degrades into canonical decomposition. TuckER (Balazevic et al., 2019) has proved that Dismult (Yang et al., 2015) and ComplEx (Trouillon et al., 2016) can be included into the framework of TuckER. In KGC task, An 3-order tensor $\boldsymbol{\mathcal{X}}$ can be decomposed into a core tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{D_e \times D_r \times D_e}$ and entity/relation embedding matrix $\boldsymbol{E}/\boldsymbol{R}$ as factor matrix. The formula of the tucker decomposition is as follows.

$$\begin{aligned}\mathcal{X} &= \boldsymbol{\mathcal{W}} \times_1 \boldsymbol{E} \times_2 \boldsymbol{R} \times_3 \boldsymbol{E} = <\boldsymbol{\mathcal{W}}; \boldsymbol{E}, \boldsymbol{R}, \boldsymbol{E}> \\ &= \sum_{d_1=1}^{D_e} \sum_{d_2=1}^{D_r} \sum_{d_3=1}^{D_e} W_{d_1 d_2 d_3} \boldsymbol{h}_{:d_1} \otimes \boldsymbol{r}_{:d_2} \otimes \boldsymbol{t}_{:d_3}\end{aligned}$$

$\times_n$ denotes the n-mode product of the tensor, which can be explained as the core tensor expanding into a matrix along the n-th dimension.

To obtain proper timestamp embedding $\boldsymbol{T}$, TuckERTNT (Shao et al., 2022) use two relation embeddings $\boldsymbol{R}$ and $\boldsymbol{R}^t$ to separately capture time-variant information and time-invariant information as follows.

$$\mathcal{X} = <\boldsymbol{\mathcal{W}}; \boldsymbol{E}, \boldsymbol{R}^t \odot \boldsymbol{T} + \boldsymbol{R}, \boldsymbol{E}>$$

Although previous works have achieved good results in the TKGC task, they may still encounter many problems. First, the learnable parameters representing the frequency of DE-SimplE may be clustered around 0, affecting the model performance (as shown in Appendix D). Second, TuckERTNT constrains the core tensor with the fourth power of L_4 norm. However, it is not guaranteed that the n-mode product of the core tensor is well-perform. Therefore, there is still space to improve the temporal tucker decomposition for the TKGC task.

4 Model

We propose a new framework, TuckER-FA, combining the FA and OR with the temporal tucker decomposition method. We input the timestamp embedding and relation embedding to the FA model to compute frequency attention weights, then weighted-sum the timestamp embedding and combine it with the relation embedding. The timestamp-enhanced relation feature and head/tail entity embedding compose the factor matrix of tucker decomposition. In learning progress, we include several regularization losses and the orthogonal regularization of the core tensor into the overall objective.

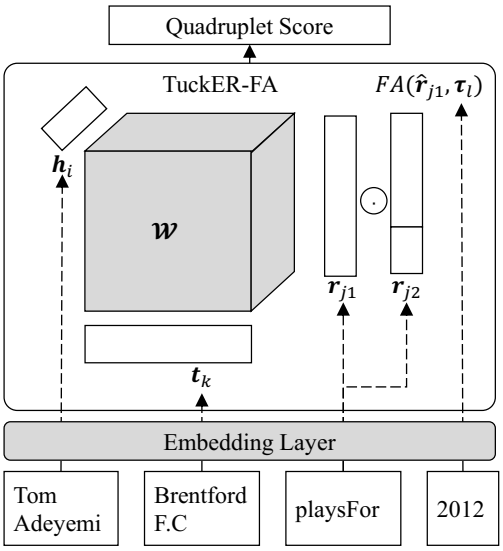


Figure 2: The score function of TuckER-FA, which can measure how likely the quadruplet is to be true. The gray part contains the learnable parameters.

4.1 Score Function

We propose a score function for TKGC based on tucker decomposition shown in Figure 2. Specifically, the formula of the score function is.

$$\mathcal{X}_{ijkl} = < \boldsymbol{\mathcal{W}}; \boldsymbol{h}_i, \boldsymbol{r}_{j1} \odot [\boldsymbol{r}_{j2} | FA(\hat{\boldsymbol{r}}_{j1}, \boldsymbol{t}_l)], \boldsymbol{t}_k >$$

We use $\rho = \frac{d_\tau}{d_r}$ to control the embedding dimension ratio between timestamps and relations. $r_{j1} \in \mathbb{R}^{d_r}$ and $r_{j2} \in \mathbb{R}^{d_r - d_\tau}$ respectively denote two relation embeddings. $\tau_l \in \mathbb{R}^{d_\tau}$ is the timestamp embedding. FA represents the Frequency Attention model. The input $\hat{\boldsymbol{r}}_{j1}$ denote the part of relation embedding which aligns with τ_l. The main amount of computation is concentrated on tucker decomposition rather than the FA model.

4.2 Frequency Attention

To capture the crucial temporal features of TKG, we propose a Frequency Attention (FA) model shown in Figure 3. We treat the evolution of timestamp embedding over time as a combination of periodic functions with different frequencies. Inspired by FcaNet (Qin et al., 2021), we use Discrete Cosine Transform (DCT) to capture the different frequency components of timestamp embeddings. In this way, we can capture the global temporal dependency of one relation and the entire timestamp.

The chronologically arranged timestamp embedding $\boldsymbol{T} \in \mathbb{R}^{N_\tau \times d_\tau}$ is viewed as d_τ different temporal signals. The direct-current (DC) component f_0 and frequency component f_k of DCT are respectively formulated as follows.

$$f_0 = \sum_{i=0}^{N_\tau - 1} T_i = GAP(T) N_\tau$$

$$f_k = \sum_{i=0}^{N_\tau - 1} T_i cos(\frac{\pi k}{N_\tau}(i + \frac{1}{2})) \quad k \in \{0, ..., N_\tau - 1\}$$

GAP represents the global average pooling operation, which always use to calculate the channel attention weight in the computer vision domain. In the TKGC task, the main calculation procedure of direct-current component frequency attention is as follows.

$$FA(\boldsymbol{\tau}_l) = \sigma(FC(GAP(T_{:d}) N_\tau)) \odot \tau_{:d}$$

The FC block represents a fully-connected layer, and σ denotes sigmoid function. It is natural to include more frequency components to calculate attention weight. Considering the limitation of computing resources, we optionally select part of the frequency components. We divide the time embedding dimension d_τ into n parts and assign a set of selected frequencies $f_0, ...f_n$ to each part. We also introduce $\hat{\boldsymbol{r}}_{j1}$, part of relation embedding aligned with τ_l, into the FA model.

$$FA(\hat{\boldsymbol{r}_j}, \boldsymbol{\tau}_l) = \sigma(FC(\hat{\boldsymbol{r}}_{j1} \odot [f_0 | f_1 | ... | f_n])) \odot \boldsymbol{\tau}_l$$

In addition, the computation complexity of the operation with finite orthogonal function bases is linear. The cost of computation of the frequency attention model is negligible compared to the cost of computation in tucker decomposition. The frequency attention weight determines how much timestamp embedding information for the corresponding dimension is retained. The FA model considers the evolution of a single relation over the entire timestamp.

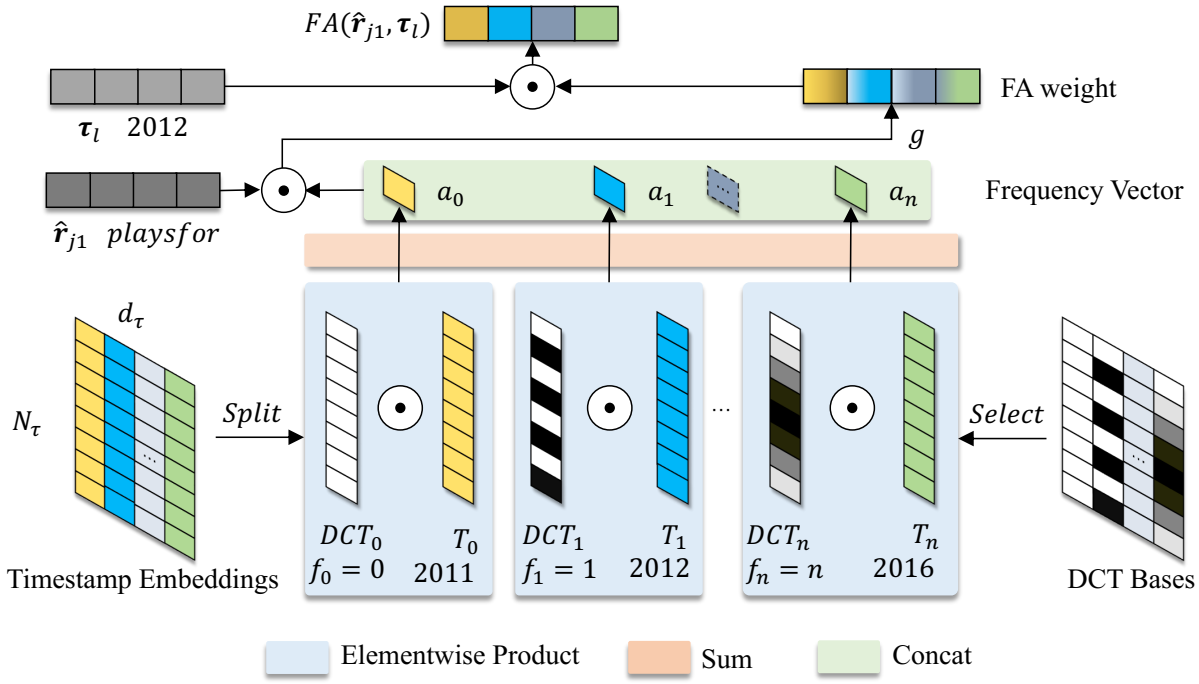


Figure 3: The framework of the Frequency Attention model, which can capture the global temporal dependency of one relation and entire timestamp. Here g is a simple two-layer MLP plus sigmoid function.

4.3 Orthogonal Regularization for Core Tensor

There have been many research results on orthogonal regularization of the matrix, such as (Miyato et al., 2018). In summary, the orthogonal regularization allows the parameter matrix to be closer to the diagonal-dominant non-singular matrix. A non-singular matrix prevents abrupt truncation changes of the feature map during matrix multiplication. Furthermore, tucker decomposition can be viewed as matrix multiplication for factor matrix with the arbitrary slice of core tensor. In other words, orthogonal regularization could be applied to core tensor multiplication.

Inspired by BigGAN (Brock et al., 2019), we heuristically propose two variants of core tensor orthogonal regularization Φ_1 and Φ_2. The baseline is the simple norm regularization Φ_0 used in TuckERTNT.

$$\Phi_0(\theta) = ||\boldsymbol{\mathcal{W}}||_4^4$$
$$\Phi_1(\theta) = ||\boldsymbol{\mathcal{W}} \odot (\mathbf{1} - \boldsymbol{\mathcal{I}})||_4^4$$
$$\Phi_2(\theta) = ||\boldsymbol{\mathcal{W}} \odot (\mathbf{1} - Proj(\boldsymbol{\mathcal{I}}))||_4^4$$

$\mathbf{1}, \boldsymbol{\mathcal{I}}, Proj(\boldsymbol{\mathcal{I}})$ is tensors with the same shape as $\boldsymbol{\mathcal{W}}$. All the elements of $\mathbf{1}$ are 1. The superdiagonal elements of $\boldsymbol{\mathcal{I}}$ are 1, and the other elements are 0. The diagonal elements of the arbitrary slice matrix of $Proj(\boldsymbol{\mathcal{I}})$ are 1, and the other elements are 0. The tucker decomposition degenerates to the CP decomposition when the super-diagonal elements of the core tensor are 1, and the rest are 0. Φ_1 regularization can restrict the result of the tucker decomposition to the neighborhood of the weighted CP decomposition. (the super-diagonal elements of the core tensor are weights, and the rest elements indicate a slight difference from weighted CP decomposition.

4.4 Other Regularization

Researchers have investigated many different kinds of embedding regularization to alleviate the overfitting problem. TNTComplex uses the third power of Nuclear-3 norm twice for temporal or non-temporal quadruplets. TIMEPLEX (Jain et al., 2020) use sampled weighted L2 regularization to avoid the overfitting problem. In our model, we use embedding regularization as ChronoR (Sadeghian et al., 2021) does, using the fourth power of $L4$ norm as embedding regularization.

$$\Omega_4(\theta) = ||\boldsymbol{h}||_4^4 + ||\boldsymbol{t}||_4^4 + ||\boldsymbol{r}_1||_4^4 + |||[\boldsymbol{r}_2|FA(\boldsymbol{\tau})]||_4^4$$

Because of the real-world time continuity, it is natural to guarantee adjacent timestamp embeddings or repeat timestamp embeddings closer in the embedding space. TuckERTNT (Shao et al., 2022)

proposes several temporal regularization to smooth the timestamp embedding. Notice that YAGO15K has many quadruplets without any timestamp, in which we artificially add a unique timestamp. This unique timestamp is excluded when computing temporal regularization terms. To be consistent with the embedding regularization, the adjacent timestamp differential regularization is as follow.

$$\Lambda_4(\theta) = \frac{1}{|\mathcal{T}|-1}\sum_{i=1}^{|\mathcal{T}|-1} ||\boldsymbol{\tau}_{i+1} - \boldsymbol{\tau}_i||_4^4$$

4.5 Loss Function

For each training data, we use instantaneous multi-class loss.

$$\mathcal{L}(\mathcal{X}_{ijkl}) = -\mathcal{X}_{ijkl} + log(\sum_{k'} exp(\mathcal{X}_{ijk'l}))$$

Considering instantaneous multi-class loss and the above three regularization term jointly, we train our model by minimizing the following loss function.

$$\mathcal{L}(\mathcal{X};\theta) = \frac{1}{|S|}\sum_{(i,j,k,l)\in S} [\mathcal{L}(\mathcal{X};\theta) + \lambda\Phi_n(\mathcal{X};\theta) + \lambda_1\Omega_4(\mathcal{X};\theta) + \lambda_2\Lambda_4(\mathcal{X};\theta))] \quad n = 1,2,3$$

where λ_1, λ_2 and λ is importance hyperparameter for tuning.

5 Experiments

5.1 Datasets and Evaluation Metrics

We choose three of the most commonly-used datasets to evaluate our model, including ICEWS14, GDELT, and YAGO15K. The detailed statistics of each dataset are shown in Table 1.

ICEWS14 is extracted from the Integrated Crisis Early Warning System (Lautenschlager et al., 2015) repository, which contains political events with daily timestamp points. This dataset, for the most part, is time-sensitive and accurate in descriptions.

Table 1: The statistics of the benchmark datasets.

	ICEWS14	GDELT	YAGO15K
#Entity	7128	500	15403
#Relation	230	20	34
#Timestamp	365	366	198
#Facts	90730	3419607	138056
Timespan	2014	2015-2016	1513-2017
Granularity	Daily	Daily	Annually
Type	Point	Point	Interval

GDELT is extracted from the Global Database of Events, Language, and Tone (Leetaru and Schrodt, 2013), which covers news data from 1979 to the present by automatically crawling. GDELT is a complicated dataset because of its abstract entity, such as government and organization.

YAGO15K (Friedland and Lim, 2018) augmented events of FB15K (Bordes et al., 2013) with time interval. YAGO15K is worst-perform in TKGC tasks because it requires the model to handle both temporal and non-temporal knowledge.

We follow the standard evaluation set in previous work, and report two standard metrics, Hit@k ($k \in \{1,3,10\}$) and filtered Mean Reciprocal Rank (MRR). They can evaluate the rank of the correct entity in the filtered candidate set. Hit@k reflects the percentage of the query whose correct tail entities are ranked within the top k candidates. Mean Reciprocal Rank, which computes the average of the reciprocal of mean rank, reflects the correct fact rank of the model. We follow the time-aware filtering (Han et al., 2021a), which means entities that cause ambiguity are removed from the candidate list for a query. We using reciprocal setting to add $(t_k, r_j^{-1}, h_i, \tau_l)$ into train set for each quadruplet (h_i, r_j, t_k, τ_l). The detailed hyperparameters of our model are shown in Appendix B.

5.2 Main Results

Table 2 shows the main temporal knowledge graph completion results. The results of other models come from the original paper. We use the bold number to indicate the existing best results. Our model slightly outperforms or ties with previous SOTA results on several metrics of ICEWS14 and YAGO3-10. On GDELT, our model achieves significant improvement results on all metrics. Compared with TuckERTNT, our model TuckER-FA has $d[(1-\rho)|\mathcal{T}| + (2+\rho)|\mathcal{R}|]$ fewer parameters and 1.1% higher MRR performance when using the same embedding dimensionality. Compared with 2500+ dimensions of entity and relation of TNTComplex and ChronoR, our model gets better results using only 400 dimensions of entity and relation.

Increasing the number of model parameters substantially improves MRR performance on the GDELT, but the improvement on the other two

Table 2: The evaluation results on ICEWS14, GDELT, and YAGO15K. For the other works, we report the best results reported in their original paper.

Model	ICEWS14				GDELT				YAGO15K			
Metric	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE	28.0	9.4	-	63.7	15.5	6.0	17.8	33.5	29.6	22.8	-	46.8
SimplE	45.8	34.1	51.6	68.7	20.6	12.4	22.0	36.6	-	-	-	-
ComplEx	47.0	35.0	54.0	71.0	21.3	13.3	22.5	36.6	36.0	29.0	36.0	45.0
TTransE	25.5	7.4	-	60.1	11.5	0.0	16.0	31.8	32.1	23.0	-	51.0
TA-DistMult	47.7	36.3	-	68.6	20.6	12.4	21.9	36.5	29.1	21.6	-	47.6
DE-SimplE	52.6	41.8	59.2	72.5	23.0	14.1	24.8	40.3	-	-	-	-
TeMP	60.7	54.5	67.3	77.4	-	-	-	-	27.5	19.1	29.7	43.7
TNTComplex	62.0	52.0	66.0	76.0	22.4	14.4	23.9	38.1	36.0	28.4	37.0	53.7
ChronoR	62.5	**54.7**	66.9	77.3	-	-	-	-	**36.6**	**29.1**	37.9	53.8
BoxTE	61.5	53.2	66.7	76.4	35.2	26.9	37.7	51.1	-	-	-	-
TuckERTNT	62.5	54.4	67.3	77.3	44.8	35.2	49.2	63.0	-	-	-	-
TuckER-FA	**62.7**	54.4	**67.7**	**78.0**	**48.6**	**39.3**	**53.2**	**66.0**	36.5	28.2	**39.2**	**54.3**

Table 3: Ablation study of FA and OR TuckER-FA.

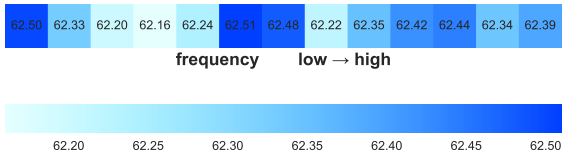
MRR	ICEWS14	GDELT	YAGO15K
ALL	62.7	48.6	36.5
w/o FA	62.0(-0.7)	47.3(-1.3)	34.8 (-1.7)
w/o OR	60.6(-2.1)	46.1(-2.5)	33.4 (-3.1)
w/o Both	58.4(-4.3)	44.9(-3.7)	32.6 (-3.9)

models is slight. Compared with ICEWS14, the GDELT dataset has nearly 38 times the number of facts, and fewer entities/relations, within the same time span. The corresponding graph of GDELT is spatially denser, with many more recurring facts. As a result, GDELT greatly enhances the global temporal dependency of relations, which is exactly our FA model focus on. Complex global temporal dependency explains TuckER-FA outstanding advantage on GDELT compared with its counterparts.

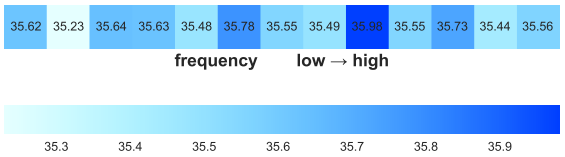
5.3 Ablation Study

Table 3 shows the ablation study of the Frequency Attention(FA) model and core tensor Orthogonal Regularization(OR) model. It can be observed that both models are valuable when working individually, and the combination of them performs even better. We can point out that the OR model is more effective than the FA model.

The single FA model increases the accuracy significantly in ICEWS14 compared with the vanilla model, which is probably because this dataset has the shortest time span and the minimal data. The improvement of the single OR model is slight in YAGO15K, and it may be attributed to the presence of facts without timestamps.



(a) ICEWS14



(b) YAGO15K

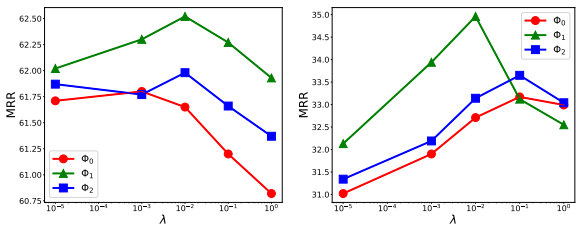
Figure 4: The MRR performance using different single frequency in the FA model. The darker colors indicate better results. Knowledge graph has inherent frequencies based on its own data distribution.

5.4 Frequency of Temporal Knowledge Graph

From the result of FcaNet (Qin et al., 2021) in the Appendix E, we can notice frequency attention model with a single direct-current component always reach the best result in the Image Classi-

Table 4: MRR Results for different test set divisions on ICEWS14 and YAGO15k.

Dataset	Total	DC	HF
ICEWS14	62.5	61.3	64.2
YAGO15k	36.5	35.1	39.4



(a) ICEWS14 (b) YAGO15K

Figure 5: The MRR performance of norm regularization and two variants of orthogonal regularization. The Φ_1 regularization without the superdiagonal elements of the core tensor performs best.

fication task. The low-frequency component is more critical than the high-frequency component because the low-frequency signal of the images provides information such as shape or size, while the high-frequency signal provides information such as edge details. However, this phenomenon does not occur in the TKGC task. Usually, it is difficult to have many repeated details in a naturally captured image. In contrast, many facts, such as the toy example, continuously change over time in TKGC. These facts determine that there exist some inherent frequencies in timestamp embedding.

Figure 4 shows the results of the frequency attention model with different single DCT bases. We use a combination of frequencies from the top six results to achieve our best results. The difference between the frequencies was insignificant, meaning the global temporal dependency spread over all frequencies. In detail, the direct-current component is the second-high result in ICEWS14 but the sixth-high in YAGO15K. The reason may be that the facts without a timestamp in YAGO15K disturb the estimation of the intrinsic frequency. The above results show that finding a suitable frequency feature with FA is helpful for the TKGC task.

To study why TuckER-FA performs differently on different datasets compared to other baselines, we count the number of the query $(h, r, ?, ?)$ occurrences over the entire timestamp. The average number of occurrences per query in GDELT is 284.7, and queries that occur more than once account for 98.8% of the training set. The statistics are 7.33, 40.8% in ICEWS14, and 4.51, 54.9% in YAGO15k, respectively. Thus, TuckER-FA gains better results on GDELT because the FA module is good at capturing the global temporal dependencies between one relation over the entire timestamp.

Table 4 show the MRR Results for two test set divisions on ICEWS14 and YAGO15k. We split the test set into two subsets, one consisting of queries that occur only once (direct-currency component, DC) and the other consisting of queries that appear multiple times (high-frequency component, HF). In conclusion, our model works much better on the high-frequency component test set. Because the GDELT dataset is almost exclusively high-frequency components, it boosts most significantly.

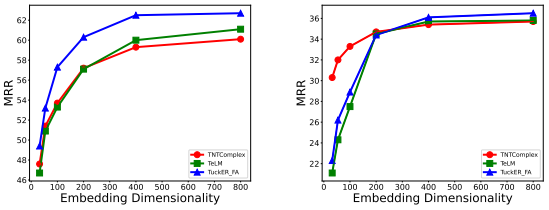
5.5 Orthogonal Regularization

Figure 5 shows the detailed comparison of three regularizations for the core tensor. In this experiment, we fix the λ_1 and λ_2 and only use the direct-current component for the frequency attention model. The Φ_2 regularization without the diagonal elements of the arbitrary slice matrix of the core tensor performs slightly better than the Φ_0 norm regularization. The Φ_1 regularization without the superdiagonal elements of the core tensor performs best.

The Φ_1 regularization increase MRR by 0.7 points on ICEWS14 and 1.8 points on YAGO15K. Top Performing Φ_1 can increase MRR to 2.1 points on ICEWS14 and 3.1 points on YAGO15K. We can point out that the relaxation of the core tensor constraint is effective.

5.6 Effect of Parameter Complexity

To compare model results more fairly, we add two experiments controlling the number of parameters or entity dimensionality between TNTcomplEx, TeLM, and TuckER-FA. A complete parameter comparison between baseline models and TuckER-FA is in Appendix A.



(a) ICEWS14 (b) YAGO15K

Figure 6: The MRR performance of different models with different embedding dimensionality. With the same embedding dimensionality, our model consistently outperforms TeLM.

Figure 6 shows our TuckER-FA consistently outperforms the baseline TNTComplEx and TeLM with the same dimensionality on ICEWS14. TNTComplEx uses a specialized matrix to represent non-timestamp embedding, which allows it to perform best in low dimensionality on YAGO15K. Our TuckER-FA can overtake in high dimensionality. The upper limit of TeLM is not as high as TuckER-FA. In summary, our model has the highest performance ceiling for TKGC.

Table 5 shows the comparison with the same parameters and 50 epochs training on ICEWS14. We limit the size of the learnable parameters to approximately 67M, which means 2110 embedding dimensions for TeLM, 4000 for TNTComplEX, and 400 for our TuckER-FA. We can point out that our TuckER-FA model achieves a significant improvement in MRR and Hit@3 and a weak improvement in the other two metrics.

Table 5: Results with approximately 67M parameters of TNTComplEx, TeLM, and TuckER-FA on ICEWS14.

Model	MRR	Hit@1	Hit@3	Hit@10
TuckER-FA	62.5	54.3	67.5	77.2
TNTComplEx	61.2	52.4	66.3	77.4
TeLM	62.1	54.2	66.7	77.0

6 Conclusion

In our work, we propose a DCT-based Frequency Attention model and two variants of Orthogonal Regularization for the core tensor of tucker decomposition. The FA model considers the global temporal dependency between one relation and the entire timestamp. Each KG has its unique inherent frequency. The OR term relaxes the constraint on the superdiagonal of the core tensor and improves the performance of tucker decomposition. TuckER-FA achieves SOTA results on three standard datasets of temporal knowledge graph completion task. There might be further discussions on an efficient frequency selection strategy or a theoretical assumption for tensor regularization.

7 Limitation

Although our method has been shown effective, it has two limitations that may be improved in the future. First, the FA model has advantages in computation but relies on an effective frequency selection strategy, which is difficult to design. We just simply select some manual frequencies for different datasets by experience. The more effective frequency selection strategy needs further exploration. Second, there is no theoretical guarantee that the orthogonal regularization can generalize to a 3-order tensor. Our OR terms are only formally consistent with matrix orthogonal regularization, which has been empirically shown effective.

Acknowledgements

This work is supported partly by the National Key R&D Program of China under Grant 2021ZD0110700, partly by the Fundamental Research Funds for the Central Universities, and partly by the State Key Laboratory of Software Development Environment.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ivana Balazevic, Carl Allen, and Timothy M. 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, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5184–5193. Association for Computational Linguistics.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544, Seattle, USA. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Pratim Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *EMNLP*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818. AAAI Press.

Shmuel Friedland and Lek-Heng Lim. 2018. Nuclear norm of higher-order tensors. *Math. Comput.*, 87(311):1255–1281.

Rishab Goel, Seyed Mehran Kazemi, Marcus A. Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *AAAI*.

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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8352–8364. Association for Computational Linguistics.

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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8104–8118. Association for Computational Linguistics.

Prachi Jain, Sushant Rathi, Mausam, and Soumen Chakrabarti. 2020. Temporal knowledge base completion: New algorithms and evaluation protocols. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3733–3747. Association for Computational Linguistics.

Jaehun Jung, Jinhong Jung, and U. Kang. 2021. Learning to walk across time for interpretable temporal knowledge graph completion. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jennifer Lautenschlager, Steve Shellman, and Michael Ward. 2015. ICEWS Event Aggregations.

Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location and tone, 1979-2012.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutional representation learning. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Siyuan Liao, Shangsong Liang, Zaiqiao Meng, and Qiang Zhang. 2021. Learning dynamic embeddings for temporal knowledge graphs. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3243–3253. Association for Computational Linguistics.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A knowledge base from multilingual wikipedias. In *Seventh Biennial Conference on Innovative Data Systems Research, CIDR 2015, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*. www.cidrdb.org.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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 Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 809–816. Omnipress.

Zequn Qin, Pengyi Zhang, Fei Wu, and Xi Li. 2021. Fcanet: Frequency channel attention networks. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 763–772. IEEE.

Ali Sadeghian, Mohammadreza Armandpour, Anthony Colas, and Daisy Zhe Wang. 2021. Chronor: Rotation based temporal knowledge graph embedding. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 6471–6479. AAAI Press.

Pengpeng Shao, Dawei Zhang, Guohua Yang, Jianhua Tao, Feihu Che, and Tong Liu. 2022. Tucker decomposition-based temporal knowledge graph completion. *Knowl. Based Syst.*, 238:107841.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Rakshit S. Trivedi, Hanjun Dai, Yichen Wang, and Le Song. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *ICML*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 417–426.

Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2016. From one point to a manifold: Knowledge graph embedding for precise link prediction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1315–1321. IJCAI/AAAI Press.

Chenjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2020. Temporal knowledge graph completion based on time series gaussian embedding. In *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part I*, volume 12506 of *Lecture Notes in Computer Science*, pages 654–671. Springer.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zhanqiu Zhang, Jianyu Cai, and Jie Wang. 2020. Duality-induced regularizer for tensor factorization based knowledge graph completion. *CoRR*, abs/2011.05816.

Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhan. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *AAAI*.

A Parameter Complexity

Table 6: Parameter complexity of different models. Here d denotes the embedding dimensionality.

DE-SimplE	$d((1+2\rho)\|\mathcal{E}\| + \|\mathcal{R}\|)$
TNTComplEx	$2d(\|\mathcal{E}\| + \|\mathcal{T}\| + 4\|\mathcal{R}\|)$
TeLM	$4d(\|\mathcal{E}\| + \|\mathcal{T}\| + \|\mathcal{R}\| + 1)$
TuckERTNT	$d(\|\mathcal{E}\| + \|\mathcal{T}\| + 4\|\mathcal{R}\|) + d^3$
TuckER-FA	$d(\|\mathcal{E}\| + \rho\|\mathcal{T}\| + (2-\rho)\|\mathcal{R}\|) + d^3$

When d is smaller the number of entity/relation/timestamp collection, tucker decomposition has considerable parameter advantage. Compared with TuckERTNT of the same type, our TuckER-FA gets better results with parameter advantage in modeling relation embedding.

B Hyperparameters

Table 7: Obtained best hyperparameters.

	ICEWS14	GDELT	YAGO15K
ρ	0.75	0.90	0.75
λ_1	1e-2	1e-1	1e-4
λ_2	1e0	1e0	1e-2
λ	1e-2	1e-4	1e-3

We implement our model based on two previous models, TuckER and TNTcomplex. Other baseline models mentioned above have yet to provide publicly available code. During the pre-processing data phase, we convert all time intervals into two different time points and consider them independent. The time intervals in YAGO15K look like "OccursUntil/OccurSince 1994". The time interval is split into two parts, "OccursSince" or "OccursUntil" merged into a relation, and the time point transforms the timestamp. Note that quadruplet without timestamps in YAGO15K also own a unique timestamp.

Although the dimensionality of entity and relation can be different, we use the same dimensionality in our experiments. For general learning settings, we set the dimensionality of entity and relation to 800, batch size to 1000, learning rate to 0.1, and dropout probability to 0.3. Each embedding initializes from 0.01 times the Standard Gaussian distribution. Moreover, the learnable parameters of the core tensor initialize from the uniform distribution from -1 to 1. The frequency attention model uses a single frequency component from f_0 to f_{12} input. We choose ratio ρ form {0.45, 0.6, 0.75, 0.9}, embedding regularization balance term λ_1 from {1e-1, 1e-2, 1e-3, 1e-4}, temporal regularization balance term λ_2 from {1e0, 1e-1, 1e-2, 1e-3}, and core tensor regularization balance term λ from {1e-2, 1e-3, 1e-4, 1e-5}. Note that the timestamp embedding dimensionality should be divisible by n, the number of the selected frequency components. We repeats the experiment three times and reported the average results.

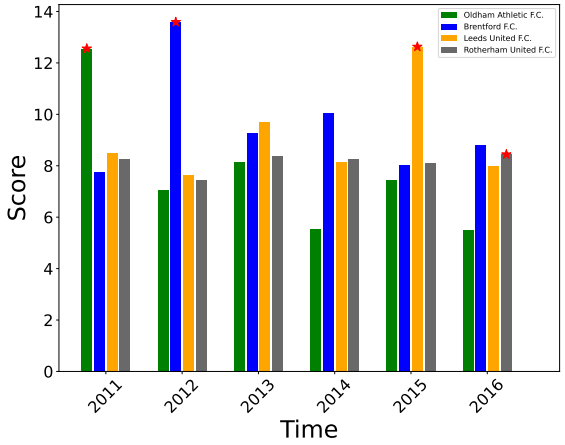
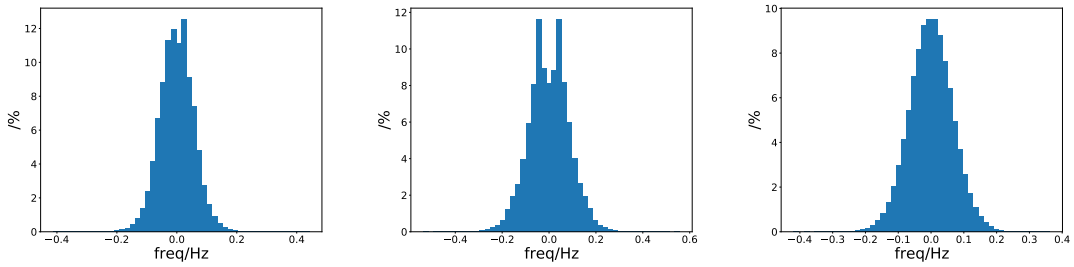


Figure 7: Scores for a set of facts (Tom Adeyemi, playsFor, {Oldham Athletic F.C.|Brentford F.C.|Leeds United F.C.|Rotherham United F.C.}, [2011-2017]) sampled from YAGO15K. The red star indicates that the tail entity is the ground truth at the corresponding timestamps.

C Visualization of Toy Example

To illustrate whether our frequency attention model captures the temporal dependencies between a relation and the entire timestamp, we visualize the scores of a selected set of facts. Figure 7 show the scores change of the toy example. The factual tail entity can consistently score high by TuckER-FA. The gray entity's score changes very little because its ground truth belongs to the test set. The ground truth of the other three color entities belongs to the train set. The facts in the same category (Tom Adeyemi, playsFor,?) change quickly, which reflects a high intrinsic frequency of global temporal dependency. The scores of ground truth are very high on timestamps where facts exist, while all entities achieve a low score on timestamps where facts do not exist. Our model can capture the fast-changing temporal dependency of facts in the same category.



(a) Year (b) Month (c) Day

Figure 8: Histogram of frequency parameters in DE-SimplE. These learnable parameters are concentrated around 0, which means that only low-frequency information about entities over time is captured.

D The Frequency of DE-SimplE

DE-SimplE divides the timestamp 2014-01-31 into three numbers representing the year, month, and day. Then, these time numbers are fed into the sine function along with the frequency and phase parameters. Figure 8 shows the histogram of learned frequency parameters of DE-SimplE on the ICEWS14 dataset. These learned parameters concentrate around 0, meaning only low-frequency information about entities over time is captured.

In previous work (Xu et al., 2020), the evolution process is divided into four different components: static component, periodicity component, trend component, and randomness component. In our opinion, the random component focuses on model robustness, and the static component focuses on the static entity or relation rather than the timestamp. The periodicity and trend components mean the temporal dependency of relations and timestamps, which can be captured by a periodic function such as cosine. If the period of the function is greater than the entire time span, then the cosine function captures the trend component. Similarly, if the period of the function is less than the entire time span, the cosine function captures the periodicity component.

E Frequency Attention for Image

The image uses 2-dimensional DCT as Figure 9, while the KG uses only one-dimensional DCT. Frequency is an essential characteristic of DCT and indicates how many repetitive structures there are in the data. The main body of the image is composed of low-frequency features, while the TKG body has high-frequency features due to repeated quadruplets.

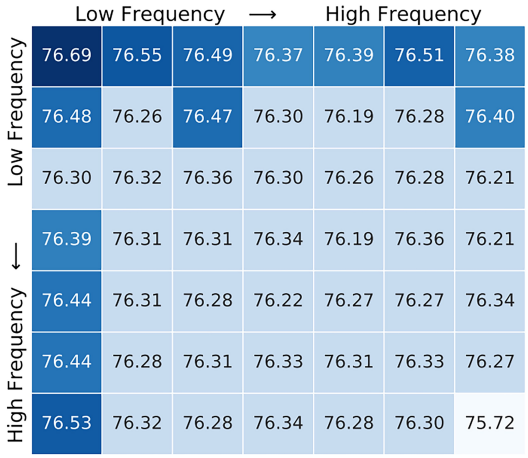


Figure 9: Top-1 accuracies of FcaNet (Qin et al., 2021) in Image Classification Results. The low-pass filtering of the spectrum has strong applicability to images. The DC component performs best.