DDPC: Dual Dynamic Presentation with Contrastive Learning for Robust Temporal Knowledge Graph Completion

Anonymous EMNLP submission

Abstract

Temporal knowledge graph completion has made significant progress, but several research gaps persist. This study addresses the challenges of temporal changes by proposing DDP and DDPC, novel dual-perspective learning frameworks that integrate static and temporal knowledge using a dual-layer embedding mechanism and a contrastive learning-enhanced version, respectively. This approach effectively captures both dynamic changes and time-invariant properties of entities and relations, optimizing the completeness and accuracy of information. Additionally, a perturbation learning mechanism is introduced to enhance the model's robustness to anomalous data and noise by simulating data perturbations during training, improving adaptability and stability in changing environments. DDPC achieves state-of-the-art results on multiple standard evaluation datasets, experimentally verifying the effectiveness of the proposed theories and methods. This study contributes to advancing the field of temporal knowledge graph completion by developing an innovative framework that integrates temporal and static perspectives, enhances robustness, and undergoes rigorous evaluations.

1 Introduction

Temporal Knowledge Graphs (TKGs) extend traditional Knowledge Graphs by incorporating time information, enabling the representation and reasoning over dynamic, time-dependent relationships between entities. Temporal Knowledge Completion (TKC) is a crucial task in TKGs that involves inferring missing facts or relationships at specific time points or intervals by leveraging existing temporal information, ultimately improving the accuracy and completeness of the graph.

Temporal Knowledge Graph Embedding (TKGE) models capture temporal dynamics in knowledge graphs by incorporating temporal information, allowing for the representation of evolving relationships over time. Notable TKGE models includeTTransE (Jiang et al., 2016), which extends TransE by embedding timestamps;

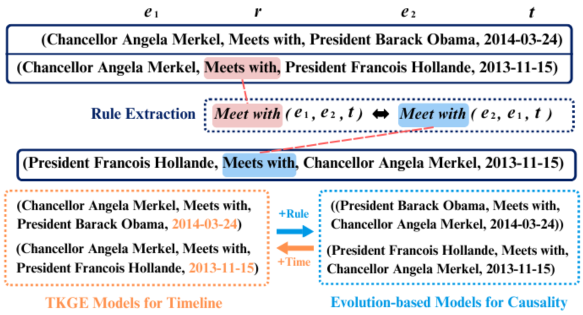

Figure 1: An example of coupling TKGE models and Evolution-based models using causality and timeline

Know-Evolve (Trivedi et al., 2017), which uses a recurrent neural network; HyTE (Dasgupta et al., 2018), which employs a hyperplane-based approach; and DE-SimplE (Goel et al., 2020), which adapts SimplE for diachronic data. Additionally, TeMP (Wu et al., 2020) uses tensor decomposition methods for temporal graph completion, and ChronoR (Sadeghian et al., 2021) incorporates relative temporal order information to improve prediction accuracy. Convolutional-based models have also shown promise in knowledge graph embeddings, utilizing convolutional neural networks (CNNs) to capture local patterns and interactions between entities and relations. ConvE (Dettmers et al., 2018) employs 2D convolutional layers to embed entities and relations into a unified space, while ConvKB (Nguyen et al., 2018) uses convolutional filters to learn features from concatenated embeddings. More recent models like HypER (Balavzevic et al., 2019) leverage hypernetworks to generate convolutional filters dynamically, and some studies (Shen et al., 2020; Niu and Li, 2023) explore integrating timeline and causality to improve model performance on completion tasks, as shown in Figure 1.

Despite the progress in temporal knowledge graph completion, significant research gaps remain. First, the lack of integration between temporal and static knowledge graphs hinders the ability to capture the dynamic and time-invariant properties of entities and relations effectively. Second, the robustness of existing models is challenged by anomalous data and noise, necessitating the development of perturbation learning mechanisms to enhance adaptability and stability. Finally, the scarcity of comprehensive evaluations across multiple datasets makes it difficult to assess the effectiveness and generalizability of proposed methods. Addressing these gaps by developing innovative frameworks that integrate temporal and static perspectives, enhance robustness, and undergo rigorous evaluations is crucial for advancing the field and creating more accurate and reliable models for temporal knowledge graph completion. The major contributions of this study are listed as follows:

- We propose a novel **D**ual **D**ynamic **P**erspective (DDP) model that integrates static and temporal knowledge, utilizing a dual-layer embedding mechanism to capture dynamic changes and time-invariant properties to optimize the completeness and accuracy of information.
- We introduce a perturbation learning mechanism that enhances the model's robustness to anomalous data and noise by simulating data perturbations during training, improving the model's adaptability, stability, and reliability in changing environments, and propose a dual **D**ynamic **D**ynamic **P**erspective with **C**ontrastive (DDPC) learning model that employs such perturbation-based contrastive learning mechanism.
- Our method achieves State-of-the-Art results on multiple standard evaluation datasets, experimentally verifying the effectiveness of our proposed theories and methods.

2 Related Works

In this section, we review the advancements in temporal knowledge graph completion through embedding-based models and discuss the emergence of contrastive learning techniques, particularly perturbation-based methods, in the context of graph data.

2.1 Embedding-based Models

Temporal knowledge graph completion (TKGC) has witnessed significant progress through the development of embedding-based models that extend traditional knowledge graph embeddings to incorporate temporal dynamics. TTransE (Leblay and Chekol, 2018), TA-DistMult (García-Durán et al., 2018), and DE-SimplE (Goel et al., 2019) are notable models that employ various techniques to capture temporal patterns and evolution of entities and relations. More advanced models, such as ATiSE (Xu et al., 2019), TComplEx (Lacroix et al., 2020), and LCGE (Niu and Li, 2023), introduce additional mechanisms like temporal regularizers, tensor factorization, logical rules, and commonsense knowledge to enhance temporal reasoning capabilities and improve prediction accuracy.

2.2 Contrastive Learning

Contrastive learning has become a powerful self-supervised representation learning framework, particularly in computer vision and natural language processing. In graph data, traditional graph contrastive learning (GCL) methods like GraphCL (You et al., 2020) and MVGRL (Hassani and Khasahmadi, 2020) utilize augmentations to generate positive pairs for contrastive loss. However, these augmentations often require manual selection or domain-specific knowledge, limiting their scalability and efficiency. SimGRACE (Xia et al., 2022) addresses these challenges by perturbing the graph neural network (GNN) encoder instead of the graph data, generating semantically consistent views without manual augmentation selection. Other perturbation strategies in contrastive learning, such as feature perturbation (Zhu et al., 2020), structural perturbation (You et al., 2021), semantic perturbation (Zhu et al., 2021), and augmentation-free methods like BGRL (Thakoor et al., 2021) and MERIT (Jin et al., 2021), have their own limitations in terms of computational cost or effectiveness in preserving graph data semantics.

2.3 Rule Learning for Knowledge Graph Completion

Logic rules are naturally suited for knowledge graph (KG) completion due to the symbolic nature of KGs. Horn rules, a common type of logic rule, take the form $a_1 \Leftarrow a_2 \wedge a_3 \wedge \cdots \wedge a_n$, where a_1 is the head atom and $a_2, \ldots, a_n$ are the body atoms. Various rule learning algorithms have been developed specifically for large-scale KGs, focusing on efficient rule searching and quality evaluation. Notable examples include AMIE+ (Galárraga et al., 2015), ScaLeKB (Chen et al., 2016), RuLES (Dong et al., 2018), AnyBURL (Meilicke et al., 2019), DRUM (Sadeghian et al., 2019), RLvLR (Omran and Tresp, 2019), and RNNLogic (Qu et al., 2021). These algorithms effectively discover meaningful rules from KGs, contributing to KG completion by inferring missing facts based on the learned rules. LCGE extends previous work on temporal rule learning, which focuses on mining static rules from knowledge graphs and converting them into dynamic temporal rules (Niu and Li, 2023).

3 Methodology

In this section, we introduce two key contributions: a dual-representation approach and a contrastive loss function, as demonstrated in Figure 1. The dual-representation approach integrates time-sensitive and time-independent representations to capture both temporal dynamics and commonsense knowledge, enhancing the model's ability to evaluate event plausibility. The contrastive loss function maximizes the agreement between positive event pairs while ensuring distinct representations for negative pairs, improving the model's robustness and accuracy.

3.1 Preliminaries

A temporal knowledge graph (TKG) is a representation that captures events along with their associated temporal information. In a TKG, each event is represented as a

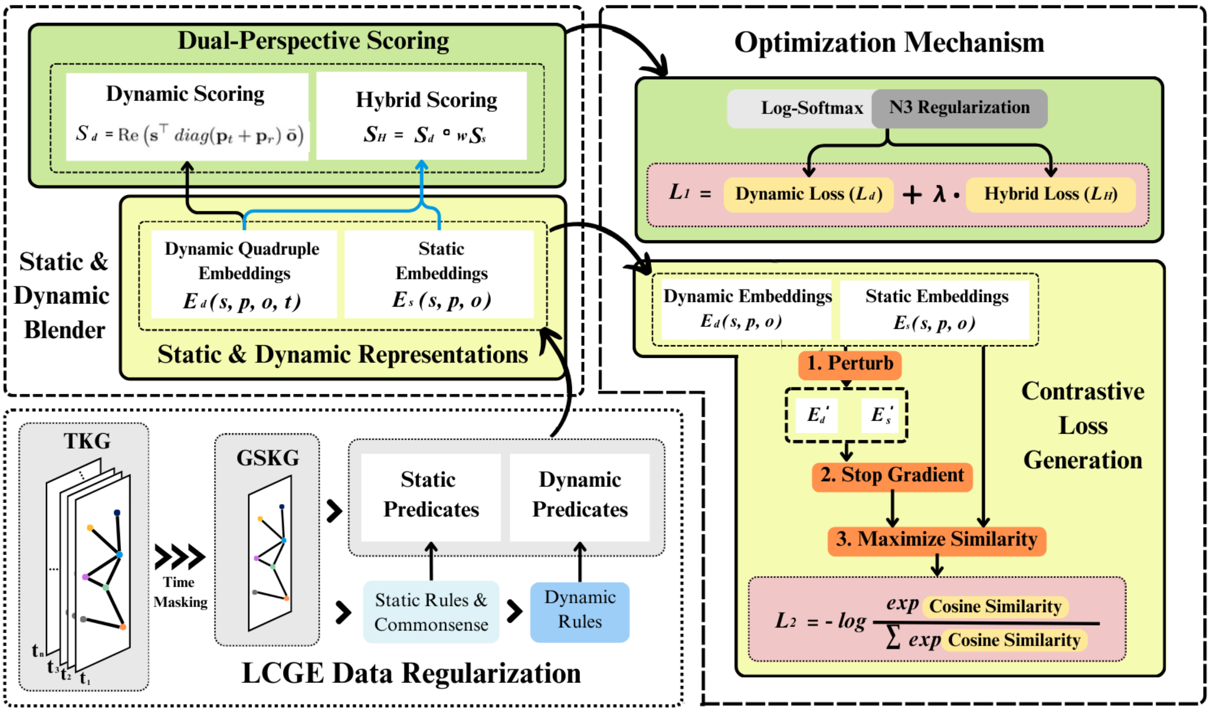

Figure 2: **Model Framework of DDP and DDPC**, using dual-dynamic perspectives for embeddings and perturbation-based contrastive learning to enhance temporal knowledge graph completion performance.

quadruple (s, p, o, t), where s, o, p, t denotes the subject of the event, the object of the event, the predicate or relationship between the subject and object, and the timestamp or time interval corresponding to the occurrence of the event, respectively. In cases where an event is associated with a time interval $[t_{start}, t_{end}]$ instead of a single timestamp, the event can be decomposed into two distinct event quadruples: (s, p, o, t_{start}), signifying the start of the event, and (s, p, o, t_{end}), representing the end of the event. This decomposition allows for a consistent representation of all events using a single timestamp t while still capturing the temporal span of events that occur over a period of time.

3.1.1 Rule-based Data Preparation

In this study, we adopt the temporal rule learning approach proposed by Niu and Li, 2023, which consists of a static-to-dynamic strategy to mine temporal rules with diverse patterns. The process involves two main stages: static rule learning and dynamic rule learning. During the static rule learning stage, the temporal information of each event in the training set is masked, converting the quadruple events into triples. These triples form a global static knowledge graph (GSKG). Static rules are then mined from the GSKG using an existing rule learning algorithm, such as AMIE+ (Galárraga et al., 2015) or AnyBURL (Meilicke et al., 2019). The dynamic rule learning stage extends the static rules into temporal rules by incorporating five well-designed temporal rule patterns based on the temporal sequences among the atoms. These patterns capture various relationships between events occurring at different timestamps. The temporal rule patterns are defined as follows:

1. If $p_1(e_1, e_2)$ holds at time t, then $p_2(e_1, e_2)$ will hold at time $t + t_1$.
2. If $p_1(e_1, e_2)$ holds at time t, then $p_2(e_1, e_2)$ will also hold at the same time t.
3. If $p_1(e_1, e_3)$ holds at time t and $p_2(e_3, e_2)$ holds at time $t + t_1$, then $p_3(e_1, e_2)$ will hold at time $t + t_1 + t_2$.
4. If $p_1(e_1, e_3)$ and $p_2(e_3, e_2)$ both hold at time t, then $p_3(e_1, e_2)$ will hold at time $t + t_1$.
5. If $p_1(e_1, e_3)$ and $p_2(e_3, e_2)$ both hold at time t, then $p_3(e_1, e_2)$ will also hold at the same time t.

, where p represents predicates that describe the type of relationship between entities, e_1, e_2, and e_3 are entities, and t denotes the times. The quality of the candidate temporal rules is evaluated using the support degree (SD), standard confidence (SC), and head coverage (HC) metrics, calculated by searching for events that satisfy the grounding of each rule. Temporal rules that meet the predefined thresholds of SC and HC are retained for the subsequent data preparation process.

Then, we employ the temporal rule-guided predicate embedding regularization (RGPR) mechanism (Niu and Li, 2023) to enhance the data preparation process for our proposed model. The RGPR mechanism leverages temporal rules mined from the knowledge graph to inject the causality among events into predicate embeddings, providing valuable information for improving the model's performance. The RGPR mechanism is based on the temporal rules obtained from the temporal rule learning module, which discovers meaningful temporal dependencies among events in the knowledge graph. These temporal rules are represented in the form of Horn clauses, capturing the causal relationships between events occurring at different timestamps. To apply the RGPR mechanism on each rule pattern, a time transfer operator T is defined to ensure that all the atoms in a rule are represented at the same time when calculating their correlations, and RGPR will guide the predicate embedding regularization. For instance, for the temporal rule pattern $p_2(e_1, e_2, t + t_1) \Leftarrow p_1(e_1, e_2, t)$ (namely, if $p_1(e_1, e_2)$ holds at time t, then $p_2(e_1, e_2)$ will hold at time $t + t_1$.), the regularization term is defined as:

$$G = |(T \circ p_{r1}) - p_{r2}| \tag{1}$$

, where p_{r1} and p_{r2} are the embeddings of predicates p_1 and p_2, respectively. We aim to infuse the predicate embeddings with the causal information captured by the temporal rules. This enhanced representation of predicates is expected to improve the model's ability to reason about the temporal dependencies and causal relationships present in the knowledge graph, ultimately leading to better performance on downstream tasks.

3.2 Dual Perspective Scoring

To effectively represent events in a temporal knowledge graph and evaluate their plausibility, we propose an approach that leverages both time-sensitive and time-independent representations deriving from predicates. The time-sensitive representation captures the temporal dynamics of events, while the time-independent representation incorporates commonsense knowledge. By combining these complementary representations, our approach enables a comprehensive understanding of events, considering both temporal and commonsense aspects, leading to improved performance in various tasks such as event prediction and temporal reasoning. To learn the time-sensitive representation of events, inspired by the TKGE model TComplEx (Timothée Lacroix and Usunier, 2020), we learn the timeliness of each event embodied with the timestamp via fourth-order tensor decomposition. Besides, the causality among events can be represented via our RGPR mechanism together with the subject and object embeddings. Given an event quadruple (s, p, o, t), the time-sensitive score function is defined as:

$$E_d(s,p,o,t) = \Re\left(\sum_{i=1}^{d}[s]_i \cdot [p \circ t + p_r]_i \cdot [\overline{o}]_i\right) \tag{2}$$

To learn the time-independent representation of commonsense associated with events, the timestamp in each event is masked to convert the event quadruple (s, p, o, t) into the factual triple (s, p, o). Motivated by some typical commonsense KGs such as ConceptNet (Robert Speer and Havasi, 2017), commonsense is represented as two concepts linked by a predicate. Therefore, we score each event in the view of commonsense via the learnable concept and predicate embeddings together with the proposed time-independent score function based on commonsense:

$$E_s(s,p,o) = \Re\left(\sum_{i=1}^{k}[s_c]_i \cdot [p_c]_i \cdot [\overline{o_c}]_i\right) \tag{3}$$

where $s_c \in C^k$, $p_c \in C^k$, and $o_c \in C^k$ represent the concept embeddings in the k-dimensional complex vector space with regard to the subject s, predicate p, and object o, respectively. Particularly, k should be set smaller than d to enhance the abstract feature of entity concept embeddings.

Finally, the hybrid scoring is concatenated from the time-sensitive score $E1$ and the time-independent score $E2$ with weights $k1$ and $k2$ to form the final score E:

$$E_H(s,p,o,t) = \sum_{m \in M} w_m E_m(s,p,o,t) \tag{4}$$

, where M is the set that includes the individual embedding functions, and w_m is the weight assigned to each function m in the set M. Specifically, this set M includes the semantic embedding function E_s and the dynamic embedding function E_d. The semantic embedding function, denoted as $E_s(s, p, o)$, captures the semantic relationships between the subject s, predicate p, and object o. On the other hand, the dynamic embedding function, $E_d(s, p, o, t)$, incorporates temporal dynamics by considering the interaction between the subject, predicate, object, and time t. By summing these functions with their respective weights w_m, the hybrid function E_H effectively integrates both static and dynamic aspects, providing a comprehensive representation that leverages the strengths of each individual embedding method.

3.3 Optimization

The optimization process consists of two parts. First, regular optimization is performed for each embedding to learn the representations. Second, contrastive learning is applied to specific embeddings to enhance the learning process and improve the model's performance. The visualized process is displayed in Figure 2, and the overall algorithm, which combines these two optimization techniques, is presented in Algorithm 1.

For the first part, we employ the log-softmax loss function and N3 regularization to design the optimization objective for training:

$$\begin{aligned} L_1 = &\alpha_1\left(|s|_3^3 + |p_t|_3^3 + |p_r|_3^3 + |o|_3^3\right) \\ &- \log\left(\sum \exp(E)\right) - \log\left(\sum \exp(E)\right) \end{aligned} \tag{5}$$

in which L_1 represents the loss functions for both the dynamic representation and hybrid representations of static and dynamic, respectively. E denotes an entity set that contains all events. α_1 is defined as the N3 regularization weights corresponding to entity embeddings and predicate embeddings, respectively. λ is the weight of commonsense representation in the overall loss function, which is applied for the trade-off between the time-sensitive and the time-independent representations of each event.

Inspired by SimGRACE (Xia et al., 2022), we design a contrastive loss function based on standard normal distribution perturbation as an optimizer regularizer. The goal of this loss function is to maximize the agreement between positive pairs (different views of the same graph) while minimizing the agreement between negative pairs (representations of different graphs). Given a mini-batch of N representations, we generate 2N representations by passing the original graph and its perturbed version through the encoder. The perturbation is computed using the following equation:

$$r' = r + \eta \cdot \text{std}(r) \cdot \mathcal{N}(0,1) \tag{6}$$

where r' is the perturbed tensor; std(r) is the standard deviation of the elements in the tensor T; $\mathcal{N}(0,1)$ is a tensor of random values drawn from a standard normal distribution with the same shape as r. Then, let r_i and r_i' represent the representations of the original and perturbed views of the i-th graph, respectively. The contrastive loss for the i-th graph, denoted as ℓ_i, is calculated as follows:

$$\ell_i = -\log \frac{\exp(\frac{\text{sim}(r_i, r_i')}{\tau})}{\sum_{i'=1}^{2N} 1_{[i' \neq i]} \exp(\frac{\text{sim}(r_i, r_{i'})}{\tau})'} \tag{7}$$

where $sim(r_i, r_i')$ represents the cosine similarity between two vectors r_i and r_i', defined as:

$$\text{sim}(r_i, r_i') = \frac{r_i^\top r_i'}{|r_i||r_i'|} \tag{8}$$

and τ is a temperature parameter that controls the scaling of the similarities. The total contrastive loss L_2 across the mini-batch is calculated by taking the average of the individual losses overall positive pairs:

$$L_2 = \frac{1}{2N} \sum_{i=1}^{N} (\ell_i + \ell_i') \tag{9}$$

where ℓ_i' represents the loss computed for the perturbed view r_i' with respect to its positive pair r_i. The overall optimization objective combines the contrastive loss with the loss for dynamic and hybrid embeddings, as shown in the following equation:

$$\mathcal{L} = \sum (L_1(E_d + \cdot E_H) + \lambda \cdot L_2(E)) \tag{10}$$

in which L represents the loss sum of dynamic and hybrid embedding, λ_1 is a hyper-parameter that controls the weight of the hybrid loss, λ_2 is a hyper-parameter that controls the weight of the hybrid loss, and T denotes the embedding set of all entities, relations, and times in the temporal knowledge graph that contains E_d, E_H, and E. Finally, the model is trained using the Adam optimizer to learn the embedding of entities, predicates, and timestamps.

Algorithm 1 Optimization Process

1: **Input:** T (event set), E (entity set), α_1, α_2, λ (regularization and loss weights), τ (temperature parameter)
2: **Output:** The embeddings of entities, relations, and timestamps.
3: Initialize model parameters
4: **for** each $(s, p, o, t) \in T$ **do**
5:     Compute loss for E_H and E_d (Eq.5)
6:     Compute contrastive loss $\mathcal{L}$:
7:     **for** each mini-batch of N representations **do**
8:         Obtain two $2N$ representations
9:         **for** each graph i in mini-batch **do**
10:             Compute perturbation r_i' of r_i (Eq.6)
11:             Compute cosine similarity sim(r_i, r_j)
12:             Compute single contrastive loss (Eq.7)
13:         **end for**
14:         Compute total contrastive loss (Eq.9)
15:     **end for**
16:     Combine L and $\lambda \cdot \mathcal{L}$ (Eq.10)
17: **end for**
18: Train model using Adam optimizer to minimize L
19: Update model parameters
20: Repeat from Line 4 until convergence

4 Experiment

In this section, we present the experimental setup, results, and analysis to evaluate the performance of our proposed model. We introduce the datasets, evaluation protocol, baselines, metrics, and implementation details used in our experiments.

4.1 Datasets

In our experiments, we employ three widely-used temporal knowledge graph (TKG) datasets: ICEWS14 (García-Durán et al., 2018), ICEWS05-15 (García-Durán et al., 2018), and Wikidata12k (Dasgupta et al., 2018). The ICEWS datasets contain political events with specific timestamps, while Wikidata12k, a subset of Wikidata (Erxleben et al., 2014), includes time annotations as either timestamps or time intervals. Following the standard practice in previous works (Lacroix et al., 2020; Xu et al., 2020b; Niu and Li, 2023), we split each dataset into training, validation, and test sets with a ratio of 80%, 10%, and 10%, respectively. This setup allows for a comprehensive evaluation of the models' performance on diverse temporal knowledge graphs. Further information on each dataset is presented in Table 1.

Dataset	Time Span	Predicate	Entity	Train	Valid	Test
ICEWS14	2014	230	6,869	72,826	8,941	8,963
ICEWS05-15	2005-2015	251	10,094	368,962	46,275	46,092
Wikidata12k	1479-2018	24	12,554	32,497	4,062	4,062

Table 1: **Statistics of the experimental datasets**, including the time span, number of predicates, entities, and facts (train, valid, and test sets). The time span indicates the range of years in which the events occur.

4.2 Baselines

To evaluate our model, we compare it with two types of related baselines: typical Knowledge Graph Embedding (KGE) models without time information, such as TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), RotatE (Sun et al., 2019), and QuatE (Zhang et al., 2019), which are widely-used benchmarks to assess our approach's effectiveness in capturing temporal information; and well-performing Temporal Knowledge Graph Embedding (TKGE) models, including TTransE (Leblay and Chekol, 2018), HyTE (Dasgupta et al., 2018), ATiSE (Xu et al., 2020a), TeRo (Xu et al., 2020c), TComplEx (Lacroix et al., 2020), TeLM (Xu et al., 2021), and the latest state-of-the-art model, LCGE (Xie et al., 2022), which handle temporal information and have shown promising results in temporal knowledge graph completion tasks. Comparing our model with these baselines demonstrates its effectiveness in capturing both temporal and commonsense information for improved temporal knowledge graph completion performance.

4.3 Experiment Setups

All experiments are conducted using PyTorch on a GeForce GTX 4090 GPU with a batch size of 1024. The thresholds for support count (SC) and head coverage (HC) in the temporal rule learning algorithm are set to 0.1 for all datasets. Hyperparameters are tuned using grid search on the validation sets. For the ICEWS14 dataset, the rank is 2000, embedding regularization is 0.005, temporal regularization is 0.01, rule regularization is 0.01, maximum epochs are 500, weight static is 0.1, learning rate is 0.1, and λ of Equation 10 is selected from [10, 2, 1, 0.5, 0.1, 0.05, 0.01]. For ICEWS05-15, the rank is 2000, embedding regularization is 0.0025, temporal regularization is 0.05, rule regularization is 1.0, maximum epochs are 500, weight static is 0.1, learning rate is 0.1, and λ Equation 10 is selected from [10, 2, 1, 0.5, 0.1, 0.05, 0.01]. For the Wikidata12k dataset, the rank is 2000, embedding regularization is 0.2, temporal regularization is 0.5, maximum epochs are 500, weight static is 0.1 and 0.07, learning rate is 0.1, and λ of Equation 10 is selected from [10, 2, 1, 0.5, 0.1, 0.01]. The Hybrid weight (w) of Equation 4 is set to 1 for all datasets.

4.4 Evaluation Metrics

The effectiveness of the proposed model is evaluated using Mean Reciprocal Rank (MRR) and Hits@k, where k $\in 1, 3, 10$, for the entity prediction task. Given a test quadruple (s, p, o, t), the object o is replaced with candidate entities e_i, and a score is computed for each candidate quadruple using the scoring function:

$$E(e_i) = E_H(s, p, e_i, t) + \lambda \cdot E_d(s, p, e_i, t) \quad (11)$$

Candidate entities are ranked based on their scores, and MRR and Hits@k are calculated as follows:

$$\text{MRR} = \frac{1}{|n|} \sum_{i=1}^{|n|} \frac{1}{rank_i} \quad (12)$$

$$\text{Hits@n} = \frac{1}{|n|} \sum i = 1^{|n|} 1(rank_i \leq n) \quad (13)$$

, where N is the total number of test instances, rank$i$ is the rank of the correct entity for the i-th test instance, and $I(\text{rank}i \leq k)$ is an indicator function. For events with time intervals in the Wikidata12k dataset, each event is converted into two events with timestamps at the interval's endpoints during training, and the score is obtained by averaging the scores of the two events during evaluation. Performance improvement is assessed by Absolute Performance Gain (APG) and Relative Performance Gain (RPG):

$$APG = P_{proposed} - P_{baseline} \quad (14)$$

$$RPG = \frac{P_{proposed} - P_{baseline}}{P_{baseline}} \times 100 \quad (15)$$

5 Result Analysis

The performance comparison of various models on the ICEWS14, ICEWS05-15, and Wikidata12k datasets is presented in Table 2. The models are evaluated using four key metrics: Mean Reciprocal Rank (MRR), Hits@10 (H@10), Hits@3 (H@3), and Hits@1 (H@1). The results highlight the effectiveness of the proposed DDP and DDPC models in capturing temporal knowledge graph information. Visualizations of the performance on each dataset are demonstrated in Appendix A. Improvements evaluated by APG and RPG methods are listed in Table 3.

5.1 Lambda Tuning Report

We conducted a hyperparameter tuning experiment to determine the optimal value of λ for each dataset. The results are presented in Figures 3, 4, and 5 for ICEWS14, ICEWS05-15, and Wikidata12k, respectively. For the ICEWS14 dataset (Figure 3), the best performance was achieved with $\lambda = 0.1$, resulting in a score of 0.805.

Models	ICEWS14				ICEWS05-15				Wikidata12k			
	MRR	H@10	H@3	H@1	MRR	H@10	H@3	H@1	MRR	H@10	H@3	H@1
TransE	0.280	0.637	-	0.094	0.294	0.663	-	0.090	0.178	0.339	0.192	0.100
DistMult	0.439	0.672	-	0.323	0.456	0.691	-	0.337	0.222	0.460	0.238	0.119
ComplEx	0.467	0.716	0.527	0.347	0.481	0.729	0.535	0.362	0.233	0.436	0.253	0.123
RotatE	0.418	0.690	0.478	0.291	0.304	0.595	0.355	0.164	0.221	0.461	0.236	0.116
QuatE	0.471	0.712	0.530	0.353	0.482	0.727	0.529	0.370	0.230	0.416	0.243	0.125
TTransE	0.255	0.601	-	0.047	0.271	0.616	-	0.085	0.172	0.329	0.185	0.096
HyTE	0.297	0.655	0.416	0.108	0.316	0.681	0.445	0.116	0.253	0.483	0.197	0.147
TeRo	0.562	0.732	0.621	0.468	0.586	0.795	0.668	0.469	0.299	0.507	0.329	0.198
ATiSE	0.550	0.750	0.629	0.436	0.519	0.794	0.606	0.378	0.252	0.462	0.288	0.148
TComplEx	0.610	0.770	0.660	0.530	0.660	0.800	0.710	0.590	0.331	0.539	0.357	0.233
TeLM	0.625	0.774	0.673	0.545	0.678	0.823	0.728	0.599	0.332	0.542	0.360	0.231
LCGE	0.667	0.815	0.714	0.588	0.730	0.866	0.776	0.655	0.429	0.677	0.495	0.304
DDP	**0.712**	**0.818**	**0.741**	**0.658**	**0.792**	**0.882**	**0.821**	**0.742**	**0.453**	**0.697**	**0.515**	**0.331**
DDPC	**0.805**	**0.885**	**0.829**	**0.762**	**0.905**	**0.950**	**0.921**	**0.879**	**0.497**	**0.712**	**0.558**	**0.3843**

Table 2: **Performance Comparison of Models on ICEWS14, ICEWS05-15, and Wikidata12k Datasets.** Metrics: Hits@1 (H1), Hits@3 (H3), and Hits@10 (H10). Results from DDP and DDPC models are listed in bold, and the previous SOTA results of LCGE are underlined.

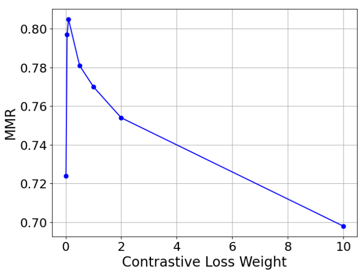

Figure 3: Performance with Different λ on ICEWS14 Dataset

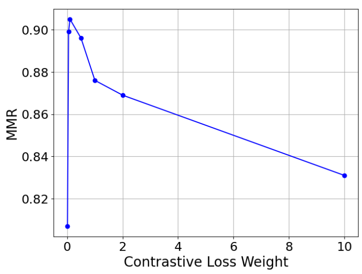

Figure 4: Performance with Different λ on ICEWS05-15 Dataset

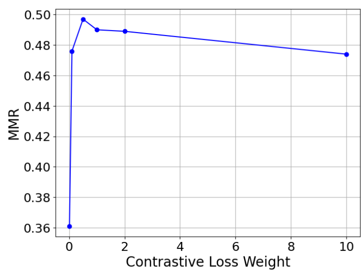

Figure 5: Performance with Different λ on Wikidata12k Dataset

The scores gradually increased as λ decreased from 10 to 0.1, but further reducing λ to 0.05 and 0.01 led to a decline in performance. Similarly, for the ICEWS05-15 dataset (Figure 4), the optimal value of λ was found to be 0.1, yielding a score of 0.905. The scores followed a similar trend as in ICEWS14, with an increase in performance as λ decreased from 10 to 0.1, followed by a slight decline when λ was further reduced to 0.05 and 0.01. In the case of the Wikidata12k dataset (Figure 5), the best score of 0.497 was obtained with $\lambda = 0.5$. The scores showed a slight improvement as λ decreased from 10 to 0.5, but setting λ to lower values such as 0.1 and 0.01 resulted in a significant drop in performance. Based on these findings, we recommend setting λ to 0.1 for both the ICEWS14 and ICEWS05-15 datasets, and to 0.5 for the Wikidata12k dataset to achieve the best performance in our experiments.

5.2 Case study on Contrastive Learning Effect

To compare the performance of DDP and DDPC, we train both models on a temporal knowledge graph dataset and evaluate their mean reciprocal rank (MRR) scores over 500 epochs on the Wikidata12k Dataset. The training progress and the resulting MRR scores are visualized in Figure 6. The red curve represents the performance of the DDP model, while the blue curve represents the performance of the DDPC model. As evident from the graph, both models exhibit a steady improvement in MRR scores as the number of epochs increases. However, the DDPC model demonstrates a faster convergence rate and consistently outperforms the DDP model throughout the training process. The perturbation-based contrastive learning mechanism employed by DDPC helps the model learn more robust and discriminative representations, leading to better performance in the temporal knowledge graph completion task.

Furthermore, the DDPC model achieves a higher final MRR score of approximately 0.497 after 500 epochs, compared to the DDP model's final MRR score of 0.453. This suggests that the incorporation of the contrastive learning mechanism in DDPC enables the model to capture more accurate and complete information from the temporal knowledge graph.

5.3 Best Results Analysis

The results analysis reveals the significant advancements achieved by the DDP and DDPC models in temporal knowledge graph completion. The DDP model, which employs a dynamic-enhanced dual perspective embed-

Metrics / Models	ICEWS14				ICEWS05-15				Wikidata12k			
	MRR	**H@1**	**H@3**	**H@10**	**MRR**	**H@1**	**H@3**	**H@10**	**MRR**	**H@1**	**H@3**	**H@10**
APG / DDP	0.045	0.003	0.027	0.070	0.062	0.016	0.045	0.087	0.024	0.020	0.020	0.027
RPG(%) / DDP	6.7	0.4	3.8	11.9	8.5	1.8	5.8	13.3	5.59	2.95	4.04	8.88
APG / DDPC	0.138	0.070	0.115	0.174	0.175	0.084	0.145	0.224	0.068	0.035	0.063	0.0803
RPG(%) / DDPC	20.7	8.6	16.1	29.6	24.0	9.7	18.7	34.2	15.8	5.2	12.7	26.4

Table 3: Performance comparison of the DDP and DDPC models using Absolute Performance Gain (APG) and Relative Performance Gain (RPG) metrics on the three datasets.

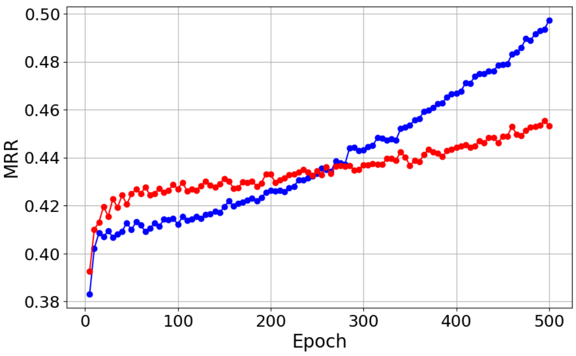

Figure 6: Comparison of DDP (in red) and DDPC (in blue) models for temporal knowledge graph completion

ding for scoring, shows notable improvements over established baseline models across all datasets. For example, in the ICEWS14 dataset, DDP achieves a MRR of 0.712, surpassing the previous state-of-the-art model LCGE, which had an MRR of 0.667. Similarly, on the ICEWS05-15 dataset, DDP records an MRR of 0.792, significantly exceeding LCGE's MRR of 0.730. The enhancements are also evident in the Hits metrics, with DDP demonstrating better performance in H@1, H@3, and H@10 across the datasets. The DDPC model, which integrates contrastive learning on top of the DDP architecture, further enhances performance, leading to the best results across all evaluated metrics and datasets.

On the ICEWS14 dataset, DDPC achieves an impressive MRR of 0.805, with H@1, H@3, and H@10 values of 0.762, 0.829, and 0.885, respectively. This trend is consistently observed in the ICEWS05-15 and Wikidata12k datasets, where DDPC sets new benchmarks with MRR values of 0.905 and 0.497, respectively, along with corresponding improvements in Hits metrics.The APG and RPG metrics further highlight the effectiveness of the DDP and DDPC models. For instance, the APG for DDPC on the ICEWS14 dataset is 0.138 for MRR and 0.174 for Hits@10, translating to RPG improvements of 20.7% and 29.6%, respectively. These gains are mirrored in the ICEWS05-15 and Wikidata12k datasets, underscoring the robustness and superiority of the DDPC model in capturing and utilizing temporal information for knowledge graph completion.

The results clearly demonstrate the superiority of the proposed DDP and DDPC models over previous state-of-the-art (SOTA) models, such as LCGE. The incorporation of dynamic temporal reasoning in the DDP and DDPC models significantly enhances their ability to capture and utilize temporal information. Additionally, the use of contrastive learning effectively improves learning performance, leading to better results across all evaluated datasets. The improvements in various metrics suggest that these models are more effective in ranking the correct entities higher and making accurate top-k predictions. Consequently, DDP and DDPC provide a more robust and comprehensive representation of temporal knowledge graphs compared to existing methods.

6 Conclusion

In this study, we introduced DDP and DDPC, novel dual-perspective learning frameworks for temporal knowledge graph completion. These models integrate static and temporal knowledge using a dual-layer embedding mechanism and a contrastive learning-enhanced version, effectively capturing dynamic changes and time-invariant properties. A perturbation learning mechanism was incorporated to enhance robustness to anomalous data and noise. Experiments on three widely used temporal knowledge graph datasets (ICEWS14, ICEWS05-15, and Wikidata12k) demonstrated the superior performance of DDP and DDPC over the previous state-of-the-art model. The dual-perspective approach, along with dynamic temporal reasoning and contrastive learning, significantly improved the models' ability to capture and utilize temporal information, leading to better performance across all evaluated metrics. DDP and DDPC contribute to advancing temporal knowledge graph completion and offer promising solutions for various downstream applications.

Limitations

The current study presents significant advancements in temporal knowledge graph completion by introducing innovative dual-layer embedding mechanisms and contrastive learning-enhanced frameworks. However, there are some limitations to consider. One limitation is that the study does not explore the potential of using large language models (LLMs) for temporal knowledge graph completion. Given the remarkable capabilities of LLMs in various natural language processing tasks, their inclusion could potentially lead to significant improvements in this domain. By not incorporating LLMs, the study leaves a gap in understanding their effectiveness compared to the proposed methods.

Another limitation is the study's primary focus on developing and evaluating specific embedding and contrastive learning techniques. While these techniques are important, the narrow focus means that other influential hyperparameters, such as learning rates, regularization parameters, and model architectures, have not been thoroughly explored. These hyperparameters can greatly impact the performance and generalizability of the models. To further optimize model performance, future research should include a comprehensive tuning of these hyper-parameters.

Lastly, although the study demonstrates the effectiveness of the proposed methods across several datasets, it does not test their generalizability to other types of temporal knowledge graphs or domains. To gain a more comprehensive understanding of the models' robustness and applicability, it would be beneficial to expand the evaluation to a broader range of datasets and applications.

As a result, while the current study makes significant contributions to temporal knowledge graph completion, future research should address the limitations by exploring the potential of LLMs, conducting comprehensive hyper-parameter tuning, and evaluating the models' generalizability across a wider range of datasets and domains.

References

Ivana Balavzevic, Carl Allen, and Timothy Hospedales. 2019. Hypernetwork knowledge graph embeddings. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 446–455. PMLR.

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*.

Xiaoyong Chen, Lise Getoor, and Benjamín Rosenfeld. 2016. Scalekb: scalable learning and inference over large knowledge bases. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 1064–1075. IEEE.

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, Brussels, Belgium. Association for Computational Linguistics.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, and Wilko Horn. 2018. Rules: Rule learning with embedding-based similarity. In *Proceedings of the 2018 ACM International Conference on Management of Data (SIGMOD)*, pages 1105–1120. ACM.

Frederic Erxleben, Markus Günther, Markus Krötzsch, Gustavo Mendez, and Denny Vrandečić. 2014. Introducing wikidata to the linked data web. In *The Semantic Web: ESWC 2014 Satellite Events*, pages 50–65.

Luis Galárraga, Christina Teflioudi, Fabian M. Suchanek, and Klaus Berberich. 2015. Fast rule mining in ontological knowledge bases with amie+. In *Proceedings of the VLDB Endowment*, volume 8, pages 1542–1553.

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

Rishab Goel, Seyed Mehran Kazemi, Marcus A Brubaker, and Pascal Poupart. 2019. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3988–3995.

Rishab Goel, Seyed Mehran Kazemi, Marcus A Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. *arXiv preprint arXiv:2008.03902*.

Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *Proceedings of the International Conference on Machine Learning*, pages 4116–4126.

Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Baobao Chang, Sujian Li, and Zhifang Sui. 2016. Towards time-aware knowledge graph completion. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1715–1724, Osaka, Japan. The COLING 2016 Organizing Committee.

Ming Jin, Yizhen Zheng, Yuan-Fang Li, Chen Gong, Chuan Zhou, and Shirui Pan. 2021. Multi-scale contrastive siamese networks for self-supervised graph representation learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. In *Proceedings of the 2020 Conference on Learning Representations (ICLR)*.

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.

Christian Meilicke, Mikhail Safavi, Steffen Staab, and Heiner Stuckenschmidt. 2019. Anytime bottom-up rule learning for knowledge graph completion. In *Proceedings of the Twenty-Eighth International Joint*

Conference on Artificial Intelligence (IJCAI), pages 3137–3143. AAAI Press.

Dat Quoc Nguyen, Thanh Tam Nguyen, and Dinh Phung Nguyen. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 327–333.

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

Peter W. Omran and Volker Tresp. 2019. Embedding logical queries on knowledge graphs. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, pages 2675–2682.

Mingyang Qu, Xuchen Yao, Guanghui Qin, and Xiang Ren. 2021. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3594–3605. Association for Computational Linguistics.

Joshua Chin Robert Speer and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*.

Ali Sadeghian, Mohammadreza Armandpour, Anthony Colas, and Daisy Zhe Wang. 2021. Chronor: Rotation based temporal knowledge graph embedding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6471–6479.

Ameen Sadeghian, Mohammadreza Armandpour, Anthony Colas, and Daisy Zhe Wang. 2019. Drum: End-to-end differentiable rule mining on knowledge graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 15321–15331.

Yaliang Shen, Xiaoxiao Ma, Jinfeng Zhang, Jiawei Su, and Xing Xie. 2020. Exploiting timeline and causality between events on knowledge graphs for storyline generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2031–2041.

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

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Remi Munos, Petar Veličković, and Michal Valko. 2021. Bootstrapped representation learning on graphs. In *Proceedings of the International Conference on Learning Representations Workshop on Geometrical and Topological Representation Learning*.

Guillaume Obozinski Timothée Lacroix and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge graph completion. In *International Conference on Learning Representations (ICLR)*.

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 of *Proceedings of Machine Learning Research*, pages 3462–3471. PMLR.

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

Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L. Hamilton. 2020. Temp: Temporal message passing for temporal knowledge graph completion.

Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z. Li. 2022. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 1070–1079, New York, NY, USA. Association for Computing Machinery.

Zhiyuan Xie, Jie Ding, Bojie Fu, Ruifeng Chen, and Sheng Chen. 2022. Latent class graph embedding for temporal knowledge graph completion. *IEEE Transactions on Knowledge and Data Engineering*.

Jiaoyan Xu, Xiao He, Yue Feng, and Carlo Zaniolo. 2021. Telm: Temporal knowledge graph completion via a linear model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Jiaoyan Xu, Kangfei Wang, Yue Feng, Zhijiang Chen, Huajun Chen, and Carlo Zaniolo. 2020a. Atise: Temporal knowledge graph completion with additive time series decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Jing Xu, Parisa Kordjamshidi, and Ryszard S. Michalski. 2020b. Temporal knowledge graph completion using a linear temporal regularizer. In *Proceedings of the 2020 Conference on Artificial Intelligence (AAAI)*.

Yanlin Xu, Chengliang Ruan, Konstantin Koroteev, Hanmeng Liu, Lingfu Ma, Yanghua Xiao, and Zhiwei Steven Wang. 2019. Temporal knowledge graph completion using a linear temporal regularizer and multivector embeddings. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3948–3954.

Yidong Xu, Yaliang Liang, Lu Zhang, Cheng Sun, and Ru Guo. 2020c. Tero: A time-aware knowledge graph embedding via temporal rotation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.

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 *International Conference on Learning Representations*.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. 2021. Graph contrastive learning automated. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. In *Proceedings of the 34th Conference on Neural Information Processing Systems*.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. In *Advances in neural information processing systems*.

Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. 2021. An empirical study of graph contrastive learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. In *Proceedings of the ICML Workshop on Graph Representation Learning and Beyond*.

A Appendix: Visualized MRR Comparsion

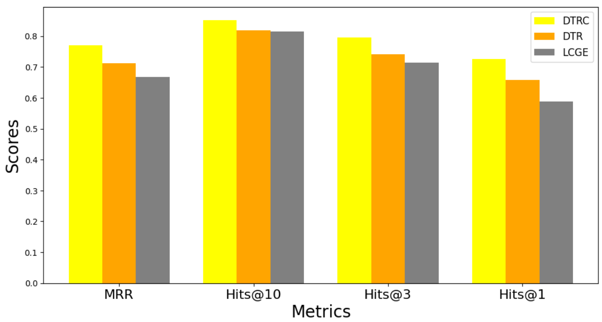

Figure 7: Performance Comparison on ICEWS14 Dataset

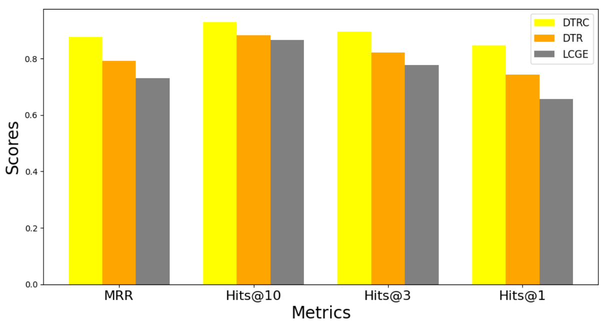

Figure 8: Performance Comparison on ICEWS05-15 Dataset

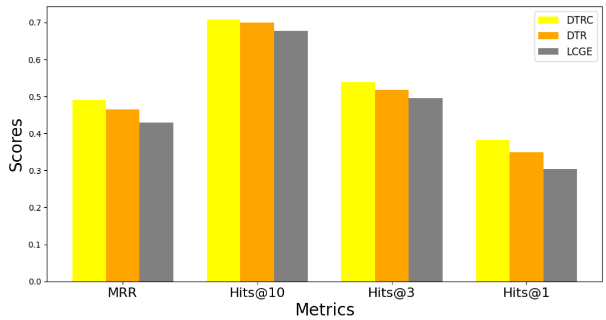

Figure 9: Performance Comparison on Wikidata12k Dataset