# Time-dependent Entity Embedding is not All You Need: A Re-evaluation of Temporal Knowledge Graph Completion Models under a Unified Framework

**Zhen Han**[*1,2], **Gengyuan Zhang**[*1], **Yunpu Ma**[†1], **Volker Tresp**[†1,2]
[1]Institute of Informatics, LMU Munich [2] Corporate Technology, Siemens AG
zhen.han@campus.lmu.de, gengyuanmax@gmail.com
cognitive.yunpu@gmail.com, volker.tresp@siemens.com

## Abstract

Various temporal knowledge graph (KG) completion models have been proposed in the recent literature. The models usually contain two parts, a temporal embedding layer and a score function derived from existing static KG modeling approaches. Since the approaches differ along several dimensions, including different score functions and training strategies, the individual contributions of different temporal embedding techniques to model performance are not always clear. In this work, we systematically study six temporal embedding approaches and empirically quantify their performance across a wide range of configurations with about 4000 experiments and 19000 GPU hours. We classify the temporal embeddings into two classes: (1) *timestamp embeddings* and (2) *time-dependent entity embeddings*. Despite the common belief that the latter is more expressive, an extensive experimental study shows that timestamp embeddings can achieve on-par or even better performance with significantly fewer parameters. Moreover, we find that when trained appropriately, the relative performance differences between various temporal embeddings often shrink and sometimes even reverse when compared to prior results. For example, TTransE (Leblay and Chekol, 2018), one of the first temporal KG models, can outperform more recent architectures on ICEWS datasets. To foster further research, we provide the first unified open-source framework for temporal KG completion models with full composability, where temporal embeddings, score functions, loss functions, regularizers, and the explicit modeling of reciprocal relations can be combined arbitrarily.

## 1 Introduction

The Knowledge Graph (KG), a graph-structured knowledge base, has gained increasing interest as a promising way to store factual knowledge. KGs represent facts in the form of triples $(s, r, o)$, e.g., (*Bob*, *livesIn*, *New York*), in which $s$ (subject) and $o$ (object) denote nodes (entities) and $r$ denotes the edge type (relation) between $s$ and $o$. Knowledge graphs are commonly static and store facts in their current state. In reality, however, the relations between entities often change over time. For example, if Bob moves to California, the triple of (*Bob*, *livesIn*, *New York*) will be invalid. To this end, temporal knowledge graphs (tKGs) have been introduced to capture temporal aspects of facts in addition to their multi-relational nature. A tKG represents a temporal fact as a quadruple $(s, r, o, t)$ by extending a static triple with time $t$, describing that this fact is valid at time $t$. Figure 2 in the appendix depicts an exemplary temporal KG. To address the inherent incompleteness of temporal KGs, Tresp et al. (2015) proposed the first tKG model. Afterwards, a line of work emerged that extends static KG completion models by adding temporal embeddings, e.g., TTransE (Leblay and Chekol, 2018), TA-TransE (García-Durán et al., 2018), DE-SimplE (Goel et al., 2019), TNTComplEx (Lacroix et al., 2020), ConT (Ma et al., 2018), and many more. The models generally consist of two parts, a temporal embedding layer to capture the evolving features of tKGs and a score function to examine the plausibility of a given quadruple.

Temporal embeddings are crucial in temporal KG completion models for storing the evolving knowledge; without them, the temporal aspect cannot be captured. The PEs can be generally categorized into three classes: (1) ***t**imestamp **e**mbeddings (TEs)*: the models learn an embedding for each discrete timestamp in the same vector space as entities and relations (Tresp et al., 2017; Leblay and Chekol, 2018; Dasgupta et al., 2018; Lacroix et al., 2020). (2) ***t**ime-dependent **e**ntity **e**mbeddings (TEEs)*: the models define entity embedding as a function that takes an entity and a timestamp as

[*] Equal contribution.
[†] Corresponding author.



*Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8104–8118
November 7–11, 2021. 

input and generates a time-dependent representation for the entity at that time (Goel et al., 2019; Xu et al., 2019; Han et al., 2020a). (3) *deep representation learning (DTRs)*: the models incorporate temporal information into advanced deep learning models, e.g., Recurrent Neural Network and Graph Neural Network, to learn time-aware representations of entities and relations (García-Durán et al., 2018). In many cases, the introduction of new temporal embedding approaches went along with new score functions and new training methods (regularization, the explicit modeling of reciprocal relations, etc.). Ablation studies were provided, but not investigated thoroughly. Besides, some temporal embedding papers introduced new datasets. They commonly tune model architecture and hyperparameters of old temporal embedding approaches on new datasets using grid search on a small grid involving hand-crafted parameter ranges or settings known to work well from prior studies. A grid suitable for one dataset may be suboptimal for another, however. It is often difficult to attribute the incremental improvements in performance reported with each new state-of-the-art (SOTA) model to the proposed temporal embeddings or other components.

In this work, we investigate the significance of previously reported temporal embeddings with several thousands of experiments and 19000 GPU hours. First, we aim to study which temporal embedding approach can generally outperform other temporal embedding approaches regardless of different score functions and different datasets. We choose one representative from bilinear score functions, i.e., SimplE (Kazemi and Poole, 2018), and one from translation-based score functions, i.e., TransE (Bordes et al., 2013). Then we benchmark six temporal embedding approaches on two subsets of ICEWS (Boschee et al., 2015) and a subset of GDELT(Leetaru and Schrodt, 2013) with the two representative score functions through an extensive set of experiments. Second, we performed an extensive benchmark study on well-known temporal KG completion models using popular model architectures and training strategies in a unified experimental setup. Following the work (Ruffinelli et al., 2020), we considered many training strategies as well as a large hyperparameter space, and we performed model selection using a quasi-random search followed by Bayesian optimization, which has been shown to be able to find good model configurations with relatively low effort.

Regarding the first aim, we surprisingly find that the TE proposed by Leblay and Chekol (2018) outperforms other temporal embedding approaches on the ICEWS subsets and achieves on-par results on GDELT. Leblay and Chekol (2018) represent timestamps in the same vector space as entities and relations and learn embeddings for each discrete timestamp. While achieving better results, the TE models only require about half of the model parameters as much of TEEs. However, the common belief is that the TEEs are more expressive and can better capture the evolving knowledge. Recall that models with TEEs learn an embedding function for each entity that takes time as input and provides an entity representation as output. In particular, it has been proven that TEEs are fully expressive for tKG completion in combination with certain score functions (Goel et al., 2019), and thus, they should perform better than TEs, which is in contrast to our findings. We argue that the sparsity of temporal KG data may cause the undesirable empirical performance of TEEs. Every entity has the same dimensionality of time-dependent embeddings, but the majority of entities are only involved in a small number of quadruples. As a result, the TEEs may suffer from the overfitting problem. To verify our assumption, we learn a **u**nique **t**emporal **e**mbedding function for all **e**ntities instead of learning entity-specific embedding functions. We refer to it as UTEE. Empirical study shows that the UTEE achieves similar or even better results than all other TEEs variants, emphasizing the overfitting problem of TEEs.

Besides, we empirically find that the performance of a fine-tuned baseline can by far exceed the performance observed in all previous studies. For example, T-TransE (Leblay and Chekol, 2018), one of the first temporal KG completion models, achieves superior performance metric in our study that is more than **doubled** to that reported in recent papers (García-Durán et al., 2018; Goel et al., 2019; Lacroix et al., 2020; Xu et al., 2019). Thus, it is competitive to or even outperforms current SOTA models such as DE-SimplE (Goel et al., 2019) and TComplEx (Lacroix et al., 2020). This suggests that training strategies significantly affect the performance of temporal KG models and are responsible for a substantial fraction of the progress made in recent years. Thus, to fairly compare the effectiveness of different temporal KG models, it is nec-

essary to evaluate them on a unified framework. To this end, our study realizes the first fair benchmarking by investigating the interplay between temporal KG interaction models, loss functions, regularization methods, the use of reciprocal relations, and other training techniques in a unified open-source framework[1]. To ensure the composability of the framework, the temporal embedding layer, score functions, and various training strategies are implemented as independent submodules. Thus, one can easily assess the individual benefit of a novel temporal embedding approach via our framework. Additionally, we perform an extensive experimental study in which well-known temporal KG models are fine-tuned by popular training strategies and a wide range of hyperparameter settings. The reported results can be directly used for comparison in future work.

## 2 Preliminaries and Related Work

### 2.1 Temporal Knowledge Graph Completion

Temporal knowledge graphs (tKGs) are multi-relational, directed graphs with labeled timestamped edges between entities. Let $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ represent a finite set of entities, relations, and timestamps, respectively. Each fact can be denoted by a quadruple $q = (e_s, r, e_o, t)$, representing a timestamped and labeled edge between a subject entity $e_s \in \mathcal{E}$ and an object entity $e_o \in \mathcal{E}$ regarding a relation $r \in \mathcal{R}$ at a timestamp $t \in \mathcal{T}$. Let $\mathcal{F}$ represents the set of all quadruples that are facts, i.e., real events in the world, the tKG completion (tKGC) is the problem of inferring $\mathcal{F}$ based on a set of observed facts $\mathcal{O}$, which is a subset of $\mathcal{F}$. Specifically, the task of tKGC is to predict either a missing subject entity $(?, r, e_o, t)$ given the other three components or a missing object entity $(e_s, r, ?, t)$.

Our study focuses solely on temporal knowledge graph embedding models for the completion task, which do not exploit temporal knowledge graph embedding models for the forecasting task (Trivedi et al., 2017; Han et al., 2020b, 2021).

### 2.2 Temporal KG Embedding Models

A tKG embedding (tKGE) model embeds each entity $e \in \mathcal{E}$ and relation $r \in \mathcal{R}$ in a vector space. To capture temporal aspects, each model either embeds discrete timestamps into a vector space or learns time-dependent representations for each entity. Besides, each model has a score function that takes the temporal information and the embeddings of the subject, relation, and object as the input and computes a score for each potential quadruple. The higher the quadruple score, the more plausible it is considered to be true by the model. Taking the object prediction as an example, we consider all entities in $\mathcal{E}$ and learn a score function $\phi(e_s, r, e_o, t) = f(\mathbf{e}_s(t), \mathbf{r}, \mathbf{e}_o(t))$, for models with TEEs and $\phi(e_s, r, e_o, t) = f(\mathbf{e}_s, \mathbf{r}, \mathbf{e}_o, \mathbf{t})$ for models with TEs. The bold symbols denote the embeddings of the corresponding entities, relation, and time.

#### 2.2.1 Temporal Embeddings

tKGE models differ in their temporal embeddings and score functions. Temporal embedding approaches come in three categories: timestamp embeddings (TEs), where the models learn a representation for each discrete timestamp; time-dependent entity embeddings (TEEs), where an entity embedding function takes time and an entity as inputs and provides a hidden representation as output; and deep temporal representations (DTRs), where the models incorporate time information into deep learning frameworks.

The best known TE is the vanilla TE (abbreviated to T by its authors) proposed by Leblay and Chekol (2018) where each timestamp is mapped in the same vector space as entities and relations. Later, Lacroix et al. (2020) introduced a new regularization scheme to smooth the representation of neighboring timestamps. Another well-known TE is HyTE (Dasgupta et al., 2018), which associates each timestamp with a corresponding hyperplane and projects the embeddings of entities and relations onto timestamp-specific hyperplanes to incorporate temporal information in entity embeddings:

$$\mathbf{e}_i(t) = \mathbf{e}_i \perp \boldsymbol{\omega}_t = \mathbf{e}_i - (\boldsymbol{\omega}_t^T \mathbf{e}_i)\boldsymbol{\omega}_t.$$

$\mathbf{e}_i$ represents the global embedding of entity $e_i$, $\perp$ represents the projection operator, and $\boldsymbol{\omega}_t$ represents the normal vector of the hyperplane associated with timestamp $t$.

A well-known variants of TEEs is the diachronic entity embeddings (DE) proposed by Goel et al. (2019) that defines the temporal embeddings of entity $e_i$ at timestamp $t$ as

$$\mathbf{e}_i^{DE}(t)[n] = \begin{cases} \mathbf{a}_{e_i}[n] & \text{if } 1 \leq n \leq \gamma d, \\ \mathbf{a}_{e_i}[n] \sin(\boldsymbol{\omega}_{e_i}[n]t + \mathbf{b}_{e_i}[n]) & \text{else.} \end{cases} \quad (1)$$

[1] https://github.com/TemporalKGTeam/A_Unified_Framework_of_Temporal_Knowledge_Graph_Models

$\mathbf{e}_i^{DE}(t)[n]$ denotes the $n^{th}$ element of the embeddings of entity $e_i$ at time $t$. $\mathbf{a}_{e_i}, \boldsymbol{\omega}_{e_i}, \mathbf{b}_{e_i}$ are entity-specific vectors with learnable parameters. The first $\gamma d$ elements of the vector in Equation 1 capture static features, and the other $(1-\gamma)d$ elements capture temporal features. ATiSE (Xu et al., 2019) is another popular TEE that adds time information into entity/relation representations by using additive time series decomposition, where the entity representation is defined as

$$\mathbf{e}_i^{ATiSE}(t) = \mathbf{e}_i + \alpha_{e_i}\mathbf{w}_{e_i}t + \boldsymbol{\beta}_{e_i}\sin(2\pi\boldsymbol{\omega}_{e_i}t) + \mathcal{N}(0, \boldsymbol{\Sigma}_{e_i}). \quad (2)$$

The term $\mathbf{e}_i + \alpha_{e_i}\boldsymbol{\omega}_{e_i}t$ is the trend component where the coefficient denotes the evolutionary rates, and the vector $\omega_{e_i}$ represents the corresponding evolutionary direction. $\boldsymbol{\beta}_{e_i}\sin(2\pi\boldsymbol{\omega}_{e_i}t)$ is the corresponding seasonal component, and the Gaussian noise term $\mathcal{N}(0, \Sigma_{e_i})$ denotes the random component. In principle, other temporal embedding approaches can also be converted into a probabilistic approach by adding Gaussian noise. Thus, to fairly compare with other temporal embeddings and simplify our study, we do not take the noise term into account. The representation of relations in ATiSE is also time-dependent and defined similarly to the entity representation.

A representative of DTRs is the TA-approach (García-Durán et al., 2018) that utilizes recurrent neural networks to learn time-aware representations of relations. Specifically, the relation representation is obtained by $\mathbf{r}^{TA}(t) = LSTM(r, t)$, where the timestamp (date) $t$ is tokenized into digits (year, month, and day). The sequence of temporal tokens and the relation $r$ is used as input to the LSTM. In addition to the five PEs mentioned above, we propose a new TEE where we learn a **u**nique **t**emporal **e**mbedding function for all **e**ntities to investigate the overfitting problem of DE. We refer to it as UTEE, which is defined as follows:

$$\mathbf{e}_i^{UTEE}(t)[n] = \begin{cases} \mathbf{a}[n] & \text{if } 1 \leq n \leq \gamma d, \\ \mathbf{a}[n]\sin(\boldsymbol{\omega}[n]t + \mathbf{b}[n]) & \text{else.} \end{cases}$$

where the amplitude vector $\mathbf{a}$, frequency vector $\boldsymbol{\omega}$, and bias $\mathbf{b}$ are identical for all entities.

#### 2.2.2 Score Functions

A large number of score functions have been developed for the KG completion task. A class of these models is the translation-based approaches corresponding to variations of TransE (Bordes et al., 2013; Wang et al., 2014; Nguyen et al., 2016) that models relations as a translation of subject to object embeddings, i.e., $s^{TransE}(e_s, r, e_o) = -||\mathbf{e}_s + \mathbf{r} - \mathbf{e}_o||_2$. Another line of work is bilinear score functions (Nickel et al., 2011; Yang et al., 2014; Trouillon et al., 2016; Kazemi and Poole, 2018) that define product-based functions over embeddings, i.e., $s^{RESCAL}(e_s, r, e_o) = \mathbf{e}_s^T\mathbf{R}\mathbf{e}_o$, where relation matrix $\mathbf{R} \in \mathbb{R}^{d\times d}$ contain weights $r_{i,j}$ that capture the interaction between the $i$-th latent factor of $\mathbf{e}_s$ and the $j$-th latent factor of $\mathbf{e}_o$. Among the bilinear models, SimplE (Kazemi and Poole, 2018) a simple yet fully expressive model that represents each entity $e_i \in \mathcal{E}$ by two vectors $\mathbf{e}_{i,s}$ and $\mathbf{e}_{i,o}$. Depending on whether $e_i$ participates in a triple as the subject or object entity, either $\mathbf{e}_{i,s}$ or $\mathbf{e}_{i,o}$ is used. To address the independence of the two vectors for each entity, SimplE takes advantage of reciprocal relations and uses $\frac{1}{2}(\langle \mathbf{e}_{i,s}, \mathbf{r}, \mathbf{e}_{j,o}\rangle + \langle \mathbf{e}_{j,s}, \mathbf{r}^{-1}, \mathbf{e}_{i,o}\rangle)$ as the score of $(e_i, r, e_j)$, where $r^{-1}$ is the reciprocal relation of $r$.

In the rest of the paper, we examine the above six temporal embeddings in terms of the two representative score functions (TransE and SimplE) on two benchmark tKG datasets. We refer to a specific combination of temporal embedding approach and score function as an *interaction model*.

### 2.3 Reciprocal Relations

Lacroix et al. (2018) and Dettmers et al. (2018) introduced the use of reciprocal relation for training knowledge graph embeddings. For every quadruple $(e_s, r, e_o, t)$ in the dataset, we add $(e_o, r^{-1}, e_s, t)$, where $r^{-1}$ denotes the reciprocal relation of $r$. The idea of reciprocal relations is to use separate scoring functions for object prediction and subject prediction. Reciprocal relations can help translation-based approaches model symmetric patterns and help bilinear approaches model anti-symmetric and inverse patterns (Kazemi and Poole, 2018).

### 2.4 Related Work

Previous benchmarking studies (Kadlec et al., 2017; Akrami et al., 2020; Rossi et al., 2021) only focus on static knowledge graph models. For example, Ruffinelli et al. (2020) and Ali et al. (2020) realize a fair benchmarking by re-implementing static KGE models and performing an extensive empirical study with a massive search space. However, they do not take temporal knowledge graph models into account. To this end, we provide a unified framework that covers relevant tKGE mod-

els and investigate the influence of temporal embeddings on model performance as well as other components. To the best of our knowledge, this is the first benchmarking study for tKGE models.

## 3 Experimental Study

In this section, we first introduce the design of our unified framework that enables us to evaluate a large set of different combinations of interaction models, loss functions, regularization methods, the usage of of explicitly modeling reciprocal relations and position-aware entity embeddings. Then we split our experimental study into two parts. In the first part, we examine six temporal embedding methods combined with two representative score functions by performing an extensive set of experiments using advanced training strategies and a wide range of hyperparameter settings via the unified framework. In the second part, we re-evaluate various well-known tKG models from prior studies. We provide evidence that several old tKG models can obtain results competitive to or even better than the SOTA when configured carefully. We present the best configuration of each model and report its best performance on each benchmark that future research can directly use for comparison.

### 3.1 Composable Unified Framework

In the proposed framework, a tKGE model is considered as a composition of six modules that can flexibly be combined: a temporal embedding layer, a static embedding layer, a score function, a loss function, a regularization method, and the usage of reciprocal relations. In particular, the framework can automatically optimize the embedding method: the temporal embeddings can be either combined with entity embeddings or relation embeddings or both; there are different ways to combine static embeddings and temporal embeddings, i.e., addition, concatenation and element-wise multiplication. The framework supports six temporal embedding approaches as introduced in Section 2.2.1, seven score functions, i.e., TransE(Bordes et al., 2013), SimplE(Kazemi and Poole, 2018), DistMult(Yang et al., 2014), three loss functions (MR, CE, and BCE), four regularization methods (L1/L2/L3-norm , and dropout), and two initialization methods (Xavier uniform and Xavier normal). For interaction models with TEs, a smoothness regularization for timestamp embeddings is applied, enforcing neighboring timestamps to have close representations (Lacroix et al., 2020). Additionally, Kazemi and Poole (2018) distinguished an entity between as a head or as a tail entity and learns two embeddings for each entity, which we term *position-aware entity embedding* and extend to all interaction models. Position-aware entity embeddings can enhance the model's expressiveness. For example, it can help Distmult (Yang et al., 2014) to model anti-symmetric relations: without it, all relations are enforced to be symmetric since $\langle h, r, t, \tau \rangle$ and $\langle t, r, h, \tau \rangle$ share the same score regardless of properties of $r$.

### 3.2 Experimental Setup

**Datasets** Integrated Crisis Early Warning System (ICEWS) (Boschee et al., 2015) dataset has established itself in the research community as representative samples of tKGs and has been widely applied in recent tKG studies. The ICEWS dataset contains information about political events with specific time annotations, e.g. (Barack Obama, visit, India, 2010-11-06). We apply our model on two subsets of the ICEWS dataset: ICEWS14 contains events in 2014, and ICEWS11-14 corresponds to the facts between 2011 to 2014. Besides, we also used a subset of the Global Database of Events, Language, and Tone (GDELT) (Leetaru and Schrodt, 2013) dataset as a benchmark. To make the extensive configuration search feasible, we extracted a subset named GDELT-m10 consisting of factual events in October, 2015. The statistics and further details are provided in Appendix C.

**Hyperparameters** We used a large hyperparameter search space to ensure that suitable hyperparameters for each model can be covered. We consider seven embedding dimensions $\{64, 100, 128, 256, 512, 1024, 2048\}$. The learning rate can be randomly selected from $(0, 0.1]$. We use separate weights for regularization of embeddings of entities, relations, and timestamps. A detailed report of the search space is provided in Appendix B.

**Interaction models** In the first part, we evaluate six temporal embedding methods combined with two representative score functions. The formulas of these twelve interaction models are listed in Table 1. Additionally, we select DE-SimplE/TransE(Goel et al., 2019), TNTComplEx(Lacroix et al., 2020), ATiSE(Xu et al., 2019), TTransE(Leblay and Chekol, 2018), TA-TransE (García-Durán et al.,

Table 1: Formulas of a given quadruple $(e_i, r, e_j, t)$. $\mathbf{e}_{i,s}$ denotes the embedding of $e_i$ when the entity is the subject while $\mathbf{e}_{i,o}$ denotes the embedding of $e_i$ when the entity is the object. In comparison, $\mathbf{e}_i$ represents the shared embedding of entity $e_i$ for both subject and object. $\mathbf{t}$, $\mathbf{r}$ represent the embedding of timestamp $t$ and relation $r$, respectively. $\perp$ represents the projection operator. $\mathbf{e}_i(t)$ denotes the temporal embedding of $e_i$ at $t$.

| Temporal Embeddings | TransE | SimplE |
|---|---|---|
| T | $\|\|\mathbf{e}_i + \mathbf{r} + \mathbf{t} - \mathbf{e}_j\|\|$ | $\frac{1}{2}(\langle \mathbf{e}_{i,s}, \mathbf{r}, \mathbf{t}, \mathbf{e}_{j,o}\rangle + \langle \mathbf{e}_{j,s}, \mathbf{r}^{-1}, \mathbf{t}, \mathbf{e}_{i,o}\rangle)$ |
| DE | $\|\|\mathbf{e}_i^{DE}(t) + \mathbf{r} - \mathbf{e}_j^{DE}(t)\|\|$ | $\frac{1}{2}(\langle \mathbf{e}_{i,s}^{DE}(t), \mathbf{r}, \mathbf{e}_{j,o}^{DE}(t)\rangle + \langle \mathbf{e}_{j,s}^{DE}(t), \mathbf{r}^{-1}, \mathbf{e}_{i,o}^{DE}(t)\rangle)$ |
| UTEE | $\|\mathbf{e}_i^{UTEE}(t) + \mathbf{r} - \mathbf{e}_j^{UTEE}(t)\ \|$ | $\frac{1}{2}(\langle \mathbf{e}_{i,s}^{UTEE}(t), \mathbf{r}, \mathbf{e}_{j,o}^{UTEE}(t)\rangle + \langle \mathbf{e}_{j,s}^{UTEE}(t), \mathbf{r}^{-1}, \mathbf{e}_{i,o}^{UTEE}(t)\rangle)$ |
| HyTE | $\|\mathbf{e}_i \perp \boldsymbol{\omega}_t + \mathbf{r} \perp \boldsymbol{\omega}_t - \mathbf{e}_j \perp \boldsymbol{\omega}_t\|$ | $\frac{1}{2}(\langle \mathbf{e}_{i,s} \perp \boldsymbol{\omega}_t, \mathbf{r} \perp \boldsymbol{\omega}_t, \mathbf{e}_{j,o} \perp \boldsymbol{\omega}_t\rangle + \langle \mathbf{e}_{j,s} \perp \boldsymbol{\omega}_t, \mathbf{r} \perp \boldsymbol{\omega}_t, \mathbf{e}_{i,o} \perp \boldsymbol{\omega}_t\rangle)$ |
| ATiSE | $\|\|\mathbf{e}_i^{ATiSE}(t) + \mathbf{r}^{ATiSE}(t) - \mathbf{e}_j^{ATiSE}(t)\|\|$ | $\frac{1}{2}(\langle \mathbf{e}_{i,s}^{ATiSE}(t), \mathbf{r}^{ATiSE}(t), \mathbf{e}_{j,o}^{ATiSE}(t)\rangle + \langle \mathbf{e}_{j,s}^{ATiSE}(t), \mathbf{r}^{-1,ATiSE}(t), \mathbf{e}_{i,o}^{ATiSE}(t)\rangle)$ |
| TA | $\|\mathbf{e}_i + \mathbf{r}^{TA}(t) - \mathbf{e}_j\|$ | $\frac{1}{2}(\langle \mathbf{e}_{i,s}, \mathbf{r}^{TA}(t), \mathbf{e}_{j,o}\rangle + \langle \mathbf{e}_{j,s}, \mathbf{r}^{-1,TA}(t), \mathbf{e}_{i,o}\rangle)$ |

2018), and HyTE (Dasgupta et al., 2018) for the second part of our study, which are the most famous tKGE models.

**Evaluation** All models are evaluated on *link prediction task*. For each test quadruple $(s, r, o, t)$, we create a subject prediction query $(?, r, o, t)$ and an object prediction query $(s, r, ?, t)$. Taking the object prediction as an example, all entities $e_i \in \mathcal{E}$ are ranked according to the score $s(s, r, e_i, t)$. We filter from the candidate list all the entities but the ground truth that form a valid quadruple with $s, r$, and $t$, i.e., the quadruple occurs either in the training, validation, or test data. We report filtered Mean Reciprocal Ranks (MRR) and Hits@1, 3, 10 averaged over subject prediction and object prediction. For detailed definitions please see Appendix A.

**Computational resources and model selection.** We perform large-scale benchmarking with about 4000 experiments and 19000 GPU hours of computation time. All experiments are run on NVIDIA Tesla T4. For each dataset and interaction model, we first randomly generate 40 different configurations from the search space using the Ax framework[2]. After the random hyperparameter search, we search 60 new configurations based on Bayesian optimization to tune the numerical hyperparameters further. Each trial runs for 100 epochs, and an early stopping strategy with a patience of 30 epochs is employed. We select the best-performing configuration according to filtered MRR on validation data. The best configuration will be further trained until its convergence.

### 3.3 Examining Temporal Embeddings

**Performance in prior studies vs. in our study.** Table 3 shows the filtered MRR and filtered

[2]https://ax.dev/

Table 2: Selected hyperparameters of best performing configurations of selected tKG models on ICEWS14. A full description of hyperparameters are reported in Table 12 in the appendix.

| | TTransE | T-SimplE | DE-TransE | DE-SimplE |
|---|---|---|---|---|
| Emb. size | 512 | 256 | 256 | 128 |
| lr. | 7e-3 | 9e-3 | 2e-3 | 4e-3 |
| loss | CE | CE | CE | BCE |
| Reciprocal | Yes | Yes | Yes | Yes |
| Position-aware ent. emb. | Yes | Yes | Yes | Yes |

Hits@1/3/10 on test data of various temporal embeddings on ICEWS14 and ICEWS11-14 datasets. We found that the relative performance differences between various temporal embeddings often shrink and sometimes even reverse compared to published results. For example, T-TransE was first run on ICEWS14 by (García-Durán et al., 2018), achieving a filtered MRR of 25.5%. This number is relatively low compared to today's standards. In comparison, T-TransE achieves a superior MRR of **55.3%** in our study, which has been improved significantly. Studies that report the lower performance number of T-TransE (i.e., 25.5%) thus do not fairly compare the temporal embedding approaches. Similar remarks hold for DE-TransE and HyTE-TransE. (Goel et al., 2019) proposed DE-TransE and report an MRR of 32.6% on ICEWS14 while it achieves an MRR of 50.8 % in our study. Similarly, the achieved MRR of HyTE-TransE on ICEWS14 is 42.9% in our study, which significantly improves the reported results (29.7%) in previous studies (Goel et al., 2019; Sadeghian et al., 2021). The results suggest that the performance of old temporal embedding approaches can be largely improved by advanced training strategies and hyperparameter-tuning, which may account for a large fraction of the progress made in recent years. Figure 1 shows the distribution of

Table 3: Link prediction results of six temporal embedding approaches with two representative score functions on ICEWS datasets: MRR (%) and Hits@1/3/10 (%). The best results in group are in bold.

| Dataset | ICEWS14 | | | | | | | | ICEWS11-14 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Score function | TransE | | | | SimplE | | | | TransE | | | | SimplE | | | |
| Temporal Embeddings | MRR | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 |
| T | **55.3** | **43.7** | **62.7** | **76.5** | **53.9** | **43.9** | 59.4 | 73.0 | **57.8** | **46.0** | **65.5** | **79.5** | **60.2** | **51.3** | **65.2** | 75.5 |
| DE | 50.8 | 38.7 | 59.0 | 72.4 | 53.9 | 42.5 | 61.2 | 74.6 | 54.1 | 42.1 | 60.9 | 77.1 | 54.2 | 42.3 | 61.0 | 67.8 |
| UTEE | 52.6 | 40.5 | 60.3 | 74.7 | 53.7 | 42.5 | **60.8** | **74.8** | 55.2 | 43.0 | 63.3 | 77.5 | 56.1 | 45.2 | 62.9 | **76.4** |
| HyTE | 47.9 | 35.8 | 54.1 | 71.8 | 52.3 | 41.9 | 58.9 | 71.4 | 48.2 | 36.3 | 54.1 | 72.0 | 54.9 | 43.1 | 61.7 | 77.4 |
| ATiSE | 47.1 | 34.7 | 53.8 | 71.2 | 46.6 | 34.7 | 53.4 | 69.7 | 51.0 | 38.8 | 57.7 | 74.5 | 49.3 | 37.5 | 56.1 | 72.2 |
| TA | 22.3 | 14.4 | 25.0 | 37.5 | 37.1 | 25.3 | 42.2 | 61.4 | 26.3 | 18.3 | 28.6 | 43.0 | 33.4 | 24.0 | 37.6 | 51.2 |

Table 4: Link prediction results of six temporal embedding approaches with two representative score functions on the GDELT-m10 dataset: MRR (%) and Hits@1/3/10 (%). The best results in group are in bold.

| Dataset | GDELT-m10 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Score function | TransE | | | | SimplE | | | |
| Temporal Embeddings | MRR | Hits@1 | Hits@3 | Hits@10 | MRR | Hits@1 | Hits@3 | Hits@10 |
| T | **31.6** | **22.7** | **34.0** | **48.9** | 30.8 | 21.6 | 33.5 | 48.8 |
| DE | 25.9 | 17.1 | 28.1 | 43.0 | **34.4** | **24.9** | **37.5** | **53.0** |
| UTEE | 26.1 | 16.9 | 28.3 | 44.1 | 28.5 | 18.9 | 30.9 | 47.4 |
| HyTE | 30.8 | 21.9 | 33.2 | 48.3 | 27.4 | 17.8 | 29.9 | 46.9 |
| ATiSE | 25.3 | 16.7 | 27.3 | 42.1 | 29.6 | 20.6 | 32.2 | 47.3 |
| TA | 11.6 | 1.0 | 16.1 | 29.8 | 19.9 | 12.4 | 21.1 | 34.3 |

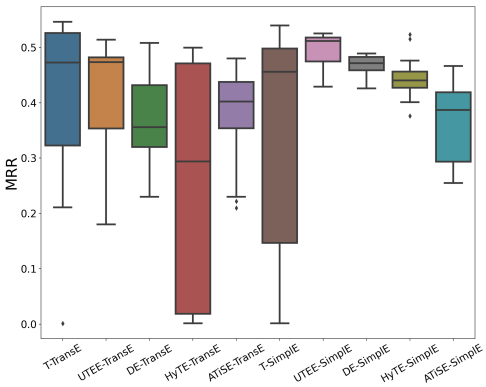


Figure 1: Distribution of filtered MRR (%) on ICEWS14 over the hyperparameter configurations explored in our study.

filtered MRR for each model on ICEWS14. Each distribution consists of 100 different hyperparameter configurations. We can see that some models show a wide dispersion, and only very few configurations achieve good results. Generally, the impact of the hyperparameter choice is more pronounced on TransE-based models (higher variance) than on SimplE-based models. The hyperparameters of the best performing models are reported in Table 2 (selected hyperparameters) and Table 12 in the appendix (all parameters). Perhaps unsurprisingly, we find that the optimum choice of hyperparameters is often model- and dataset-dependent. Thus, a grid search on a small search space is not suitable to compare model performance because the result may be considerably affected by the specific grid points being used. Besides, we find that the use of **r**eciprocal **r**elations (RR) and **p**osition-aware **e**ntity **e**mbeddings (PEE) often improve model performance. To investigate their impacts, we conduct ablation studies where we do not use RR (or PEE) and keep other hyperparameters same to the best configuration. We report the reduction of filtered metrics in Table 5, which confirms our findings.

**TE vs. TEE** Since the timestamp embeddings (TE) are independent of entities, they can only capture global patterns at each timestamp. In comparison, the time-dependent entity embedding approaches (DE, ATiSE) learn entity-specific temporal functions (e.g., frequency, amplitude, etc.) as shown in Equation 1 and 2. The time-dependent entity embeddings are expected to capture entity-specific temporal features, and thus, being more expressive. However, we see that the simple timestamp embedding approach (T) proposed by (Leblay and Chekol, 2018) achieves overall the best performance. In particular, it outperforms the time-dependent entity embedding approaches (DE, ATiSE), which is in contrast to the common belief. Table 6 provides the number of learnable parameters of each model, showing that the interaction model with timestamp embeddings (T) has significantly fewer model parameters than time-dependent entity embeddings (DE, ATiSE). We argue that the existing time-dependent entity embeddings are overfitting to temporal signals. To this end, we propose the unique time-dependent entity embeddings (UTEE), where we learn a unique (global) entity embedding function for all entities to investigate the overfitting problem of DE. In other words, all entities have the same temporal embedding part. Notably, the model parameter of DE is often more than three times than UTEE. As shown

Table 5: Impact of reciprocal relations and position-aware entity embeddings on ICEWS14. The number in parenthesis shows the performance reduction if the best configuration doesn't use the reciprocal relation or position-aware embeddings.

| | **with/without reciprocal relation** | | | | **with/without position-aware entity embeddings** | | | |
|---|---|---|---|---|---|---|---|---|
| Models | MRR(%) | Hits@1(%) | Hits@3(%) | Hits@10(%) | MRR(%) | Hits@1(%) | Hits@3(%) | Hits@10(%) |
| T-TransE | 51.7 (-3.6) | 39.4(-4.3) | 59.1(-3.6) | 75.0(-1.5) | 31.2(-24.1) | 10.6(-33.1) | 44.8(-17.9) | 68.4(-8.1) |
| HyTE-TransE | 40.1(-7.0) | 28.4(-9.9) | 44.9(-12.2) | 64.3(-7.7) | 27.7(-19.4) | 9.6(-32.3) | 38.2(-20.7) | 62.2(-7.5) |
| HyTE-SimplE | 41.2(-11.1) | 28.9(-13.0) | 47.3(-11.6) | 65.4(-4.3) | 51.2(-1.1) | 40.9(-1.0) | 57.7(-1.2) | 70.2(-1.2) |

in Table 3, the UTEE achieves competitive or even better performance with both translation-based and bilinear score functions on both datasets. Additionally, Table 4 shows the evaluation metrics on the GDELT-m10 dataset. Compared to the ICEWS datasets, the number of entities and relations on GDELT-m10 is much fewer while the amount of timestamped edges is about three times more than the ICEWS14 dataset. Thus, the data sparsity issue is alleviated in the GDELT-m10 dataset. Since TEE approaches need dense data for training, their performance has been improved on the GDELT-m10 dataset, which is better than TEs. The results suggest that even though DE has theoretical full expressiveness and provides more freedom degrees of the temporal movements of each entity representation, their performance would deteriorate significantly on sparse data. We tried to add regularization to entity-specific parameters, e.g., amplitude and frequency, and adjust the portion $\gamma$ of the temporal embeddings. However, there are no significant improvements. Thus, the time-dependent entity embeddings need to be revisited to realize their theoretical expressiveness.

Table 6: Model parameters number: million (M).

| Dataset | **ICEWS14** | | **ICEWS11-14** | | **GDELT-m10** | |
|---|---|---|---|---|---|---|
| Score function | **TransE** | **SimplE** | **TransE** | **SimplE** | **TransE** | **SimplE** |
| T | 7.72M | 3.86M | 7.89M | 3.94M | 0.55M | 0.55M |
| UTEE | 7.54M | 3.77M | 28.57M | 7.14M | 0.55M | 0.55M |
| DE | 16.2M | 12.01M | 18.45M | 14.6M | 0.82M | 2.11M |
| HyTE | 3.86M | 3.86M | 1.54M | 3.94M | 0.55M | 0.27M |
| ATiSE | 18.9M | 4.72M | 8.94M | 6.98M | 0.67M | 0.67M |
| TA | 0.78M | 0.78M | 0.5M | 0.5M | 0.22M | 0.22M |

**Findings on other temporal embeddings.** Besides, we find HyTE is sensitive to the choice of score functions. With the translation-based score function (TransE), HyTE only achieves a relatively low number by today's standards while it obtains a competitive number with the bilinear score function (SimplE). This suggests that the score function has a considerable impact on model performance and may account for a large fraction of the progress. Thus, if a new temporal embedding technique is proposed, it should be evaluated on different score functions to assess its benefits. Additionally, we see that the TA-approach (García-Durán et al., 2018) achieves overall relatively low numbers by today's standards, showing its limited capacity. Moreover, we find that the performance of ATiSE considerably deteriorates in our study compared to the prior study. The difference is that we do not cover the Gaussian noise component in our study. This result suggests that taking temporal uncertainty into account would significantly improve the tKG models. Thus, it is worth extending other deterministic KG models into probabilistic approaches.

### 3.4 Benchmarking tKGE Models

Table 7: Link prediction results of well-known tKG models on ICEWS14. The number outside the parentheses is the performance achieved in our study. The number in the parentheses is the best performance results obtained in prior studies. We list the references indicate where the performance number was reported: TTransE/TA-TransE (García-Durán et al., 2018), HyTE/DE-TransE/DE-SimplE (Goel et al., 2019), TNTComplEx (Lacroix et al., 2020), ATiSE (Xu et al., 2019). For ATiSE and HyTE, we use the same score function (KL divergence and TransE, respectively) as reported in their original papers.

| Models | MRR (%) | Hits@1 (%) | Hits@10 (%) |
|---|---|---|---|
| TTransE | 55.3(25.5) | 43.7(7.4) | 76.5(60.1) |
| HyTE | 47.9(29.7) | 35.8(10.8) | 71.8(65.5) |
| DE-TransE | 50.8(32.6) | 38.7(12.4) | 72.4(68.6) |
| DE-SimplE | 53.9(52.6) | 42.5(41.8) | 74.6(72.5) |
| ATiSE | 55.1(55.0) | 42.5(43.6) | 75.0(75.0) |
| TNTComplEx | 60.6(62) | 51.6(52) | 77.3(76) |
| TA-TransE | 26.3(27.5) | 18.3(9.5) | 43.0(62.5) |

Table 7 depicts the best performance of well-known tKGE models from prior studies (numbers in the parentheses) and that found in our study (numbers outside the parentheses). The configu-

ration of the best performing models are reported in Table 13 in the appendix. First, we find that the performance of a single model can vary wildly across studies. For example, DE-TransE, T-TransE, and HyTE have been significantly improved using advanced training strategies and hyperparameter-tuning. Besides, we see that some recent models cannot consistently outperform old models in contrast to the conclusion in prior studies. In particular, T-TransE, which constitutes one of the first tKGE models, achieves results competitive to advanced models, i.e., ATiSE and DE-SimplE, in our study. Even compared to TNTComplEx, which is a very large models with 25.12 million learnable parameters (3 times more than TTransE), the performance difference is not large. We provide explanation for the performance gap between our study and prior study regarding TA-TransE and TNTComplEx in Appendix D.

## 4 Conclusion

We assess well-known temporal embeddings of tKGE models via an extensive experimental study. We found that when trained appropriately, the naive timestamp embedding approach performs similarly or even outperforms the more advanced time-dependent entity embedding (TEE) approaches, which is in contrast to the results in prior studies. We contribute to the community in at least two ways: $i$) we provide a unified framework to enable an insightful assessment for novel temporal embedding approaches; $ii$) reveal the weakness of TEE approaches.

## References


Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and Chengkai Li. 2020. Realistic re-evaluation of knowledge graph completion methods: An experimental study. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 1995–2010.

Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Mikhail Galkin, Sahand Sharifzadeh, Asja Fischer, Volker Tresp, and Jens Lehmann. 2020. Bringing light into the dark: A large-scale evaluation of knowledge graph embedding models under a unified framework. *arXiv preprint arXiv:2006.13365*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

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

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

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

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. 2019. Diachronic embedding for temporal knowledge graph completion. *arXiv preprint arXiv:1907.03143*.

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

Zhen Han, Yunpu Ma, Peng Chen, and Volker Tresp. 2020a. Dyernie: Dynamic evolution of riemannian manifold embeddings for temporal knowledge graph completion. *arXiv preprint arXiv:2011.03984*.

Zhen Han, Yunpu Ma, Yuyi Wang, Stephan Günnemann, and Volker Tresp. 2020b. Graph hawkes neural network for forecasting on temporal knowledge graphs. *arXiv preprint arXiv:2003.13432*.

Woojeong Jin, Changlin Zhang, Pedro Szekely, and Xiang Ren. 2019. Recurrent event network for reasoning over temporal knowledge graphs. *arXiv preprint arXiv:1904.05530*.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. Knowledge base completion: Baselines strike back. *arXiv preprint arXiv:1705.10744*.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in neural information processing systems*, pages 4284–4295.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Timothee Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. *ICLR preprint https://openreview.net/pdf?id=rke2P1BFwS*.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. *arXiv preprint arXiv:1806.07297*.

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. International World Wide Web Conferences Steering Committee.

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.

Yunpu Ma, Volker Tresp, and Erik A Daxberger. 2018. Embedding models for episodic knowledge graphs. *Journal of Web Semantics*, page 100490.

Sameh K Mohamed, Vít Novácek, Pierre-Yves Vandenbussche, and Emir Muñoz. 2019. Loss functions in knowledge graph embedding models. In *DL4KG@ ESWC*, pages 1–10.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. Stranse: a novel embedding model of entities and relationships in knowledge bases. *arXiv preprint arXiv:1606.08140*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, pages 809–816.

Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. 2021. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(2):1–49.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2020. You {can} teach an old dog new tricks! on training knowledge graph embeddings. In *International Conference on Learning Representations*.

Ali Sadeghian, Mohammadreza Armandpour, Anthony Colas, and Daisy Zhe Wang. 2021. Chronor: Rotation based temporal knowledge graph embedding. *arXiv preprint arXiv:2103.10379*.

Volker Tresp, Cristóbal Esteban, Yinchong Yang, Stephan Baier, and Denis Krompaß. 2015. Learning with memory embeddings. *arXiv preprint arXiv:1511.07972*.

Volker Tresp, Yunpu Ma, Stephan Baier, and Yinchong Yang. 2017. Embedding learning for declarative memories. In *European Semantic Web Conference*, pages 202–216. Springer.

Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 3462–3471. JMLR. org.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. International Conference on Machine Learning (ICML).

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*.

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

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhan. 2020. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. *arXiv preprint arXiv:2012.08492*.

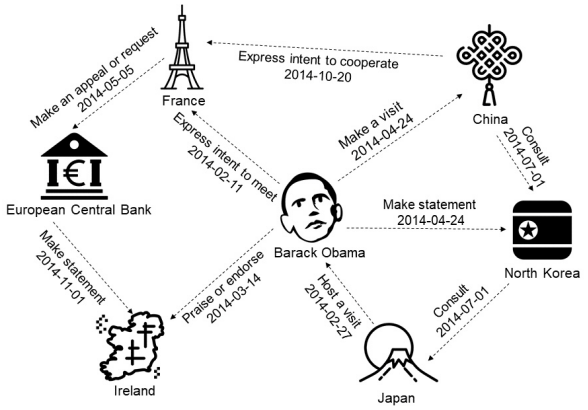


Figure 2: Exemplary temporal KG: nodes represent entities and edges their respective relations.

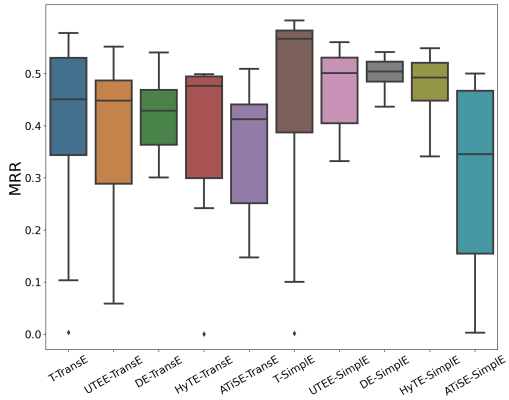


Figure 3: Distribution of filtered MRR (%) on ICEWS11-14 over the hyperparameter configurations explored in our study.

## Appendix

Table 8: Link prediction results of well-known temporal KG models on ICEWS11-14: MRR (%) and Hits@1/3/10 (%).

| Models | MRR | Hits@1 | Hits@3 | Hits@10 |
|---|---|---|---|---|
| TTransE | 57.8 | 46.0 | 65.5 | 79.5 |
| HyTE | 49.8 | 37.6 | 56.2 | 74.0 |
| DE-TransE | 54.1 | 42.1 | 60.9 | 77.1 |
| DE-SimplE | 54.2 | 42.3 | 61.0 | 67.8 |
| TNTComplEx | 63.5 | 55.4 | 68.5 | 78.8 |
| ATiSE | 53.3 | 40.3 | 61.4 | 77.9 |

## A Evaluation Metrics

For each test quadruple $(e_s, r, e_o, t)$, we create a subject prediction query $(?, r, e_o, t)$ and an object prediction query $(e_s, r, ?, t)$. Let $\psi_{e_s}$ and $\psi_{e_o}$ represent the rank for ground truth subject $e_s$ and ground truth object $e_o$ of the subject prediction query and object prediction query, respectively. We evaluate our models using standard metrics across the link prediction literature: *mean reciprocal rank (MRR)*: $\frac{1}{2\cdot|\mathcal{G}_{test}|}\sum_{q\in\mathcal{G}_{test}}(\frac{1}{\psi_{e_s}}+\frac{1}{\psi_{e_o}})$ and *Hits@k*$(k \in \{1, 3, 10\})$: the percentage of times that the true entity candidate appears in the top $k$ of ranked candidates.

There are two common filtering settings. The first one is following the ranking technique described in (Bordes et al., 2013), where we remove from the list of corrupted **triples** all the **triples** that appear either in the training, validation, or test set. We name it *static filtering*. Trivedi et al. (2017), Jin et al. (2019), and Zhu et al. (2020) use this filtering setting for reporting their results on temporal KG forecasting. However, this filtering setting is not appropriate for evaluating the link prediction on temporal KGs. For example, there is a test quadruple (Barack Obama, visit, India, 2015-01-25), and we perform the object prediction (Barack Obama, visit, ?, 2015-01-25). We have observed the quadruple (Barack Obama, visit, Germany, 2013-01-18) in training set. According to the *static filtering*, (Barack Obama, visit, Germany) will be considered as a genuine triple at the timestamp **2015-01-25** and will be filtered out because the triple (Barack Obama, visit, Germany) appears in the training set in the quadruple (Barack Obama, visit, Germany, 2015-01-18). However, the triple (Barack Obama, visit, Germany) is only temporally valid on 2013-01-18 but not on 2015-01-25. To this end, another filtering scheme was introduced, which is more appropriate for the link forecasting task on temporal KGs. We name it *time-aware filtering*. In this case, we only filter out the triples that are genuine at the timestamp of the query. In other words, if the triple (Barack Obama, visit, Germany) does not appear at the query time of 2015-01-25, the quadruple (Barack Obama, visit, Germany, 2015-01-25) is considered as corrupted. In this paper, we focus on time-aware filtering.

## B Additional Information of Hyperparameter Search Space

**Loss functions** Various loss functions are used in training temporal knowledge graphs. Dasgupta et al. (2018); Leblay and Chekol (2018) used margin ranking (MR) loss for training, where each pair consists of a positive quadruple and one of its neg-

ative quadruple. The margin $\eta$ is a hyperparameter. Goel et al. (2019); García-Durán et al. (2018) treat the entity prediction task as a categorical classification problem and utilize the cross entropy (CE) loss to align the model distribution and the data distribution. Han et al. (2020a) proposed to use binary cross entropy (BCE) loss that applies a sigmoid function to the score of each positive or negative quadruples and takes the cross entropy between the resulting probability and that quadruple's label as the loss. It has been shown in (Ruffinelli et al., 2020; Mohamed et al., 2019) that the loss function has a significant impact on the performance of static KGE models. To provide additional evidence on temporal KGE models, we search the best choice of loss functions for each model on each dataset.

**Regularization methods** L2 regularization is widely used in literature (Leblay and Chekol, 2018). Besides, (Dasgupta et al., 2018) proposed to use L1-norm in the regularization term. And (Lacroix et al., 2020) used L3-norm for CP-decomposition. Additionally, (Lacroix et al., 2020) proposed a smoothness regularization for timestamp embeddings that enforce neighboring timestamps to have close representations. Moreover, (Goel et al., 2019) used dropout in its hidden layers. AiTSE normalized the static embeddings $\mathbf{e}_i$, the trend component $\mathbf{w}_{e_i}$ to unit norm after each update.

**Other hyperparameters** For models with diachronic entity embedding as its temporal encoding heads, we extend the static feature ratio as an extra searchable hyperparameter. The negative sample ratio of the negative sampling policy is 500. Namely, for each positive sample (s, p, o, t), we corrupt the subject and object entity via uniformly sampling from $T$, where $T = \{(s', p, o, t)|s' \in \mathcal{E}\backslash s\} \cup (\{(s', p, o, t)|t' \in \mathcal{E}\backslash o\}$. We set our batch size to be 512. Besides, since Adam (Kingma and Ba, 2014) optimizer performs well for the majority of the models (Ali et al., 2020), we decided to progress only with Adam in order to reduce the computational costs. Additionally, for translational models, we set the margin $\gamma$ to be 100 in the score function.

## C Datasets

Dataset statistics including subset split information are described in Table 11. We follow the data preprocessing method used in the original papers. For example, DE-SimplE (Goel et al., 2019) takes the date (day/month/year) as timestamp input while AiTSE (Xu et al., 2019) converts dates into consecutive integers.

Table 9: The average runtime of each training epoch: seconds (s).

| Dataset | ICEWS14 | | ICEWS11-14 | | GDELT-m10 | |
|---|---|---|---|---|---|---|
| Score function | **TransE** | **SimplE** | **TransE** | **SimplE** | **TransE** | **SimplE** |
| T- | 99s | 64s | 80s | 105s | 290s | 321s |
| UTEE- | 128s | 75s | 450s | 262s | 1230s | 638s |
| DE- | 196s | 145s | 375s | 302s | 438s | 518s |
| HyTE- | 208s | 212s | 146s | 105s | 360s | 382s |
| ATiSE- | 317s | 75s | 212s | 175s | 380s | 390s |
| TA- | 730s | 365s | 730s | 365s | 2696s | 3751s |

## D Reproducibility Studies

We were not able to reproduce the results of TA-TransE on ICEWS14. A reason might be differences in the implementation details of the frameworks used to train and evaluate the models. Since there exists no official implementation for TA-TransE, it is not possible to check the implementation difference. Also, García-Durán et al. (2018) did not report the full setup, which impedes the reproduction of results. For example, the regularization method and initialization method have not been reported, which can have a significant effect on the results.

Lacroix et al. (2020) provides an official implementation of TNTComplEx. However, we were not able to reproduce the same metric number as reported in their paper. Similarly, Sadeghian et al. (2021) also did not successfully reproduce the results of TNTComplEx. The initialization of the embeddings might be a reason.

## E Average Runtime of each Approach

Table 9 shows the average runtime of each training epoch for each interaction model.

Table 10: Hyperparameter search space used in our study.

| Hyperparameter | Search space |
|---|---|
| Embedding | |
| Embedding dimension | {64, 100, 128, 256, 512, 1024, 2048} |
| Embedding initialization | {Xavier Uniform, Xavier Normal} |
| Training | |
| Reciprocal relation | {True, False} |
| Position-aware entity embeddings | {True, False} |
| Loss function | {CE, BCE, MR} |
| Learning rate | (0.0, 0.1] |
| Regularization | |
| Entity regularization type | {None, L1, L2, L3} |
| Entity regularization weight | (0.0, 0.1] |
| Relation regularization type | {None, L1, L2, L3} |
| Relation regularization weight | (0.0, 0.1] |
| Timestamp smoothness regularization weight | (0.0, 0.1] |
| Dropout | [0.0, 0.6] |

| Data set | $N_{train}$ | $N_{valid}$ | $N_{test}$ | $N_{ent}$ | $N_{rel}$ | $N_{timestamp}$ | Time granularity |
|---|---|---|---|---|---|---|---|
| ICEWS14 | 72826 | 8941 | 8963 | 7128 | 230 | 365 | day |
| ICEWS11-14 | 118766 | 14859 | 14756 | 6738 | 235 | 1461 | day |
| GDELT-m10 | 221132 | 27608 | 27926 | 50 | 20 | 30 | day |

Table 11: Dataset Statistics

Table 12: Best configurations of six temporal embeddings with two score functions on ICEWS14, ICEWS11-14 and GDELT-m10.

| | | Emb. dim. | Emb. init. | SE ratio | Recip. rel. | Pos-aware entity | Learning rate | Loss func. | Ent. reg. type | Ent. reg. weight | Rel. reg. type | Rel. reg. weight | Temp. smooth. weight. | Dropout |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ICEWS14 | T-TransE | 512 | Xavier Uniform | - | True | True | 0.0075324 | CE | - | - | - | - | 1.9513135e-17 | 0.5 |
| | UTEE-TransE | 512 | Xavier Uniform | - | True | True | 0.0010000 | CE | None | - | None | - | - | 0.5 |
| | DE-TransE | 256 | Xavier Uniform | 0.57 | True | True | 0.0019251 | CE | None | - | None | - | - | 0.4 |
| | HyTE-TransE | 128 | Xavier Uniform | - | True | True | 0.0080112 | CE | L2 | 0.0368056 | L2 | 0.0762292 | 0.0381662 | 0.6 |
| | ATiSE-TransE | 256 | Xavier Uniform | - | True | True | 0.0029397 | CE | | - | - | - | - | 0.2 |
| | T-SimplE | 256 | Xavier Uniform | - | True | True | 0.0090146 | CE | None | - | None | - | 0.0000881 | 0.6 |
| | UTEE-SimplE | 256 | Xavier Uniform | - | True | True | 0.0030251 | BCE | - | - | - | - | - | 0.5 |
| | DE-SimplE | 128 | Xavier Uniform | 0.42 | True | True | 0.0038826 | BCE | - | - | - | - | - | 0.4 |
| | HyTE-SimplE | 256 | Xavier Uniform | - | True | True | 0.0089471 | CE | None | - | None | - | 0.0315673 | 0.2 |
| | ATiSE-SimplE | 64 | Xavier Uniform | - | True | True | 0.0006407 | BCE | L3 | 0.0165849 | L3 | 0.0148400 | - | 0.2 |
| ICEWS11-14 | T-TransE | 512 | Xavier Uniform | - | False | True | 0.0018173 | CE | - | - | - | - | 0.0115133 | 0.4 |
| | UTEE-TransE | 2048 | Xavier Normal | - | True | True | 0.0003236 | CE | None | - | L2 | 0.0968473 | - | 0.0 |
| | DE-TransE | 256 | Xavier Uniform | 0.46 | True | True | 0.0026801 | CE | - | - | - | - | - | 0.6 |
| | HyTE-TransE | 100 | Xavier Uniform | - | True | True | 0.0084563 | CE | L2 | 0.0548569 | L3 | 0.0848771 | 0.0718383 | 0.0 |
| | ATiSE-TransE | 128 | Xavier Uniform | - | True | True | 0.0013809 | CE | L3 | 0.0011405 | L2 | 0.0182759 | 0.0115133 | 0.2 |
| | T-SimplE | 256 | Xavier Uniform | - | True | True | 0.0015910 | CE | - | - | - | - | 0.0049301 | 0.4 |
| | UTEE-SimplE | 512 | Xavier Normal | - | True | True | 0.0048310 | CE | - | - | - | - | - | 0.6 |
| | DE-SimplE | 128 | Xavier Normal | 0.07 | True | True | 0.0040708 | CE | None | - | None | - | - | 0.6 |
| | HyTE-SimplE | 100 | Xavier Normal | - | True | True | 0.0077529 | CE | None | - | None | - | 0.0751268 | 0.5 |
| | ATiSE-SimplE | 100 | Xavier Uniform | - | True | True | 0.0006454 | CE | L2 | 0.0397851 | L3 | 0.0262544 | - | 0.4 |
| GDELT-m10 | T-TransE | 512 | Xavier Normal | - | TRUE | TRUE | 0.0005205 | CE | None | - | L3 | 0.0861472 | 0.0123214 | 0.2 |
| | UTEE-TransE | 512 | Xavier Uniform | - | TRUE | TRUE | 0.0018018 | BCE | None | - | None | - | - | 0.4 |
| | DE-TransE | 100 | Xavier Normal | 0.10 | TRUE | TRUE | 0.0021739 | CE | None | - | None | - | - | 0.4 |
| | HyTE-TransE | 512 | Xavier Normal | - | TRUE | TRUE | 0.0007655 | CE | L3 | 0.0128709 | L2 | 0.0440400 | 0.0755946 | 0.2 |
| | ATiSE-TransE | 128 | Xavier Normal | - | TRUE | TRUE | 0.0001836 | CE | L2 | 0.0322397 | L2 | 0.0854999 | - | 0.4 |
| | T-SimplE | 512 | Xavier Normal | - | TRUE | TRUE | 0.0086667 | CE | L3 | 0.0039186 | L3 | 0.0466820 | 0.0730538 | 0.2 |
| | UTEE-SimplE | 512 | Xavier Normal | - | TRUE | TRUE | 0.0006767 | CE | L2 | 0.0035275 | None | - | - | 0.4 |
| | DE-SimplE | 256 | Xavier Normal | 0.66 | TRUE | TRUE | 0.0021578 | BCE | None | - | None | - | - | 0.4 |
| | HyTE-SimplE | 256 | Xavier Normal | - | TRUE | TRUE | 0.0046384 | CE | L3 | 0.0034211 | L2 | 0.0986175 | 0.0143311 | 0.2 |
| | ATiSE-SimplE | 128 | Xavier Normal | - | TRUE | TRUE | 0.0059198 | CE | L2 | 0.0002838 | L2 | 0.0819929 | - | 0.4 |

Table 13: Best configurations of well-known temporal knowledge graph models on ICEWS14 and ICEWS11-14.

| | | Emb. dim. | Emb. init. | SE ratio | Recip. rel. | Pos-aware entity | Learning rate | Loss func. | Ent. reg. type | Ent. reg. weight | Rel. reg. type | Rel. reg. weight | Temp. smooth. weight. | Dropout |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ICEWS14 | T-TransE | 512 | Xavier Uniform | - | True | True | 0.0075324 | CE | - | - | - | - | 1.9513135e-17 | 0.5 |
| | DE-TransE | 256 | Xavier Uniform | 0.57 | True | True | 0.0019251 | CE | None | - | None | - | - | 0.4 |
| | DE-SimplE | 128 | Xavier Uniform | 0.42 | True | True | 0.0038826 | BCE | - | - | - | - | - | 0.4 |
| | TNTComplEx | 1560 | - | - | True | - | 0.1 | CE | n3 | 0.01 | - | - | 0.01 | - |
| | ATiSE | 500 | - | - | False | - | 0.00003 | LMR | - | - | - | - | - | - |
| | HyTE | 100 | Xavier Uniform | - | True | True | 0.0084563 | CE | L2 | 0.0548569 | L3 | 0.0848771 | 0.0718383 | 0.0 |
| ICEWS11-14 | TTransE | 512 | Xavier Uniform | - | False | True | 0.0018173 | CE | - | - | - | - | 0.0115133 | 0.4 |
| | DE-TransE | 256 | Xavier Uniform | 0.46 | True | True | 0.0026801 | CE | - | - | - | - | - | 0.6 |
| | DE-SimplE | 128 | Xavier Normal | 0.07 | True | True | 0.0040708 | CE | None | - | None | - | - | 0.6 |
| | TNTComplEx | 1560 | - | - | True | - | 0.0859516 | CE | n3 | 0.001 | - | - | 0.1 | - |
| | ATiSE | 500 | - | - | False | - | 0.00003 | LMR | - | - | - | - | - | - |
| | HyTE | 100 | Xavier Normal | - | True | True | 0.0077529 | CE | None | - | None | - | 0.0751268 | 0.5 |
| GDELT-m10 | TTransE | 512 | Xavier Normal | - | TRUE | TRUE | 0.0005205 | CE | None | - | L3 | 0.0861472 | 0.0123214 | 0.2 |
| | DE-TransE | 100 | Xavier Normal | 0.10 | TRUE | TRUE | 0.0021739 | CE | None | - | None | - | - | 0.4 |
| | DE-SimplE | 256 | Xavier Normal | 0.66 | TRUE | TRUE | 0.0021578 | BCE | None | - | None | - | - | 0.4 |
| | TNTComplEx | 1560 | - | - | True | - | 0.1499999 | CE | n3 | 0.001 | - | - | 0.1 | - |
| | ATiSE | 512 | - | - | True | - | 0.00003 | LMR | - | - | - | - | - | - |
| | HyTE | 512 | Xavier Normal | - | TRUE | TRUE | 0.0007655 | CE | L3 | 0.0128709 | L2 | 0.0440400 | 0.0755946 | 0.2 |