A Dual-task Learning Model for Temporal Knowledge Graph Entity Alignment

Jingwei Cheng[1,2], Xihao Wang[1,2], and Fu Zhang[1,2](✉)

[1] School of Computer Science and Engineering, Northeastern University, China
{chengjingwei,zhangfu}@mail.neu.edu.cn, wangxihao106@163.com
[2] Key Laboratory of Intelligent Computing in Medical Image of Ministry of Education, Northeastern University, China

Abstract. The goal of entity alignment (EA) is to discover equivalent entity pairs across different knowledge graphs (KGs), which is crucial for knowledge fusion and integration. Temporal knowledge graphs (TKGs) have gained increasing attention by extending traditional KGs with the introduction of timestamps. State-of-the-art studies on temporal-aware EA have demonstrated the benefits of incorporating temporal information from TKGs. However, we argue against the inclusion of both temporal and relational information in entity embeddings, as the temporal and relational information in most TKGs may interfere with each other, thereby limiting the performance. Therefore, we propose a Dual-task learning model for temporal knowledge graph entity alignment (DT-TEA) that involves both a relational task and a temporal task. Firstly, the embeddings of both tasks are updated by optimizing a dynamic loss function. Then, a coarse-grained and a fine-grained similarity matrix are generated using the entity embeddings from two tasks as well as a temporal overlap matrix, which is then used for entity alignment. Extensive experiments on four real-world TKG datasets demonstrate that our proposed model outperforms state-of-the-art methods significantly. The code is available at https://github.com/w-oo/WORK/tree/master.

1 Introduction

Knowledge Graphs (KGs) are knowledge repositories that integrate data by means of a graph structure. They are typically represented as triples (e_h, r, e_t), where e_h, r, and e_t represent the head entity, the relation, and the tail entity respectively. KGs have been widely applied in various downstream applications such as information retrieval [5], recommendation systems [27], and question answering [16,17]. However, these KGs are specific to their respective domains, and their design and construction are not standardized. Due to the fact that different KGs are constructed from different data sources, heterogeneity and redundancy issues exist among them. This scenario presents a chance to integrate various knowledge graphs.

Entity Alignment (EA) plays a vital role in knowledge fusion. It aligns entities from different KGs that refer to the same real-world objects. Given two

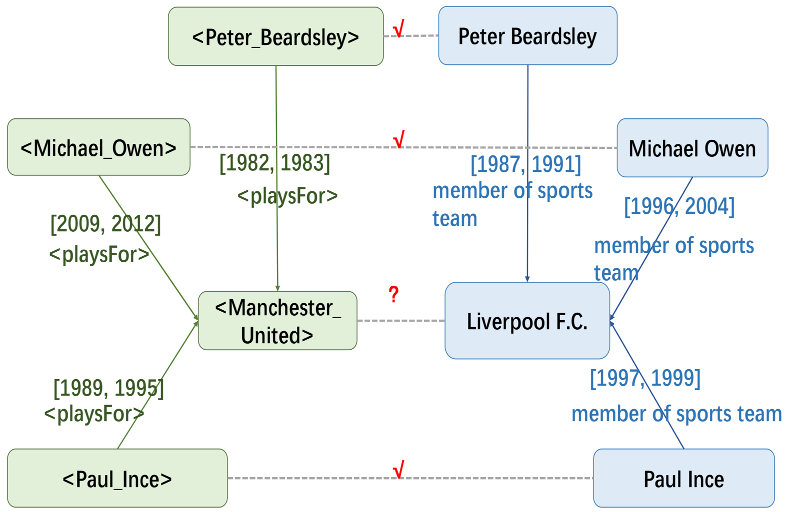


Fig. 1. An example of wrong alignment of entities with similar structures in different TKGs.

KGs and a small set of pre-aligned seed entities (also known as entity pairs), EA identifies all possible alignments between them. Existing embedding-based studies [9,12,15,20] have demonstrated their effectiveness of performing EA. They embed entities into a low-dimensional vector space and obtain equivalent entity pairs based on pairwise distances between vectors.

In the real world, certain facts occur at specific times. To incorporate temporal information indicating when events happen, YAGO2 [6] and Wikidata [22] have introduced temporal information into triples. Therefore, in TKGs, facts are extended to quadruplets (e_h, r, e_t, t), where t represents a timestamp. Fig. 1 provides an example of entity alignment between TKGs, where nodes represent entities and edges represent the temporal or relational connections between them. As shown Fig. 1, *<Peter Beardsley>* plays for *<Manchester United>* from 1982 to 1983. It is worth noting that this time interval applies to the *<playsFor>* relations, and the timestamps in TKGs can be viewed as an alternative representation of relations.

Most existing embedding-based entity alignment methods rely heavily on graph structural information while do not take into account the temporal information in KGs. This omission can lead to erroneous alignments of similar entities. Taking Fig. 1 as an example, the entities *<Manchester_ United>* and *Liverpool F.C.* have similar neighbors and relations. Disregarding temporal information, existing entity alignment methods tend to mistakenly identify <*Manchester_ United*> and *Liverpool F.C.* as the same entity.

To leverage the temporal information in TKGs, TEA-GNN [24] proposed a time-aware GNN for EA between TKGs, which integrates temporal information into entity embeddings through a time-aware attention mechanism. TREA [25] introduced time embeddings and relation embeddings into entity embeddings and developed a temporal relation attention mechanism. However, these methods have inherent limitations. When performing computations on both relational information and temporal information within a single GAT [21] and using them as attention weights, some entities can assign optimal attention weights to adjacent entities based solely on temporal information. However, the introduction of relational information can affect the optimal weight distribution, and similarly, temporal information can also influence the relations. Therefore, interference may arise between temporal and relational information, limiting the performance of the model.

To tackle the aforementioned challenge, we propose a **D**ual-**t**ask Learning Model for **T**emporal Knowledge **E**ntity **A**lignment, called DTTEA. DTTEA utilizes a dual-task paradigm that involves both a relational task and a temporal task. The relational task focuses on learning embeddings of entities and relations based on the structure and relation information in TKGs. On the other hand, the temporal task aims to learn embeddings of entities and time-aware relations by leveraging the structure and timestamp information available. Firstly, the embeddings of both tasks are updated by optimizing a dynamic loss function. This ensures that the learned embeddings are well-suited for both the relational and temporal aspects of the alignment task. Then, the similarity matrices for both tasks are obtained from the entity embeddings of the two tasks and are fused with the temporal overlap matrix through coarse-grained and fine-grained fusion. This results in a coarse-grained similarity matrix and a fine-grained similarity matrix. Finally, the optimal pairs of entities are identified through the application of the Sinkhorn operation.

In summary, this paper makes the following main contributions.

- The DTTEA model introduces a dual-task model for relational task and temporal task, which is the first to apply multi-task learning to TKG entity alignment task.
- DTTEA overcomes the limitations of existing EA models by efficiently incorporating temporal information. Furthermore, it effectively addresses the issue of interference between temporal and relational information in existing temporal knowledge graph entity alignment (TEA) models, thereby significantly improving the overall performance.
- We propose two fusion methods that effectively combine the similarity matrices of the temporal task and relational task, and the temporal overlap matrix, overcoming the mutual interference between temporal and relation, and improving the model performance of graph neural networks on temporal and relational embeddings.

– We conduct extensive experiments on four public datasets, and validate the superiority of DTTEA over previous SOTA EA methods on TKGs.

2 Related Work

2.1 Translation-based model

MTransE [3] uses TransE to represent different KGs as independent embeddings and learns the transformation between KGs using five alignment models. In addition to learning entity structural embeddings based on TransE using relation triples, JAPE [18] combines entity attribute embeddings with structural embeddings for entity alignment. IPTransE [28] embeds individual KGs using PTransE [10] and integrates three modules based on translation, linear transformation, and parameter sharing to jointly embed different KGs. BootEA [19] proposes a bootstrapping process that iteratively adds potential aligned entities to the training data to improve the performance of entity alignment. These methods align entities independently using triples, lacking the utilization of global structural information.

2.2 GNN-based Model

Due to the strong modeling capability of Graph Neural Networks (GNNs) in capturing global structure of KGs, several GNN-based models have been proposed and achieved promising performance in the EA task. GCN-align [23] utilizes GCN [8] to encode entities into a unified vector space and aligns entities based on their structural and attribute embeddings. However, GCN-align fails to effectively exploit the relational features present in the knowledge graph. MRAEA [14] introduces a meta-relation-aware EA method, where a relation-aware self-attentive GNN is employed to learn entity embeddings and obtain aligned entities. RREA [15] designs a GNN with relation reflection to acquire relation-specific embeddings that facilitate EA.

2.3 Temporal Knowledge Graph Alignment

STEA [1] introduces a simple GNN to learn entity embeddings and employs a time-aware matching mechanism to compute the temporal similarity of entities in the TKG, DualMatch [11] transforms EA on the TKG into a weighted graph matching problem. However, neither of these approaches utilizes time embeddings to extend the model, which limits the potential for aligning entities in the temporal dimension. LightTEA [2] proposes an effective and efficient non-neural EA model for TKGs, leveraging three-view label propagation (LP) to improve EA performance. Since it relies solely on a non-neural network approach, which leads to limitations in their performance. TEA-GNN [24] proposes a time-aware attention mechanism based on correlation and timestamp embedded neighborhood computation, assigning different weights to different nodes through an

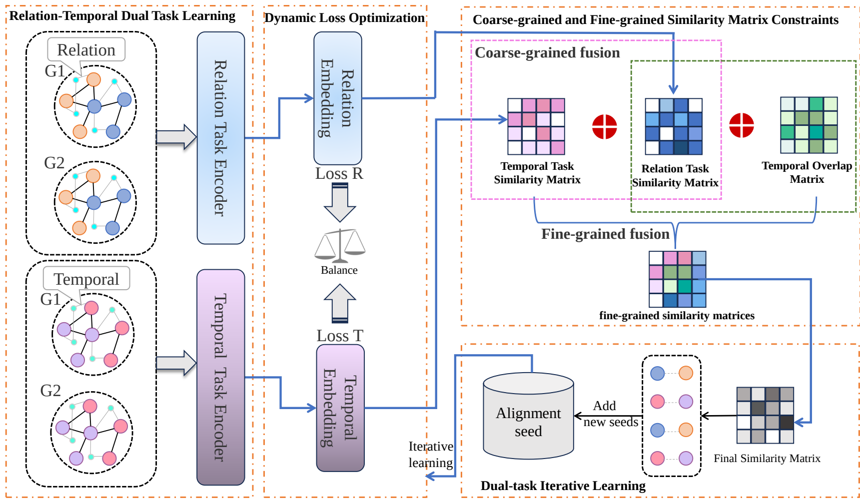


Fig. 2. The overal DTTEA model consists of (1) Relational-Temporal Dual Task Learning, (2) Dual-task Dynamic Loss Optimization, (3) Coarse-grained and Fine-grained Similarity Matrix Constraints, and (4) Dual-task Iterative Learning.

orthogonal transformation matrix. TREA [25] utilizes an attention mechanism based on temporal relations to integrate the relational and temporal features of links between entities, thereby enhancing EA. However, these models have not addressed the issue of mutual interference between time and relation. we present a neural network-based TEA model as an effective solution to address these limitations.

3 Task Definition

A TKG is represented as $G = (E, R, T, Q)$, where E, R and T represent the sets of entities, relations and timestamps, respectively, $Q \subset E \times R \times E \times T$ is the set of factual quadruples. Give two TKGs $G_1 = (E_1, R_1, T_1, Q_1), G_2 = (E_2, R_2, T_2, Q_2)$, and alignment seed set $S = \{(e_{1i}, e_{2j}) \in E_1 \times E_2 | e_{1i} \equiv e_{2j}\}$ where $\equiv$ denotes the equivalence relation. Specifically, the timestamps in the two sets are merged into a uniform time stamp set $T^* = T_1 \bigcup T_2$ for both two TKGs. Therefore, the task of EA between TKGs aims to obtain new aligment entities set P based on the alignment seed S between $G_1 = (E_1, R_1, T^*, Q_1)$ and $G_2 = (E_2, R_2, T^*, Q_2)$.

4 The Proposed Method

We propose a dual-task learning model for temporal knowledge graph entity alignment called DTTEA. DTTEA utilizes a dual-task paradigm that involves both a relational task and a temporal task. The relational task focuses on learning embeddings of entities and relations based on the structure and relation information found in TKGs. On the other hand, the temporal task aims to learn embeddings of entities and time-aware relations by leveraging the structure and timestamp information available. DTTEA enhances the accuracy of TEA by simultaneously optimizing these two tasks.

Fig. 2 illustrates the model of DTTEA. DTTEA consists of four parts: (1) Relational-Temporal Dual Task Learning, (2) Dual-task Dynamic Loss Optimization, (3) Coarse-grained and Fine-grained Similarity Matrix Constraints, and (4) Dual-task Iterative Learning.

4.1 Relational-Temporal Dual Task Learning

The representation of a temporal fact can be expressed as a quadruple (e_h, r, e_t, t), where the time information t can be represented in various forms, such as specific time points, and time intervals. A time interval can be depicted as $[t_b, t_e]$, where t_b and t_e represent the actual start and end times of a temporal fact, respectively. A time point is a time interval where $t_b = t_e$. We represent a begin or end time as $[t_b, t_0]$ or $[t_0, t_e]$ where t_0 is an unknown timestamp. Some studies [26,15] suggest that both structural information and relational information are beneficial for entity representation. Similarly, we consider timestamps as a special type of relation. Therefore, we decompose each quadruple $(e_h, r, e_t, [t_b, t_e])$ into relational triples and temporal triples. The obtained relation triples are (e_h, r, e_t) and (e_h, r^{-1}, e_t), r^{-1} represents the inverse relation. The obtained time triples are (e_h, t_b, e_t) and (e_h, t_e, e_t), t_b and t_e are temporal relations that are derived from timestamps and share the same symbols.

In the relational task, entities and relations are embedded into a vector space $\mathbb{R}^k$, which are denoted as $\boldsymbol{H}_e^r \in \mathbb{R}^{|E| \times k}$ and $\boldsymbol{H}_r \in \mathbb{R}^{|R| \times k}$, where k represents the embedding dimension. In the temporal task, entities and temporal relations are embedded into another vector space $\mathbb{R}'^k$ for the temporal task, denoted as $\boldsymbol{H}_e^t \in \mathbb{R}'^{|E| \times k}$ and $\boldsymbol{H}_t \in \mathbb{R}'^{|T| \times k}$. The embeddings of the entity e_i and relation r_j for the relational task are represented as $\boldsymbol{h}_{e_i,r}$ and $\boldsymbol{h}_{r_j}$, respectively. The embeddings of the entity e_i and relation t_j for the temporal task are represented as $\boldsymbol{h}_{e_i,t}$ and $\boldsymbol{h}_{t_j}$. The Graph Attention Network (GAT) is an extension of the conventional Graph Convolutional Network (GCN). The embeddings of the relational task and temporal task for entity e_i are calculated by the following equations:

$$\boldsymbol{h}_{e_i,r}^{l+1} = \sigma\left(\sum_{e_j \in \mathcal{N}_{e_i}^e} \sum_{r_k \in R_{ij}} \alpha_{ijk}^l \boldsymbol{M}_{r_k} \boldsymbol{h}_{e_i,r}^l\right) \tag{1}$$

$$\boldsymbol{h}_{e_i,t}^{l+1} = \sigma\left(\sum_{e_j \in \mathcal{N}_{e_i}^e} \sum_{t_k \in T_{ij}} \beta_{ijk}^l \boldsymbol{M}_{t_k} \boldsymbol{h}_{e_i,t}^l\right) \tag{2}$$

where $\sigma()$ is a non-linear activation function ReLU. $\boldsymbol{M}_{r_k} \in \mathbb{R}^{k \times k}$ and $\boldsymbol{M}_{t_k} \in \mathbb{R}^{k \times k}$ represent the relation reflection matrix and the temporal reflection matrix of the relation r_k and the temporal relation t_k. $\mathcal{N}^e_{e_i}$ represents the neighboring entity set of e_i. The relational task attention coefficient α_{ijk} and temporal-task attention coefficient β_{ijk}, are computed by the following equations:

$$\gamma^l_{ijk} = \boldsymbol{v}_r^T[\boldsymbol{h}^l_{e_i,r}||\boldsymbol{M}_{r_k}\boldsymbol{h}^l_{e_j,r}||\boldsymbol{h}_{r_j}] \tag{3}$$

$$\tau^l_{ijk} = \boldsymbol{v}_t^T[\boldsymbol{h}^l_{e_i,t}||\boldsymbol{M}_{t_k}\boldsymbol{h}^l_{e_j,t}||\boldsymbol{h}_{t_j}] \tag{4}$$

$$\alpha^l_{ijk} = \left(\frac{\exp(\gamma^l_{ijk})}{\sum_{e_j\in\mathcal{N}^e_{e_i}}\sum_{r_k\in R_{ij}}\exp(\gamma^l_{ijk})}\right) \tag{5}$$

$$\beta^l_{ijk} = \left(\frac{\exp(\tau^l_{ijk})}{\sum_{e_j\in\mathcal{N}^e_{e_i}}\sum_{t_k\in T_{ij}}\exp(\tau^l_{ijk})}\right) \tag{6}$$

where $\boldsymbol{v}_r$ and $\boldsymbol{v}_t$ are shared relational-task and temporal-task attention weight vectors, and T represents the transpose operation, l represents the number of layers. To create a global graph representation, we stack multiple layers of GNNs to capture multi-hop neighborhood information. For each task, we concatenate the embeddings from different layers to obtain the output embeddings of entity e_i in each task,

$$\boldsymbol{h}^{out}_{e_i,r} = \left[\boldsymbol{h}^0_{e_i,r}||...||\boldsymbol{h}^l_{e_i,r}\right] \tag{7}$$

$$\boldsymbol{h}^{out}_{e_i,t} = \left[\boldsymbol{h}^0_{e_i,t}||...||\boldsymbol{h}^l_{e_i,t}\right] \tag{8}$$

To obtain a comprehensive representation of the entity e_i, we further concatenate the average embedding of neighboring relation embeddings of entity e_i with the output embedding of entity e_i in the relational task. Similarly, we concatenate the average embedding of temporal relation embeddings with the entity embedding in the temporal task. This results in the final entity embedding for both tasks, i.e.,

$$\boldsymbol{h}^{Mul}_{e_i,r} = \left[\boldsymbol{h}^{out}_{e_i,r}||\frac{1}{|\mathcal{N}^r_{e_i}|}\sum_{r_m\in\mathcal{N}^r_{e_i}}\boldsymbol{h}_{r_m}\right] \tag{9}$$

$$\boldsymbol{h}^{Mul}_{e_i,t} = \left[\boldsymbol{h}^{out}_{e_i,t}||\frac{1}{|\mathcal{N}^t_{e_i}|}\sum_{t_n\in\mathcal{N}^t_{e_i}}\boldsymbol{h}_{t_n}\right] \tag{10}$$

where $\mathcal{N}^r_{e_i}$ represents the neighboring relation set of e_i and $\mathcal{N}^t_{e_i}$ represents the neighboring temporal relation set of e_i.

4.2 Dual-task Dynamic Loss Optimization

The optimization objective of an embedding-based entity alignment model aims to ensure that entities from each alignment seed possess similar representations.

We use L1 distance to measure the difference between entities $e_i \in E_1$ and $e_j \in E_2$ of two tasks as follows,

$$d_r(e_i, e_j) = ||\boldsymbol{h}^{Mul}_{e_i,r} - \boldsymbol{h}^{Mul}_{e_j,r}|| \tag{11}$$

$$d_t(e_i, e_j) = ||\boldsymbol{h}^{Mul}_{e_i,t} - \boldsymbol{h}^{Mul}_{e_j,t}|| \tag{12}$$

We utilize the following margin-based ranking loss function to train the model on both tasks.

$$L_r = \sum_{(e_i,e_j)\in S} \sum_{(e'_i,e'_j)\in S'} \max\left(d_r(e_i, e_j) - d_r(e'_i, e'_j) + \lambda, 0\right) \tag{13}$$

$$L_t = \sum_{(e_i,e_j)\in S} \sum_{(e'_i,e'_j)\in S'} \max\left(d_t(e_i, e_j) - d_t(e'_i, e'_j) + \lambda, 0\right) \tag{14}$$

where λ represents the margin hyperparameter, (e_i, e_j) represent the positive pair, (e'_i, e'_j) represent the negative pair obtained by randomly replacing one of (e_i, e_j), and S' is the set of negative pairs. The dual-task learning loss function of DTTEA is as follows,

$$L_{final} = \mu L_r + \theta L_t \tag{15}$$

where L_r is the loss of the relational task, L_t represents the loss of the temporal task. μ and θ are hyperparameters that balance the importance of losses. In order to optimize the loss more quickly and robustly, we apply the principles of probabilistic modeling to automatically learn the optimal task weights [7].

4.3 Coarse-grained and Fine-grained Similarity Matrix Constraints

Upon the embeddings for the temporal task and the relational task from Equations (9) and (10), we obtain the similarity matrices for the relational task $S_{emb\_r} \in \mathbb{R}^{|P|\times|P|}$ and the similarity matrices for temporal task $S_{emb\_t} \in \mathbb{R}^{|P|\times|P|}$. They are obtained through the dot product of entity embeddings acquired from the temporal task and the relational task between the two knowledge graphs. In order to achieve a more accurate alignment by sufficiently balancing the temporal task and the relational task, and to ensure the mutual constraint and complementarity between temporal and relational information, the coarse-grained constraint matrix is defined as follows:

$$S_{c\_TR} = \omega S_{emb\_t} + (1 - \omega) S_{emb\_r} \tag{16}$$

where ω is a hyper-parameter to balance the matrix of relational aspect and temporal aspect. Inspired by STEA [1], we additionally introduce a temporal overlap matrix as it has the potential to enhance the similarity for entity alignment. First, we construct a time dictionary for each entity. For each quadruple, if the time information is a single time point, it is added to the time dictionary of the corresponding entity. If the time information is a time interval, the start time and end time of the interval are added to the time dictionary of the corresponding entity. Then, we calculate the temporal overlap similarity between entities e_i and e_j as follows,

$$s^t_{e_i e_j} = \frac{2 \times c}{a + b} \tag{17}$$

where a and b represent the numbers of timestamps for entities e_i and e_j in the time dictionary, and c represents the number of overlapping timestamps between entities e_i and e_j in the dictionary. By calculating the temporal overlap smilarity for all entities, we obtain the temporal overlap matrix S_{t_ovlp}. In order to fully utilize the temporal overlap matrix, we proposed two combination methods, TRO and TROR, where TRO combines the S_{c_TR} with the temporal overlap matrix. TROR includes part of the coarse-grained matrix, adding weights from the temporal overlap matrix and the relational similarity matrix on top of the S_{c_TR}. The fine-grained similarity matrix is defined as follows:

$$S_{f\_TRO} = (1 - \eta) S_{c\_TR} + \eta S_{t\_ovlp} \tag{18}$$

$$\begin{aligned} S_{f\_TROR} = \omega S_{emb\_t} &+ \eta S_{t\_ovlp} \\ &+ (1 - \omega)(1 - \eta) S_{emb\_r} \end{aligned} \tag{19}$$

where S_f represents a composite of the fine-grained similarity matrix.

To further improve the effectiveness of the model and achieve one-to-one alignment, we adopt a similar approach to LightEA [2] by transforming the decoding process of EA into an assignment problem. We employ the Sinkhorn [4] operator to obtain an approximate solution and enhance the EA. The S_{final} is defined as follows:

$$S_{final} = Sinkhorn(S_f) \tag{20}$$

The Sinkhorn algorithm [4] is defined as follows:

$$\begin{aligned} S^0(\boldsymbol{X}) &= \exp(\boldsymbol{X}), \\ S^q(\boldsymbol{X}) &= \mathcal{N}_c(\mathcal{N}_r(S^{q-1}(\boldsymbol{X}))), \\ Sinkhorn(\boldsymbol{X}) &= \lim_{q \to \infty} S^q(\boldsymbol{X}), \end{aligned} \tag{21}$$

where $\mathcal{N}_r(\boldsymbol{X}) = \boldsymbol{X} \oslash (\boldsymbol{X} \mathbf{1}_N \mathbf{1}_N^T)$ and $\mathcal{N}_c = \boldsymbol{X} \oslash (\mathbf{1}_N \mathbf{1}_N^T \boldsymbol{X})$ are the row and column-wise normalization operators of a matrix, $\oslash$ represents the elementwise division, $\mathbf{1}_N$ is a column vector of ones.

4.4 Dual-task Iterative Learning

We adopt the iterative strategy proposed by STEA [1]. If the entities e_i and e_j are mutually nearest neighbors, then the pair (e_i, e_j) is considered new alignment entities and will be added into the training set of the next iteration. The difference of this approach lies in the integration of both coarse-grained and fine-grained similarity matrices through the Sinkhorn operation to enhance the distinguishability of entity similarities.

Table 1. Statistics of original datasets.

Dataset	$\|E_1\|$	$\|E_2\|$	$\|R_1\|$	$\|R_2\|$	$\|T^*\|$	$\|Q_1\|$	$\|Q_2\|$	$\|P\|$
DICEWS-1K/200	9,517	9,537	247	246	4,017	307,552	307,553	7,566/8,366
YAGO-WIKI50K-5K/1K	49,629	49,222	11	30	245	221,050	317,814	44,172/48,172

5 Experiments

5.1 Datasets

To evaluate the capability of the DTTEA, we conducted experiments on four widely used public datasets. The detailed statistical data for all datasets can be found in Table 1.

DICEWS-1K/200 [24] are built based on the event knowledge graph ICEWS05-15, which contains events from 2005 to 2015. DICEWS-200 and DICEWS-1K use 200 and 1000 seed entity pairs for alignment, respectively.

YAGO-WIKI50K-5K/1K [24]: YAGO-WIKI50K consists of YAGO-WIKI50K-1K and YAGO-WIKI50K-5K, which use 1000 and 5000 seed entity pairs for alignment, respectively. It is a hybrid dataset derived from Wikidata and YAGO, representing a large number of entities with different surface forms. Additionally, the dataset includes timestamps represented in various forms such as specific time points, start or end times, and time intervals.

5.2 Experimental Setting

We have selected 11 competitive knowledge graph alignment methods as baselines. They can be categorized into the following groups:

(1) Supervised methods: Models without temporal information: MTransE[3], JAPE[18], GCN-align[23], MRAEA[14], Dual-AMN[12], LightEA[13]. Models with temporal information: TEA-GNN[24], TREA[25], STEA[1].

(2) Semi-supervised methods: Models without temporal information: RREA[15]. Models with temporal information: Dual-Match[11].

The hyperparameter settings for the models are as follows. We set $\eta = 0.3$. For the datasets DICEWS and YAGO-WIKI50K, we set the epoch to 600 and 1200, respectively. We also set the factor ω to 0.1 and 0.3 to balance the relation and temporal tasks. We set embedding dimension $k = 100$, dropout rate $dr = 0.3$, number of layers $l = 2$, learning rate $lr = 0.005$, and the number of iterations for the Sinkhorn operator $m = 10$. The hyperparameter settings for the baseline methods are based on the configurations reported in their original publications.

Our experiments are conducted on a workstation equipped with an AMD EPYC 7543 32-Core Processor, 80GB memory, and the NVIDIA A40 / 48GB. In our experiments, to increase robustness, the seed pairs are randomly selected for each experiment. The best performance is indicated in bold, and the second best performance is indicated with an underline.

Table 2. The results of DTTEA and the baselines on the DICEWS and YAGO-WIKI50K datasets. Examples marked with an asterisk (*) indicate cases without iterations. The best results are in bold, and the second best results are indicated with an underline. DTTEA$_{TRO}$ and DTTEA$_{TROR}$ represent the combination of the similarity matrix of the two tasks and temporal overlap matrix using Equation (18) and (19).

Categories	Models	DICEWS-1K			DICEWS-200			YAGO-WIKI50K-5K			YAGO-WIKI50K-1K		
		Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
Supervised	MtransE	10.1	24.1	0.150	6.7	17.5	0.104	24.2	47.7	0.322	1.2	6.7	0.033
	JAPE	14.4	29.8	0.198	9.8	21.0	0.138	27.1	48.8	0.345	10.1	26.2	0.157
	GCN-Align	20.4	46.6	0.291	16.5	36.3	0.231	51.2	71.1	0.581	21.7	39.8	0.279
	MRAEA	67.5	87.0	0.745	47.6	73.3	0.564	80.6	91.3	0.848	62.3	80.1	0.685
	Dual-AMN	71.6	89.3	0.779	66.8	85.4	0.733	89.7	96.4	0.922	75.5	89.0	0.834
	LightEA	78.5	91.8	0.833	72.1	87.8	0.779	94.8	97.9	0.960	87.8	94.5	0.902
	TEA-GNN	88.7	94.7	0.911	87.6	94.1	0.902	87.9	96.1	0.909	72.3	87.1	0.775
	TREA	91.4	96.6	0.933	91.0	96.0	0.927	94.0	98.9	0.958	84.0	93.7	0.885
	STEA*	92.8	96.0	0.941	92.7	96.1	0.941	93.5	98.6	0.954	88.7	96.6	0.916
	Dual-Match	95.3	97.3	0.961	95.3	97.4	0.961	**98.1**	**99.6**	0.986	94.7	98.4	0.961
	DTTEA$^*_{TRO}$	95.6	97.3	0.964	95.0	97.1	0.959	87.4	95.2	0.909	79.6	90.1	0.844
	DTTEA$^*_{TROR}$	91.7	94.9	0.932	90.5	94.3	0.923	97.1	99.1	0.980	80.3	90.5	0.849
Semi-supervised	RREA	83.8	93.6	0.875	84.2	93.7	0.878	95.1	98.0	0.963	93.8	97.0	0.951
	STEA	94.5	96.7	0.954	94.3	96.8	0.954	96.1	99.2	0.974	94.3	**98.9**	0.962
	DTTEA$_{TRO}$	**95.7**	**97.4**	**0.965**	**95.4**	**97.5**	**0.964**	96.1	98.8	0.974	95.3	98.6	0.968
	DTTEA$_{TROR}$	94.9	96.1	0.956	**95.4**	96.6	0.960	**98.1**	**99.6**	**0.987**	**95.8**	98.7	**0.971**

5.3 Results and Analysis

Table 2 shows the performance of our proposed model and all baseline models in entity alignment between temporal knowledge graphs. We refer to these two approaches as Method TRO and Method TROR. From Table 2, it can be observed that our dual-task learning approach in DTTEA exhibits significant superiority over other baseline models.

For DICEWS, DTTEA$_{TRO}$ achieve the best performance, indicating that the combination of similarity matrices and temporal overlap matrix using Method TRO is most suitable for this dataset. It also confirms that temporal tasks and relational information are equally important in the dataset alignment process. DTTEA$_{TROR}$ also outperforms the iterative model STEA in terms of $Hits@1$, and it performs better than the state-of-the-art model dual-match on DICEWS-200, validating the effectiveness of Method TROR on this dataset. The performance of the DTTEA$^*_{TRO}$ model is also superior to the iterative STEA. Additionally, for DICEWS-1K, even without using the iterative strategy, DTTEA$^*_{TRO}$ outperforms other models, confirming that DTTEA can achieve good performance even without the iterative strategy.

DTTEA$_{TROR}$ achieves the best performance on YAGO-WIKI, further confirming that the TROR method is more suitable for the dataset. It validates that relational tasks are more important than temporal tasks in this dataset. Even without the iterative strategy, DTTEA$^*_{TROR}$ performs better than STEA on YAGO-WiKI-5k. For YAGO-WiKI-1k, the DTTEA$_{TRO}$ model with Method TRO combination also outperforms other models, validating the effectiveness of Method TRO.

Table 3. The result of ablation study

Models	DICEWS-200			Models	YAGO-WIKI50K-5K		
	Hits@1	Hits@10	MRR		Hits@1	Hits@10	MRR
DTTEA$_{TRO}$	**95.4**	**97.5**	**0.964**	**DTTEA$_{TROR}$**	**98.1**	**99.6**	**0.987**
w/o REL	92.6	96.5	0.944	**w/o REL**	94.5	98.1	0.961
w/o TIME	95.2	97.4	0.962	**w/o TIME**	94.9	98.2	0.964
w/o OVLP	78.6	88.2	0.831	**w/o OVLP**	91.7	96.2	0.938
w/o SH	95.3	96.5	0.959	**w/o SH**	96.9	99.0	0.978

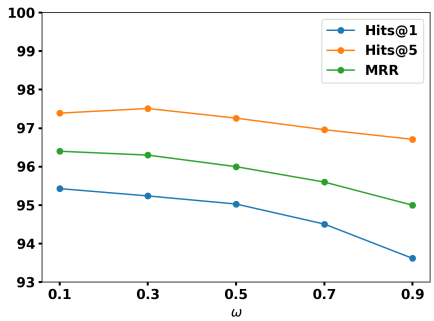
5.4 Ablation Experiments

Table 3 show the ablation study of the experiment. DTTEA consists of four key components: (1) Relational Task Module (REL), (2) Temporal Task Module (TIME), (3) temporal overlap matrix (OVLP), (4) Sinkhorn operation (SH). As shown in the table 3, (w/o REL) represents removing the relational task embedding and (w/o TIME) represents removing the temporal task embedding, (w/o OVLP) represents without using temporal overlap matrix, (w/o SH) represents without using sinkhorn operator. Removing any of these modules results in a performance decline for DTTEA. Removing the Relational Task Module (w/o REL) leads to a performance decline, emphasizing the importance of the relational task. Similarly, a slight performance decrease is observed when the Temporal Task Module is removed (w/o TIME), indicating the beneficial effect of incorporating temporal task information into EA. When the temporal overlap matrix is removed (w/o OVLP), there is a significant performance drop, highlighting the enhancement provided by the temporal overlap matrix in EA. Without the Sinkhorn operation (w/o SH), DTTEA's performance slightly decreases, indicating the effectiveness of the Sinkhorn operation.

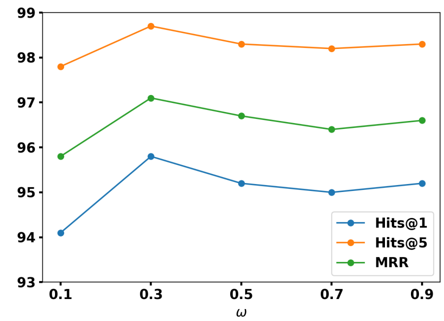
5.5 Hyper-parameter Study

The DTTEA model has two hyperparameters, ω and η, which are used to balance the two tasks and the temporal overlap matrix, respectively. For ω and η, we varied them within the range of 0 to 1 with an interval of 0.2 to study their impact. We conducted parameter adjustments on two datasets, with the DICEWS-200 dataset using the DTTEA$_{TRO}$ model and the YAGO-WIKI50K-1K dataset using the DTTEA$_{TROR}$ model. The experiments involved adjusting the values of both ω and η. The Fig. 3 shows the experimental results for different ω and η values on the two datasets.

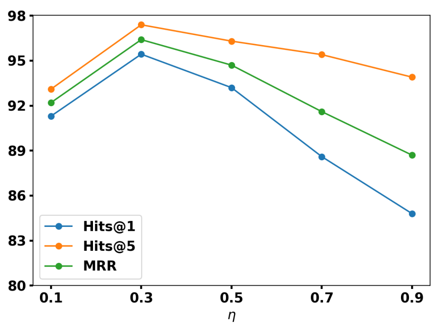
According to the experimental results, Fig. 3(a) shows that on the DICEWS-200 dataset, as the hyperparameter ω increases, the hits@1 metric gradually decreases. Therefore, we set ω to 0.1. According to Fig. 3(b), the YAGO-WIKI50K-1K dataset achieves the best hits@1 performance when the hyperparameter ω is set to 0.3. Fig. 3(c) and Fig. 3(d) indicate that the optimal performance can be



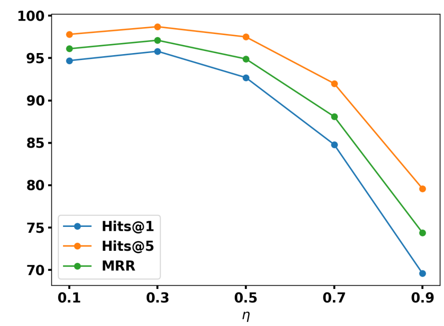
(a) On the DICEWS-200.



(b) On the YAGO-WIKI50K-1K.



(c) On the DICEWS-200.



(d) On the YAGO-WIKI50K-1K.

Fig. 3. Fig (a) and Fig (c) respectively show the results of using different balance factors ω and η on the DICEWS-200 dataset through DTTEA$_{TRO}$. On the other hand, Fig (b) and Fig (d) respectively depict the results of using different balance factors ω and η on the YAGO-WIKI50K-1K dataset through DTTEA$_{TROR}$.

achieved when the parameter η of the temporal overlap matrix is set to 0.3 in both datasets.

5.6 Seed-generation Study

Table 4 illustrates the number of seeds generated per iteration. We compare DTTEA$_{TRO}$ with STEA [1]. As shown in the table, after five iterations, DTTEA$_{TRO}$ generates more seeds than STEA. Additionally, after the iteration 1, DTTEA$_{TRO}$ generates 5.55% more seeds compared to STEA, validating the effectiveness of the iterative strategy in maximizing the model's performance with fewer iterations.

6 Conclusion

In this work, we demonstrate a method to address interference in entity alignment of temporal knowledge graphs through a dual-task learning approach. To

Table 4. The number of seeds generated per iteration on the DICEWS-200 dataset.

Models	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Sum
STEA	7706	432	111	52	28	8329
DTTEA$_{TRO}$	8134	164	40	18	8	8364

tackle this issue, we reframe the entity alignment problem as a multitask learning problem in temporal knowledge graphs (TKGs) and propose DTTEA. The model consists of four essential components: (1) Relation-Temporal Dual Task Learning, (2) Dual-Task Dynamic Loss Optimization, (3) Coarse-Grained and Fine-Grained Similarity Matrix Constraints, and (4) Dual-Task Iterative Learning. These modules are combined to enhance the performance of entity alignment, with validation that each module contributes to improving the model's performance. Extensive experiments conducted on public datasets demonstrate that the model significantly outperforms state-of-the-art methods. Furthermore, the model achieves good performance even without using iterations. When iterations are employed, the model maximizes its performance with fewer iterations. In future work, we will explore the use of additional information in temporal knowledge graphs, such as entity names and descriptions. We will also investigate methods for entity alignment in spatial-temporal knowledge graphs.

Acknowledgments. The authors sincerely thank the anonymous reviewers for their valuable comments and suggestions, which improved the paper. The work is supported by the National Natural Science Foundation of China (62276057), and Sponsored by CAAIMindSpore Open Fund, developed on OpenI Community.

References

1. Cai, L., Mao, X., Ma, M., Yuan, H., Zhu, J., Lan, M.: A simple temporal information matching mechanism for entity alignment between temporal knowledge graphs. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 2075–2086 (2022)
2. Cai, L., Mao, X., Xiao, Y., Wu, C., Lan, M.: An effective and efficient time-aware entity alignment framework via two-aspect three-view label propagation. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. pp. 5021–5029 (2023)
3. Chen, M., Tian, Y., Yang, M., Zaniolo, C.: Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. pp. 1511–1517 (2017)
4. Cuturi, M.: Sinkhorn distances: lightspeed computation of optimal transport. In: Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2. pp. 2292–2300 (2013)
5. Dietz, L., Kotov, A., Meij, E.: Utilizing knowledge graphs for text-centric information retrieval. In: The 41st international ACM SIGIR conference on research & development in information retrieval. pp. 1387–1390 (2018)

6. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: Yago2: A spatially and temporally enhanced knowledge base from wikipedia. Artificial intelligence **194**, 28–61 (2013)
7. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7482–7491 (2018)
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2016)
9. Li, C., Cao, Y., Hou, L., Shi, J., Li, J., Chua, T.S.: Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 2723–2732 (2019)
10. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 705–714 (2015)
11. Liu, X., Wu, J., Li, T., Chen, L., Gao, Y.: Unsupervised entity alignment for temporal knowledge graphs. In: Proceedings of the ACM Web Conference 2023. pp. 2528–2538 (2023)
12. Mao, X., Wang, W., Wu, Y., Lan, M.: Boosting the speed of entity alignment 10×: Dual attention matching network with normalized hard sample mining. In: Proceedings of the Web Conference 2021. pp. 821–832 (2021)
13. Mao, X., Wang, W., Wu, Y., Lan, M.: LightEA: A scalable, robust, and interpretable entity alignment framework via three-view label propagation. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 825–838 (2022)
14. Mao, X., Wang, W., Xu, H., Lan, M., Wu, Y.: Mraea: an efficient and robust entity alignment approach for cross-lingual knowledge graph. In: Proceedings of the 13th International Conference on Web Search and Data Mining. pp. 420–428 (2020)
15. Mao, X., Wang, W., Xu, H., Wu, Y., Lan, M.: Relational reflection entity alignment. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 1095–1104 (2020)
16. Qiu, Y., Wang, Y., Jin, X., Zhang, K.: Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. In: Proceedings of the 13th international conference on web search and data mining. pp. 474–482 (2020)
17. Saxena, A., Tripathi, A., Talukdar, P.: Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In: Proceedings of the 58th annual meeting of the association for computational linguistics. pp. 4498–4507 (2020)
18. Sun, Z., Hu, W., Li, C.: Cross-lingual entity alignment via joint attribute-preserving embedding. In: The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21–25, 2017, Proceedings, Part I 16. pp. 628–644. Springer (2017)
19. Sun, Z., Hu, W., Zhang, Q., Qu, Y.: Bootstrapping entity alignment with knowledge graph embedding. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. pp. 4396–4402 (2018)
20. Sun, Z., Wang, C., Hu, W., Chen, M., Dai, J., Zhang, W., Qu, Y.: Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 222–229 (2020)

21. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
22. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)
23. Wang, Z., Lv, Q., Lan, X., Zhang, Y.: Cross-lingual knowledge graph alignment via graph convolutional networks. In: Proceedings of the 2018 conference on empirical methods in natural language processing. pp. 349–357 (2018)
24. Xu, C., Su, F., Lehmann, J.: Time-aware graph neural network for entity alignment between temporal knowledge graphs. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 8999–9010 (2021)
25. Xu, C., Su, F., Xiong, B., Lehmann, J.: Time-aware entity alignment using temporal relational attention. In: Proceedings of the ACM Web Conference 2022. pp. 788–797 (2022)
26. Yang, H.W., Zou, Y., Shi, P., Lu, W., Lin, J., Sun, X.: Aligning cross-lingual entities with multi-aspect information. arXiv (2019)
27. Zhou, K., Zhao, W.X., Bian, S., Zhou, Y., Wen, J.R., Yu, J.: Improving conversational recommender systems via knowledge graph based semantic fusion. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1006–1014 (2020)
28. Zhu, H., Xie, R., Liu, Z., Sun, M.: Iterative entity alignment via joint knowledge embeddings. In: IJCAI. vol. 17, pp. 4258–4264 (2017)