DECRL: A Deep Evolutionary Clustering Jointed Temporal Knowledge Graph Representation Learning Approach

Qian Chen, Ling Chen*
State Key Laboratory of Blockchain and Data Security
College of Computer Science and Technology
Zhejiang University
{qianchencs,lingchen}@cs.zju.edu.cn

Abstract

Temporal Knowledge Graph (TKG) representation learning aims to map temporal evolving entities and relations to embedded representations in a continuous low-dimensional vector space. However, existing approaches cannot capture the temporal evolution of high-order correlations in TKGs. To this end, we propose a **D**eep **E**volutionary **C**lustering jointed temporal knowledge graph **R**epresentation **L**earning approach (**DECRL**). Specifically, a deep evolutionary clustering module is proposed to capture the temporal evolution of high-order correlations among entities. Furthermore, a cluster-aware unsupervised alignment mechanism is introduced to ensure the precise one-to-one alignment of soft overlapping clusters across timestamps, thereby maintaining the temporal smoothness of clusters. In addition, an implicit correlation encoder is introduced to capture latent correlations between any pair of clusters under the guidance of a global graph. Extensive experiments on seven real-world datasets demonstrate that DECRL achieves the state-of-the-art performances, outperforming the best baseline by an average of 9.53%, 12.98%, 10.42%, and 14.68% in MRR, Hits@1, Hits@3, and Hits@10, respectively.

1 Introduction

Temporal Knowledge Graphs (TKGs) are collections of human temporal evolving knowledge [35], which are widely utilized in various fields, e.g., information retrieval [19], natural language understanding [3], and recommendation systems [32]. A TKG represents events in the form of quadruples (s, r, o, t), where s and o denote the subject and object entities, respectively, r denotes the relation between s and o, and t represents the timestamp [35]. Event prediction in TKGs is an important task that predicts future events according to historical events [17]. TKG representation learning aims to map temporal evolving entities and relations to embedded representations in a continuous low-dimensional vector space. Due to the complex temporal dynamics and multi-relations within TKGs, TKG representation learning poses great challenges to the research community.

In recent years, many TKG representation learning approaches have used graph neural networks [33] (GNNs) to model pairwise entity correlations [14, 1]. Furthermore, some researchers have leveraged derived structures, e.g., communities [39], entity groups [28], and hypergraphs [26, 29], to model high-order correlations among entities, i.e., the simultaneous correlations among three or more entities.

These approaches, however, lack the capability to capture the temporal evolution of high-order correlations in TKGs. As a kind of dynamic graph, TKGs inherently feature the temporal evolution

*Corresponding author: Ling Chen

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1752

of high-order correlations. For example, within TKGs focused solely on countries, a country entity may be affiliated with different international political organizations at different timestamps, with these organizations experiencing membership adjustment over time. In addition, the membership adjustment in international political organizations does not shift suddenly. Typically, numerous events occur before the formal adjustment of their membership, gradually pushing memberships away from or drawing them closer to international political organizations. For instance, the UK's official departure from the European Union (EU) was preceded by numerous events that incrementally estranged it from the EU.

To address the aforementioned deficiencies, a **D**eep **E**volutionary **C**lustering jointed temporal knowledge graph **R**epresentation **L**earning approach (**DECRL**) is proposed for event prediction in TKGs. To the best of our knowledge, DECRL is the first work that integrates deep evolutionary clustering approaches into TKGs, which jointly optimizes TKG representation learning with evolutionary clustering to capture the temporal evolution of high-order correlations. Our main contributions are outlined as follows:

- We propose a deep evolutionary clustering module to capture the temporal evolution of high-order correlations among entities, where clusters represent the high-order correlations between multiple entities. Furthermore, a cluster-aware unsupervised alignment mechanism is introduced to ensure precise one-to-one alignment of soft overlapping clusters across timestamps, maintaining the temporal smoothness of clusters over successive timestamps.
- We propose an implicit correlation encoder to capture latent correlations between any pair of clusters, which defines the interaction intensities between clusters to form a cluster graph. In addition, a global graph, constructed from all events of training set, is introduced to guide the assignment of different interaction intensities to different cluster pairs.
- We evaluate DECRL on seven real-world datasets. The experimental results demonstrate that DECRL achieves the state-of-the-art (SOTA) performance. It outperforms the best baseline by an average of 9.53%, 12.98%, 10.42%, and 14.68% in MRR, Hits@1, Hits@3, and Hits@10, respectively.

2 Related work

2.1 TKG representation learning approach

GNNs and variants of recurrent neural networks (RNNs) are commonly integrated to model graph structural information and temporal dependency within TKGs, respectively [1, 34, 16, 14]. For example, TeMP [34] and RE-GCN [16] both utilize relation-aware GCN [24] to model the influence of neighbor entities and apply GRUs to model the temporal dependency. TiRGN [14] employs multi-relational GNNs to model graph structural information, and utilizes GRUs to capture the temporal dependency across sequential timestamps. However, these approaches often resort to stacking multiple layers to model the influence of distant neighbors, which may lead to the over-smoothing problem.

Recent research advancements have introduced paths [15, 8, 18] to enhance the modeling capability of the latent pairwise correlations between entities. For example, xERTE [8] utilizes a time-aware graph attention mechanism to obtain and exploit local subgraphs. TLogic [18] employs a time-based random walk strategy to acquire logical paths. Furthermore, some researchers have leveraged derived structures, e.g., communities [39], entity groups [28], and hypergraphs [29], to model high-order correlations among entities. For example, EvoExplore [39] utilizes a temporal point process enhanced by a hierarchical attention mechanism to establish dynamic communities for modeling latent correlations among entities. DHyper [29] utilizes hypergraph neural networks to model high-order correlations among entities and among relations.

While the promising results of introducing derived structures, existing approaches lack the capability to capture the temporal evolution of high-order correlations in TKGs. To address this gap, we propose DECRL, which is the first work to integrate deep evolutionary clustering approaches into TKGs. DECRL jointly optimizes TKG representation learning with evolutionary clustering to capture the temporal evolution of high-order correlations.

2.2 Deep evolutionary clustering approach

Existing deep evolutionary clustering approaches employ different strategies for discovering stable clusters [7, 21, 36, 20, 38, 37]. For example, DYNMOGA [7] employs a deep evolutionary clustering approach designed to identify evolving communities within dynamic networks, which treats the process as a multi-objective optimization problem aimed at cluster stability. sE-NMF [21] implements linear fusion of adjacency matrices across timestamps to enhance the temporal stability of clusters. Both DNETC [36] and CDNE [20] use constructed temporal smoothness loss functions to ensure a consistent and stable evolution of clustering results through successive timestamps. TRNNGCN [37] introduces a decay matrix to assess the influence of historical clustering results, thereby stabilizing the current clustering configuration.

These approaches straightforwardly incorporate prior clustering results with current timestamp results over time under the presumption that clustering results are relatively stable across different timestamps. However, the nuances between similar clusters are difficult to recognize in this way. In TKG representation learning, it is necessary to discern the subtle differences in high-order correlations. To bridge this gap, in our work, a cluster-aware unsupervised alignment mechanism is introduced, which ensures the precise one-to-one alignment of soft overlapping clusters across timestamps, thereby maintaining the temporal smoothness of clusters over successive timestamps.

3 Preliminaries

Definition 1 (TKG). A TKG is a set of events formalized as $\mathcal{G} = \{(s, r, o, t) \mid s, o \in \mathcal{E}, r \in \mathcal{R}, t \in \mathcal{T}\}$, where $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ denote the set of entities, relations, and timestamps, respectively. $\mathcal{G}^t$ denotes the set of events at timestamp t.

Definition 2 (Entity Graph). The entity graph is constructed based on $\mathcal{G}^t$, which is a multi-relational graph, and can be denoted as $G_{\mathrm{e}}^t = (V_{\mathrm{e}}^t, E_{\mathrm{e}}^t)$, where V_{e}^t and E_{e}^t denote the set of nodes and edges of the entity graph at timestamp t, respectively. The nodes in V_{e}^t represent entities, while the edges in E_{e}^t represent relations between entities at timestamp t.

Definition 3 (Cluster Graph). The cluster graph is constructed based on G_{e}^t, which is a fully connected graph, and can be denoted as $G_{\mathrm{c}}^t = (V_{\mathrm{c}}^t, E_{\mathrm{c}}^t)$, where V_{c}^t and E_{c}^t denote the set of nodes and edges of the cluster graph at timestamp t, respectively. The nodes and edges of cluster graphs represent clusters and the latent correlations between them, respectively, where each cluster represents the high-order correlations among entities.

Task (Event Prediction). Given a query $(s, ?, o, t)$, the event prediction task in TKGs aims to predict the conditional probability of all relations with the subject entity s, the object entity o, and the historical event sets $\mathcal{G}^{1:T-1}$, which can be denoted as $p(\hat{\boldsymbol{r}}|s, o, \mathcal{G}^{1:T-1})$, where T denotes the number of historical timestamps.

4 Methodology

4.1 Overview

The proposed DECRL approach is illustrated in Figure 1. At each timestamp, entity and relation representations updated by the cluster graph message passing module are merged with input representations from the previous timestamp using a time residual gate, serving as the input for the current timestamp. The multi-relational interactions among entities are modeled by the relation-aware GCN. The deep evolutionary clustering module captures the temporal evolution of high-order correlations among entities, maintaining the temporal smoothness of clusters over successive timestamps through an unsupervised alignment mechanism. The implicit correlation encoder captures the latent correlations between any pair of clusters, enabling message passing within the cluster graph to update entity representations. Finally, the attentive temporal encoder captures temporal dependencies among entity and relation representations across timestamps, integrating them with temporal information for future event prediction. The computational complexity of DECRL can be found in Appendix A.1.

4.2 Evolutionary clustering module

The entity graph $G_{\mathrm{e}}^t = (V_{\mathrm{e}}^t, E_{\mathrm{e}}^t)$ is constructed based on the historical events at timestamp t, which can be used to capture the structural dependency among concurrent events. Since the entity graph

https://doi.org/10.52202/079017-1752

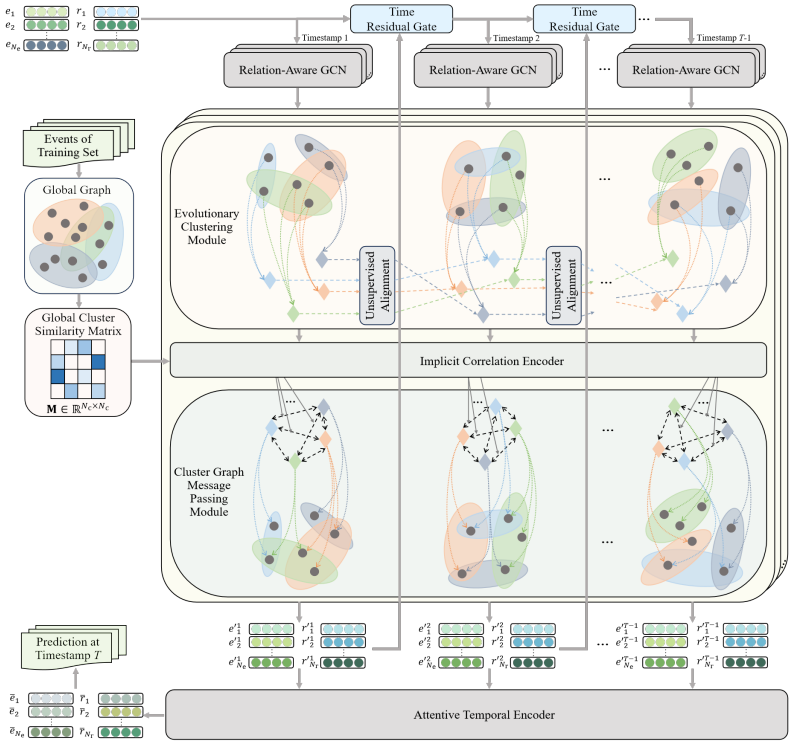


Figure 1: The framework of DECRL.

is a multi-relational graph, relation-aware GCN [24] is utilized to model the structural dependency, which is formulated as:

$$\boldsymbol{h}_{\mathrm{o}}^{t,l+1} = \mathrm{RReLU}\left(\frac{1}{d_{\mathrm{o}}} \sum_{(s,r),\exists(s,r,o)\in\mathcal{G}^t} \boldsymbol{W}_1(\boldsymbol{h}_{\mathrm{s}}^{t,l} + \boldsymbol{h}_{\mathrm{r}}^{t,l}) + \boldsymbol{W}_2 \boldsymbol{h}_{\mathrm{o}}^{t,l}\right) \tag{1}$$

where $\boldsymbol{h}_{\mathrm{s}}^{t,l}$, $\boldsymbol{h}_{\mathrm{o}}^{t,l}$, and $\boldsymbol{h}_{\mathrm{r}}^{t,l}$ denote the l^{th} layer representations of subject entity s, object entity o, and relation r at timestamp t, respectively. $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are learnable parameters for aggregating neighbor entity representations and self-loop, respectively. d_{o} denotes the in-degree of object entity o. Initially, entities are assigned random representations based on their in-degree, such that entities with the same in-degree have the same initial representations, thereby efficiently accelerating the update process of entity representations. Relations are given with the random initial representations. For simplicity, the subscript l of variables is omitted in the following sections without causing ambiguity.

The representation of the relation r at timestamp t is influenced by the r-related entity representations at timestamp t and its representation at timestamp $t-1$. The updating of the relation representation is formulated as $\boldsymbol{h}_{\mathrm{r}}^{t} = \mathrm{pooling}[\boldsymbol{e}_{\mathcal{V}_{\mathrm{r}}^t}^{t}; \boldsymbol{h}_{\mathrm{r}}^{t-1}]$, where $[;]$ denotes the concatenation operation. pooling denotes the mean pooling operation. $\boldsymbol{e}_{\mathcal{V}_{\mathrm{r}}^t}^{t}$ denotes r-related entity representations at timestamp t, where $\mathcal{V}_{\mathrm{r}}^t = \{i|(i,r,o,t)\ or\ (s,r,i,t) \in \mathcal{G}^t\}$.

A country entity may be affiliated with different international political organizations at different timestamps within TKGs. Thus, the fuzzy c-means clustering method [23] is utilized to obtain the soft overlapping clusters, which outputs the membership matrix between entities and clusters by optimizing the object function as follows:

$$J = \sum_{i=1}^{N_{\mathrm{e}}} \sum_{j=1}^{N_{\mathrm{c}}} u_{i,j}^{t,m} \left(1 - \frac{\langle \boldsymbol{e}_i^t, \boldsymbol{\mu}_j^t \rangle}{\|\boldsymbol{e}_i^t\| \|\boldsymbol{\mu}_j^t\|}\right), s.t. \sum_{j=1}^{N_{\mathrm{c}}} u_{i,j}^{t,m} = 1, 0 < \sum_{i=1}^{N_{\mathrm{e}}} u_{i,j}^{t,m} < N_{\mathrm{e}} \tag{2}$$

where N_{e} and N_{c} are the number of entities and clusters, respectively. $u_{i,j}^t$ denotes the membership degree of the entity representation $\boldsymbol{e}_i^t$ to the cluster centroid $\boldsymbol{\mu}_j^t$ at timestamp t. m denotes the fuzzy

smoothing hyper-parameter, which is typically set to a value greater than 1. $\langle\cdot\rangle$ denotes the dot product. $\|\cdot\|$ denotes the norm of a vector. Then, the representation of clusters is formulated as:

$$\boldsymbol{c}_j^t = \sum_{\boldsymbol{e}_i^t \in V_{\mathrm{e}}^t} u_{i,j}^t \boldsymbol{e}_i^t, \quad u_{i,j}^t = \frac{\left(\frac{\langle \boldsymbol{e}_i^t, \boldsymbol{\mu}_j^t \rangle}{\|\boldsymbol{e}_i^t\| \|\boldsymbol{\mu}_j^t\|}\right)^{\frac{1}{m-1}}}{\sum_{k=1}^{N_{\mathrm{c}}} \left(\frac{\langle \boldsymbol{e}_i^t, \boldsymbol{\mu}_k^t \rangle}{\|\boldsymbol{e}_i^t\| \|\boldsymbol{\mu}_k^t\|}\right)^{\frac{1}{m-1}}} \tag{3}$$

where $\boldsymbol{c}_j^t$ denotes the representation of cluster j at timestamp t, weighted by the membership degree of different entities to the cluster.

The maximum weight matching can be found in polynomial time using the Hungarian algorithm [11]. Herein, a cluster-aware Hungarian matching algorithm is introduced to ensure a smooth one-to-one alignment of clusters across timestamps. The cluster-aware Hungarian matching algorithm quantifies the similarity between clusters of consecutive timestamps, i.e., $t-1$ and t, which creates an affinity matrix A, where each element $a_{j,k}$ represents the similarity between the cluster j at timestamp $t-1$ and the cluster k at timestamp t:

$$a_{j,k} = \cos(\boldsymbol{c}_j^{t-1}, \boldsymbol{c}_k^t) \tag{4}$$

With the affinity matrix established, the cluster-aware Hungarian algorithm seeks an optimal assignment that maximizes the sum of similarities across matched clusters. It is formulated as an optimization problem: $\max_\pi \sum_{j=1}^{N_{\mathrm{c}}} a_{j,\pi(j)}$, where π is the permutation function representing the one-to-one alignment between clusters of consecutive timestamps. Therefore, $\pi(j)$ denotes the index of the cluster at timestamp t matched with the cluster j at timestamp $t-1$.

Subsequently, the fused cluster representations are computed by taking a weighted combination of the cluster representations at consecutive timestamps, which is formulated as:

$$\boldsymbol{c}_j^t = \beta \boldsymbol{c}_j^{t-1} + (1-\beta)\boldsymbol{c}_{\pi(j)}^t \tag{5}$$

where β is the fusion weight, which indicates the relative contribution of each cluster representation at consecutive timestamps to the fused cluster representation.

What's more, it is essential to ensure the temporal smoothness of these cluster representations across consecutive timestamps. To this end, a temporal smoothness loss is introduced to penalize significant deviations of cluster representations across consecutive timestamps, which is formulated as:

$$\mathcal{L}_{\mathrm{temporal}} = \sum_{t=1}^{T-1} \sum_{j=1}^{N_{\mathrm{c}}} \left(1 - \frac{\langle \boldsymbol{c}_j^{t-1}, \boldsymbol{c}_j^t \rangle}{\|\boldsymbol{c}_j^{t-1}\| \|\boldsymbol{c}_j^t\|}\right) \tag{6}$$

4.3 Cluster graph message passing module

Due to the intricate relationships and alliances formed among international political organizations, it is crucial to capture the latent correlations between clusters for event prediction. Herein, an implicit correlation encoder is proposed to capture the latent correlations between any pair of clusters. The cluster graph is designed to be a fully connected graph to avoid missing any relevant information between clusters. The representations of the latent correlations are formulated as:

$$\boldsymbol{s}_{i,j}^t = \mathrm{ReLU}\left(\varphi(\boldsymbol{c}_i^t, \boldsymbol{c}_j^t)\right) \tag{7}$$

where $\boldsymbol{c}_i^t$ and $\boldsymbol{c}_j^t$ are representations of the cluster i and j at timestamp t, respectively. φ is the transformation function, which is implemented by the multi-layer perceptron.

It is worth noting that clusters exhibit varying degrees of interaction, characterized by different intensities of latent correlations. Thus, we quantify the intensity of latent correlations between clusters, which is formulated as:

$$q_{i,j}^t = \sigma\left(\mathrm{Conv}(\boldsymbol{s}_{i,j}^t)\right) \tag{8}$$

where σ is the sigmoid function, which ensures the resulting correlation intensity bounded between 0 and 1. $\mathrm{Conv}(\cdot)$ represents the convolution operation.

The intensity of the latent correlations between clusters varies in the short term and intensifies over time; a higher frequency of interactions indicates a stronger latent correlation. To this end, we

 https://doi.org/10.52202/079017-1752

construct a global graph, which is a static graph composed of all events from the training set. The spectral clustering approach is then applied to this global graph to obtain global clusters $\boldsymbol{c}_i^{\text{global}}$. The similarity between global clusters is used to enhance the intensity of latent correlations between cluster pairs, which is formulated as:

$$\widehat{q}_{i,j}^t = q_{i,j}^t \times m_{i,j}^t, \quad m_{i,j}^t = \frac{\langle \boldsymbol{c}_{\pi^t(i)}^{\text{global}}, \boldsymbol{c}_{\pi^t(j)}^{\text{global}} \rangle}{\|\boldsymbol{c}_{\pi^t(i)}^{\text{global}}\| \|\boldsymbol{c}_{\pi^t(j)}^{\text{global}}\|} \tag{9}$$

where $\boldsymbol{c}_{\pi^t(i)}^{\text{global}}$ and $\boldsymbol{c}_{\pi^t(j)}^{\text{global}}$ are global clusters matched with the cluster i and j at timestamp t, respectively. $m_{i,j}^t$ is the similarity between the global cluster i and j at timestamp t. $\widehat{q}_{i,j}^t$ is the enhanced intensity of latent correlations, which also integrates structural information from the global graph.

In the message passing process of the cluster graph, the stronger the intensity of the latent correlation, the more critical the latent correlation is, and vice versa. The aggregation operation is formulated as:

$$\boldsymbol{v}_{c_i}^t = \sum_{i \neq j} \widehat{q}_{i,j}^t \boldsymbol{s}_{i,j}^t \tag{10}$$

where $\widehat{q}_{i,j}^t$ and $\boldsymbol{s}_{i,j}^t$ denote the enhanced intensity and the representation of latent correlation calculated by Equation 9 and Equation 7, respectively. Then, the representation of clusters is updated through:

$$\widehat{\boldsymbol{c}}_i^t = \varphi(\boldsymbol{v}_{c_i}^t, \boldsymbol{c}_i^t) \tag{11}$$

To implement information transfer from clusters to individual entities, we use the membership matrix from Section 4.2. Entity representations are updated based on their associated cluster representations, which are formulated as $\boldsymbol{e}_i'^t = \sum_{j=1}^{N_c} \boldsymbol{u}_{i,j}^t \widehat{\boldsymbol{c}}_j^t$, where N_c denotes the number of clusters.

4.4 Time residual gate

The time residual gate is introduced to combine updated entity and relation representations with input representations of the current timestamp through a weighted mechanism. This approach preserves inherent characteristics while capturing the temporal evolution of entities and relations. For simplicity, the subscripts i and j of variables are omitted in the following sections without causing ambiguity. The final updated representations at timestamp t are formulated as:

$$\boldsymbol{H}^t = \boldsymbol{X}^t \otimes \boldsymbol{H}_\theta^t + (1 - \boldsymbol{X}^t) \otimes \boldsymbol{H}^{t-1}, \quad \boldsymbol{X}^t = \sigma(\boldsymbol{W}_3 \boldsymbol{H}^{t-1} + \boldsymbol{b}) \tag{12}$$

where $\boldsymbol{H}_\theta^t$ denotes the entity or relation representations updated by the cluster graph message passing module at timestamp t. $\boldsymbol{H}^{t-1}$ denotes the output of entity or relation representations at timestamp $t-1$, and also is the input of timestamp t. The time residual gate, i.e., $\boldsymbol{X}^t$, determines the inherent characteristics to be preserved, where σ is the sigmoid function. $\boldsymbol{W}_3$ and $\boldsymbol{b}$ are learnable parameters.

4.5 Attentive temporal encoder

In the context of sequence modeling, the attentive temporal encoder is introduced to capture the temporal dependency among the final updated representations across timestamps. The position-enhanced representations of entity and relation are formulated as:

$$\boldsymbol{z}^t = [\boldsymbol{H}^t; \Phi(t)], \quad \Phi(t) = \sqrt{\frac{1}{d}}[\cos(\omega_1 \tau^t), \cos(\omega_2 \tau^t), ..., \cos(\omega_d \tau^t)] \tag{13}$$

where $\boldsymbol{H}^t$ is the final updated representation of entity or relation at timestamp t. $[;]$ denotes the concatenation operation. $\Phi(t)$ is the time position encoder. d denotes the value of the representation dimension. ω_1 to ω_d are learnable parameters. τ^t is the timestamp. Then, the temporal dependency is captured by a position-enhanced self-attention mechanism based on representation sequence $\boldsymbol{z}^{1:T-1} = \{\boldsymbol{z}^1, \boldsymbol{z}^2, ..., \boldsymbol{z}^{T-1}\}$. For simplicity, the subscript t of variables is omitted in the following sections. The integrated entity or relation representation is formulated as:

$$\beta_{m,n} = \frac{\langle \boldsymbol{W}_q \boldsymbol{z}_m, \boldsymbol{W}_k \boldsymbol{z}_n \rangle}{\sqrt{d}}, \quad \alpha_{m,n} = \frac{\exp(\beta_{m,n})}{\sum_{i=1}^{T-1} \exp(\beta_{m,i})}, \qquad \overline{\boldsymbol{h}}_m = \sum_{n=1}^{T-1} \alpha_{m,n} \boldsymbol{z}_{m,n} \tag{14}$$

where $\boldsymbol{W}_q$ and $\boldsymbol{W}_k$ are learnable parameters. $\alpha_{m,n}$ is the attention weight. $\overline{\boldsymbol{h}}_m$ is the integrated representation of entity or relation, which integrates the temporal information.

4.6 Event prediction

ConvTransE [25] is employed as the decoder of DECRL, which predicts the probability of each relation between an entity pair, which is formulated as:

$$p(\hat{\boldsymbol{r}}|s, o, \mathcal{G}^{1:T-1}) = \sigma(\boldsymbol{H}_{\mathrm{r}}\mathrm{ConvTransE}(\bar{\boldsymbol{e}}_{\mathrm{s}}, \bar{\boldsymbol{e}}_{\mathrm{o}})) \tag{15}$$

where $\hat{\boldsymbol{r}}$ is the probability vector of relations. σ is the sigmoid function. $\boldsymbol{H}_{\mathrm{r}}$ is the relation representation matrix, each row of which corresponds to an integrated relation representation $\overline{\boldsymbol{r}}$. $\mathrm{ConvTransE}(\cdot)$ contains a one-dimensional convolution layer and a fully connected layer.

The event prediction training objective is minimizing the cross-entropy loss, which is formulated as:

$$\mathcal{L}_{\mathrm{TKG}} = -\frac{1}{S}\sum_{i=1}^{S}\sum_{j=1}^{N_{\mathrm{r}}}\{y_{i,j}\log p_{i,j} + (1-y_{i,j})\log(1-p_{i,j})\} \tag{16}$$

where S and N_{r} denote the numbers of samples in the training set and relations, respectively. $y_{i,j}$ denotes the label of relation j for sample i, of which the element is 1 if the event occurs, otherwise 0. $p_{i,j}$ is the predicted probability of relation j for sample i, calculated by Equation 15.

Finally, the total loss for DECRL is formulated as:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{\mathrm{TKG}} + \lambda\mathcal{L}_{\mathrm{temporal}} \tag{17}$$

where λ is a hyper-parameter that controls the trade-off between event prediction and temporal smoothness, which is bounded between 0 and 1.

5 Experiments

In this section, we evaluate the performance of DECRL through comprehensive experiments. We first describe the datasets and experimental settings, followed by a comparison with 12 SOTA TKG representation learning approaches. Next, we present an ablation study and hyper-parameter sensitivity analysis (see Appendix A.5). Finally, we offer case studies showcasing the effectiveness of DECRL.

5.1 Datasets and experimental settings

We evaluate DECRL on seven real-world datasets: ICEWS14 [30], ICEWS18 [9], ICEWS14C [29], ICEWS18C [29], GDELT [13], WIKI [12], and YAGO [9]. The ICEWS14 and ICEWS18 datasets span January 1, 2014, to December 31, 2014, and January 1, 2018, to October 31, 2018, respectively. We derive ICEWS14C and ICEWS18C datasets by filtering the original datasets to focus exclusively on events involving countries. The GDELT dataset spans January 1, 2018, to January 31, 2018. The WIKI and YAGO datasets are subsets of the Wikipedia history and YAGO3 [22], respectively. Dataset statistics and experimental settings are detailed in Appendices A.2 and A.3.

5.2 Comparison with baselines

DECRL is compared against 12 SOTA TKG representation learning approaches, categorized into shallow encoder-based approaches, i.e., TTransE [12] and HyTE [5], DNN-based approaches, i.e., RE-NET [9], Glean [6], TeMP [34], RE-GCN [16], DACHA [4], and TiRGN [14], and derived structure-based approaches, i.e., TITer [27], EvoExplore [39], GTRL [28], and DHyper [29]. All baselines are evaluated following consistent experimental settings with well-tuned hyper-parameters. The detailed descriptions of compared baselines can be found in Appendix A.4.

Tables 1, 2, and 3 display the MRR and Hits@1/3/10 results of event prediction on ICEWS14, ICEWS18, ICEWS14C, ICEWS18C, and GDELT datasets. "*" denotes the statistical superiority of DECRL over compared approaches based on pairwise t-tests at a 95% confidence level. The best results are highlighted in bold, while the second-best results are underlined. Notably, DECRL consistently exhibits superior performance, yielding average improvements of 13.24%, 12.98%, 10.42%, and 14.68% in MRR and Hits@1/3/10 metrics, respectively. These findings affirm the efficacy of integrating deep evolutionary clustering for capturing the temporal evolution of high-order correlations among entities. Furthermore, it is evident that approaches leveraging derived structures

https://doi.org/10.52202/079017-1752

Table 1: The performance of DECRL and the compared approaches on ICEWS14 and ICEWS14C

	ICEWS14				ICEWS14C			
Approach	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TTransE (WWW 2018)	23.79*	14.24*	29.17*	34.56*	11.79*	13.24*	19.97*	24.88*
HyTE (EMNLP 2018)	25.12*	18.15*	30.15*	45.37*	22.17*	18.15*	27.28*	35.37*
RE-NET (EMNLP 2020)	45.77*	37.98*	49.07*	58.87*	43.27*	36.97*	47.08*	55.19*
Glean (KDD 2020)	42.20*	36.86*	47.68*	52.39*	40.24*	34.62*	45.48*	50.09*
TeMP (EMNLP 2020)	46.04*	39.07*	49.84*	59.74*	44.17*	37.37*	47.78*	55.66*
RE-GCN (SIGIR 2021)	45.56*	38.09*	50.37*	62.44*	41.76*	36.67*	45.37*	51.74*
DACHA (TKDD 2022)	45.44*	37.88*	49.47*	58.69*	44.26*	37.59*	44.18*	53.19*
TiRGN (IJCAI 2022)	46.07*	39.83*	52.17*	63.95*	44.73*	38.13*	49.77*	60.91*
TITer (EMNLP 2021)	46.12*	39.08*	50.76*	60.39*	44.86*	39.37*	48.84*	55.79*
EvoExplore (KBS 2022)	47.71*	40.68*	52.37*	65.94*	49.77*	40.12*	54.37*	65.83*
GTRL (TKDE 2023)	46.25*	40.11*	51.09*	65.79*	50.95*	40.31*	52.09*	64.89*
DHyper (TOIS 2024)	56.15*	43.76*	65.46*	85.89*	54.16*	41.45*	62.03*	75.35*
DECRL	**62.61**	**48.73**	**70.57**	**93.03**	**58.55**	**44.62**	**66.52**	**82.06**
Improvement	**11.50%**	**11.36%**	**7.81%**	**8.31%**	**8.11%**	**7.65%**	**7.24%**	**8.91%**

Table 2: The performance of DECRL and the compared approaches on ICEWS18 and ICEWS18C

	ICEWS18				ICEWS18C			
Approach	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TTransE (WWW 2018)	11.96*	13.97*	12.79*	24.33*	9.84*	10.29*	11.04*	18.89*
HyTE (EMNLP 2018)	21.85*	16.86*	25.64*	41.86*	22.23*	16.27*	25.68*	33.39*
RE-NET (EMNLP 2020)	42.25*	33.81*	44.98*	52.72*	41.05*	32.87*	42.78*	50.43*
Glean (KDD 2020)	37.11*	34.15*	42.56*	47.35*	35.58*	32.26*	40.44*	46.49*
TeMP (EMNLP 2020)	43.24*	38.77*	45.04*	55.94*	43.08*	36.07*	43.18*	53.03*
RE-GCN (SIGIR 2021)	41.56*	37.59*	44.34*	57.42*	40.27*	36.35*	41.75*	49.25*
DACHA (TKDD 2022)	43.87*	37.11*	47.47*	57.69*	40.11*	36.11*	46.17*	52.37*
TiRGN (IJCAI 2022)	44.27*	38.13*	50.66*	62.90*	43.57*	37.23*	47.67*	54.44*
TITer (EMNLP 2021)	45.44*	39.78*	48.77*	58.73*	44.07*	38.85*	46.44*	49.79*
EvoExplore (KBS 2022)	46.65*	40.05*	50.07*	58.35*	47.33*	38.96*	49.37*	56.15*
GTRL (TKDE 2023)	46.35*	40.95*	51.19*	60.18*	49.33*	40.15*	53.39*	60.74*
DHyper (TOIS 2024)	54.22*	42.16*	63.26*	75.38*	52.11*	41.04*	60.03*	73.22*
DECRL	**63.30**	**50.13**	**70.72**	**90.82**	**61.37**	**46.28**	**67.01**	**86.79**
Improvement	**16.75%**	**18.90%**	**11.79%**	**20.48%**	**17.77%**	**12.76%**	**11.63%**	**18.53%**

Table 3: The performance of DECRL and the compared approaches on GDELT. "OOM" and "TLE" indicate out of memory and a single epoch exceeded 24 hours

Approach	MRR	Hits@1	Hits@3	Hits@10
TTransE (WWW 2018)	8.62*	7.73*	11.03*	23.34*
HyTE (EMNLP 2018)	10.63*	8.39*	14.23*	28.79*
RE-NET (EMNLP 2020)	17.55*	11.73*	18.14*	35.52*
Glean (KDD 2020)	15.60*	10.35*	17.61*	37.40*
TeMP (EMNLP 2020)	19.19*	11.07*	19.84*	40.52*
RE-GCN (SIGIR 2021)	20.84*	10.80*	21.09*	43.65*
DACHA (TKDD 2022)	21.91*	11.27*	17.49*	47.13*
TiRGN (IJCAI 2022)	24.61*	13.78*	25.66*	49.02*
TITer (EMNLP 2021)	TLE	TLE	TLE	TLE
EvoExplore (KBS 2022)	18.53*	10.74*	19.45*	42.07*
GTRL (TKDE 2023)	22.44*	12.48*	18.03*	50.82*
DHyper (TOIS 2024)	OOM	OOM	OOM	OOM
DECRL	**27.58**	**15.74**	**29.16**	**59.54**
Improvement	**12.07%**	**14.22%**	**13.64%**	**17.16%**

Table 4: The performance of DECRL and the compared approaches on WIKI and YAGO with MRR

Approach	WIKI	YAGO
TiRGN (IJCAI 2022)	99.04	99.30
DHyper (TOIS 2024)	99.38	99.31
DECRL	**99.67**	**99.56**
Improvement	**0.29%**	**0.25%**

Table 5: The performance of DECRL and the variants on ICEWS14

Method	MRR	Hits@1	Hits@3	Hits@10
DECRL-w/o-alignment	59.16 (-3.5)	43.92 (-4.8)	67.57 (-3.0)	92.31 (-0.7)
DECRL-w/o-fusion	57.98 (-4.3)	41.90 (-6.8)	66.97 (-3.6)	92.00 (-1.0)
DECRL-w/o-ICE	60.53 (-2.8)	46.33 (-2.4)	68.29 (-2.3)	91.14 (-1.9)
DECRL-w/o-global-graph	58.67 (-3.9)	43.45 (-5.3)	69.46 (-1.1)	91.07 (-2.0)
DECRL-w/o-$\mathcal{L}_{\text{temporal}}$	58.74 (-3.9)	44.43 (-4.3)	65.33 (-5.2)	89.85 (-3.2)
DECRL	**62.61**	**48.73**	**70.57**	**93.03**

generally outshine those relying on DNNs, which in turn surpass approaches employing shallow encoders.

Since most approaches do not report event prediction results on WIKI and YAGO datasets, we compare DECRL with TiRGN and DHyper, both of which provide results on these datasets, as shown in Table 4. Furthermore, TiRGN and DHyper are the SOTA DNN-based approach and the SOTA derived structure-based approach, respectively. As a result, DECRL shows improvements of 0.29% and 0.25% over the runner-up, DHyper, on WIKI and YAGO datasets, respectively, underscoring its effectiveness. Detailed entity prediction results are presented in Appendix A.6 due to space limitation.

5.3 Ablation study

To dissect the contributions of each module within DECRL, we conduct the ablation study on ICEWS14. The detailed descriptions of variants are as follows:

- **DECRL-w/o-alignment**: Removing unsupervised alignment mechanism.
- **DECRL-w/o-fusion**: Removing fusion operation between clusters across timestamps.
- **DECRL-w/o-ICE**: Removing implicit correlation encoder, resulting in a fully connected graph where all edges are assigned uniform weights.
- **DECRL-w/o-global-graph**: Removing the guidance of the global graph on the assignment of different interaction intensities to different cluster pairs.
- **DECRL-w/o-$\mathcal{L}_{\text{temporal}}$**: Removing the temporal smoothness loss term.

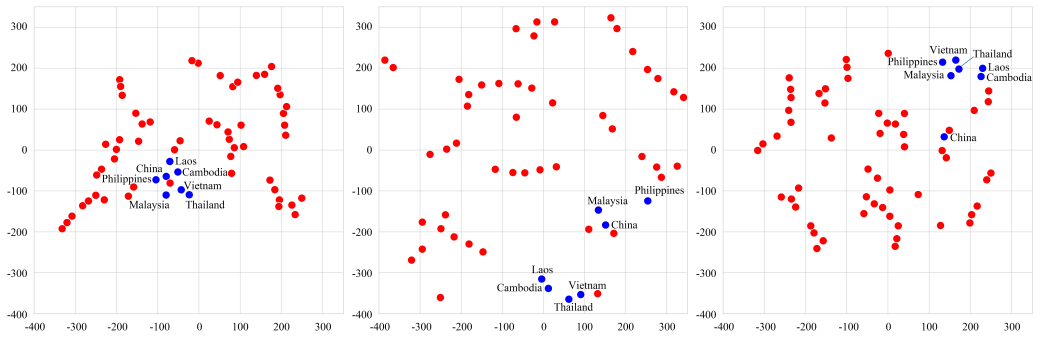
Table 5 displays the MRR and Hits@1/3/10 results of DECRL and the variants on ICEWS14. From the ablation study, several conclusions can be drawn: Firstly, the most substantial performance degradation is observed in the DECRL-w/o-fusion variant, which indicates the effectiveness of capturing the temporal evolution of high-order correlations among entities. Moreover, the performance degradation of the DECRL-w/o-global-graph variant justifies the effectiveness of leveraging structural information from the global graph for guiding the assignment of different interaction intensities to different cluster pairs. Lastly, the performance degradation of DECRL-w/o-alignment and DECRL-w/o-$\mathcal{L}_{\text{temporal}}$ variants shows the importance of preserving the temporal smoothness of clusters over successive timestamps.

5.4 Case study

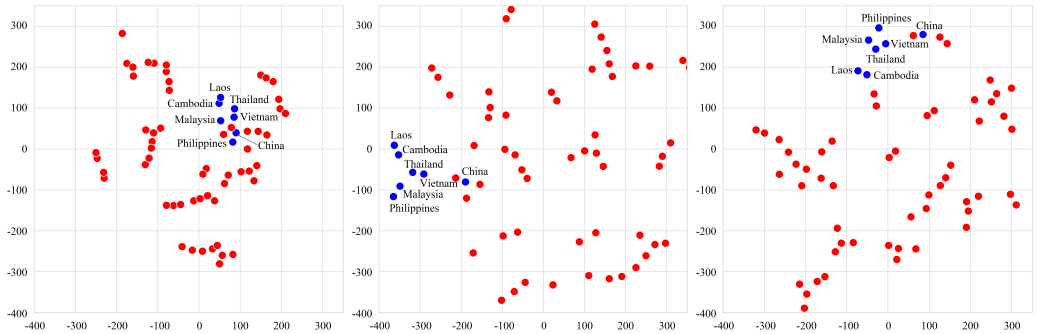
To demonstrate the efficacy of DECRL, we conducted a case study with two variants: DECRL-w/o-alignment and DECRL-w/o-fusion, both of which are elaborated in Sub-section 5.3. In Figure 2, we utilize t-SNE [31] for the visualization of entity representations on ICEWS14C, where red dots represent individual entities in the TKGs, with their groupings indicating entity clusters. Notably, the entities marked in Figure 2, i.e., China, Thailand, Vietnam, Laos, Malaysia, Philippines, and Cambodia, are countries in East Asia participating in the Belt and Road Initiative. "Middle" and "Final" denote the entity representations obtained after training at the penultimate epoch and the final epoch, respectively, for different variants.

By comparing the visualizations in the first row of sub-figures in Figure 2, we observe that DECRL exhibits the best entity clustering phenomena during the mid-training phase. By comparing the visualizations in the second row of sub-figures in Figure 2, Figure 2d (Final DECRL) showcases pronounced clustering phenomena, and inter-cluster distances indicate significant differences in

https://doi.org/10.52202/079017-1752



(a) Middle DECRL (b) Middle DECRL-w/o-alignment (c) Middle DECRL-w/o-fusion



(d) Final DECRL (e) Final DECRL-w/o-alignment (f) Final DECRL-w/o-fusion

Figure 2: The visualization of entity representations on ICEWS14C.

clusters, i.e., international political organizations. Figure 2e (Final DECRL-w/o-alignment) demonstrates moderate clustering phenomena among countries, but inter-cluster distances are not significant enough. This observation implies that the lack of precise one-to-one alignment may lead to proximity between different clusters. Figure 2f (Final DECRL-w/o-fusion, which only models high-order correlations without temporal evolution) exhibits increased inter-cluster distances, with less distinct clustering phenomena. The comparison between Figure 2d (Final DECRL) and Figure 2f (Final DECRL-w/o-fusion) shows that capturing temporal evolution leads to better entity representations, as indicated by larger inter-cluster distances and tighter intra-cluster groupings. Comparing the first and third rows of Figure 2 further highlights the training progression, where modeling temporal evolution gradually enhances cluster separation and tightens the grouping of entities within clusters. This demonstrates that the capability of DECRL to model the temporal evolution of high-order correlations significantly enhances its ability to capture more nuanced cluster representations. Additional case study details are provided in Appendix A.7.

6 Conclusions and limitations

In this paper, a **D**eep **E**volutionary **C**lustering jointed temporal knowledge graph **R**epresentation **L**earning approach (**DECRL**) is proposed for event prediction in TKGs, which jointly optimizes TKG representation learning with evolutionary clustering to capture the temporal evolution of high-order correlations. Comprehensive experiments are conducted on seven real-world datasets, including the comparison with baselines, ablation study, hyper-parameter sensitivity analysis, and case studies, which demonstrate the superior performance of DECRL. However, this study overlooks the continuous temporal evolution of diverse high-order correlations. Currently, the implicit correlation encoder assumes uniform correlations across all clusters, which may not reflect the complexity of real-world interactions between political organizations. In future work, we will address this issue by developing a multi relation-aware inter-cluster correlation encoder. Furthermore, DECRL currently models the temporal evolution of high-order correlations by considering only the previous and current timestamps. Future work will focus on explicitly modeling the continuous influence of high-order correlations from different past timestamps on the current timestamp.

https://doi.org/10.52202/079017-1752

Acknowledgement

This work was supported by the Science Foundation of Donghai Laboratory (Grant No. DH-2022ZY0013).

References

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

[2] James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David D Cox. Hyperopt: A python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1):014008, 2015.

[3] Jiaao Chen, Jianshu Chen, and Zhou Yu. Incorporating structured commonsense knowledge in story completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6244–6251, 2019.

[4] Ling Chen, Xing Tang, Weiqi Chen, Yuntao Qian, Yansheng Li, and Yongjun Zhang. DACHA: A dual graph convolution based temporal knowledge graph representation learning method using historical relation. *ACM Transactions on Knowledge Discovery from Data*, 16(3):1–18, 2021.

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

[6] Songgaojun Deng, Huzefa Rangwala, and Yue Ning. Dynamic knowledge graph based multi-event forecasting. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1585–1595, 2020.

[7] Francesco Folino and Clara Pizzuti. An evolutionary multiobjective approach for community discovery in dynamic networks. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1838–1852, 2013.

[8] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *Proceedings of the International Conference on Learning Representations*, pages 1–24, 2020.

[9] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 6669–6683, 2020.

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

[11] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[12] Julien Leblay and Melisachew Wudage Chekol. Deriving validity time in knowledge graph. In *Proceedings of the Web Conference*, pages 1771–1776, 2018.

[13] Kalev Leetaru and Philip A Schrodt. GDELT: Global data on events, location, and tone, 1979–2012. In *Proceedings of ISA Annual Convention*, volume 2, pages 1–49, 2013.

[14] Yujia Li, Shiliang Sun, and Jing Zhao. TiRGN: Time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2152–2158, 2022.

[15] Zixuan Li, Xiaolong Jin, Saiping Guan, Wei Li, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. Search from history and reason for future: Two-stage reasoning on temporal knowledge graphs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 4732–4743, 2021.

 https://doi.org/10.52202/079017-1752

[16] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 408–417, 2021.

[17] Kangzheng Liu, Feng Zhao, Hongxu Chen, Yicong Li, Guandong Xu, and Hai Jin. Da-Net: Distributed attention network for temporal knowledge graph reasoning. In *Proceedings of the ACM International Conference on Information & Knowledge Management*, pages 1289–1298, 2022.

[18] Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. TLogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4120–4127, 2022.

[19] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 2395–2405, 2018.

[20] Lijia Ma, Yutao Zhang, Jianqiang Li, Qiuzhen Lin, Qing Bao, Shanfeng Wang, and Maoguo Gong. Community-aware dynamic network embedding by using deep autoencoder. *Information Sciences*, 519:22–42, 2020.

[21] Xiaoke Ma and Di Dong. Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks. *IEEE Transactions on Knowledge and Data Engineering*, 29(5):1045–1058, 2017.

[22] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *Proceedings of Conference on Innovative Data Systems Research*, 2013.

[23] Yue Pu, Wenbin Yao, and Xiaoyong Li. EM-IFCM: Fuzzy c-means clustering algorithm based on edge modification for imbalanced data. *Information Sciences*, 659:120029, 2024.

[24] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *Proceedings of the European Semantic Web Conference*, pages 593–607, 2018.

[25] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3060–3067, 2019.

[26] Zongjiang Shang and Ling Chen. MSHyper: Multi-scale hypergraph transformer for long-range time series forecasting. *arXiv preprint arXiv:2401.09261*, 2024.

[27] Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. TimeTraveler: Reinforcement learning for temporal knowledge graph forecasting. *arXiv preprint arXiv:2109.04101*, 2021.

[28] Xing Tang and Ling Chen. GTRL: An entity group-aware temporal knowledge graph representation learning method. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–16, 2023.

[29] Xing Tang, Ling Chen, Hongyu Shi, and Dandan Lyu. DHyper: A recurrent dual hypergraph neural network for event prediction in temporal knowledge graphs. *ACM Transactions on Information Systems*, pages 1–25, 2024.

[30] Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. Know-Evolve: Deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of International Conference on Machine Learning*, pages 3462–3471, 2017.

[31] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11):2579–2605, 2021.

[32] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 950–958, 2019.

[33] Binqing Wu, Weiqi Chen, Wengwei Wang, Bingqing Peng, Liang Sun, and Ling Chen. WeatherGNN: Exploiting meteo- and spatial-dependencies for local numerical weather prediction bias-correction. In *Proceedings of the International Joint Conference on Artificial Intelligence*, page 2433–2441, 2024.

[34] Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L Hamilton. TeMP: Temporal message passing for temporal knowledge graph completion. *arXiv preprint arXiv:2010.03526*, 2020.

[35] Yi Xu, Junjie Ou, Hui Xu, and Luoyi Fu. Temporal knowledge graph reasoning with historical contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4765–4773, 2023.

[36] Min Yang, Xiaoliang Chen, Baiyang Chen, Peng Lu, and Yajun Du. DNETC: Dynamic network embedding preserving both triadic closure evolution and community structures. *Knowledge and Information Systems*, 65(3):1129–1157, 2023.

[37] Yuhang Yao and Carlee Joe-Wong. Interpretable clustering on dynamic graphs with recurrent graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4608–4616, 2021.

[38] Jingyi You, Chenlong Hu, Hidetaka Kamigaito, Kotaro Funakoshi, and Manabu Okumura. Robust dynamic clustering for temporal networks. In *Proceedings of the ACM International Conference on Information & Knowledge Management*, pages 2424–2433, 2021.

[39] Jiasheng Zhang, Shuang Liang, Yongpan Sheng, and Jie Shao. Temporal knowledge graph representation learning with local and global evolutions. *Knowledge-Based Systems*, 251:109234, 2022.

 https://doi.org/10.52202/079017-1752

Table 6: The statistics of datasets

Dataset	#Entity	#Relation	#Training set	#Validation set	#Test set	#Time interval
ICEWS14	7128	230	74,845	8,514	7,371	24 hours
ICEWS14C	205	171	35,665	7,369	7,068	24 hours
ICEWS18	23,033	256	373,018	45,995	49,545	24 hours
ICEWS18C	208	164	34,497	4,412	4,661	24 hours
GDELT	7,691	240	1,734,399	238,765	305,241	15 mins
WIKI	12,554	24	539,286	67,538	63,110	1 year
YAGO	10,623	10	161,540	19,523	20,026	1 year

Table 7: The final choices of hyper-parameter values

Hyper-parameter	ICEWS14	ICEWS14C	ICEWS18	ICEWS18C	GDELT	WIKI	YAGO
N_{c}	14	6	18	8	16	18	16
$N_{\text{DECRL layer}}$	2	2	5	2	5	2	1
$N_{\text{historical window}}$	9	7	4	10	2	2	1
λ	0.2	0.2	0.2	0.2	0.2	0.2	0.2

A Appendix

A.1 Complexity analysis

The time complexity of the relation-aware GCN is $O(N_{\mathrm{e}}N_{\mathrm{r}}D^2)$, where N_{e} and N_{r} are the numbers of entities and relations, respectively. D is the dimension of representations. The time complexity of the evolutionary clustering module is $O(N_{\mathrm{e}}N_{\mathrm{c}}D + N_{\mathrm{c}}^3 + N_{\mathrm{c}}D)$, where N_{c} is the number of clusters. The time complexity of the cluster graph message passing module is $O(N_{\mathrm{c}}^2D^2)$. The time complexity of the time residual gate is $O(D^2)$. For the attentive temporal encoder, the time complexity is $O(T^2D)$, where T is the length of the history. For the event prediction module, the time complexity is $O(D)$. The total complexity of DECRL is $O(N_{\mathrm{e}}N_{\mathrm{r}}D^2 + N_{\mathrm{c}}^3 + N_{\mathrm{c}}^2D^2 + T^2D)$.

A.2 Statistics of datasets

All datasets are split into training (80%), validation (10%), and test (10%) sets following [16]. The statistics of these datasets are summarized in Table 6.

A.3 Experimental settings

DECRL is implemented in Python using PyTorch and trained on one NVIDIA RTX 3080 GPU with 10GB memory. We leverage the Neural Network Intelligence (NNI) toolkit[2] to automatically identify the optimal hyper-parameter values. The search space for the number of clusters N_{c}, the number of DECRL layers $N_{\text{DECRL layer}}$, the length of historical windows $N_{\text{historical window}}$, and the value of λ range from 1 to 20 with the step of 2, 1 to 5 with the step of 1, 1 to 14 with the step of 1, and 0.1 to 0.5 with the step of 0.1, respectively. The final hyper-parameter values are presented in Table 7. For NNI configurations, the maximum number of trials is set to 30, and the optimization algorithm used is the Tree-structured Parzen Estimator [2].

We utilize the Adam [10] optimizer with an initial learning rate of 0.01. The batch size is set to 16. The representation dimension is set to 200. The hidden sizes for the time residual gate and the attentive temporal encoder are both set to 200. The results reported are the averages across five independent runs. The evaluation metrics used in this paper include Mean Reciprocal Rank (MRR) and Hits@1/3/10, which represent the proportion of correct predictions ranked within the top 1, 3, and 10 positions, respectively, all expressed as percentages. Higher Hits@k and MRR scores indicate better performance.

[2]https://github.com/microsoft/nni

A.4 Description of baselines

Shallow encoder-based approaches:

- **TTransE** [12] extends the TransE by incorporating timestamps as corresponding representations.
- **HyTE** [5] models timestamps as corresponding hyperplanes.

DNN-based approaches:

- **RE-NET** [9] leverages GCNs to model the influence of neighbor entities and uses RNNs to capture temporal dependencies among events.
- **Glean** [6] employs CompGCN to model the influence of neighbor entities and utilizes GRUs to capture temporal dependencies among representations.
- **TeMP** [34] utilizes relation-aware GCN to model the influence of neighbor entities and employs a frequency-based gating GRU to capture temporal dependencies among inactive events.
- **RE-GCN** [16] uses relation-aware GCN to model the influence of neighbor entities and employs an autoregressive GRU to capture temporal dependencies among events.
- **DACHA** [4] introduces a dual graph convolution network to obtain entity representations and uses a self-attentive encoder to model temporal dependencies among relations.
- **TiRGN** [14] is the SOTA approach, using a multi-relational GCN to capture graph structure information and a double recurrent mechanism to model temporal dependencies.

Derived structure-based approaches:

- **TITer** [27] incorporates temporal agent-based reinforcement learning to search paths and obtain entity representations via the inductive mean.
- **EvoExplore** [39] establishes dynamic communities to model the influence of neighbor entities.
- **GTRL** [28] uses entity group modeling to capture the influence of distant and unreachable entities and employs GRUs to model temporal dependencies among representations.
- **DHyper** [29] is the SOTA approach, which utilizes hypergraph neural networks to model high-order correlations among entities and among relations.

A.5 Hyper-parameter sensitivity analysis

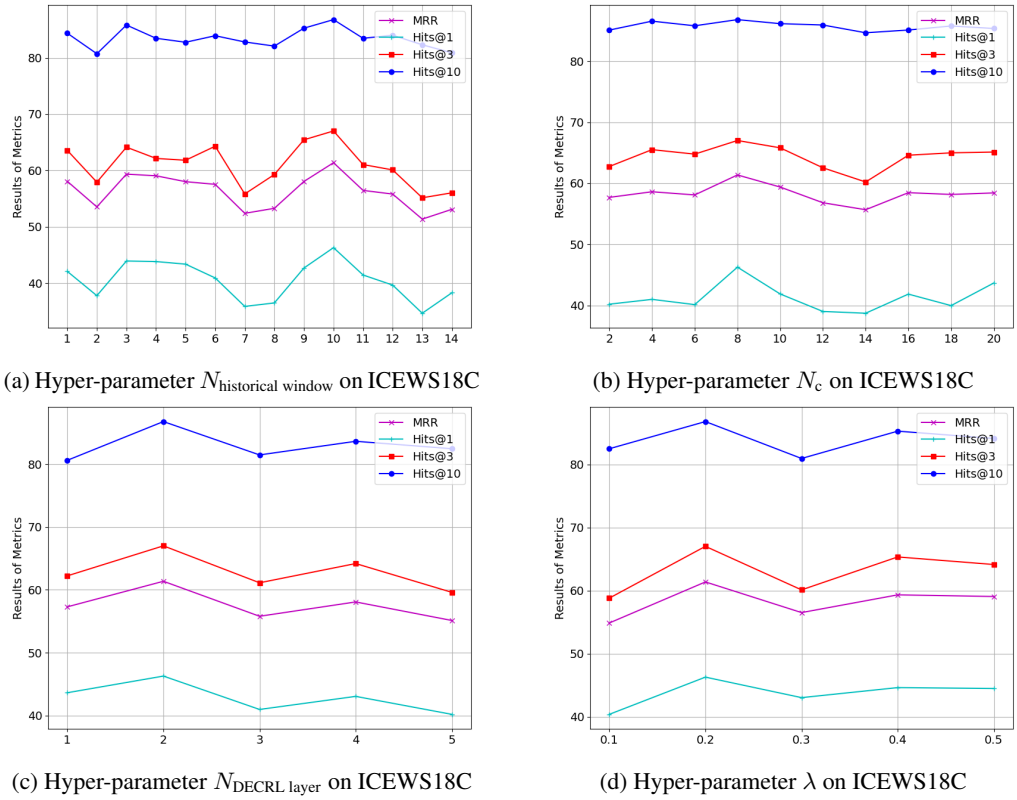
In this sub-section, we study the hyper-parameter sensitivity of DECRL on ICEWS18C, including the length of historical windows $N_{\text{historical window}}$, the number of clusters N_{c}, the number of DECRL layers $N_{\text{DECRL layer}}$, and the value of λ. We show the performance changes by varying the hyper-parameter values in Figure 3.

We present the impact of $N_{\text{historical window}}$ in Figure 3a. The results indicate that as $N_{\text{historical window}}$ increases, the performance of DECRL gradually improves, peaking at the window length of 10. Beyond this point, performance declines rapidly, likely due to the inclusion of excessive irrelevant information in longer historical windows, which negatively impacts the performance of DECRL.

In Figure 3b, we present the effect of N_{c}. The performance of DECRL remains relatively stable across most metrics as N_{c} increases. Hits@1 shows a slight initial increase until $N_{\text{c}} = 8$ and then stabilizes. MRR, Hits@3, and Hits@10 remain consistent, indicating that the performance of DECRL is not substantially affected by variations of N_{c}.

We show the impact of $N_{\text{DECRL layer}}$ and λ in Figures 3c and 3d, respectively. In Figure 3c, performance metrics reach their peak when $N_{\text{DECRL layer}} = 2$. Beyond this point, the performance of DECRL declines, likely due to increased model complexity, which raises the risk of over-fitting. Similarly, in Figure 3d, the performance metrics peak at $\lambda = 0.2$, which suggests the need to set the appropriate trade-off between event prediction loss and temporal smoothness loss.

https://doi.org/10.52202/079017-1752



(a) Hyper-parameter $N_{\text{historical window}}$ on ICEWS18C

(b) Hyper-parameter N_{c} on ICEWS18C

(c) Hyper-parameter $N_{\text{DECRL layer}}$ on ICEWS18C

(d) Hyper-parameter λ on ICEWS18C

Figure 3: Results of hyper-parameters changes of DECRL on ICEWS18C.

Table 8: The entity prediction performance of DECRL and the compared approaches on GDELT. "OOM" and "TLE" indicate out of memory and a single epoch exceeded 24 hours. * indicates that DECRL is statistically superior to the compared approaches according to pairwise t-test at a 95% significance level. The best results are in bold and the second best results are underlined

Approach	MRR	Hits@1	Hits@3	Hits@10
TTransE (WWW 2018)	10.04*	6.31*	19.23*	28.08*
HyTE (EMNLP 2018)	18.41*	5.16*	21.75*	30.47*
RE-NET (EMNLP 2020)	20.78*	15.37*	23.16*	35.59*
Glean (KDD 2020)	21.14*	19.29*	27.28*	36.37*
TeMP (EMNLP 2020)	36.23*	<u>31.37</u>*	39.47*	48.45*
RE-GCN (SIGIR 2021)	21.19*	19.53*	22.92*	35.71*
DACHA (TKDD 2022)	31.05*	28.11*	40.12*	48.12*
TiRGN (IJCAI 2022)	23.64*	20.95*	26.88*	40.26*
TITer (EMNLP 2021)	TLE	TLE	TLE	TLE
EvoExplore (KBS 2022)	23.61*	17.44*	32.37*	41.47*
GTRL (TKDE 2023)	**39.23***	**33.37***	<u>42.45</u>*	<u>51.49</u>*
DHyper (TOIS 2024)	OOM	OOM	OOM	OOM
DECRL	<u>37.24</u>	28.60	**46.69**	**66.47**
Improvement	**-5.07%**	**-14.29%**	**9.99%**	**29.09%**

Table 9: Top 5 relations predicted by TiRGN, DHyper, and DECRL

Test sample	TiRGN	DHyper	DECRL
(Russia, ?, Ukraine, 2014/12/16)	Engage in diplomatic cooperation	Criticize or denounce	Accuse
	Make empathetic comment	Sign formal agreement	Criticize or denounce
	Apologize	Reduce or break diplomatic relations	Use conventional military force
	Appeal for economic aid	Engage in negotiation	Appeal for economic aid
	Host a visit	Praise or endorse	Impose embargo, boycott, or sanctions
(the United States, ?, Russia, 2014/12/17)	Make statement	Impose embargo, boycott, or sanctions	Disapprove
	Make a visit	Sign formal agreement	Sign formal agreement
	Sign formal agreement	Accuse	Impose embargo, boycott, or sanctions
	Express intent to meet or negotiate	Engage in negotiation	Express intent to yield
	Impose embargo, boycott, or sanctions	Use conventional military force	Use unconventional violence

A.6 Entity prediction performance

We have conducted additional experiments to evaluate the performance of DECRL on the future entity prediction task, as shown in Table 8. Although DECRL does not achieve the SOTA performance in terms of MRR and Hits@1 metrics, it achieves the best results in Hits@3 and Hits@10 metrics. It is important to note that DECRL is not specifically designed for entity prediction tasks. Nevertheless, these results demonstrate the effectiveness and robustness of DECRL, particularly in capturing a broader range of relevant entities.

A.7 Case study

Table 9 presents the top 5 relations predicted by the SOTA DNN-based approach TiRGN, the SOTA derived structure-based approach DHyper, and our proposed approach DECRL for two test samples on ICEWS14C. The test samples pertain to the conflict between Russia and Ukraine that began in February 2014. During this period, Russia deployed military forces to the Crimean region of Ukraine, which resulted in widespread condemnations and sanctions from several countries, including the United States and various European nations. Correct predictions are underlined in Table 9. Compared to TiRGN and DHyper, DECRL predicts more correct relations and ranks the correct predictions higher. The results indicate that by modeling the temporal evolution of the high-order correlations among entities, DECRL can achieve more accurate prediction results.

A.8 Societal impacts

In this paper, a **D**eep **E**volutionary **C**lustering jointed temporal knowledge graph **R**epresentation **L**earning approach (**DECRL**) is proposed for event prediction in temporal knowledge graphs, which offers more accurate and credible results. We summarize the positive and possible negative societal impacts as follows:

Positive societal impacts:

- **Optimizing social governance**: Temporal knowledge graph event prediction can help governments and relevant institutions foresee potential future events, enabling them to take preemptive measures and optimize social governance.
- **Supporting business decisions**: Companies can use the prediction results for market analysis and business decisions, enhancing their competitiveness.
- **Enhancing public safety**: By predicting potential threats and dangerous events, relevant departments can allocate resources in advance to ensure public safety.
- **Advancing academic research**: This research can promote progress in the fields of temporal knowledge graphs and event prediction, providing new directions and methods for academic studies.

Negative societal impacts:

- **Misuse risks**: Accurate predictive technology might be exploited by malicious actors to forecast and manipulate events.

https://doi.org/10.52202/079017-1752

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope (see Abstract and Introduction).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the work (see Conclusions and limitations).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper provides the full set of assumptions and a complete (and correct) proof (see Section 4).

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information needed to reproduce the main experimental results of the paper (see Appendix A.3).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

https://doi.org/10.52202/079017-1752

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We upload source code and datasets on Anonymous GitHub (see Appendix A.3).

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results (see Appendices A.2 and A.3).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports appropriate information about the statistical significance of the experiments (see Sub-section 5.2).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computer resources and time complexity analysis (see Appendices A.1 and A.3).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses both potential positive societal impacts and negative societal impacts of the work performed (see Appendices A.8).

    Guidelines:

https://doi.org/10.52202/079017-1752

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The datasets used in this work are public and out of risk. Our approach poses no such risks, not involving pretrained language models or image generators.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, are properly credited, and the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The instruction of dataset/code/model is provided on the project homepage at the anonymous Github. The URL of the anonymous repository please see Abstract.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

https://doi.org/10.52202/079017-1752

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.