RCTEA: Richness-guided Co-training for Temporal Entity Alignment

Anonymous ACL submission

Abstract

Temporal Entity Alignment (TEA), which aims to identify equivalent entities across Temporal Knowledge Graphs (TKGs), is crucial for integrating knowledge facts from multiple sources. However, existing TEA models often fail to capture the orthogonal yet complementary effect between structural and temporal features, and typically overlook the importance of information richness—a key factor for effective message passing in the neural feature encoders. To address these limitations, we propose a RCTEA framework that jointly models both structural and temporal aspects of the TKGs for entity alignment. Specifically, we design a richness-guided attention mechanism along with an adaptive weighting strategy to facilitate effective feature fusion. To ensure robust alignment despite noisy entity contexts, we introduce a dual-view neighborhood consensus algorithm that jointly refines the feature encoders to enforce local structural consistency of the predicted alignments. Extensive experiments demonstrate the superiority of RCTEA, achieving state-of-the-art performance on public TEA benchmarks.

1 Introduction

A Knowledge Graph (KG) is a structured knowledge base that captures real-world knowledge to support data-driven applications such as information retrieval (Kobayashi and Takeda, 2000), information extraction (Sarawagi et al., 2008), and recommendation (Jiang et al., 2024; Lü et al., 2012). With the growing use of KGs derived from diverse sources, integrating knowledge from multiple KGs has become crucial. Entity Alignment (EA), the process of identifying equivalent entities across KGs, plays a central role in KG fusion.

Given the dynamic and complex nature of entity-wise interactions, the incorporation of temporal information into KGs has led to the emergence of Temporal Knowledge Graphs (TKGs) (Trisedya et al., 2019; Zeng et al., 2020; Ge et al., 2021; Xin et al., 2022; Liu et al., 2023a). TKGs extend traditional triples by including timestamps, enabling a richer representation of dynamic relationships among entities over time. However, this advancement also introduces the challenging task of Temporal Entity Alignment (TEA), as illustrated in Figure 1. Existing TEA models (Xu et al., 2021, 2022; Liu et al., 2023b; Cai et al., 2022, 2023) often treat temporal features in a simplified way, lacking dedicated mechanisms for temporal refinement and noise mitigation. Specifically, (Xu et al., 2021, 2022) treat temporal information as a type of relation for feature propagation and rely on straightforward feature concatenation during inference. In contrast, (Liu et al., 2023b; Cai et al., 2022, 2023) employ temporal encoders to facilitate temporal knowledge transfer, but they are often computationally expensive and sensitive to temporal heterogeneity. Although (Li et al., 2025) introduces a temporal attention mechanism and a heterogeneity refinement strategy to enhance temporal knowledge propagation, it does not explicitly address the challenges of complex feature interactions.

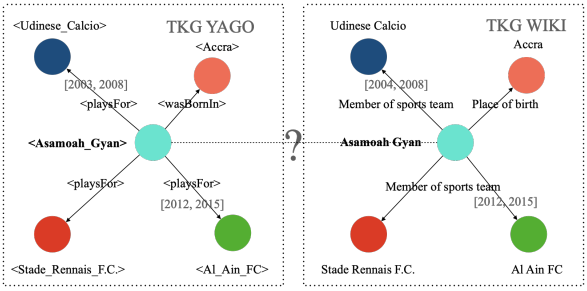

Figure 1: Example of temporal entity alignment.

Issue 1: Existing TEA models are typically GNN-based, requiring message propagation between neighboring entities to refine the representation learning. However, they often overlook the importance of feature richness in the local neighborhood when learning entity embeddings. Our preliminary study (Appendix D.1) reveals that prop-

agating the supervision signals of low-rich entities to their neighbors naturally hinders the model's ability to capture precise and informative features, resulting in suboptimal feature representations and inferior alignment results. This calls for a mechanism to distinguish entities based on feature richness in the message-passing process.

Issue 2: Most TEA models combine structural and temporal features using basic methods like concatenation or averaging, which limits their ability to adaptively emphasize the most informative signals for each entity pair. Structural and temporal features provide orthogonal but complementary insights for entity alignment, and their relative importance may vary depending on the specific context. According to our preliminary study (Appendix D.1), structural encoders perform better on entities with richer neighborhood, while temporal encoders are more effective for temporally-diverse entities, highlighting the need for entity-specific and richness-based weighting to integrate both features into the alignment decision.

To address these issues, we propose RCTEA, a novel Richness-guided Co-training framework for Temporal Entity Alignment. In particular, we design a novel richness-guided attention mechanism to facilitate effective message propagation in the feature encoders by quantifying the importance of each neighbor based on its structural and temporal richness. We also introduce a dynamic weighting strategy to adaptively integrate both structural and temporal features for embedding learning. To enhance model robustness, we propose a dual-view neighborhood consensus algorithm that jointly refines the feature encoders by enforcing local consistency in the predicted alignments. Our model can be further extended to a semi-supervised iterative training framework, where the training signals are progressively expanded with an effective bi-directional seed selection mechanism to eliminate noisy pseudo-labels. Our main contributions are summarized below:

- We introduce novel dual-aspect feature encoders that fully leverage the orthogonal yet complementary features-structural and temporal-to learn more informative and comprehensive entity embeddings. The feature encoders are jointly refined via dual-view neighborhood consensus.
- We highlight the importance of richness for feature encoders, and design richness-guided attention and adaptive weighting strategies to enhance the effectiveness of representation learning by dynamically prioritizing informative features.
- Extensive experiments on publicly available TEA datasets clearly demonstrate the superiority of our RCTEA model, achieving state-of-the-art alignment performance.

2 Related Work

Recent studies have recognized the importance of temporal information for entity alignment. Models like TEA-GNN (Xu et al., 2021) and TREA (Xu et al., 2022) treat temporal point embeddings as a medium for temporal representation, yet they tend to treat temporal information similarly to relational data, thus overlooking its unique properties. The following methods explore temporal information as an attribute: STEA (Cai et al., 2022) encodes temporal information using a temporal dictionary, while DualMatch (Liu et al., 2023b) incorporates a temporal encoder to enhance message propagation. MGTEA (Zeng et al., 2024) further adopts a similar strategy to exploit fine-grained temporal cues. However, these approaches are often computationally expensive, largely due to the need to construct and process temporal-aspect similarity matrices. LightTEA (Cai et al., 2023), an extension of LightEA (Mao et al., 2022), introduces a lightweight model for TKG alignment, but the temporal module leads to limited improvement on existing datasets (Xu et al., 2021). HTEA (Li et al., 2025) introduces an L1-weighted feature selection matrix and a temporal attention module to enhance the propagation of temporal information. Furthermore, a heterogeneity module is applied iteratively to alleviate the impact of heterogeneous temporal triples. Nevertheless, current TEA models still fail to effectively capture the interaction between structural and temporal features, and suffer from the noise issue on both sides.

3 Problem Formulation

Definition 1 (Temporal Knowledge Graph)**.** *A temporal knowledge graph (TKG) is a directed graph, denoted by* $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$*, where* $\mathcal{E}, \mathcal{R}$ *and* $\mathcal{T}$ *represent the set of entities, relations and temporal points, respectively;* $\mathcal{F} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T} \times \mathcal{T}$ *is a collection of facts in the form of* $f = (e_h, r, e_t, I)$*. Specifically,* $I = [t_s, t_e]$ *is the temporal interval during which the head entity* e_h *has a valid relation* r *with the tail entity* e_t*, where* $t_s \in \mathcal{T}$ *(resp.* $t_e \in \mathcal{T}$*) represents the start time (resp. end time)*

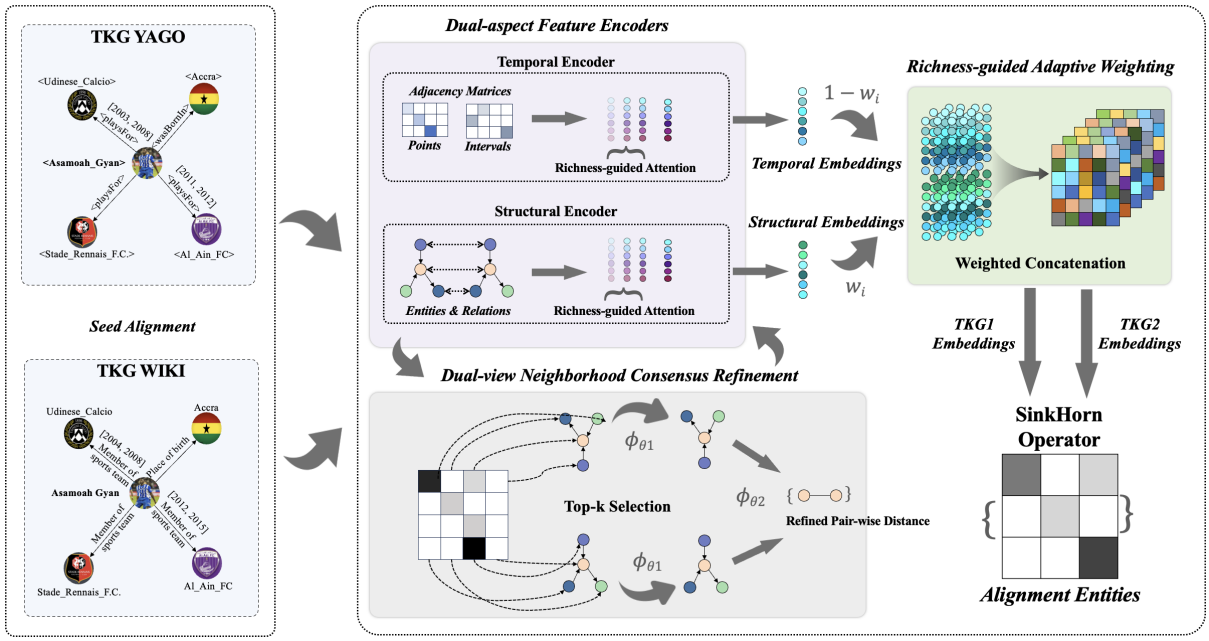

Figure 2: RCTEA framework overview.

of the fact. A placeholder ($\sim$) is used when t_s or t_e does not exist. $\mathcal{I} \subseteq \mathcal{T} \times \mathcal{T}$ denotes the set of temporal intervals.

Definition 2 (Temporal Entity Alignment)**.** *Given two TKGs $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$ and $\mathcal{G}' = (\mathcal{E}', \mathcal{R}', \mathcal{T}', \mathcal{F}')$ and a set of pre-aligned entity pairs $\mathcal{S} = \{(e, e') \mid e \equiv e', e \in \mathcal{E}, e' \in \mathcal{E}'\}$ as the seed alignment, where $\equiv$ denotes entity equivalence, temporal entity alignment (TEA) aims to identify all possible newly matched entity pairs: $\{(e, e') \mid e \equiv e', e \in \mathcal{E}, e' \in \mathcal{E}', (e, e') \notin \mathcal{S}\}$.*

4 The RCTEA Model

Figure 2 presents an overview of our proposed RCTEA framework, which consists of three key modules: (1) Dual-aspect feature encoders that learns entity embeddings from both structural and temporal perspectives, along with a richness-guided attention mechanism to enhance the reliability of message propagation in the GNNs; (2) A richness-guided adaptive weighting strategy that balances both features with dynamically determined weights to generate a unified entity representation for alignment; (3) A denoising algorithm based on dual-view neighborhood consensus to refine the feature encoders by capturing locality-aware alignment patterns.

4.1 Dual-aspect Feature Encoders

As discussed earlier, the structural and temporal cues of a TKG provide orthogonal yet complementary insights for the alignment decision, making it necessary to build *dual-aspect feature encoders* that learn both the separate feature representations and their joint effect. In particular, we decompose the structural and temporal perspectives of a TKG by introducing two neural models based on the Graph Attention Network (GAT) architecture for feature encoding. These include: a structural encoder $\mathcal{M}_{stru}$ that utilizes entity- and relation-level input to learn the structural and semantic representations of a TKG, and a temporal encoder $\mathcal{M}_{temp}$ that incorporates both temporal points and intervals to encode the temporal behaviors of entities. Each encoder is designed to receive and propagate distinct types of information from the TKG. Beyond separate feature learning, we also construct a mixed encoder $\mathcal{M}_{mix}$ that jointly encodes all types of features and balances them through an adaptively weighted combination.

4.1.1 Structural and Temporal Encoders

Given a TKG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$, we extract four types of features (i.e., entities $\mathcal{E}$, relations $\mathcal{R}$, temporal points $\mathcal{T}$ and temporal intervals $\mathcal{I}$) and generate type-specific bipartite adjacency matrices $\mathbf{A}_{(\mathcal{C})}$ from the TKG, where $\mathcal{C} \in \{E, R, T, I\}$. $\mathbf{A}_{(\mathcal{C})}$ indicates the importance of each feature c for learning the embedding of entity e_i. We utilize log-normalized initial representation to dampen the impact of high-frequency features (Li et al., 2025):

$$a_{ic} = \frac{(\log(|\mathcal{F}_{ic}| + 1))}{\sum_{c' \in \mathcal{C}_{e_i}} (\log(|\mathcal{F}_{ic'}| + 1))} \quad (1)$$

$\mathcal{C}_{e_i}$ denotes the set of features contained by entity e_i; $\mathcal{F}_{ic}$ is the set of facts involving both entity e_i and feature c. For example, when $\mathcal{C} = E$, $\mathcal{C}_{e_i} = \mathcal{N}_{e_i}$ represent all the neighboring entities that e_i connects to. Similarly, when $\mathcal{C} = I$, $\mathcal{C}_{e_i} = \mathcal{I}_{e_i}$ indicates all the temporal intervals that e_i exists a relation with other entities.

Both the structural and temporal encoders learn the type-specific entity embeddings layer-by-layer in a GAT-like manner. Specifically, the type-specific layer propagation is defined as follows:

$$h^0_{e_{i(\mathcal{C})}} = \left(\mathbf{A}_{(\mathcal{C})}[e_i]\right)^\top \mathbf{F}_{(\mathcal{C})} \quad (2)$$

$$h^{l+1}_{e_i(\mathcal{C})} = \sigma\Big(\sum_{e_j \in \mathcal{N}_{e_i}} \beta^l_{ij(\mathcal{C})} \mathbf{W}^l_{(\mathcal{C})} h^l_{e_j(\mathcal{C})}\Big), l \geq 0 \quad (3)$$

Here, $\mathbf{F}(\mathcal{C})$ denotes the type-specific initialization matrix, while $\mathbf{A}(\mathcal{C})[e_i]$ represents the bipartite feature selection weights for entity e_i with respect to type $\mathcal{C}$, as defined in Eq. 1. To enable message passing across neighboring entities, multi-layer feature propagation is performed as follows: β^l_{ij} denotes the adaptive attention weight between entities e_i and e_j at layer l (detailed in the next section), $\boldsymbol{W}^l_{(\mathcal{C})}$ is the learnable type-specific transformation matrix, and $\sigma(\cdot)$ is the ReLU activation function.

After L-hop feature propagation and aggregation, we concatenate the embeddings learned at each layer to obtain the final type-specific feature representation for entity e_i as outlined below:

$$h_{e_i(\mathcal{C})} = [h^0_{e_i(\mathcal{C})} || h^1_{e_i(\mathcal{C})} || \cdots || h^L_{e_i(\mathcal{C})}] \quad (4)$$

The structural encoder $\mathcal{M}_{stru}$ concatenates the entity- and relation-specific embeddings to obtain the structural representation of entity e_i. Similarly, the temporal encoder $\mathcal{M}_{temp}$ combines the embeddings of both temporal points and temporal intervals, as below:

$$h_{e_i(stru)} = [h_{e_i(E)} || h_{e_i(R)}] \quad (5)$$

$$h_{e_i(temp)} = [h_{e_i(T)} || h_{e_i(I)}] \quad (6)$$

4.1.2 Mixed Encoder

Considering the complementary effect between structural and temporal perspectives of a TKG, we further build a mixed encoder $\mathcal{M}_{mix}$ to jointly obtain the dual-view feature representation of entity e_i through a weighted concatenation of both features obtained by $\mathcal{M}_{stru}$ and $\mathcal{M}_{temp}$ respectively:

$$h_{e_i} = [w_i h_{e_i(stru)} \oplus (1 - w_i) h_{e_i(temp)}] \quad (7)$$

All encoders are trained with the same procedure. Given two TKGs $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$ and $\mathcal{G}' = (\mathcal{E}', \mathcal{R}', \mathcal{T}', \mathcal{F}')$, along with their seed alignments $\mathcal{S} = \{(e_i, e'_i) \,|e_i \equiv e'_i, e_i \in \mathcal{E}, e'_i \in \mathcal{E}'\}$, we formulate a bi-directional probability-based loss over entity pairs, where both directions of alignment (i.e., from one TKG to the other and vice versa) are considered symmetrically to enhance robustness and alignment quality (Chen et al., 2023).

$$p(e_i, e'_i) = \frac{\gamma(e_i, e'_i)}{\gamma(e_i, e'_i) + \sum_{e_j \in \mathcal{N}^{ng}} \gamma(e_i, e_j)} \quad (8)$$

where $\gamma(e_i, e_j) = exp(h_{e_i} \cdot h^T_{e_j}/\tau)$ and τ is the temperature hyperparameter. $\mathcal{N}^{ng}$ is the in-batch negative samples to enhance training efficiency. To consider both directions of entity alignment, the overall loss function is defined as follows:

$$\mathcal{L}_{align} = -log(p(e_i, e'_i) + p(e'_i, e_i))/2 \quad (9)$$

4.2 Richness-guided Attention and Adaptive Weighting

Existing neighborhood weighting methods usually rely on naive GAT mechanisms (Velickovic et al., 2018) or relation-based approaches (Mao et al., 2020b), which tend to be less interpretable and fail to capture distinctive feature contributions. Meanwhile, current TEA models (Xu et al., 2021, 2022) typically incorporate various feature embeddings through a simple concatenation and feed the combined representations into alignment modules to predict entity mappings. However, our preliminary findings show that the richness of structural and temporal information varies significantly across entities, which should influence the weighting strategy during alignment.

To capture this signal, we propose a *richness-guided attention mechanism* along with an *entity-wise adaptive weighting scheme* that balances the contributions of structural and temporal features. Recognizing that structural and temporal information contribute differently to each entity's representation, our approach adaptively prioritizes the more informative aspect for each entity, thereby enhancing alignment accuracy. Specifically, we introduce the concept of reference embedding and define a corresponding richness measurement, which quantifies the feature richness of each entity. These measurements form the foundation for both the richness-guided attention and adaptive weighting mechanisms used in our model.

4.2.1 Reference Embeddings

When counterpart entities exhibit similar structural or temporal characteristics (such as node degrees, neighborhood distributions or temporal behaviors), their embeddings can reliably capture underlying similarities. To leverage this, we propose an effective strategy for measuring structural and temporal richness. Specifically, we introduce a global reference entity e_{ref} to simulate the entity with extremely low richness, e.g., without any neighbors or temporal behaviors. We obtain its corresponding type-specific representations, called reference embeddings $h_{e_{ref}(\mathcal{C})}$, similarly as the other entities using $\mathcal{M}_{stru}$ and $\mathcal{M}_{temp}$. These reference embeddings serve as representative proxy vectors that reflect the typical semantics of entities with low richness in the corresponding feature space. For example, the structural reference embeddings $h_{e_{ref}(\mathcal{C})}$ where $\mathcal{C} \in \{E, R\}$ quantify the degree-based richness of entity e_i, using the number of facts it participates in as the count between e_i and the reference entity e_{ref} (Eq.10). The reference entity's temporal aspect embeddings are none-temporal embeddings (point and interval).

$$|F_{ie_{ref}(E)}|, |F_{ie_{ref}(R)}| = \sum_{e_j \in \mathcal{N}_{e_i}} |F_{ie_j}| \quad (10)$$

4.2.2 Richness Measurement

We introduce a new metric, called *reference similarity*, to quantify the feature richness of a specific entity e_i. Since the reference entity e_{ref} is designed to simulate an entity prototype with extremely low richness, a low embedding similarity between e_i and e_{ref} naturally indicates that entity e_i is structurally or temporally rich. To capture this signal, we estimate e_i's feature richness by calculating its feature-specific reference similarity with regard to the corresponding reference embedding. Formally,

$$\mathcal{S}_{e_i(\mathcal{C})} = \cos(h_{e_i(\mathcal{C})}, h_{e_{ref}(\mathcal{C})}) \quad (11)$$

4.2.3 Richness-guided Feature Propagation

During the multi-layer message propagation of our dual-aspect feature encoders (i.e., Eq. 3), we introduce a richness-guided attention weight $\beta^l_{ij(\mathcal{C})}$ to quantify the importance of each neighbor e_j for learning the representation of entity e_i, as illustrate in Figure 3. The underlying intuition is that feature-specific richness varies across neighbors, while it is more valuable to propagate information from structurally or temporally rich neighbors. For example,

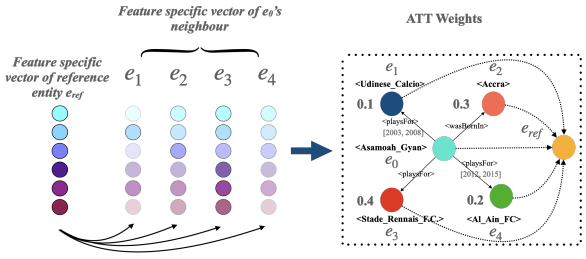

Figure 3: Example of richness-guided attention weights.

our preliminary study (Appendix D.1) proves that entities with higher degree are more likely to be correctly aligned, indicating that the training signals provided by these structurally rich neighbors are more reliable for learning robust entity embeddings. Similarly, a neighbor with diverse range of temporal features tends to carry more distinctive information for determining the entity alignment. Based on this observation, we formally define the richness-guided attention weight for each layer's feature propagation as follows:

$$\beta^l_{ij(\mathcal{C})} = \frac{\exp\left(-\mathcal{S}^l_{e_j(\mathcal{C})}\right)}{\sum_{e_k \in \mathcal{N}_{e_i}} \exp\left(-\mathcal{S}^l_{e_k(\mathcal{C})}\right)}, l \geq 0 \quad (12)$$

Here, $\mathcal{S}^l_{e_j(\mathcal{C})}$ at each feature propagation layer l denotes the feature-specific reference distance obtained by Eq. 11.

4.2.4 Richness-guided Feature Concatenation

Our preliminary study (Appendix D.1) also reveals that a straightforward concatenation of the structural and temporal features may degrade the alignment performance, calling for an adaptive weighting mechanism that can dynamically adjust the relative importance of both features for each specific alignment case. Similarly, these weights can be inferred based on the entity's structural and temporal richness. Inspired by (Chen et al., 2023) which applies weighted concatenation for balancing multimodal features, we design a richness-guided adaptive weighting scheme to balance the contributions of structural and temporal features for learning the embedding of a specific entity.

Specifically, we utilize the reference similarity between an entity's type-specific embedding and the reference embedding to construct an initial *reference matrix* $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} \mathcal{S}_{e_1(E)} & \mathcal{S}_{e_1(R)} & \mathcal{S}_{e_1(T)} & \mathcal{S}_{e_1(I)} \\ \mathcal{S}_{e_2(E)} & \mathcal{S}_{e_2(R)} & \mathcal{S}_{e_2(T)} & \mathcal{S}_{e_2(I)} \\ \vdots & \ddots & \vdots & \\ \mathcal{S}_{e_n(E)} & \mathcal{S}_{e_n(R)} & \mathcal{S}_{e_n(T)} & \mathcal{S}_{e_n(I)} \end{bmatrix} \quad (13)$$

The four columns represent the reference similarity for the four types of features, i.e., entities, relations, temporal points and intervals, respectively.

We then apply a single multilayer perceptron (MLP) to perform a non-linear transformation on the initial reference matrix $\mathbf{x}$ to infer the adaptive entity-wise concatenation weights used in the mixed encoder $\mathcal{M}_{mix}$ (i.e., Eq. 7). Formally,

$$\mathbf{h} = \phi(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1), \quad \mathbf{w} = \sigma(\mathbf{W}_2\mathbf{h} + \mathbf{b}_2) \quad (14)$$

ϕ and σ denotes MLP and sigmoid function, where $\mathbf{h}$ is the intermediate result, and $w_i \in \mathbf{w}$ is the weight obtained for each entity e_i. $\boldsymbol{W_1}$, $\boldsymbol{W_2}$, $\boldsymbol{b_1}$, $\boldsymbol{b_2}$ are learnable parameters.

4.3 Neighborhood Consensus De-noising and Iterative Training

To enhance the alignment compatibility among entities and their neighbors (i.e., neighbors of aligned entities should also be matched), we employ a neighborhood consensus model (Fey et al., 2020) to refine the dual-aspect feature encoders $\mathcal{M}_{stru}$ and $\mathcal{M}_{temp}$ for de-noising the learned entity embeddings, ensuring that aligned entities share similar local structures. Building on the complementary nature of structural and temporal features, we further introduce a *dual-view neighborhood consensus strategy*, enabling mutual enhancement between the two modalities through joint training.

In particular, given two TKGs $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$ and $\mathcal{G}' = (\mathcal{E}', \mathcal{R}', \mathcal{T}', \mathcal{F}')$, we generate their structural and temporal feature embeddings using the corresponding encoders $\mathcal{M}_{stru}$ and $\mathcal{M}_{temp}$, and jointly refine both encoders by concatenating their learned embeddings:

$$\mathbf{H} = [\mathbf{H}_{stru} \,||\, \mathbf{H}_{temp}] \quad (15)$$

We adopt the FAISS framework (Douze et al., 2024) to perform efficient top-k embedding retrieval from large-scale TKGs, as the top-k candidates typically contain the most informative alignment signals:

$$\hat{\mathbf{S}} = Softmax(Faiss_k[\mathbf{H} \cdot \mathbf{H}']) \quad (16)$$

where $\hat{\mathbf{S}}$ is the top-k retrieved probability score matrix, and exp is the softmax function. We then apply a multi-layer perceptron (MLP) to iteratively refine the similarity matrix, leveraging the injective node distances derived from the propagated features of the two TKGs to minimize the achieved embedding distances for two counterpart embeddings (Details in Appendix A.1).

Our RCTEA model can be further extended to an iterative training framework, denoted as RCTEA$^+$, which progressively incorporates high-quality seeds for model training. To this end, we design a *dual-aspect bi-directional seed selection* mechanism that exploits the complementary nature of the two orthogonal features—structural and temporal information—to collaboratively eliminate the influence of noisy pseudo-labels (Details in Appendix A.2).

5 Experiments

5.1 Experimental Settings

Datasets. We conduct experiments on both homogeneous and heterogeneous TEA benchmarks (dataset statistics in Appendix C.1). (Li et al., 2025) identifies a key limitation of existing TEA benchmarks—neglect of temporal heterogeneity across TKGs—and constructs a more realistic YAGO-WIKI180K dataset for evaluation. (Zeng et al., 2024) introduces BTEA, a fine-grained TEA dataset that incorporates hybrid temporal triples and further refines temporal information to the date level. We present experimental results on YAGO-WIKI180K and BTEA in this section, while additional evaluations on the homogeneous TEA datasets are reported in Appendix D.2. We also provide the detailed parameter settings in Appendix C.2, for better reproducibility of the experiments.

Baselines. We compare RCTEA with eight state-of-the-art TEA approaches, including two structural embedding models and six temporal-aware models. 1) *time-unaware*: Dual-AMN (Mao et al., 2021) and LightEA (Mao et al., 2022); 2) *time-aware*: TEA-GNN (Xu et al., 2021), STEA (Cai et al., 2022), DualMatch (Liu et al., 2023b), LightTEA (Cai et al., 2023), MGTEA(Zeng et al., 2024), and HTEA(Li et al., 2025). We ignore models that utilize various side information (entity names, entity descriptions, attributes) for a fair comparison.

Evaluation Metrics. Building on previous research, we use Mean Reciprocal Rank (MRR) and Hits at Top-N (H@N) as evaluation metrics. H@N measures the precision of the top-N retrieved entities, while MRR is calculated using the average reciprocal rank of all entities.

All experiments are conducted on a Linux cluster equipped with an AMD EPYC 3 Milan CPU, NVIDIA H100 GPU, and 1,500GB of RAM. The experiments are repeated five times, and the mean results are reported.

Table 1: Overall TEA performance on YAGO-WIKI180K (2000 seeds). The p-value is the result of one sample t-test between RCTEA and their corresponding strong baselines.

Model	All			Non-tem (50.4%)			Sparse-tem (43%)			Dense-tem (6.6%)		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
Dual-AMN	.381	.330	.476	.159	.110	.253	.572	.521	.668	.824	.768	.928
LightEA	.376	.329	.464	.169	.121	.260	.562	.516	.647	.742	.691	.838
STEA	.308	.277	.366	.007	.004	.112	.496	.457	.570	.911	.880	.963
TEA-GNN	.352	.307	.436	.145	.101	.227	.519	.472	.605	.843	.794	.931
LightTEA	.361	.317	.442	.150	.108	.231	.548	.505	.628	.744	.696	.833
DualMatch	.388	.342	.474	.157	.103	.259	.577	.538	.650	.912	.882	.966
MGTEA	.405	.355	.499	.185	.126	.300	.583	.541	.661	.920	.893	.967
HTEA[1]	.404	.354	.500	.183	.128	.290	.591	.544	.679	.876	.846	.930
RCTEA	**.459**	**.413**	**.545**	**.226**	**.168**	**.336**	**.655**	**.618**	**.723**	**.960**	**.947**	**.983**
p-value	1e-4	3e-5	7e-4	5e-3	2e-3	2e-3	7e-5	3e-5	3e-4	3e-3	2e-3	1e-2
RCTEA$^+$ (iter)	**.468**	**.423**	**.554**	**.232**	**.173**	**.344**	**.669**	**.633**	**.735**	**.971**	**.960**	**.987**
p-value	6e-5	2e-5	4e-4	3e-3	1e-3	2e-3	3e-5	1e-5	1e-4	2e-3	1e-3	1e-2

Table 2: Overall TEA performance on BETA .

Model	BETA (10%)		BETA (30%)	
	H@1	H@10	H@1	H@10
Dual-AMN	.566	.695	.589	.757
LightEA	.492	.673	.595	.744
STEA	.506	.637	.556	.681
TEA-GNN	.557	.709	.609	.751
LightTEA	.615	.727	.666	.769
DualMatch	.650	.765	.689	.796
MGTEA	**.686**	.795	.711	.813
HTEA	.647	.783	.660	.772
RCTEA	.681	**.797**	**.713**	**.816**
p-value	3e-2	1e-2	5e-3	5e-3
RCTEA$^+$ (iter)	**.707**	**.813**	**.720**	**.819**
p-value	8e-3	7e-3	3e-3	4e-3

5.2 Main Results

We consider the following research question in this section: **RQ1: Can RCTEA enhance TEA performance over existing baselines?** We compare RCTEA with existing TEA models and report their alignment performance on the overall YAGO-WIKI180K dataset and various subsets of different characteristics: *Dense-tem* with rich temporal features, *Sparse-tem* with scarce temporal information, and *Non-tem* without any temporal facts. As presented in Table 2, RCTEA consistently achieves the state-of-the-art performance on all variants of the YAGO-WIKI180K dataset, surpassing HTEA and MGTEA, the second-best models, by nearly 6% on all evaluation metrics. In particular, RCTEA outperforms HTEA and MGTEA on the Non-tem, Sparse-tem and Dense-tem dataset variants by around 4%, 7% and 5%, respectively, attributed to our design of richness-guided attention and adaptive weighting mechanisms that fully utilize the feature richness to enhance representation learning. The alignment accuracy further increases when the RCTEA model is iteratively trained with high-quality pseudo-labels (# iterations = 2 by default), i.e., RCTEA$^+$ in Table 2.

For the BETA dataset, even without leveraging fine-grained temporal information (e.g., date annotations), our model achieves superior performance across all seed settings at both 10% and 30%. In particular, RCTEA attains competitive Hit@1 performance in low-resource settings and outperforms other baselines in the remaining settings, despite not employing the time-consuming temporal encoders used in MGTEA and DualMatch. This further demonstrates the effectiveness and efficiency of our approach. Moreover, the iterative variant RCTEA$^+$ also yields notable performance gains, especially in the low-resource setting.

5.3 Ablation Study

We further explore the contribution of each component in our RCTEA framework, and present the detailed results in Table 3.

RQ2: Is it necessary to combine all features in RCTEA? We evaluate the importance of each type of features, i.e., entities, relations, temporal points, and temporal intervals, by excluding them separately from the RCTEA model. As observed from the first part of Table 3, the alignment performance drops in each case, highlighting the effectiveness of our dual-aspect feature encoders to integrate both structural and temporal perspectives for representation learning.

RQ3: Can the richness-guided attention mecha-

[1]The reported results for HTEA differ from those in the original paper, as we standardize the output embedding size across all models to ensure a fair comparison.

Table 3: Ablation study on YAGO-WIK180K (H@1).

Model	All	Non	Sparse	Dense
RCTEA	.4135	.1684	.6183	.9465
RCTEA w/o. E	.1933	.0013	.3300	.6758
RCTEA w/o. R	.3588	.1033	.5713	.9213
RCTEA w/o.T	.3874	.1377	.6004	.9031
RCTEA w/o. I	.3807	.1349	.5905	.8868
RCTEA w/o. ET	.3937	.1416	.6082	.9147
RCTEA w/o. RT	.3936	.1381	.6117	.9197
RCTEA w/o. TT	.3942	.1416	.6095	.9170
RCTEA w/o. IT	.3922	.1381	.6094	.9129
RCTEA w. GAT	.3825	.1402	.5881	.8887
RCTEA w. RST	.3919	.1520	.5986	.8726
RCTEA w. EW	.3919	.1371	.6099	.9142
RCTEA w/o. DW	.3951	.1427	.6103	.9166
RCTEA w/o. NC	.4065	.1607	.6120	.9414
RCTEA w/o. DUAL	.4120	.1710	.6125	.9416

nism enhance alignment performance? We examine the effectiveness of richness-guided attention weights for feature propagation by selectively disabling them. Variants RCTEA w/o. ET, RT, TT, and IT represent the removal of attention weights for entities, relations, temporal points, and temporal intervals, respectively. The experimental results confirm that each feature-specific attention contributes positively to the overall model, validating the effectiveness of our multi-view attention design. We further compare the proposed richness-guided attention with other attention mechanisms used by existing TEA models, including naïve GAT attention (RCTEA w. GAT) and relation-specific attention (RCTEA w. RST). Our approach consistently outperforms both variants, showcasing the importance of feature richness for attention learning.

RQ4: Is adaptive weighting necessary for feature fusion? We evaluate the effectiveness of the dynamic weighting strategy by modifying our mixed feature encoder $\mathcal{M}_{mix}$ with equal weights (RCTEA w. EW) or removing the dynamic weighting module entirely (RCTEA w/o. DW). Experimental results demonstrate that our adaptive weighting mechanism enhances overall alignment performance, surpassing the equally weighted version by a large extent.

RQ5: What is the impact of dual-view neighborhood consensus algorithm? Finally, we investigate the importance of neighborhood consensus for entity alignment. RCTEA w/o. NC disables the neighborhood consensus module, while RCTEA w/o. DUAL removes the dual-view feature representation used in the consensus process. Experimental results indicate that both components are crucial for effective feature refinement and alignment inference.

5.4 Efficiency Study

Finally, we compare RCTEA with existing TEA approaches in terms of model efficiency, considering the following two research questions: **RQ6: Can RCTEA balance alignment accuracy and model efficiency? RQ7: Can the iterative training of RCTEA$^+$ further enhance alignment performance, and at what cost?** From Figure 4(a), we can see that RCTEA achieves comparable time consumption with the most efficient model LightTEA and HTEA, while further enhances the alignment accuracy, which validates our model's superiority. Figure 4(b) indicates that the training time increses linearly with the iterative training, and the accuray improves mostly between the first and second iterations.

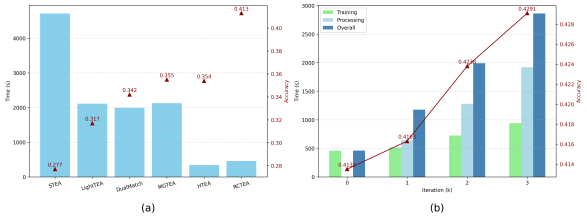

Figure 4: Efficiency evaluation on YAGO-WIKI180K. Specifically: (a) compares the time consumption and overall performance across different models; (b) illustrates the relationship between model iterations and performance for our approach.

6 Conclusion

Existing TEA models tend to underestimate the importance of relational and temporal features, as well as the inherent noise present in both. Our proposed RCTEA model features an efficient split-combined training framework and a dynamic weighting mechanism that adaptively balances different feature aspects. To mitigate noise, we introduce a joint neighborhood consensus strategy that enhances the representativeness of both views. Additionally, our dual-aspect seed selection method significantly reduces noise in general seed selection. Extensive experiments on TEA datasets validate the efficiency and effectiveness of our approach.

Limitation

This paper explores a richness measurement to guide graph neural networks in propagating and integrating informative features, thereby providing

explicit alignment signals to the model. However, when the TKGs evolve over time, the corresponding feature richness also needs to be incrementally refined, which may incur significant extra cost for frequently-updated TKGs. Hence, extending the proposed RCTEA model to evolving TKGs constitutes an important direction for future work. Moreover, investigating how to effectively balance or integrate various attention mechanisms—such as standard GAT, relation-specific attention, and other variants—is another promising research direction.

References

Li Cai, Xin Mao, Youshao Xiao, Changxu Wu, and Man Lan. 2023. 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*, pages 5021–5029.

Li Cai, Xinnian Mao, Meirong Ma, Hao Yuan, Jianchao Zhu, and Man Lan. 2022. A simple temporal information matching mechanism for entity alignment between temporal knowledge graphs. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2075–2086.

Zhuo Chen, Jiaoyan Chen, Wen Zhang, Lingbing Guo, Yin Fang, Yufeng Huang, Yichi Zhang, Yuxia Geng, Jeff Z Pan, Wenting Song, and 1 others. 2023. Meaformer: Multi-modal entity alignment transformer for meta modality hybrid. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3317–3327.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *arXiv preprint arXiv:2401.08281*.

Matthias Fey, Jan E Lenssen, Christopher Morris, Jonathan Masci, and Nils M Kriege. 2020. Deep graph matching consensus. *arXiv preprint arXiv:2001.09621*.

Congcong Ge, Xiaoze Liu, Lu Chen, Baihua Zheng, and Yunjun Gao. 2021. Make it easy: An effective end-to-end entity alignment framework. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 777–786.

Wei Jiang, Xinyi Gao, Guandong Xu, Tong Chen, and Hongzhi Yin. 2024. Challenging low homophily in social recommendation. *arXiv preprint arXiv:2401.14606*.

Mei Kobayashi and Koichi Takeda. 2000. Information retrieval on the web. *ACM computing surveys (CSUR)*, 32(2):144–173.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*, pages 2863–2872. PMLR.

Jiayun Li, Wen Hua, Fengmei Jin, and Xue Li. 2025. Htea: Heterogeneity-aware embedding learning for temporal entity alignment. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 982–990.

Bing Liu, Tiancheng Lan, Wen Hua, and Guido Zuccon. 2023a. Dependency-aware self-training for entity alignment. In *Proceedings of the Sixteenth ACM International conference on web search and data mining*, pages 796–804.

Xiaoze Liu, Junyang Wu, Tianyi Li, Lu Chen, and Yunjun Gao. 2023b. Unsupervised entity alignment for temporal knowledge graphs. In *Proceedings of the ACM Web Conference 2023*, pages 2528–2538.

Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. 2012. Recommender systems. *Physics reports*, 519(1):1–49.

Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. 2021. Boosting the speed of entity alignment 10×: Dual attention matching network with normalized hard sample mining. In *Proceedings of the Web Conference 2021*, pages 821–832.

Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. 2020a. 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*, pages 420–428.

Xin Mao, Wenting Wang, Huimin Xu, Yuanbin Wu, and Man Lan. 2020b. Relational reflection entity alignment. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1095–1104.

Xinnian Mao, Wenting Wang, Yuanbin Wu, and Man Lan. 2022. 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*, pages 825–838.

Sunita Sarawagi and 1 others. 2008. Information extraction. *Foundations and Trends® in Databases*, 1(3):261–377.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2019. Entity alignment between knowledge graphs using attribute embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 297–304.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. *stat*, 1050:4.

Kexuan Xin, Zequn Sun, Wen Hua, Bing Liu, Wei Hu, Jianfeng Qu, and Xiaofang Zhou. 2022. Ensemble semi-supervised entity alignment via cycle-teaching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4281–4289.

Chengjin Xu, Fenglong Su, and Jens Lehmann. 2021. 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*, pages 8999–9010.

Chengjin Xu, Fenglong Su, Bo Xiong, and Jens Lehmann. 2022. Time-aware entity alignment using temporal relational attention. In *Proceedings of the ACM Web Conference 2022*, pages 788–797.

Weixin Zeng, Xiang Zhao, Wei Wang, Jiuyang Tang, and Zhen Tan. 2020. Degree-aware alignment for entities in tail. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 811–820.

Weixin Zeng, Jie Zhou, and Xiang Zhao. 2024. Benchmarking challenges for temporal knowledge graph alignment. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 3103–3112.

A Model Details

A.1 Neighborhood Consensus De-noising

To enhance alignment consistency between entities and their local neighborhoods, we incorporate a neighborhood consensus model (Fey et al., 2020) to refine the dual-aspect encoders $\mathcal{M}_{stru}$ and $\mathcal{M}_{temp}$. This approach mitigates noise in the learned embeddings by encouraging aligned entities to exhibit similar neighborhood structures. Leveraging the complementary nature of structural and temporal features, we further propose a *dual-view neighborhood consensus strategy* that enables mutual reinforcement between the two modalities through joint training. This design facilitates bidirectional enhancement between structural and temporal signals, thereby improving the quality and robustness of the final entity representations.

In particular, given two TKGs $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$ and $\mathcal{G}' = (\mathcal{E}', \mathcal{R}', \mathcal{T}', \mathcal{F}')$, we first generate their structural and temporal feature embeddings using the corresponding encoders $\mathcal{M}_{stru}$ and $\mathcal{M}_{temp}$:

$$\mathbf{H}_{stru}, \mathbf{H}'_{stru} = \mathcal{M}_{stru}(\mathbf{A}_{(E)}, \mathbf{A}_{(R)}) \quad (17)$$

$$\mathbf{H}_{temp}, \mathbf{H}'_{temp} = \mathcal{M}_{temp}(\mathbf{A}_{(T)}, \mathbf{A}_{(I)}) \quad (18)$$

$\mathbf{A}_{(\mathcal{C})}$ is the bipartite adjacency matrix of the TKG for a specific feature type $\mathcal{C} \in \{E, R, T, I\}$.

Instead of relying on top-k entity retrieval from a single encoder to de-noise the corresponding feature based on neighborhood consensus, we jointly refine both encoders by concatenating their learned embeddings:

$$\mathbf{H}, \mathbf{H}' = [\mathbf{H}_{stru} \,||\, \mathbf{H}_{temp}], [\mathbf{H}'_{stru} \,||\, \mathbf{H}'_{temp}] \quad (19)$$

This approach leverages the complementary perspectives of the two encoders to enhance overall alignment performance. We adopt the FAISS framework (Douze et al., 2024) to perform efficient top-k embedding retrieval from large-scale TKGs, as the top-k candidates typically encapsulate the most informative signals for entity alignment:

$$\hat{\mathbf{S}} = Softmax(Faiss_k[\mathbf{H} \cdot \mathbf{H}']) \quad (20)$$

The resulting top-k probability score matrix, denoted by $\hat{\mathbf{S}}$, is computed using a softmax function over the retrieved similarities.

Finally, the consensus model utilizes a multilayer perceptron (MLP) to iteratively refine the similarity matrix, leveraging the injective node distances derived from the propagated features of the two TKGs to minimize the achieved embedding distances for two counterpart embeddings, as detailed below:

$$\boldsymbol{O}, \boldsymbol{O}' = \boldsymbol{\Phi}_{\theta_2}(\boldsymbol{I}_{|\mathcal{E}|}, \boldsymbol{A}_{(E)}),\ \boldsymbol{\Phi}_{\theta_2}(\hat{\boldsymbol{S}}^T \boldsymbol{I}_{|\mathcal{E}|}, \boldsymbol{A}_{(E)}) \quad (21)$$

$$\hat{S}^{l+1}_{i,j} = \hat{S}^{l}_{i,j} + \boldsymbol{\Phi}_{\theta_2}(d_{i,j}),\ \text{where } l \geq 0 \quad (22)$$

$$\mathbf{S}^{\mathbf{L}} = Softmax(\hat{\mathbf{S}}^{\mathbf{L}}) \quad (23)$$

Here, $d_{ij} = \vec{o_i} - \vec{o_j}$ measures the distance between two nodes, $\mathbf{A}_{(E)}$ is the adjacency matrix, $\mathbf{O}$ and $\mathbf{O}'$ are the reconstructed embeddings for the two TKGs, d_{ij} is the distance between the two reconstructed counterpart embeddings, $\mathbf{I} \sim \mathcal{N}(0, 1)$ is used for randomized embeddings, $\mathbf{\Phi}_{\theta_1}$ is a simple GNN, and $\mathbf{\Phi}_{\theta_2}$ is the defined MLP.

To train the dual-view neighborhood consensus module, we maximize the log-normalized probability scores of the seed alignments for refining the structural and temporal encoders, as defined below:

$$\mathcal{L}_{nc} = -\sum_{(e_i, e'_i) \in \mathcal{S}} log(S^L_{i,i'}) \quad (24)$$

A.2 RCTEA$^+$ with Iterative Training

Our RCTEA model can be further extended to an iterative training framework, denoted as RCTEA$^+$, which progressively incorporates high-quality seeds for model training. To this end, we design a *dual-aspect bi-directional seed selection* mechanism to exploit the complementary nature of the two orthogonal features—structural and temporal information—to collaboratively eliminate the influence of noisy pseudo-labels. In particular, we employ the two specialized feature encoders: $\mathcal{M}_{stru}$ for identifying structural-based seed alignments and $\mathcal{M}_{temp}$ for temporal-based matched pairs. To maintain high-quality seed selection, we impose bi-directional constraints (Mao et al., 2020a) derived from both structural and temporal views, ensuring the consistency and reliability of the selected seeds. The complete seed selection procedure is outlined in Algorithm 1:

Algorithm 1: Seed Selection

Input: S_r, S_t: structural and temporal similarity matrices, P: pre-aligned seed pairs

Output: $\hat{P}$: selected seed pairs

$\hat{P} \leftarrow P$;
for $e \in E$ **do**
  $e_r^* \leftarrow \arg\max_{e'} S_r(e, e')$, $e_t^* \leftarrow \arg\max_{e'} S_t(e, e')$;
  if $\text{argmax}_{e''} S_r[e'', e_r^*] = e$ ***and*** $\text{argmax}_{e''} S_t[e'', e_t^*] = e$ ***and*** $e_r^* = e_t^*$ **then**
    $\hat{P} \leftarrow \hat{P} \cup \{(e, e_r^*)\}$;
return $\hat{P}$;

A.3 Sinkhorn Operator

The Sinkhorn operator utilizes a fast and effective way to compute the assignment problem. It iteratively normalizes rows and columns to generate results in a doubly stochastic matrix, as follows:

$$
\begin{aligned}
&Sinkhorn^0(S) = \exp(S),\\
&Sinkhorn^m(S) = \mathcal{N}_c(\mathcal{N}_r(Sinkhorn^{m-1}(S))),\\
&Sinkhorn(S) = \lim_{x\to\infty} Sinkhorn^m(S)
\end{aligned}
$$

where $\mathcal{N}_r(S) = S \oslash (S1_N 1_N^T)$, $\mathcal{N}_c(S) = S \oslash (1_N 1_N^T S)$ are row and column-wise normalization operators of a matrix, $\oslash$ represents the element-wise division, and 1_N is a column vector of ones.

B Proof of the Richness Measurement

Theorem 1 (Effect of Neighborhood Diversity on Reference Weight Contribution)**.** *Let an entity have a neighborhood degree of fixed size N and neighborhood number n, where w_r of its count are reference entities. We define the log-normalized weight assigned to the reference embedding as:*

$$\alpha_r = \frac{\log(w_r + 1)}{\sum_{i=1}^{n} \log(w_i + 1)}$$

Consider two cases:

- ***Low-diversity case:*** *All $N - w_r$ non-reference degrees are identical (i.e., low diversity).*
- ***High-diversity case:*** *All $N - w_r$ non-reference degrees are unique (i.e., high diversity).*

Then, under log-normalization, the weight α_r assigned to the reference embedding is strictly smaller in the high-diversity case than in the low-diversity case, provided that $N - w_r > 1$. Consequently, the contribution of the reference embedding to the target entity representation is lower under high-diversity conditions, increasing the distance between the target entity representation and the reference embedding.

Proof. Let N be the total number of neighbors' degrees (fixed), and w_r be the count of reference degrees. We define the normalized weight assigned to the reference embedding in each case.

Case 1: Low Diversity (all $N - w_r$ non-reference degrees are the same):

$$\alpha_r^{\text{low}} = \frac{\log(w_r + 1)}{\log(w_r + 1) + \log(N - w_r + 1)}$$

Case 2: High Diversity (all $N - w_r$ non-reference degrees are distinct):

$$\alpha_r^{\text{high}} = \frac{\log(w_r + 1)}{\log(w_r + 1) + (N - w_r)\log(2)}$$

We want to show that:

$$\alpha_r^{\text{high}} < \alpha_r^{\text{low}}$$

This is equivalent to proving:

$$(N - w_r)\log 2 > \log(N - w_r + 1)$$

This inequality holds for $N - w_r > 1$, since for all $x \geq 2$:

$$x \log 2 > \log(x + 1)$$

Therefore:

$$\alpha_r^{\text{high}} < \alpha_r^{\text{low}} \quad \text{whenever} \quad N - w_r > 1$$

Thus, increasing neighborhood diversity (with a fixed reference count w_r) lowers the normalized weight α_r, reducing the contribution of the reference embedding and increasing its relative distance to the target entity representation. □

Theorem 2 (Increasing Neighborhood Size Reduces Reference Embedding Contribution)**.** *Let the final embedding e of an entity be computed from a set of n neighborhood embeddings $\{e_i\}_{i=1}^n$ and a reference embedding e_r as follows:*

$$e = \frac{1}{Z}\left(\sum_{i=1}^{n} \log(w_i+1)e_i + \log(w_r+1)e_r\right)$$
$$\textit{where} \quad Z = \log(w_r+1) + \sum_{i=1}^{n} \log(w_i+1)$$

Assume each neighborhood embedding e_i is a mixture:

$$e_i = \alpha e_r + (1-\alpha)e_i^{other}, \quad \textit{where } \alpha \in [0,1)$$

Then, the total contribution of the reference embedding e_r to e is:

$$W_{e_r} = \frac{\log(w_r+1) + \sum_{i=1}^{n} \alpha \log(w_i+1)}{\log(w_r+1) + \sum_{i=1}^{n} \log(w_i+1)}$$

If we assume all neighborhood counts are equal, i.e., $\log(w_i+1) = x > 0$ for all i, and $\log(w_r+1) = A$, then:

$$W_{e_r}(n) = \frac{A + n\alpha x}{A + nx}$$

Then, $W_{e_r}(n)$ is a strictly decreasing function of n for fixed $A > 0$, $x > 0$, and $\alpha \in [0,1)$. That is, as the number of neighbors increases, the contribution of the reference embedding decreases.

Proof. Let us define:

$$W(n) = \frac{A + n\alpha x}{A + nx}$$

We compute the derivative of $W(n)$ with respect to n:

$$\frac{dW}{dn} = \frac{d}{dn}\left(\frac{A + n\alpha x}{A + nx}\right) = \frac{\alpha x(A+nx) - x(A+n\alpha x)}{(A+nx)^2}$$

We simplify the numerator:

$$\begin{aligned} &\alpha x(A+nx) - x(A+n\alpha x) \\ &= \alpha xA + \alpha n x^2 - xA - n\alpha x^2 \\ &= \alpha xA - xA \\ &= xA(\alpha - 1) < 0 \quad \text{since } \alpha < 1 \end{aligned}$$

Thus, $\frac{dW}{dn} < 0$, meaning $W(n)$ is strictly decreasing with n. Therefore, as the number of neighbors increases, the normalized contribution of the reference embedding e_r to the final embedding e decreases. □

Theorem 3 (Smaller Reference Portion Reduces Cosine Similarity with Reference Embedding)**.** *Let $e_r \in \mathbb{R}^d$ be a unit-norm reference embedding, i.e., $\|e_r\| = 1$. Let $e^{other} \in \mathbb{R}^d$ be an embedding orthogonal to e_r, i.e., $\langle e_r, e^{other}\rangle = 0$. We define the mixed embedding as:*

$$e(\alpha) = \alpha e_r + (1-\alpha)e^{other}, \quad \textit{with } \alpha \in [0,1]$$

Then, the cosine similarity between $e(\alpha)$ and e_r is strictly increasing with α. In other words, decreasing the reference portion α results in a lower cosine similarity with the reference embedding.

Proof. We compute the cosine similarity between $e(\alpha)$ and e_r:

$$\cos(\theta) = \frac{\langle e(\alpha), e_r\rangle}{\|e(\alpha)\| \cdot \|e_r\|} = \frac{\langle \alpha e_r + (1-\alpha)e^{\text{other}}, e_r\rangle}{\|e(\alpha)\|}$$

Using orthogonality: $\langle e_r, e^{\text{other}}\rangle = 0$, and $\|e_r\| = 1$, we get:

$$\cos(\theta) = \frac{\alpha}{\|e(\alpha)\|}$$

Compute the norm:

$$\|e(\alpha)\|^2 = \|\alpha e_r + (1-\alpha)e^{\text{other}}\|^2 = \alpha^2 + (1-\alpha)^2\|e^{\text{other}}\|^2$$

Let $C = \|e^{\text{other}}\|^2 > 0$. Then:

$$\cos(\theta) = \frac{\alpha}{\sqrt{\alpha^2 + (1-\alpha)^2 C}}$$

Now consider the function:

$$f(\alpha) = \frac{\alpha}{\sqrt{\alpha^2 + (1-\alpha)^2 C}}$$

This function is strictly increasing with $\alpha \in [0,1]$ for any constant $C > 0$. Therefore, decreasing α reduces the cosine similarity between $e(\alpha)$ and e_r. □

Table 4: Dataset Statistics: #Entities, # Relations, and #Facts indicate the number of entities, relations, and facts in two TKGs. Additionally, $|\mathcal{S}|$: number of ground-truth aligned entity pairs; $|\mathcal{P}|$: number of temporal points; λ_t: proportion of temporal triples; λ_{TH}: proportion of heterogeneous temporal triples.

Dataset	#Entities	#Relations	#Facts	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{P}\rvert$	λ_{t_1}	λ_{t_2}	λ_{TH}
DICEWS	9,517 | 9,573	247 | 246	307,552 | 307,553	8,566	4,017	1	1	0
YAGO-WIKI50K	49,629 | 49,222	11 | 30	221,050 | 317,814	49,172	245	1	1	0
YAGO-WIKI20K	19,493 | 19,929	32 | 130	83,583 | 142,568	19,462	405	.634	.825	0
YAGO-WIKI180K	187,987 | 187,977	32 | 261	924,935 | 1,636,020	187,977	1,000	.218	.265	.082
BETA	42,666 | 42,297	257 | 45	199,879 | 162,320	40,364	967	.645	.350	.390[2]

C Experimental Settings

C.1 TEA Datasets

In this section, we present the specifications and construction procedures of various publicly available TEA datasets, including the homogeneous datasets (DICEWS, YAGO-WIKI50K, YAGO-WIKI20K) and the more realistic and challenging heterogeneous datasets (YAGO-WIKI180K and BETA). Table 4 compares the statistics of the five TEA datasets.

DICEWS (Xu et al., 2021). Integrated Crisis Early Warning System (ICEWS) is a publicly available, large-scale event-based database that contains political events with specific time annotations extracted from millions of real-world news stories. ICEWS05-15 is a subset of ICEWS, consisting of 100,904 entities, 251 relations, 4,017 timestamps, and 461,329 facts from 2005 to 2015. It is commonly used as a TKG benchmark dataset in the community. DICEWS is built based on ICEWS05-15. Initially, ICEWS05-15 is randomly divided into two subsets, $\mathcal{Q}_1$ and $\mathcal{Q}_2$, of similar size, with an overlap ratio of 50% in the number of shared quadruples between $\mathcal{Q}_1$ and $\mathcal{Q}_2$. However, this dataset includes all temporal triples and inherently ignores the temporal heterogeneity issue (Li et al., 2025), as all non-temporal triples are excluded from the common ICEWS05-15 dataset.

YAGO-WIKI50K (Xu et al., 2021). Wikidata is a free and open knowledge base that stores structured data from Wikipedia. Similarly, YAGO is an open-source knowledge base extracted from Wikipedia and WordNet. Both contain a large number of identical entities represented in different surface forms, and some facts are associated with temporal information in various formats, such as timestamps and time intervals. (Lacroix et al., 2018) constructed a large-scale TKG dataset from Wikidata, consisting of 432,715 entities, 407 relations, and 1,724 timestamps (retaining only year information) by filtering out high-frequency entities and relations. The entire dataset includes over 7 million triples in total, with around 10% associated with temporal information. Based on this, YAGO-WIKI50K is built from YAGO and Wikidata, starting with the top 50,000 entities selected based on their frequencies in the dataset and linked to their corresponding Wikidata counterparts using QIDs. Two TKGs are generated by filtering out facts where entities appear in only one KG. Temporal information is then attached from the Wikidata part to the YAGO counterpart triples, and non-temporal triples are removed from the filtered TKGs. The YAGO-WIKI50K dataset is a fully temporal dataset without temporal heterogeneity, as the temporal information in the YAGO part is derived from the Wikidata counterpart triples.

YAGO-WIKI20K (Xu et al., 2022). YAGO-WIKI20K is constructed using a similar procedure as YAGO-WIKI50K, with the primary difference being that the number of selected Wikidata entities is reduced to 20,000 while retaining the non-temporal facts in the two TKGs. Consequently, YAGO-WIKI20K is a temporally-hybrid dataset yet without the issue of temporal heterogeneity.

YAGO-WIKI180K (Li et al., 2025). The YAGO-WIKI180K dataset is constructed using the same Wikidata source as the YAGO-WIKI50K and YAGO-WIKI20K datasets, with a key distinction in how temporal information is handled. The YAGO triples are extracted from YAGO3 (Suchanek et al., 2007). All available YAGO triples are first retrieved, followed by the extraction of temporal information from associated file records that contain timestamp details for each fact. This procedure enables the temporal information to be derived directly from the YAGO portion, rather than incorporating temporal data from the WIKI counterpart. Using available QID links, the YAGO and WIKI

[2]BETA's heterogeneous portion of temporal triples is measured at the date level.

TKGs are aligned by selecting triples in which YAGO entities have corresponding counterparts in the WIKI graph. The resulting dataset includes approximately 19,000 shared entities, with 32 and 261 relations on the YAGO and WIKI sides, respectively, and includes a partial set of temporal triples. Compared to other existing TEA datasets, this construction process yields nearly four times as many entities. Furthermore, due to the independent sourcing of temporal information from the two KGs, the dataset exhibits temporal heterogeneity in 8.2% of the aligned triples. This level of heterogeneity more accurately reflects the variation commonly encountered in real-world settings.

BETA (Zeng et al., 2024). Unlike all the other datasets which standardizes timestamps to year-level granularity, BETA retains multi-granular temporal information, with 28% of temporal facts in the Wikidata subset and 14% in the YAGO subset containing precise month-day annotations. This fine-grained temporal data introduces more diverse information in alignment tasks. In terms of entity distribution, BETA exhibits a more realistic spread: while most entities in YAGO-WIKI50K are associated with a single dominant temporal relation (e.g., member_of_sports_team), BETA contains entities with no temporal facts as well as those linked to multiple temporal relations similar to YAGO-WIKI180K. This distribution supports the design of new alignment scenarios beyond the original single-relation setting. Additionally, relation frequency in BETA is more balanced: whereas member_of_sports_team and plays_for account for over 98% of all quadruples in YAGO-WIKI50K, the top five relations in BETA collectively contribute to less than 35%, reflecting a broader and more domain-general knowledge structure. These characteristics make BETA a more challenging and representative benchmark for the TEA task.

C.2 Parameter Settings

We report the detailed parameter settings for the YAGO-WIKI180K and BETA datasets to support better reproducibility of the experimental results.

For the YAGO-WIKI180K dataset, we set the embedding dimension to $d = 50$, GNN depth to $l = 2$, number of attention layers to $L = 1$, batch size to $b = 512$, and dropout rate to drop $= 0.3$. The model is optimized using RMSprop with a learning rate $\eta = 0.005$. For the neighborhood consensus module, we select the top-$k = 15$ most similar entities. The randomized embedding dimension is set to $d_r = 32$, and the number of MLP propagation steps to $s = 5$, with the MLP comprising 2 layers and a hidden dimension of 32. For the dynamic weighting module, the MLP also consists of 2 layers but uses a smaller hidden dimension of 4. In the semi-supervised setting, the number of training iterations is set to $t = 2$ to balance effectiveness and efficiency. The training schedule includes $E_{\text{rel}} = 20$, $E_{\text{temp}} = 60$, $E_{\text{mix}} = 5$, and $E_{\text{neigh}} = 90$ epochs for the respective modules.

For the BETA dataset, the main differences lie in the neighborhood consensus component, where we set the MLP propagation steps to $s = 25$ and select the top-$k = 45$ most similar entities. The training epochs are adjusted accordingly: $E_{\text{rel}} = 8$, $E_{\text{temp}} = 16$, $E_{\text{mix}} = 1$, and $E_{\text{neigh}} = 110$.

D Experimental Results

D.1 Preliminary Study

We conduct a preliminary study on the YAGO-WIKI180K dataset to examine the influence of feature richness on alignment performance, In particular, we group entities based on richness levels of their core features: structural (number of neighboring entities or relation types) and temporal (number of temporal points/intervals), and evaluate the alignment accuracy (Hit@1) of different groups respectively. As illustrated in Table 5, entities with lower feature richness tend to generate less reliable alignments. Propagating their supervision signals to neighboring entities naturally hinders the model's ability to capture precise and informative features, resulting in suboptimal representation.

Table 5: Impact of feature richness (Hit@1).

Feature	Low-richness	Medium-richness	High-richness
Entity	.0424	.2109	.2730
Relation	.0005	.0009	.0022
Temporal	.0026	.0354	.4675

Table 6: Uniquely correct cases of feature encoders: Hop-1/2 sizes, temporal values indicate number of hop-1/2 neighbor and different temporal values

Feature	# Correct	Hop-1 Size	Hop-2 Size	# Temp
Structural	9525	5	758	1
Temporal	1058	5	533	3
Mixed	9464	4.5	533.5	1

To further investigate the effectiveness of different features in the alignment process, we conduct another preliminary study on the YAGO-WIKI180K dataset, comparing the alignment accuracy of three feature encoders: structural, temporal, and mixed (a simple concatenation of both fea-

Table 7: TEA results on DICEWS and WY50K. The best results are highlighted in bold. Underline indicates the second-best results.

	DICEWS (1K)			DICEWS (200)			YAGO-WIKI50K (5K)			YAGO-WIKI50K (1K)		
MODEL	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
Dual-AMN	.779	.716	.893	.733	.668	.854	.922	.897	.964	.834	.755	.890
LightEA	.833	.785	.918	.779	.721	.878	.960	.948	.979	.902	.878	.945
STEA	.941	.928	.960	.941	.927	.961	.954	.935	.986	.916	.887	.966
TEA-GNN	.911	.887	.947	.902	.876	.941	.909	.879	.961	.775	.723	.871
TREA	.933	.914	.966	.927	910	960	.958	.940	.989	.885	.840	.937
LightTEA	.959	.952	.970	.955	.949	.966	**.990**	**.986**	**.997**	.977	.969	.989
DualMatch	**.961**	.953	**.973**	**.961**	**.953**	**.974**	.986	.981	.996	.961	.947	.984
MGTEA	-	-	-	.960	.951	**.974**	-	-	-	.960	.947	.982
HTEA	.943	.932	.959	.928	.914	.949	.979	.971	.991	.950	.931	.980
RCTEA	.955	.946	.970	.947	.936	.964	.987	.982	.995	.968	.957	.987
RCTEA^{+}(iter)	**.961**	**.954**	**.973**	.958	.950	.970	.987	.981	.996	**.979**	**.971**	**.992**

tures). Table 6 reports the number of entity pairs that each encoder aligns correctly on its own while the other encoders fail. Hop-1/2 Size denotes the average number of entities in the hop-1/2 neighborhood, reflecting the structural richness of the central entity; # Temp represents the average number of temporal values that the central entity is associated with, indicating its temporal richness. The results in Table 6 verify the necessity of entity-specific weighting for feature fusion and, moreover, illustrate a clear correlation between feature richness and model performance: Structural encoder is more effective for alignment cases with rich neighborhood, while temporal encoder excels in matching temporally-diverse entities.

D.2 Experimental Results on Homogeneous TEA Datasets

In this section, we evaluate the performance of our model, RCTEA, on two widely used homogeneous TEA datasets: DICEWS and YAGO-WIKI50K, with certain number of seed alignments. As shown in Table 7, RCTEA achieves competitive performance in these relatively easy datasets, ranking among the top two models across various experimental settings. Notably, both the iterative and base versions of RCTEA exhibit strong representational capacity, even without relying on an explicit time-consuing temporal encoder, which significantly enhances model efficiency (Mao et al., 2022; Liu et al., 2023b).

D.3 Richness-stratified Analysis

In this section, we present an experiment to evaluate the effectiveness of our proposed richness measurement. Specifically, we compare the similarity between type-specific entity embeddings and their corresponding type-level reference embeddings. For each feature type, entities are divided into three groups—low, medium, and high richness—based on their respective richness scores. In particular, we define three richness levels: low, medium, and high, according to the following bins:

- **Entity (E):** $[0, 3)$, $[3, 8)$, $[8, +\infty)$
- **Relation (R):** $[0, 2)$, $[2, 4)$, $[4, +\infty)$
- **Temporal point (T):** $[0, 1)$, $[1, 5)$, $[5, +\infty)$
- **Temporal interval (I)** $[0, 3)$, $[3, 5)$, $[5, +\infty)$

Table 8 reveals that entities with higher feature richness tend to exhibit smaller similarity compared to the reference embedding, whereas feature-sparse entities remain larger. This observation supports our hypothesis that feature-rich entities possess more diverse and distinctive embedding represent than the typical one. Furthermore, the proposed similarity metric proves to be effective in

Table 8: Grouped similarity of feature-specific embedding and reference embedding in YAGO-WIKI180K (Metric: Average cosine similarity).

Feature	High-rich	Med-rich	Low-rich
Entity	-0.0240	-0.0137	0.0737
Relation	-0.2080	-0.0424	-0.0376
Temporal point	-0.0452	0.0298	0.4373
Temporal interval	0.0202	0.0358	0.2862

Table 9: Edge weight after richness-guided attention.

Entity	Neighbour Size	Attention
Mexico_national_football_team	307	0.3457
Toros_Neza	50	0.3348
Mexico_national_under-17	33	0.3195

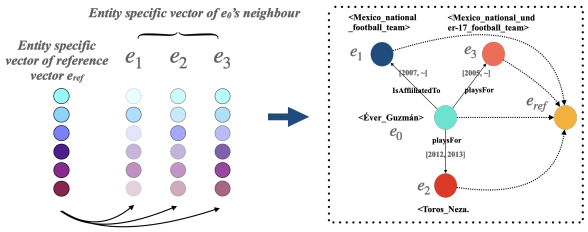

Figure 5: The triple details for entity *Éver_Guzmán*.

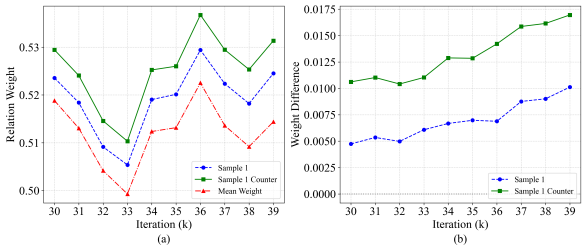

Figure 6: (a) Relation weighting for selected entity vs. Iteration. (b) Average weighting difference vs. Iteration.

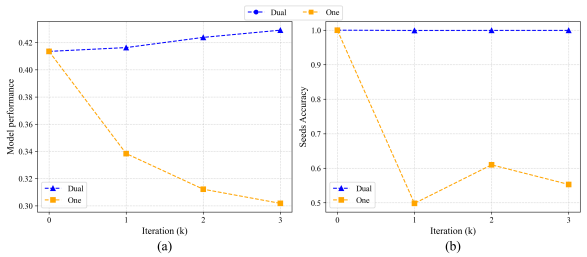

Figure 7: (a) Model Performance vs. Iteration, illustrating the impact of iterations on overall model performance. (b) Seed Accuracy vs. Iteration, showing how iterations affect overall seed accuracy.

quantifying the potential information richness of feature-specific embeddings.

D.4 Case Study for Richness-guided Attention

In this section, we focus on the richness-guided attention mechanism, specifically in the context of the entity feature. As a case study, we examine the entity *Éver_Guzmán* from the YAGO TKG, which corresponds to index 81890.

Details of this entity's associated triples are illustrated in Figure 5, where it is shown to have three neighbors. As presented in Table 9, our attention mechanism successfully captures the nuanced differences among the neighboring entities, with the assigned attention weights being approximately proportional to the number of neighbors associated with each.

D.5 Case Study for Adaptive Weighting

In this section, we analyze the dynamic weighting behavior of structural and temporal features during training in YAGO-WIKI180K. We focus on a representative example—entity index 2000—along with its ground-truth counterpart and the average Relation feature weights across all entities from iteration 30 to 39. As shown in Figure 6 (a), relation features consistently receive slightly higher weights than temporal features, reflecting their richer and less noisy characteristics. Although the weights fluctuate across iterations, they remain centered around 0.5, indicating the model's ability to adaptively balance feature contributions. Figure 6 (b) shows the weight difference between the selected entity and its counterpart, with similar trends suggesting the model captures consistent patterns that aid accurate alignment. This study highlights the effectiveness of the dynamic weighting strategy throughout the training process.

D.6 Seed Accuracy Analysis

We investigate the effectiveness of dual-aspect bi-directional seed accuracy in this section. Specifically, we compare the performance of using $\mathcal{M}_{stru}$ and $\mathcal{M}_{temp}$ for seed refinement (Dual) versus only using $\mathcal{M}_{mix}$ for seed refinement in YAGO-WIKI180K. As shown in Figure 7(a), the overall performance consistently improves with increasing iterations under dual-aspect refinement. In contrast, using only the $\mathcal{M}_{mix}$ model for seed refinement leads to a significant decline in performance. This trend is also evident in Figure 7(b), where the seed accuracy remains nearly perfect for the dual-view split-model approach but deteriorates substantially when relying solely on the $\mathcal{M}_{mix}$ model. These results highlight the critical role of seed quality in determining overall model performance.