TGformer: A Graph Transformer Framework for Knowledge Graph Embedding

Fobo Shi, Duantengchuan Li, Xiaoguang Wang, Bing Li, and Xindong Wu, *Fellow, IEEE*

***Abstract*—Knowledge graph embedding is efficient method for reasoning over known facts and inferring missing links. Existing methods are mainly triplet-based or graph-based. Triplet-based approaches learn the embedding of missing entities by a single triple only. They ignore the fact that the knowledge graph is essentially a graph structure. Graph-based methods consider graph structure information but ignore the contextual information of nodes in the knowledge graph, making them unable to discern valuable entity (relation) information. In response to the above limitations, we propose a general graph transformer framework for knowledge graph embedding (TGformer). It is the first to use a graph transformer to build knowledge embeddings with triplet-level and graph-level structural features in the static and temporal knowledge graph. Specifically, a context-level subgraph is constructed for each predicted triplet, which models the relation between triplets with the same entity. Afterward, we design a knowledge graph transformer network (KGTN) to fully explore multi-structural features in knowledge graphs, including triplet-level and graph-level, boosting the model to understand entities (relations) in different contexts. Finally, semantic matching is adopted to select the entity with the highest score. Experimental results on several public knowledge graph datasets show that our method can achieve state-of-the-art performance in link prediction.**

***Index Terms*—Knowledge graph, graph transformer, link prediction.**

I. Introduction

Knowledge Graph Embedding (KGE) serves as an efficient method for representing multi-relational graph [1], which finds applications in numerous downstream tasks. These tasks rely on knowledge modeling as a potent approach to incorporate common-sense knowledge. For instance, KGE is valuable in recommendation systems [2], [3] and enhancing large language models [4], [5].

Knowledge Graphs (KGs) like WordNet [6] and Freebase [7] offer publicly available datasets for storing information in the form of (*subject entity, relation, object entity*), typically seen as ordered sequences. In KGs, it's common for an entity to appear in multiple different triplets. In real-world KGs, entities often appear multiple times in distinct triplets, and we refer to such entities as **co-occurring entities**. However, these entities usually assume different roles in various triplets, potentially creating ambiguities when interpreting them in different contexts [8]. Therefore, it is necessary to construct corresponding knowledge representations for different entities and relational contexts, in order to capture the meaning of each triplet in its unique context.

A plethora of remarkable KGE models have been introduced. Early KGE models primarily operated at the triplet level and included models like TransE [9], TuckER [10], ConvE [11], and InteractE [12]. These models aim to construct their feature representations by focusing on the interactions between different subject entities and relations.

For instance, let's consider the prediction of the triplet (*Taylor Swift*, *PlaceOfBirth*, ?) as depicted in Fig. 1(a). In this scenario, the birthplace of *Taylor Swift* is predicted based solely on the subject entity *Taylor Swift* and the relation *PlaceOfBirth* within the triplet. However, despite their reliance on entity-relation interactions, these approaches often struggle to deliver satisfactory results when confronted with intricate relations (e.g., one-to-many, many-to-many relations, etc.) within knowledge graphs. Models like ComplEx [13] and RotatE [14] offer theoretical underpinnings for entity-relation interactions, rooted in relational patterns. These models aim to theoretically embody a variety of relational patterns, encompassing symmetry/antisymmetry, inversion, and composition. Nonetheless, extensive experiments discussed in [15] have uncovered a disconnect between these theoretically complete models and their practical performance. This difference can be attributed to the locality of entity-relation interactions. *It is difficult to obtain knowledge representations that satisfy multiple relation patterns through a single entity-relation interaction solely* [16]. As a result, recent research has increasingly focused on graph neural networks that can capture global knowledge representations utilising graph-level structure.

In recent years, there has been a growing trend towards seeking global entity information within graph-level structures [17], [18] to mitigate the limitations of localized interactions. These methods leverage graph neural networks (GNNs) [19], [20], [21] to represent the knowledge graph's topology, treating entities as nodes and relations as edges. Several studies [18], [22], [23] assert that incorporating GNNs can effectively capture graph-level structures, outperforming triplet-level KGE models. This is primarily attributed to the multi-layer GNN's capacity

Received 17 January 2024; revised 15 August 2024; accepted 15 October 2024. Date of publication 4 November 2024; date of current version 26 November 2024. This work was supported by the Major Project of the National Social Science Fund, China under Grant 21&ZD334. Recommended for acceptance by X. Yi. *(Corresponding author: Duantengchuan Li.)*

Fobo Shi, Duantengchuan Li, and Xiaoguang Wang are with the School of Information Management, Wuhan University, Wuhan 430072, China, and also with the Intelligent Computing Laboratory for Cultural Heritage, Wuhan University, Wuhan 430072, China (e-mail: foboshi99@gmail.com; dtclee1222@gmail.com; wxguang@whu.edu.cn).

Bing Li is with the School of Computer Science, Wuhan University, Wuhan 430072, China (e-mail: bingli@whu.edu.cn).

Xindong Wu is with the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, Hefei 230009, China (e-mail: xwu@hfut.edu.cn).

Digital Object Identifier 10.1109/TKDE.2024.3486747

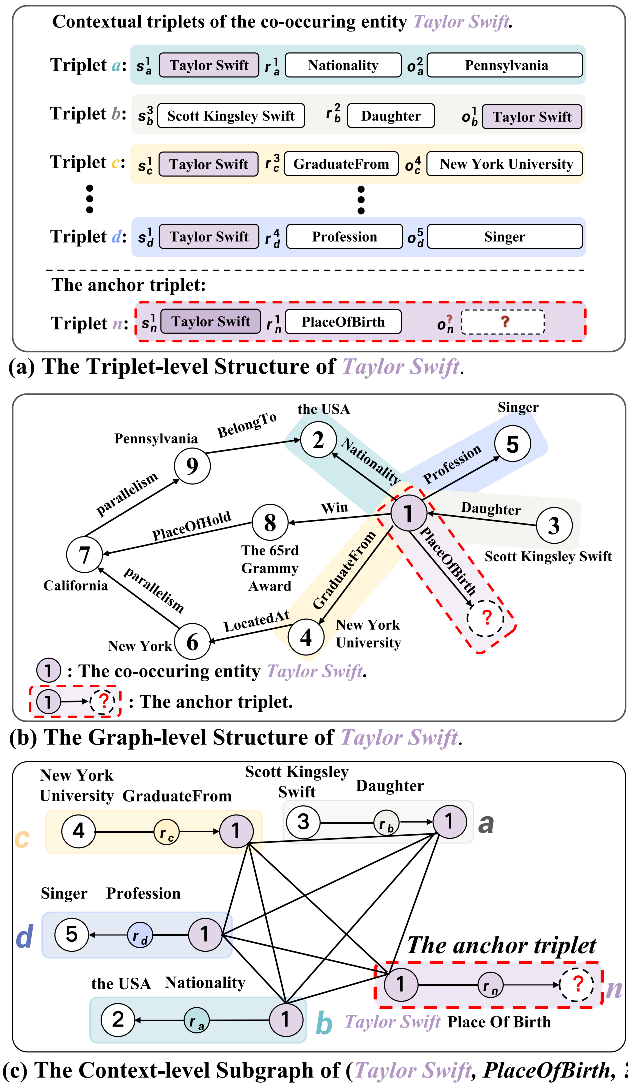

Fig. 1. Examples of knowledge graph triplet-level and graph-level structures with *Taylor Swift* as a co-occuring node in the anchor triplet (the triplet to be predicted)are in (a) and (b), respectively. (c) is the anchor triplet (*Taylor Swift, PlaceOfBirth, ?*) as the context-level subgraph of the node, which contains the triplet-level and graph-level structure of *Taylor Swift*.

for aggregating information and expanding the scope of entity-relation interactions.

Typically, traditional GNN-based KGE methods [8], [18], [24] follow a direct aggregation approach. They gather information from multi-hop neighbors in various contextual triplets to construct co-occurring entity representations. More precisely, as illustrated in Fig. 1(b), when predicting the triplet (*Taylor Swift*, *PlaceOfBirth*, ?), GNN-based techniques [17], [18], [24] consider all the multi-hop neighbors of the co-occurring entity *Taylor Swift* when modeling her representation. However, this approach inevitably introduces entity-independent noise into the entity representations. For example, features from entities like *California*, *New York University*, and *New York* are incorporated into *Taylor Swift*'s representation, potentially affecting the ranking of the correct answer, *Pennsylvania*, as these entities contain toponymically relevant features similar to *Pennsylvania*.

Based on the above analysis, there are two main challenges for existing KGE models to the comprehension of contextual information. On the one hand, the entity-relation interaction modeling approach based on the triplet-level structure is a partial interaction approach, as the entity-relation interactions of each sample only appear in a single triplet. Even though methods such as [12], [25] have demonstrated that increasing the number of entity interactions can boost the performance of the model, it is still essentially a short-range partial interaction, which makes it difficult to capture the complete global contextual information of the entities. On the other hand, although the graph operation used by the GNN-based KGE model is able to learn the interaction information of the anchor node with all its neighbors, it is difficult to extend the interactions globally due to the over-smoothing [26] limitation of the multi-layer graph neural network. Meanwhile, the indiscriminate way of considering the information aggregation of all neighbors incorporates a lot of unnecessary noise information [27], which leads to a bad performance.

To overcome the challenges posed by the aforementioned KGE models, we introduce a novel graph transformer framework, TGformer,[1] designed for knowledge graph embedding. TGformer's primary objective is to extend entity-relation interactions from the triplet level to a global scale while minimizing the aggregation of noisy information when capturing contextual triplets. The process begins by constructing a context-level subgraph(as shown in Fig. 1(c)) for each predicted triplet, centered around the co-occurring entity, which serves as its contextual anchor. These subgraphs are then fed into the triplet-level transformer module, which learns semantic features of the co-occurring entity within various contextual contexts. To address the issue of noisy contextual triplets within subgraphs, we implement a graph-level transformer that integrates relational features from the knowledge graph, thereby introducing an inductive bias. In parallel, when modeling the embedding of the co-occurring entity for the predicted triplets, we choose to aggregate co-occurring entities from different triplets rather than neighboring entities. This approach helps mitigate the inherent heterogeneity of knowledge graphs. Ultimately, we employ semantic matching to score the predicted entities, with the entity receiving the highest score becoming the final prediction.

In summary, the main contributions of this article are as follows:

- A novel graph transformer framework is proposed to build knowledge embeddings with triplet-level and graph-level structural features in the static and temporal KGs. As far as we know, it is the first to adopt a graph transformer to model KGE tasks.
- To alleviate the hindrance of noise information caused by graph-level aggregation, we construct a context-level subgraph for each to-be-predicted triplet, with prediction triplets as an anchor linking neighbor triplets that have similar contexts.
- we design a knowledge graph transformer network (KGTN), which successfully extends entity-relation interactions at the triplet-level to the global.

[1]Code will be available at https://github.com/dacilab/TGformer.

- After comprehensive experiments and extensive analysis of the static and temporal KG datasets, the TGformer model outperforms the state-of-the-art baseline model in the link prediction task.

II. Related Work

A. Knowledge Graph Embedding

The existing knowledge graph embedding models are mainly divided into three families: triplet-level structure models, graph-level structure models, and transformer-based models and their variants.

1) Triplet-Level Structure Models: Triplet-level structure models predict the corresponding missing entities by modeling tensor operations between entity embeddings and relation embeddings to simulate entity-relation interactions. TransE [9] is inspired by the translation invariance of word representations [28] and defines the structure of sequences within a triple as $\|\mathbf{s}+\mathbf{r}-\mathbf{o}\|_{L_1/L_2}$. In order to capture different relational patterns, ComplEx [13] projects the embedding of entities and relations into the complex domain, which theoretically demonstrates that tensor operations on the complex domain can satisfy multiple relational patterns of entity-relation interactions. Another interesting perspective is to view knowledge graph embedding as a three-way tensor decomposition problem, where TuckER [10] employs the Tucker tensor decomposition to inverse the entity-relation interaction process. At present, deep learning technology is widely used in various fields [29], [30], [31]. Inspired by image convolution in computer vision, ConvE [11] projects the spliced subject s and relation r into the embedding space of the object entity o by employing a convolutional neural network. Other similar KGE models, including DistMult [32], RotatE [14], HypER [33], InteractE [12], and NFE [34] which utilize different kinds of operators to incorporate the relation r into the embedding of subject entity s. However, all of these approaches deal independently with triplet-level structures within a single triplet. As a result, the interaction between entities and relations is localized, ignoring the associations between rich contextual triplets, thereby generating low-quality embeddings.

2) Graph-Level Structure Models: Intuitively, Graph Neural Networks (GNNs) appear to be a highly promising category of models for acquiring knowledge representations of entities and relations in knowledge graphs (KGs). Notable methods in this category include R-GCN [17], SACN [24], CompGCN [18], and SE-GNN [23]. Given the topological similarities between GNNs and KGs, it is a natural choice to employ GNNs for KGE tasks. R-GCN [17] aggregates neighbor information by using different diagonal matrices for various relations. To understand the significance of different neighbors, SACN [24] calculates weights during neighbor aggregation. CompGCN [18] takes into account both in-degree and out-degree nodes when aggregating neighbors, effectively extending GCN from undirected graphs to directed graphs like knowledge graphs. SE-GNN [23], aiming to enhance knowledge graph embeddings, introduces three levels of semantic evidence for entities, relations, and triplets. These semantic cues are used to design the neighborhood schema for the GNN aggregation function, resulting in more comprehensive knowledge representations. Furthermore, alternative approaches [16], [35], rooted in GNNs, have also successfully explored knowledge embeddings with graph-level structure by devising diverse neighbor aggregation techniques. However, after extensive experimentation on the relationship between adjacency tensors and graph-level structure, [36] found that while Graph Convolutional Neural Networks (GCNs) can indeed produce effective knowledge embeddings, their primary strength within the KGE model lies not in the capacity to model graph-level structure.

3) Transformer-Based Models and Their Variants: CoKE [37] focuses on embedding entities using their textual descriptions as a reference. To imbue the model with human-like comprehension of textual descriptions, [38] introduced a variant of the transformer to identify logical rules within the knowledge graph. N-Former [39] adopts knowledge distillation, concurrently leveraging transformer [40] and BERT [41] models, to learn entity and relation embeddings. These models incorporate textual descriptions of entities and relations as supplementary information, enhancing their ability to grasp the semantics of entities and relations in the human context.

Addressing the scalability challenges in knowledge graphs, [42] devised a transformer architecture with numerous self-attention heads to capture interactions between entities and relations in multiple directions. Consequently, these approaches often exhibit superior performance in language-based knowledge graphs like WN18 and WN18RR. However, it's important to note that these models are typically characterized by their large scale, necessitating significant storage space and computational resources to accommodate the additional textual auxiliary data. Additionally, transformer-based techniques tend to compress graph structures into sequential structures during knowledge representation learning, potentially causing the loss of valuable graph structure information inherent in KGs.

B. Graph Transformer

Transformer [40] is a recently emerged model that is extremely suitable for processing sequence-based data structures, and its powerful model representation has attracted a lot of attention in the fields of computer vision and natural language processing. Some existing works have successfully applied the transformer-based model to process data with Euclidean structure, such as text [41], and image [43]. The success of the transformer for euclidean structured data has led to an attempt to utilize it for more complex non-euclidean structured data, such as social networks, knowledge graphs, etc. Ying et al. [28] argued that the core of applying transformer architecture to graph data is how to encode graph structures correctly. They designed Graphormer to encode three graph structures, spatial encoding, edge encoding, and centrality encoding, into the attention mechanism and achieved better performance in graph representation learning. However, these existing approaches are proposed for homogeneous graphs, while the knowledge graph is a heterogeneous graph with complex and diverse types of entities and relations. Few models have been constructed for such heterogeneous graph relations as knowledge graphs. Therefore, how

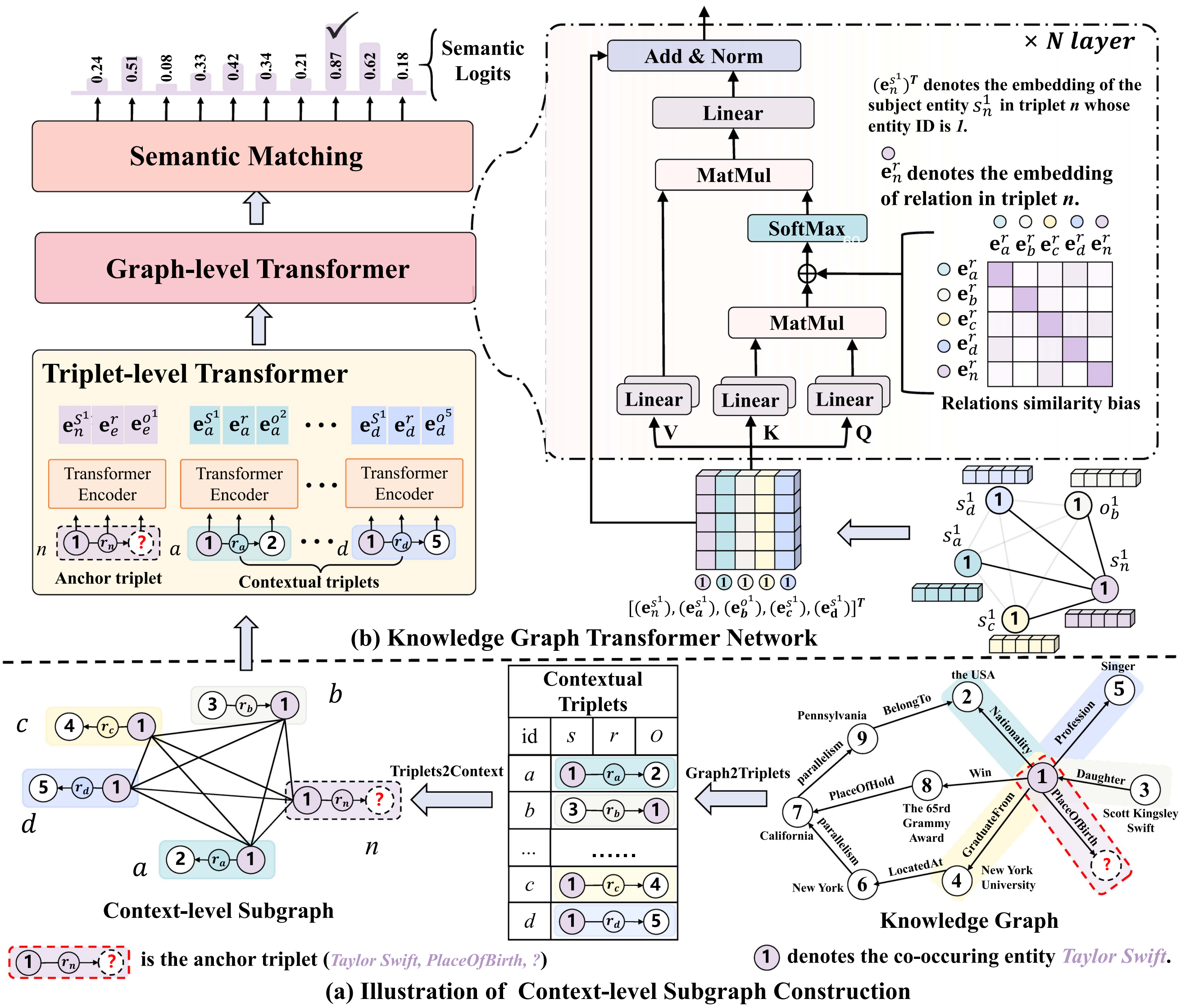

Fig. 2. A graph transformer framework for knowledge graph embedding. It consists of two components: (a) is the context-level subgraph construction process. The architecture of Knowledge Graph Transformer Network in (b).

to effectively perform feature aggregation for heterogeneous nodes of the knowledge graph is the key to enriching knowledge representation.

III. Our Approach

In this section, we introduce TGformer to build knowledge embedding with triplet-level and graph-level structural features in the knowledge graph. As shown in Fig. 2, we first construct a context-level subgraph to fully consider the multi-context information of subject entities and relations in the anchor triplets (Section III-A). On this basis, the Knowledge Graph Transformer Network (KGTN) is designed to enrich the embedding representation of entities and relations from two levels of triplet and graph structure, respectively (Section III-B). To fill the gap between graph-level representations and triplet-level representations in the knowledge graph, we employ a semantic matching approach as a decoder to score the embedding of the entity (Section III-C). Finally, The optimization and inference process of TGformer is described in Section III-D.

A. Context-Level Subgraph Construction

In real-world KGs, it is a common phenomenon that the same entity may appear multiple times in different triplets, and we call such entities **the co-occurring entity**. To facilitate the knowledge graph transformer network to aggregate neighbor triplets, we adopt whether there are co-occurring between triplets in KG as the judgment criterion to construct context-level neighbor subgraphs for **the anchor triplet** (the triplet to be predicted).

As shown in Fig. 2(a), the subgraph construction at the triplet level can be divided into two parts. Specifically, (1) Graph2Triplets: In this part, the subject entity of the anchor triplet n is treated as a co-occurring entity s_n^1. The other triplets within KG that contain the co-occurring entity s^1 are found, and these triplets are regarded as the neighbors of the anchor triplet

n. (2) Triplets2Context: For triplets with co-occurring entity s_n^1, a subgraph is constructed, which considers the information of the entities in different contexts. This completes the transformation of KG, a directed heterogeneous graph, into an undirected context-level subgraph.

B. Knowledge Graph Transformer Network

To extend entity-relation interactions from anchor triplets to a global scale and concurrently address the noise issue associated with GNN-based aggregation, we've introduced a Knowledge Graph Transformer Network (KGTN). This network comprises two key components: a triplet-level transformer and a graph-level transformer. The triplet-level transformer is designed to model more robust entity-relation interactions, thereby facilitating a deeper understanding of the significance of the co-occurring entity within different contextual triplets. On the other hand, the graph-level transformer's purpose is to expand entity-relation interaction information to a global context by aggregating semantic content pertaining to the co-occurring entity across various contexts. Additionally, in order to mitigate the impact of noise during the aggregation process, the relation embedding within the context triplet serves as a distinguishing feature for valuable contextual triplets.

1) Triplet-Level Transformer: Transformer [40] and BERT [41] have shown excellent capabilities in encoding data with sequence-type structures. Inspired by the MLM task in BERT, given the anchor triplet $j:(s_j^i, r_j, o_j^?)$. We replace the predicted entity with the mask token $[mask]$ to get the triplet $(s_j^i, r_j, o_j^{[mask]})$ as the input to the transformer encoder. The formula is as follows:

$$\begin{cases} \left[\mathbf{e}_j^{s^i}, \mathbf{e}_j^{r}, \mathbf{e}_j^{o^{[\text{mask}]}}\right] = \text{Tformer}\left(\mathbf{s}_j^i, \mathbf{r}_j, \mathbf{o}_j^{[\text{mask}]}\right), \text{SKG} \\ \left[\mathbf{e}_j^{s^i}, \mathbf{e}_j^{r}, \mathbf{e}_j^{o^{[\text{mask}]}}, \mathbf{e}_j^{t}\right] = \text{Tformer}\left(\mathbf{s}_j^i, \mathbf{r}_j, \mathbf{o}_j^{[\text{mask}]}, \mathbf{t}_j\right), \text{TKG} \end{cases}, \tag{1}$$

where $\mathbf{e}_j^{s^i}$, $\mathbf{e}_j^{r}$, and $\mathbf{e}_j^{o^{[mask]}} \in \mathbb{R}^{d\times 1}$ denote the embedding of the co-occurring entity $\mathbf{s}_j^i$, relation $\mathbf{r}_j$, and the entity $\mathbf{o}_j^{[mask]}$ in the anchor triplet j, respectively. $\text{Tformer}(\cdot)$ denotes the triplet-level transformer we designed, which is similar to the transformer encoder. It is worth noting that the process of encoding $\mathbf{e}_j^{o^{[mask]}}$ considers only the triplet-level structure information, ignoring the fact that the co-occurring entity s^i appears not only in the anchor triplet but as well as in the contextual triplets.

Therefore, we attempt to learn the embedding of the co-occurring entity s_j^i of the anchor triplet j based on its contextual triplets. Specifically, given the set of contextual triplets

$$\begin{aligned} N_{con\_triplets} = \{ (s_k^i, r_k, o_k^l) | i, \\ l = 1, \ldots, |\mathcal{E}|, k = 1, \ldots, |\mathcal{T}| \}, \end{aligned} \tag{2}$$

where s_k^i and o_k^l represent the subject entity and object entity within the contextual triplet k, with their corresponding entity IDs denoted as i and l. r_k stands for the relation in triplet k. Concerning the contextual triplets, we substitute the entity ID of the co-occurring entity in these triplets with a masking token, denoted as $[mask]$. Subsequently, we input this modified triplet into the triplet-level transformer to derive the embedding of the co-occurring entity within the contextual triplet.

If the co-occurring entity is the subject entity in the contextual triplets, that is

$$\left[\mathbf{e}_k^{s^{[mask]}}, \mathbf{e}_k^{r}, \mathbf{e}_k^{o^l}\right] = \text{Tformer}\left(\mathbf{s}_k^{[mask]}, \mathbf{r}_k, \mathbf{o}_k^l\right). \tag{3}$$

If the co-occurring entity is the object entity in the neighbor triplets, that is

$$\left[\mathbf{e}_k^{s^i}, \mathbf{e}_k^{r}, \mathbf{e}_k^{o^{[mask]}}\right] = \text{Tformer}\left(\mathbf{s}_k^i, \mathbf{r}_k, \mathbf{o}_k^{[mask]}\right), \tag{4}$$

where $\mathbf{e}_k^{s^{[mask]}}$ and $\mathbf{e}_k^{o^{[mask]}}$ represent the embeddings of the co-occurring entities $\mathbf{s}_k^{[mask]}$ and $\mathbf{o}_k^{[mask]}$ within different contextual triplets following the triplet-level transformer. By modeling the anchor triplet and its associated contextual triplets, we can obtain the embeddings of co-occurring entities in various contexts. Given that entity-relation interactions occur within each individual triplet, the knowledge representation of the co-occurring entity, derived from the anchor triplet and its contextual triplets, is limited and may only be applicable to the current context. To facilitate the learning of co-occurring entity knowledge embeddings that can be employed across multiple contexts, we have designed graph-level transformers to aggregate the embeddings of co-occurring entities from different contexts. In the following sections, we will elucidate the process of aggregating graph-level structural information from triplets.

2) Graph-Level Transformer: Entity-relation interactions typically exhibit spatial limitations in triplet-based models, which process each triplet independently, and thus may fail to capture the rich graph structure information in KGs. We design a graph-level transformer that extends entity-relation interactions to the global level. The features of entities and relations are then employed as inductive biases, which helps the model incorporate graph structure information into the transformer while effectively mitigating the introduction of noisy information due to global graph operations.

Knowledge Graph Structure Learning: As shown in Fig. 2(b), we connect the embedding of the co-occurring entity in the anchor triplet and its contextual triplet together as the input to the graph-level transformer. Since the order of the input sequences has no substantial effect on learning the embedding of the co-occurring entity (there is no obvious ordering property of the context information in the triplet-level subgraph), the graph-level transformer removes the positional encoding of the input sequences in the traditional transformer. The formula is that

$$\mathbf{H}^{(0)} = \left[\mathbf{e}_k^{s^i}, \mathbf{e}_j^{s^i}, \ldots \mathbf{e}_n^{s^i}\right]^{\mathrm{T}}, \tag{5}$$

where $\mathbf{H}^{(0)} \in \mathbb{R}^{n\times d}$ denotes the layer 0 input to the graph-level transformer, which is composed of the $d-$dimensional embedding of n the co-occurring entities in the anchor triplet and its contextual triplets. $\mathbf{e}_k^{s^i}$ is the embedding of the co-occurring entity in the anchor triplet, $\mathbf{e}_j^{s^i}$ and $\mathbf{e}_n^{s^i}$ are the embeddings of the co-occurring entity in the contextual triplets. The number of selected contextual triplets will vary depending on the dataset, which will be discussed in detail in the experimental section.

On this basis, we model knowledge graph structure information from two perspectives, namely entity-level and relation-level. Among them, the entity-level similarity is used as the adjacency matrix of the subgraph, and the relation-level similarity as knowledge graph inductive bias.

Entity-level similarity. The most valuable co-occurring entity embedding is determined by calculating the similarity between the co-occurring entity in the anchor triplet and the co-occurring entities in the contextual triplets. This similarity can be computed as follows:

$$\alpha_{jk}^{ent} = \frac{\exp\left(\left(\mathbf{e}_k^{s^i}\right)^{\mathrm{T}}\mathbf{e}_j^{s^i}\right)}{\sum_{\mathbf{e}_k^{s^i}\in N_{ent}}\exp\left(\left(\mathbf{e}_k^{s^i}\right)^{\mathrm{T}}\mathbf{e}_j^{s^i}\right)}, \tag{6}$$

where N_{ent} denotes the set of co-occurring entity embeddings of the contextual triplets. $\mathbf{e}_j^{s^i}$ and $\mathbf{e}_k^{s^i}$ denote the embedding of the co-occurring entity s^i in the anchor triplet j and the contextual triplet k, respectively. α_{jk}^{ent} is the similarity of the co-occurring entity in triplets j and k.

For relation similarity bias, it determines the importance of the co-occurring entity in different triplets by evaluating the similarity between the relation embedding $\mathbf{e}_j^r$ of the anchor triplet and the relation embedding $\mathbf{e}_k^r$ of contextual triplets, which is defined as follows:

$$b_{jk}^{rel} = \frac{\exp\left(\left(\mathbf{e}_k^{r}\right)^{\mathrm{T}}\mathbf{e}_j^{r}\right)}{\sum_{\mathbf{e}_k^{r}\in N_{rel}}\exp\left(\left(\mathbf{e}_k^{r}\right)^{\mathrm{T}}\mathbf{e}_j^{r}\right)}, \tag{7}$$

where N_{rel} denotes the set of relation embeddings of the contextual triplets. $\mathbf{e}_j^r$ and $\mathbf{e}_k^r$ denote the relation embedding in the anchor triplet j and the contextual triplet k, respectively. b_{jk}^{rel} is the similarity of the relation embedding in triplets j and k.

Graph-level Transformer Layer: The basic architecture of the graph-level transformer is similar to that of the classical Transformer [40] encoder. The main structure of the Graph-level transformer layer consists of the multi-head self-attention (MHA) and the feed-forward network (FFN). Layer normalisation (LN) [44] is used after MHA and FFN. The details are as follows:

$$\mathbf{H}'^{(l)} = \mathrm{LN}\left(\mathrm{MHA}\left(\mathbf{H}^{(l-1)}\right)\right) + \mathbf{H}^{(l-1)}, \tag{8}$$

$$\mathbf{H}^{(l)} = \mathrm{LN}\left(\mathrm{FFN}\left(\mathbf{H}'^{(l)}\right)\right) + \mathbf{H}'^{(l)}, \tag{9}$$

where $\mathbf{H}^{(l)}$ and $\mathbf{H}'^{(l)}$ are the outputs of the MHA and FFN of the lth graph-level transformer layer, respectively. In addition, We design a novel MHA that takes into account the structure of the knowledge graph, which uses the above relational similarity as the inductive bias of the transformer. That is:

$$\mathbf{Q} = \mathbf{H}\cdot\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{H}\cdot\mathbf{W}_K, \quad \mathbf{V} = \mathbf{H}\cdot\mathbf{W}_V, \tag{10}$$

where the input $\mathbf{H}$ (equivalent to $\mathbf{H}^{(0)} = [\mathbf{e}_k^{s^i}, \mathbf{e}_j^{s^i}, \ldots \mathbf{e}_n^{s^i}]^T$ in (5) if it is the input of the first layer) is projected by $\mathbf{W}_Q \in \mathbb{R}^{d\times d_Q}$, $\mathbf{W}_K \in \mathbb{R}^{d\times d_K}$ and $\mathbf{W}_V \in \mathbb{R}^{d\times d_V}$ into the three corresponding embedding representations of $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$. In order to incorporate entity similarity and relation similarity into the self-attention weights, we designed the calculation as follows:

$$\mathrm{Attention}(\mathbf{H}) = \mathrm{softmax}(\mathbf{A})\cdot\mathbf{V}, \tag{11}$$

$$\mathbf{A} = \frac{\mathbf{Q}\cdot\mathbf{K}^{\mathrm{T}}}{\sqrt{d_K}} + \mathbf{B}, \tag{12}$$

$$A_{jk} = \frac{\left(\left(\mathbf{e}_k^{s^i}\right)^{\mathrm{T}}\mathbf{W}_Q\right)\left(\left(\mathbf{e}_k^{s^i}\right)^{\mathrm{T}}\mathbf{W}_Q\right)^{\mathrm{T}}}{\sqrt{d_K}} + b_{jk}^{rel}, \tag{13}$$

where A_{jk} is the (j, k)-element of the attention weight $\mathbf{A}$, which denotes the co-occurring entity similarity between anchor triplet j and its contextual triplet k. b_{jk}^{rel} is the (j, k)-element of the inductive bias $\mathbf{B}$, which denotes the relation similarity between anchor triplet j and its contextual triplet k.

TABLE I
CLASSICAL SEMANTIC MATCHING APPROACH

Methods	Semantic Matching Approach	Ent. & Rel.
Addition [9]	$\lVert\mathbf{e}_s + \mathbf{r} - \mathbf{e}_o\rVert_p$	$\mathbf{e}, \mathbf{r} \in \mathbb{R}^d$
Multiplication [32]	$\langle\mathbf{e}_s, \mathbf{r}, \mathbf{e}_o\rangle$	$\mathbf{e}, \mathbf{r} \in \mathbb{R}^d$
ConvE [11]	$f\left(\mathrm{vec}\left(f\left([\overline{\mathbf{e}_s}; \overline{\mathbf{r}}] \star \mathbf{\Omega}\right)\right)\mathbf{W}\right)\mathbf{e}_o$	$\mathbf{e}, \mathbf{r} \in \mathbb{R}^d$
TuckER [10]	$\mathbf{W} \times_1 \mathbf{e}_s \times_2 \mathbf{r} \times_3 \mathbf{e}_o$	$\mathbf{e}, \mathbf{r} \in \mathbb{R}^d$

C. Semantic Matching

This section describes how to use the output embedding of the triplet-level and graph-level transformer to perform the task of searching from $(s, r, ?)$ to o or from $(?, r, o)$ to s for prediction. This process is called the Knowledge Graph Completion (KGC) task.

We have selected several classical methods from traditional KGE for semantic matching to match queries (s, r) and answers o, including addition [11], multiplication [32], ConvE [11], and TuckER [10]. The semantic matching methods adopted by these models are detailed in Table I. Through extensive experiments, we have observed that the semantic matching approach of TuckER enables our TGformer to achieve superior performance on most datasets. For a more in-depth understanding, we refer the reader to the experimental section, and here we present the semantic matching approach of TGformer as follows:

$$\mathbf{e}_{\text{both}}^{o^{[\text{mask}]}} = \begin{cases} \mathrm{Add}(e^s, r) = \mathbf{e}_k^{s^i} + \mathbf{e}_k^r \\ \mathrm{Mul}(e^s, r) = \mathbf{e}_k^{s^i} \times \mathbf{e}_k^r \\ \mathrm{ConvE}(e^s, r) = f\left(\left[\overline{\mathbf{e}_k^{s^i}}; \overline{\mathbf{e}_k^{r}}\right] \star \mathbf{\Omega}\right) \\ \mathrm{TuckER}(e^s, r) = \boldsymbol{\mathcal{W}} \times_1 \mathbf{e}_k^{s^i} \times_2 \mathbf{e}_k^r \end{cases}, \tag{14}$$

where $\mathbf{e}_{both}^{o^{[mask]}}$ denotes the predicted entity embedding that contains both triplet-level and graph-level features. $\mathrm{Add}(\cdot)$ and $\mathrm{Mul}(\cdot)$ represent addition and multiplication semantic matching approaches.

D. Optimization and Inference

In this section, to assess the quality of the embedding of the predicted entity, we employ a semantic matching approach to score the embedding of the co-occurring entity in the anchor

TABLE II
STATISTICAL INFORMATION FOR STATIC AND TEMPORAL KNOWLEDGE GRAPHS

Dataset	#Entity	#Relation	#Train	#Valid	#Test	Time span(year)	Category
FB15k	14,951	1,345	483,142	50,000	59,071	-	Static
WN18	40,943	18	141,442	5,000	5,000	-	Static
FB15k-237	14,541	237	272,115	17,535	20,466	-	Static
WN18RR	40,943	11	86,835	3,034	3,134	-	Static
UMLS	135	46	5,216	652	661	-	Static
Kinship	1024	25	8,544	1,068	1,074	-	Static
ICEWS14	6,869	230	72,826	8,941	8,963	2014	Temporal
ICEWS05-15	10,094	251	368,962	46,275	46,092	2005-2015	Temporal
YAGO11k	10,623	10	16,406	2,050	2,051	-431-2844	Temporal
Wikidata12k	12,554	14	32,497	4,062	4,062	19-2020	Temporal

triplet and its contextual triplets. If only triplet-level features are considered, the embedding $\mathbf{e}_i^{o^{[mask]}}$ of the co-occurring entity of the anchor triplet and its contextual triplets, is calculated as follows:

$$\mathbf{q}_{triplet} = \text{softmax}\left(\mathbf{M}_{ent} \cdot \mathbf{e}_i^{o^{[mask]}}\right), \tag{15}$$

where $\mathbf{M}_{ent} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ denotes the entity embedding matrix, the parameters of the fully connected layer are $\mathbf{W}_1$ and $\mathbf{b}_1$, $\sigma(\cdot)$ is the activation function, and $\mathbf{q}_{triplet}$ denotes the scoring of the predicted entity with triplet-level features. The embedding $\mathbf{e}_{both}^{o^{[mask]}}$ containing both triplet-level and graph-level features is scored as follows:

$$\mathbf{q}_{graph} = \text{softmax}\left(\mathbf{M}_{ent} \cdot \mathbf{e}_{both}^{o^{[mask]}}\right), \tag{16}$$

where $\mathbf{W}_2$ and $\mathbf{b}_2$ are the parameters of the fully connected layer. $\mathbf{q}_{graph}$ denotes the scoring of the predicted entity with graph-level and triplet-level features. The process of training TGformer needs to consider both triplet and graph structural features. Therefore, the cross-entropy loss function with label smoothing for optimization is applied to speed up model convergence. That is written as

$$\begin{cases} L_{triplet} = \text{CrossEntropy}\left(\mathbf{y}_{label}, \mathbf{q}_{triplet}\right) \\ L_{graph} = \text{CrossEntropy}\left(\mathbf{y}_{label}, \mathbf{q}_{graph}\right), \end{cases} \tag{17}$$

where $\mathbf{y}_{label}$ is the id of the ground truth in the entity set. Combining triplet-level loss and graph-level loss function, the TGformer loss is defined as

$$L_{KGE} = \sum_{(s,r,o)\in\mathcal{T}} \left(L_{triplet} + L_{graph}\right). \tag{18}$$

It is worth noting that TGfromer does not negatively sample during the training process, which also boosts the efficiency of the model. Following CompGCN [18], for subject entity prediction: given $(?, r, o)$, predict s, we convert the subject entity prediction to object entity prediction: given $(s, r, ?)$, predict o.

IV. EXPERIMENTS

A. Experimental Setup

Datasets: Six static and four temporal KG datasets are selected, namely, FB15k-237 [45], WN18RR [11], WN18 [9], FB15k [9], UMLS [11], Kinship [46], ICEWS14 [47], ICEWS05-15 [47], YAGO11k [48], Wikidata12k [48]. The statistical information of these datasets is given in Table II. The specific dataset details are as follows:

1) The FB15k-237 [45] dataset is a subset of FB15k [9]. Toutanova and Chen [45] first pointed out that WN18 and FB15k have a test set leakage problem. Therefore, they extracted FB15k-237 from FB15k.
2) The WN18RR [11] dataset is a subset of WN18 [9]. All inverse relations in the WN18 dataset were removed by Dettmers et al. [11] to obtain the WN18RR dataset.
3) The FB15k [9] dataset is a subset of Freebase [7], which contains 14951 entities and 1345 relations. It describes knowledge about movies, sports, etc.
4) The WN18 [9] dataset is a subset of WordNet [6], which contains 40943 entities and 18 relations. It is a linguistic knowledge graph with rich linguistic knowledge.
5) The UMLS [11] dataset is a mini-KG containing medical semantic knowledge, which consists of 135 entities and 46 relations.
6) The Kinship [46] dataset covers the relationships between people who are members of a particular tribe and is made up of 10,686 triplets. It consists of 104 entities representing tribal members and 26 relation types representing kinship relations.
7) The ICEWS14 [47] dataset is a specialized subset of the Integrated Crisis Early Warning System dataset, focusing on political events that occurred in the year 2014. It provides a rich source of event-based data with specific time points.
8) The ICEWS05-15 [47] dataset captures a decade's worth of political events, spanning from 2005 to 2015, as part of the ICEWS dataset. It offers a comprehensive look at event data over an extended period.
9) The YAGO11k [48] dataset is an extraction from the larger YAGO3 knowledge graph, curated to highlight facts that incorporate temporal information. It presents

time annotations in various formats, such as specific dates, start and end times, and intervals.

10) The Wikidata12k [48] dataset is a refined subset of the expansive Wikidata knowledge graph, selected to emphasize temporally annotated facts. It features a diverse range of time expressions, from individual time points to broader temporal intervals.

Evaluation Metrics: Given knowledge graph $\mathcal{G} = (s, r, o)$, the model performance is measured by ranking the obtained scores in terms of Mean Reciprocal Rank (MRR), and Hitting Rate before k (Hits@$k, k \in \{1, 3, 10\}$). Hits@k is the ranking of the true entity of the triplet in the candidate entities. If the ranking is less than or equal to k, then it is recorded as 1; otherwise, it is recorded as 0. The formula is as follows:

$$\text{Hist@}k : \frac{1}{|\mathcal{T}_{test}|} \sum_{t_i \in \mathcal{T}_{test}} f\left(rank_i\right), \tag{19}$$

$$f(x) = \begin{cases} 1, & x <= k \\ 0, & x > k \end{cases}, \tag{20}$$

where $\mathcal{T}_{test}$ is the test dataset, $|\mathcal{T}_{test}|$ is the number of triplet in the test dataset. t_i represents the ith predicted triplet. $rank_i$ is the ranking of the correct entity among the candidate entities. MRR is the sum of the inverse of the subject entity prediction and object entity prediction rankings by replacing them with the inverse of their rankings. That is:

$$\text{MRR} : \frac{1}{2\,|\mathcal{T}_{test}|} \sum_{t_i \in \mathcal{T}_{test}} \left(\frac{1}{rank_i^s} + \frac{1}{rank_i^o}\right), \tag{21}$$

where $rank_i^s$ and $rank_i^o$ represent the ranking of the correct subject and object entity, separately.

Baselines: To evaluate the effectiveness of TGformer, the datasets presented above will be regarded as experimental datasets. The following three categories of excellent models will be adopted as the comparison approaches for TGFormer, which are Triplet-based models, Graph-based models, and Transformer-based models. The Triplet-based models include TransE [9], DistMult [32], ComplEx [13], RotatE [14], TuckER [10], ConvE [11], HypER [33], QuatE [57], InteractE [12], DualE [58], B-CP [50], CirlularE [51], and NFE [34]. Graph-based models are provided for R-GCN [17], KBGAT [22], SACN [24], CompGCN [18], DisenKGAT [8], REP-OTE [52], SE-GNN [23], SHGNet [16], DRGI [35]. Transformer-based models are a powerful category of models that have recently arisen. For fairness, some models that do not utilize the text of entities as auxiliary information are chosen as comparison methods, such as StAR [53], SAttLE [42], CoKE [37], Ruleformer [38], N-Former [39], Knowformer [54], and SDFormer [55].

Hyperparameters: For the TGformer model, different combinations of hyperparameters are tried to acquire suitable configurations for the test and validation datasets. The range of grid search hyperparameters was as follows: the dimension of entity and relation embedding range [100, 200, 256, 300, 400, 512], batch size range [256, 512, 1024, 2048], learning rate range [0.001, 0.0001, 0.0002, 0.0004]. The number of transformer layers and multi-attention heads is set in the ranges [1, 2, 3, 4, 5], and [1, 2, 4, 6, 8], respectively. The number of neighbor triplet in [0, 1, 2, 3, 4, 5]. Semantic matching method TuckER has a hidden dropout range of [0, 0.1, 0.2, 0.3, 0.4, 0.5]. In order to achieve better model performance, the parameters of TGformer are set differently for different datasets, as shown in Table III. All experiments were performed on a computer server equipped with Quadro RTX 8000.

B. Main Results and Analyses

1) Evaluation on the Static KGs: The outcomes of the link prediction task are presented in Table IV, illustrating the impressive performance of Tformer and its variants, particularly TGformer, in comparison to the baseline models on both the FB15k-237 and WN18RR datasets. TGformer consistently achieves optimal or near-optimal results across all evaluation metrics in these datasets. Notably, our method significantly outperforms both triplet-based and graph-based methods. On WN18RR, in comparison to transformer-based models like Ruleformer [38] and N-Former [39], TGformer exhibits substantial improvements with a 3.6% increase in MRR, a 6.3% increase in Hits@1, and a 3.5% increase in Hits@3, while Ruleformer and N-Former show improvements of 1.4%, 2.7%, and 1.6% in the respective metrics. Even on FB15k-237, TGformer consistently delivers competitive results across all evaluation metrics compared to the best-performing models in other approaches.

The reason for a stagnant but stable performance on FB15k-237 may be that the fusion of multilevel graph structure bias is still a challenge for a more complex structured dataset like FB15k-237. Specifically, as shown in Table II, there are only 14541 entities in over 300,000 triplets in FB15k237, but there are 40943 entities in less than 100,000 triplets in WN18RR. Therefore, the graph structure of FB15k-237 is dense. This would cause the number of co-occurring entities generated in FB15k-237 to be much more than WN18RR for the TGformer model, which is not conducive to discovering the truly valuable neighbor triplets. Furthermore, referring to [76], only considering the information in the training set, many triplets in the FB15k-237 dataset are unpredictable. Exploring more efficient methods for bias injection of graph structure in dense graphs is left for future work.

As shown in Table V, for the two larger datasets FB15k and WN18, TGformer achieves optimal results on all metrics. Specifically, on the FB15k dataset, TGformer improves the MRR by 2.5% and 4.4% over the triplet-based method DualE and the Graph-based method DRGI, respectively. For the two small-scale datasets UMLS and Kinship, it can be seen from Table VI that TGformer also achieves optimal results on all metrics. Specifically, on the UMLS and Kinship datasets, our TGformer outperformed the DRGI of the Graph-based method by 6.9% and 8.5% on MRR, respectively. The high performance of TGformer on both large-scale and small-scale datasets proves that TGformer has excellent generalisation capabilities.

2) Evaluation on the Temporal KGs: The results of the link prediction task, as detailed in Table VII, demonstrate the exceptional performance of TGformer. When compared to the

TABLE III
THE DETAILED MODEL PARAMETERS OF TGFORMER IN DIFFERENT DATASETS

Dataset	Fundamental parameters			Parameters of TGformer					Parameters of TuckER	
	Batch Size	Learning Rate	Label Smoothing	Embedding Size	# of Contextual Triplets	# of Attention Head	# of TGformer Layer	Input Dropout	Hidden1 Dropout	Hidden2 Dropout
FB15k-237	2048	2e-4	0.8	256	3	4	2	0.7	0.4	0.5
WN18RR	1024	2e-4	0.8	256	1	2	3	0.3	0.3	0.5
FB15k	2048	2e-4	0.8	768	2	6	2	0.7	0.2	0.1
WN18	1024	2e-4	0.8	256	4	2	3	0.3	0.3	0.0
UMLS	1024	2e-4	0.8	400	2	4	1	0.3	0.5	0.1
Kinship	1024	2e-4	0.8	200	2	2	1	0.3	0.1	0.4

TABLE IV
LINK PREDICTION RESULTS ON FB15K-237 AND WN18RR

Model	FB15k-237				WN18RR			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
Triplet-based								
TransE [9]	0.288	0.192	0.325	0.478	0.192	0.021	0.367	0.435
DistMult [32]	0.180	0.101	0.191	0.347	0.322	0.243	0.371	0.463
ComplEx [13]	0.233	0.145	0.266	0.408	0.380	0.337	0.423	0.472
ConvE [11]	0.324	0.236	0.355	0.502	0.432	0.403	0.442	0.489
TuckER [10]	0.354	0.262	0.388	0.535	0.463	0.435	0.478	0.516
RotatE [14]	0.318	0.216	0.358	0.525	0.474	0.425	0.490	0.571
B-CP [50]	0.295	0.210	0.324	0.483	0.480	0.450	0.490	0.530
CirlularE [51]	0.347	0.252	0.385	0.538	0.470	0.422	0.491	0.562
NFE-1 [34]	0.362	0.270	0397	0.548	0.484	0.441	0.496	0.569
Graph-based								
R-GCN [17] ♢	0.249	0.151	0.264	0.417	0.226	0.157	0.269	0.376
KBGAT [22] ♡	0.157	-	-	0.331	0.412	-	-	0.554
SACN [24] ♢	0.350	0.260	0.390	0.540	0.470	0.430	0.480	0.540
CompGCN [18]	0.353	0.262	0.388	0.534	0.469	0.435	0.485	0.533
DisenKGAT [8]	0.368	0.275	0.407	0.553	0.486	0.441	0.502	0.578
REP-OTE [52]	0.354	0.262	0.388	0.540	0.488	0.439	0.505	0.588
SE-GNN [23]	0.365	0.272	0.400	0.553	0.486	0.442	0.505	0.569
SHGNet [16] ♢	0.355	0.268	0.395	0.544	0.476	0.448	0.496	0.549
DRGI [35] ♢	0.362	0.270	0.399	0.549	0.479	0.445	0.496	0.543
Transformer-based								
CoKE [37] ♢	0.364	0.272	0.400	0.549	0.484	0.450	0.496	0.553
StAR [53] ♢	0.296	0.205	0.322	0.482	0.401	0.243	0.491	**0.709**
Ruleformer [38] ♢	0.338	0.241	0.390	0.535	0.476	0.428	0.492	0.571
SAttLE-TwoMult [42] ♢	0.360	0.268	0.396	0.545	0.491	0.454	0.508	0.558
N-Former [39] ♢	**0.372**	0.277	**0.421**	0.556	0.486	0.443	0.501	0.578
Knowformer [54]	0.366	0.271	0.406	0.555	0.488	0.448	0.504	0.567
SDFormer [55]	0.356	0.264	0.390	0.541	0.458	0.425	0.471	0.528
Tformer(ours)	0.369	0.274	0.407	0.553	0.481	0.445	0.495	0.555
TGformer(ours)	**0.372**	**0.279**	0.410	**0.557**	**0.493**	**0.455**	**0.509**	0.566

The optimal performance is indicated in bold and the second-best is underlined. The results of ♢ are from the original articles. The results for ♡ are derived from [49], and other results are obtained by our reproduction.

baseline models on the ICEWS14 and ICEWS05-15 datasets, TGformer consistently secures the highest scores across all metrics. Specifically, on the ICEWS14 dataset, TGformer shows significant enhancements over leading models such as QDN [66] and TPAR [68], with a 10.7% boost in MRR, a 16.0% boost in Hits@1, and an 8.1% increase in Hits@3. Comparatively, QDN and TPAR report increases of 9.4%, 15.3%, and 6.7% in the same metrics, respectively. Similarly, on the ICEWS05-15 dataset, TGformer maintains a strong competitive edge, delivering results that rival the top-performing models from other methodologies across all evaluated metrics.

The link prediction results on the temporal knowledge graphs YAGO11k and Wikidata12k (for details, see Table VIII) have demonstrated that TGformer is well-adapted to scenarios involving temporal knowledge graphs. Compared to the baseline models on the YAGO11k and Wikidata12k datasets, TGformer has shown significant improvements in most of the evaluation metrics, especially in Hits@1. Specifically, on the YAGO11k dataset, TGformer has achieved notable enhancements over leading models such as TGeomE++ [65] and PTBox [74], with increases of 4.8% and 28.4% in MRR, and 14.4% and 3.6% in Hits@3, respectively. Similarly, on the Wikidata12k dataset, TGformer has maintained a strong competitive edge, with specific improvements of 41.1% and 62.1% in MRR, and 35.6% and 48.3% in Hits@3, respectively. These experimental results substantiate that TGformer is adept at adapting to the

TABLE V
LINK PREDICTION RESULTS ON FB15K AND WN18

Model	FB15k				WN18			
	MRR	Hits			MRR	Hits		
		@1	@3	@10		@1	@3	@10
TransE [9]	0.463	0.297	0.578	0.749	0.495	0.113	0.888	0.943
TorusE [56]	0.733	0.674	0.771	0.832	0.947	0.941	0.954	0.960
ConvE [11]	0.657	0.558	0.723	0.831	0.943	0.935	0.946	0.956
R-GCN [17]	0.651	0.541	0.736	0.825	0.814	0.686	0.928	0.955
QuatE [57]	0.770	0.700	0.821	0.878	0.949	0.941	0.954	0.960
RotatE [14]	0.797	0.746	0.830	0.884	0.949	0.944	0.952	0.955
HypER [33]	0.790	0.734	0.829	0.885	0.951	0.947	0.955	0.958
TuckER [10]	0.795	0.741	0.833	0.892	0.953	0.949	0.955	0.958
DualE [58]	0.813	0.766	0.850	0.896	0.952	0.946	0.956	**0.962**
B-CP [50]	0.733	0.660	0.793	0.870	0.945	0.941	0.948	0.956
DRGI [35]	0.798	0.746	0.854	0.896	0.948	0.943	0.950	0.957
TGformer(ours)	**0.833**	**0.784**	**0.868**	**0.910**	**0.954**	**0.949**	**0.958**	**0.962**

TABLE VI
LINK PREDICTION RESULTS ON UMLS AND KINSHIP

Model	UMLS				Kinship			
	MRR	Hits			MRR	Hits		
		@1	@3	@10		@1	@3	@10
TransE [9]	0.615	0.391	0.807	0.945	0.211	0.093	0.252	0.470
DistMult [32]	0.164	0.061	0.135	0.403	0.480	0.377	0.491	0.708
ConvE [11]	0.836	0.764	0.917	0.946	0.772	0.665	0.858	0.950
R-GCN [17]	0.481	0.318	0.559	0.835	0.401	0.225	0.473	0.822
RotatE [14]	0.822	0.703	0.932	0.969	0.738	0.617	0.827	0.954
SACN [24]	0.856	0.764	0.946	0.985	0.799	0.699	0.878	0.964
DRGI [35]	0.898	0.838	0.948	0.988	0.847	0.765	0.915	0.981
TGformer($ours$)	**0.960**	**0.934**	**0.983**	**0.994**	**0.919**	**0.872**	**0.966**	**0.986**

TABLE VII
LINK PREDICTION RESULTS ON ICEWS14 AND ICEWS05-15

Model	ICEWS14				ICEWS05-15			
	MRR	Hits			MRR	Hits		
		@1	@3	@10		@1	@3	@10
TeMP [59]	0.607	0.484	0.684	0.840	0.680	0.553	0.769	0.913
TNTComplEx [60]	0.620	0.520	0.660	0.760	0.670	0.590	0.710	0.810
T-GAP [61]	0.610	0.509	0.677	0.790	0.670	0.568	0.743	0.845
BoxTE [62]	0.613	0.528	0.664	0.763	0.667	0.582	0.719	0.820
RotateQVS [63]	0.591	0.507	0.642	0.754	0.633	0.529	0.709	0.813
TeAST [64]	0.637	0.560	0.682	0.782	0.683	0.604	0.732	0.829
TGeomE++ [65]	0.629	0.546	0.680	0.780	0.686	0.605	0.736	0.833
QDN [66]	0.643	0.567	0.688	0.784	0.692	0.611	0.743	0.838
HyGNet [67]	0.645	0.568	0.691	0.783	0.693	0.612	0.747	0.837
TPAR [68]	0.651	0.570	0.697	0.804	0.693	0.622	0.753	**0.859**
TGformer($ours$)	**0.712**	**0.658**	**0.744**	**0.813**	**0.761**	**0.709**	**0.790**	**0.859**

temporal knowledge graph context and has achieved outstanding performance.

3) Complex Relations Modeling: There are numerous complex relations within KGs, and accurately predicting these complex relations is a crucial aspect of model evaluation. To further assess the model's performance, we evaluated TGformer against several representative and advanced baselines in predicting missing entities on FB15K-237. Following the classification of relations in KGs into four types, namely 1-to-1, 1-to-N, N-to-1, and N-to-N, as per TransE [9], we conducted experiments to assess the performance, as presented in Table IX, with the best-performing values marked in **bold**.

In summary, TGformer demonstrates exceptional performance in predicting missing entities for complex relations such as 1-to-N and N-to-N relations. Specifically, within the 1-to-N relations category, the TGformer model achieves over 3% and better prediction results than the best baseline in terms of MRR. For N-to-N relations, TGformer outperforms DisenGAT [8], and SE-GNN [23] by 2.2% and 2.3% in MRR, respectively. An interesting observation is the significant performance gap

TABLE VIII
LINK PREDICTION RESULTS ON YAGO11K AND WIKIDATA12K

Model	YAGO11k				Wikidata12k			
	MRR	Hits			MRR	Hits		
		@1	@3	@10		@1	@3	@10
TeRo [69]	0.187	0.121	0.197	0.319	0.299	0.198	0.329	0.507
ATiSE [70]	0.185	0.126	0.189	0.301	0.252	0.148	0.288	0.462
TeLM [71]	0.191	0.129	0.194	0.321	0.332	0.231	0.360	0.542
RotateQVS [63]	0.189	0.124	0.199	0.323	-	-	-	-
BiQCap [72]	0.195	0.126	0.206	0.332	0.312	0.216	0.343	0.517
DuCape [73]	0.197	<u>0.136</u>	<u>0.228</u>	**0.348**	0.297	0.205	0.335	0.509
TGeomE++ [65]	0.195	0.130	0.196	0.326	<u>0.333</u>	<u>0.232</u>	<u>0.362</u>	<u>0.546</u>
QDN [66]	<u>0.198</u>	0.131	0.201	0.328	-	-	-	-
PTBox [74]	0.162	-	0.222	0.347	0.290	-	0.331	0.527
TGformer($ours$)	**0.208**	**0.156**	**0.230**	0.308	**0.470**	**0.409**	**0.491**	**0.585**

TABLE IX
PREDICTION RESULTS OF COMPLEX RELATIONS ON FB15K-237 DATASET

Model	1-to-1		1-to-N		N-to-1		N-to-N	
	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10
TransE [9]	0.486	0.576	0.417	0.522	0.268	0.408	0.308	0.537
DistMult [32]	0.213	0.464	0.402	0.491	0.244	0.378	0.287	0.491
ComplEx [13]	0.366	0.521	0.416	0.517	0.263	0.394	0.303	0.522
ConvE [11] †	0.370	0.508	0.427	0.567	0.257	0.397	0.318	0.531
RotatE [14]	0.490	0.586	0.425	0.526	0.272	0.410	0.313	0.542
InteractE [12] †	0.377	0.547	0.442	0.537	0.27	0.394	0.334	0.547
CompGCN [18]	0.455	0.597	0.446	0.538	0.274	0.404	0.335	0.545
DisenGAT [8]	**0.499**	**0.630**	0.459	0.569	0.281	0.420	0.343	0.559
SE-GNN [23]	0.409	0.586	0.460	0.569	0.282	0.411	0.344	0.559
HousE [75]	0.471	0.549	0.456	0.558	0.285	0.408	0.346	0.562
TGformer($ours$)	0.374	0.555	**0.474**	**0.570**	**0.289**	**0.427**	**0.352**	**0.568**

The results of † are from InteractE [12].

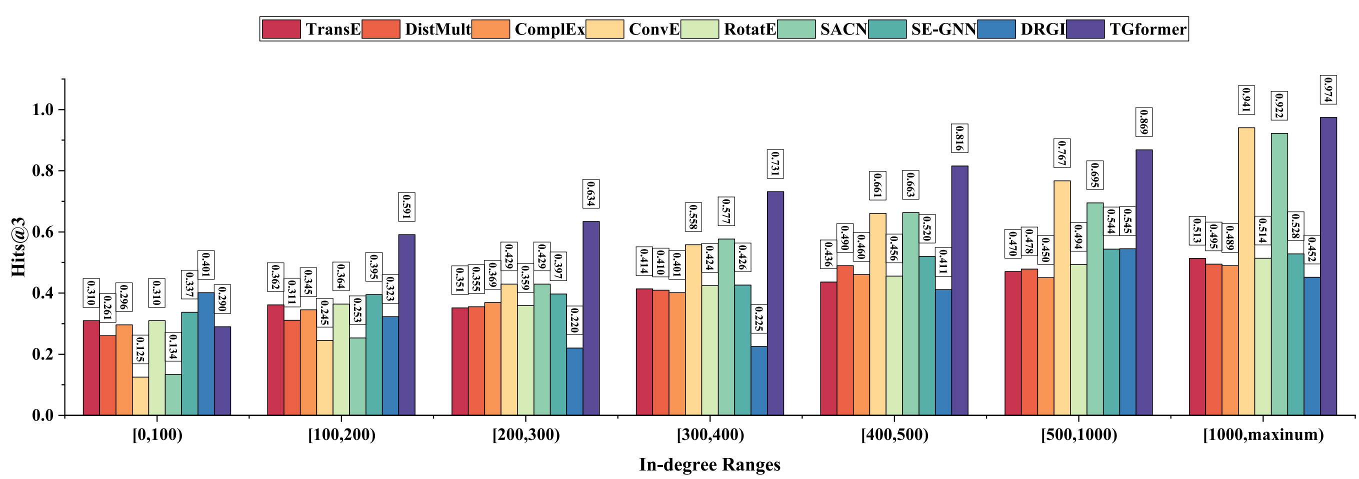

Fig. 3. Comparison of Hits@3 predictions for triplets with different indegree nodes in FB15k-237.

between TGformer's performance on 1-to-1 and the other complex relations(1-to-N, N-to-1, N-to-N). This phenomenon can be attributed to TGformer's selection of contextual triplets based on the subject entity. It's worth noting that this performance gap might be related to the inherent limitations of the link prediction task itself, which is based on the prediction of object entities given (subject entity, relation, ?). Nonetheless, the overall performance of TGformer on FB15k-237 remains competitive.

To conduct a more detailed analysis of TGformer's performance in modeling complex relations, we conducted link prediction experiments for each type of complex relation in WN18RR. The specific experimental results are presented in

TABLE X
COMPARISON OF SINGLE-TYPE RELATION MRR PERFORMANCE ON THE WN18RR DATASET

Relation Types	Relation Name	Model								
		TGformer *(ours)*	TransE [9]	DistMult [32]	ComplEx [13]	RotatE [14]	ConvE [11]	CompGCN [18]	SE-GNN [23]	NFE-1 [34]
1-to-1	similar_to	**1.000**	0.294	1.000	1.000	1.000	1.000	0.917	0.722	1.000
	verb_group	**0.974**	0.364	0.891	0.955	0.962	0.962	0.917	0.857	0.974
1-to-N	has_part	0.193	0.121	0.156	**0.228**	0.190	0.122	0.163	0.196	0.206
	member_meronym	0.282	0.185	0.163	0.238	0.251	0.167	0.225	**0.291**	0.255
	member_of_domain_region	0.340	0.126	0.181	0.126	0.241	0.366	0.392	0.360	**0.398**
	member_of_domain_usage	**0.346**	0.111	0.235	0.244	0.278	0.258	0.335	0.337	0.308
N-to-1	hypernym	**0.180**	0.060	0.093	0.140	0.156	0.087	0.139	0.178	0.160
	instance_hypernym	**0.401**	0.286	0.349	0.357	0.334	0.347	0.390	0.396	0.356
	synset_domain_topic_of	**0.386**	0.153	0.336	0.314	0.364	0.285	0.349	0.373	0.372
N-to-N	also_see	**0.654**	0.231	0.591	0.564	0.631	0.630	0.641	0.600	0.641
	derivationally_related_form	**0.957**	0.440	0.954	0.950	0.954	0.952	0.954	0.947	0.954

Table X. For each relation in WN18RR, TGformer consistently outperformed previous methods across various evaluation metrics. Notably, TGformer excels in predicting 1-to-1, N-to-1 and N-to-N relations. In particular, for N-to-1 relations, TGformer competes effectively with the current leading open-source method, SE-GNN [23]. Specifically, for relations such as $also\_see$ and $synset\_domain\_topic\_of$, TGformer improved the MRR scores by 2.0% and 3.5%, respectively. These results effectively demonstrate that TGformer enhances the model's performance across most types of complex relations.

4) Analysis of Nodes With Different Indegree Ranges: In knowledge graphs, a node's degree functions as a quantifiable measure for gauging the comprehensiveness and diversity of the contextual landscape within which the node is positioned. Consequently, for a more in-depth analysis of TGformer's performance, it is imperative to investigate its correlation with node degrees. As depicted in Fig. 3, we compare TGformer's performance with that of several advanced Knowledge Graph Embedding (KGE) models in the FB15k-237 dataset across various degree ranges. Notably, TGformer demonstrates remarkable superiority in all in-degree ranges (except [0, 100)). Furthermore, it is discernible that TGformer exhibits improved performance as the in-degree of the nodes increases. These observations highlight TGformer's proficiency in effectively capturing contextual relationships within knowledge graphs, demonstrating optimal performance with an enriched contextual environment.

C. Ablation Study

1) Influence of Different Structural Modules: To assess the contribution of each component, we conducted ablation experiments on TGformer, as presented in Table XI. It is evident that TGformer, with all components, exhibited superior performance across all metrics in both the FB15k-237 and UMLS datasets. The substantial drop in performance when removing graph-level features underscores the importance of incorporating knowledge graph structures. Notably, in FB15k-237 and UMLS, graph-level features outweigh the semantic matching module's significance, with the latter playing a more prominent role in UMLS.

2) Effect of Different Approaches of Semantic Matching: In TGformer, Different semantic matching methods are also a key factor in the performance of the model. To explore how to utilize semantic matching optimally, we applied four different approaches to compute the semantic matching score by referring to a variety of previous excellent triplet-based models [10], [11]. We evaluated the impact of these four semantic matching methods on the performance of TGformer on FB15k237, WN18RR, and UMLS. The results are shown in Table XII. The experimental results show that the TGformer with TuckER to calculate the semantic matching score works best.

TABLE XI
ABLATION STUDY ON FB15K-237 AND UMLS

Models	FB15k-237			UMLS		
	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10
TGformer	**0.372**	**0.279**	**0.556**	**0.960**	**0.934**	**0.994**
w/o RB	0.369	0.274	0.555	0.959	0.932	0.993
w/o SM	0.368	0.273	0.554	0.955	0.931	0.993
w/o GL	0.369	0.274	0.553	0.953	0.930	0.991

"RB," "SM," and "GL" refer to relation bias, semantic matching, and graph-level features respectively.

TABLE XII
THE IMPACT OF HOW DIFFERENT SPECIES OF SEMANTIC MATCHING METHODS ON THE FB15K-237, WN18RR, AND UMLS

Model	FB15k-237		WN18RR		UMLS	
	MRR	Hits@1	MRR	Hits@1	MRR	Hits@1
Add	0.368	0.271	0.486	0.450	0.956	0.933
Mul	0.369	0.274	0.489	0.451	0.958	0.930
ConvE	0.370	0.275	0.490	0.452	0.957	0.929
TuckER	**0.372**	**0.279**	**0.493**	**0.455**	**0.960**	**0.933**

"Add" and "Mul" represent addition and multiplication calculation methods.

3) Effect of the Number of Contextual Triplets: Since the co-occurring entity may exist in more than one contextual triplet, TGformer randomly selects the same number of different Contextual triplets to model the embedding of the co-occurring entity in each training round. When constructing a graph-level representation of the co-occurring entity, randomly selecting contextual triplets is a data enhancement strategy that allows the co-occurring entity to incorporate multi-context information to

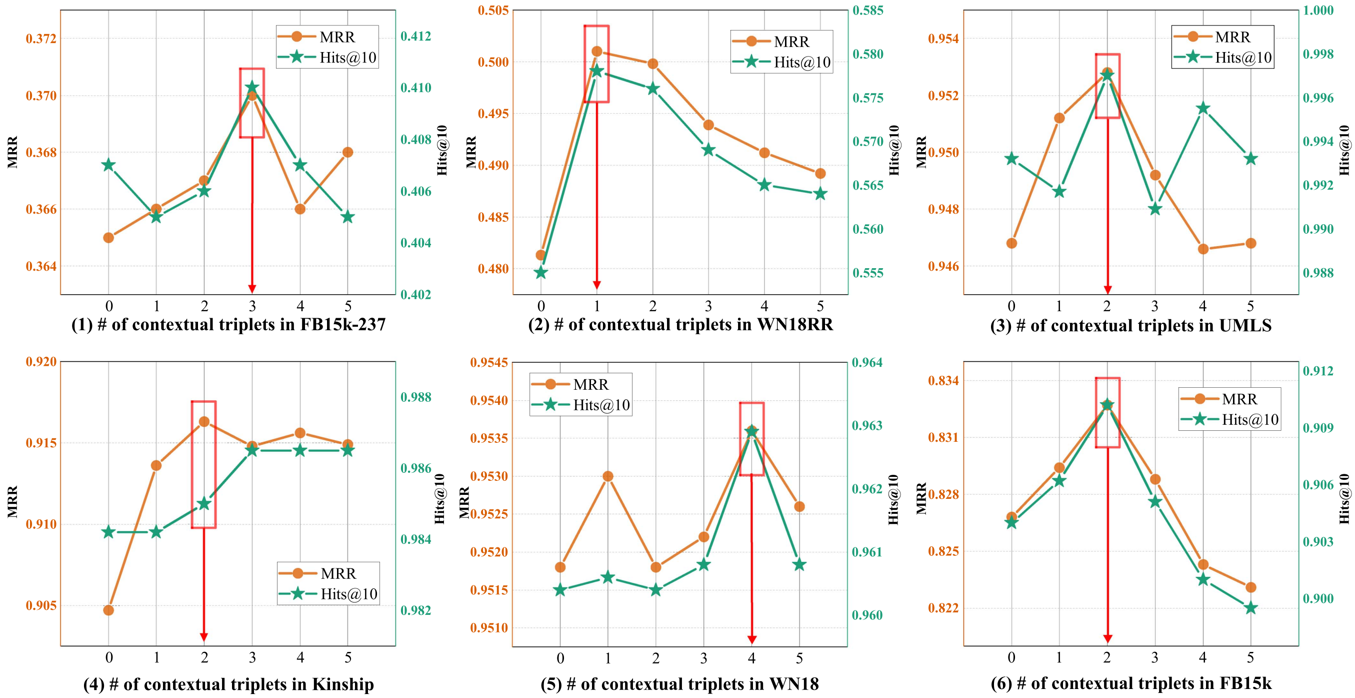

Fig. 4. Effect of the number of contextual triplets.

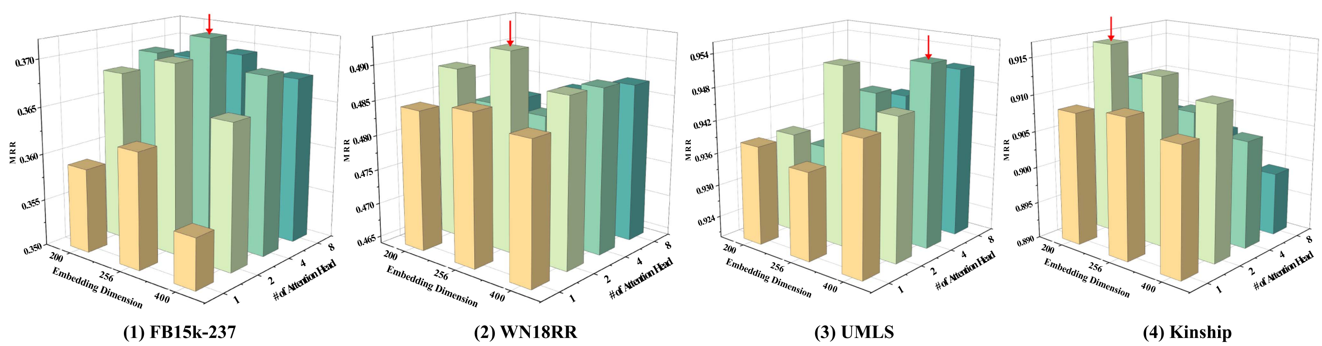

Fig. 5. Effect of embedding dimension & the number of attention head.

increase the generalizability of TGformer. Fig. 4 illustrates the effect of different choices of the number of contextual triplets on the performance of TGformer for different datasets. Intuitively, to obtain better model performance, the number of contextual triplets chosen for the relatively sparse KGs like WN18RR is less than that of the denser FB15k-237. Specifically, for WN18RR and FB15k-237, the number of contextual triplets chosen for TGformer modeling the embedding of the co-occurring entity is chosen to be 2 and 3, respectively.

4) Effect of Embedding Dimension & the Number of Attention Head: TGformer is a transformer-based model. In transformer, the embedding dimension and the number of attention heads are closely related, and the appropriate embedding dimension and the number of attention heads are crucial for the model performance. In order to explore their correlation with model performance, we conducted a number of experiments on different datasets to find the optimal embedding dimension and number of attention heads for each KG dataset. As shown in Fig. 5, the optimal embedding dimension and number of attention heads are $(256, 4)$ and (256,2) for FB15k-237 and WN18RR, respectively. For UMLS and Kinship, the optimal embedding dimension and number of attention heads are (400,4) and (200,2), respectively.

V. Conclution

This paper presents a graph transformer framework for knowledge graph embedding, which simultaneously fuses triple-level and graph-level structural information to learn entity and relation embeddings. Concretely, a context-level subgraph is built for each to-be-predicted triplet, with triplets as nodes linking contextual triplets that have co-occurring entities. On this basis, we design a knowledge graph transformer network (KGTN) including a triplet-level transformer and a graph-level transformer. KGTN understands the entities (relations) under different semantic contexts while embedding the structural knowledge of the static or temporal knowledge graphs into the transformer to learn the optimized feature representation. Extensive experiments demonstrate that TGformer outperforms other state-of-the-art baselines on the link prediction task due to the fusion of triplet-level information and graph-level information.

In the future, we plan to incorporate textual description information of entities and relations as auxiliary information in the model to simulate the semantic embedding of entities (relations) in the real world. In addition, we further explore the impact of the quality and efficiency of negative sampling on KGE.

References

[1] D. Li, F. Shi, X. Wang, C. Zheng, Y. Cai, and B. Li, "Multi-perspective knowledge graph completion with global and interaction features," *Inf. Sci.*, vol. 666, 2024, Art. no. 120438.

[2] C. Liu et al., "Social-enhanced explainable recommendation with knowledge graph," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 2, pp. 840–853, Feb. 2024.

[3] G. Tang et al., "EditKG: Editing knowledge graph for recommendation," in *Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2024, pp. 112–122.

[4] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying large language models and knowledge graphs: A roadmap," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3580–3599, Jul. 2024.

[5] Y. Wen, Z. Wang, and J. Sun, "MindMap: Knowledge graph prompting sparks graph of thoughts in large language models," in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., 2024, pp. 10370–10388.

[6] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, pp. 39–41, 1995.

[7] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 1247–1250.

[8] J. Wu et al., "DisenKGAT: Knowledge graph embedding with disentangled graph attention network," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 2140–2149.

[9] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.

[10] I. Balazevic, C. Allen, and T. Hospedales, "TuckER: Tensor factorization for knowledge graph completion," in *Proc. 2019 Conf. Empir. Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 5185–5194.

[11] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2018, Art. no. 221.

[12] S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal, and P. Talukdar, "InteractE: Improving convolution-based knowledge graph embeddings by increasing feature interactions," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 3009–3016.

[13] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 2071–2080.

[14] Z. Sun, Z. Deng, J. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," in *Proc. 7th Int. Conf. Learn. Representations*, 2019, pp. 1–18. [Online]. Available: https://openreview.net/forum?id=HkgEQnRqYQ

[15] L. Jin, Z. Yao, M. Chen, H. Chen, and W. Zhang, "A comprehensive study on knowledge graph embedding over relational patterns based on rule learning," in *Proc. 22nd Int. Semantic Web Conf.*, Athens, Greece, 2023, pp. 290–308.

[16] Z. Li, Q. Zhang, F. Zhu, D. Li, C. Zheng, and Y. Zhang, "Knowledge graph representation learning with simplifying hierarchical feature propagation," *Inf. Process. Manage.*, vol. 60, no. 4, 2023, Art. no. 103348.

[17] M. S. Schlichtkrull et al., "Modeling relational data with graph convolutional networks," in *Proc. 15th Extended Semantic Web Conf.*, 2018, pp. 593–607.

[18] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–16. [Online]. Available: https://openreview.net/forum?id=BylA_C4tPr

[19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–14. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[20] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[21] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017, pp. 1025–1035.

[22] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning attention-based embeddings for relation prediction in knowledge graphs," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 4710–4723.

[23] R. Li et al., "How does knowledge graph embedding extrapolate to unseen data: A semantic evidence view," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 5781–5791.

[24] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, "End-to-end structure-aware convolutional networks for knowledge base completion," in *Proc. 33th AAAI Conf. Artif. Intell.*, 2019, pp. 3060–3067.

[25] X. Jiang, Q. Wang, and B. Wang, "Adaptive convolution for multi-relational learning," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 978–987.

[26] Q. Li, Z. Han, and X. Ming Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, Art. no. 433.

[27] Z. Hu, V. G. Basulto, Z. Xiang, R. Li, and J. Z. Pan, "HyperFormer: Enhancing entity and relation interaction for hyper-relational knowledge graph completion," in *Proc. 32st ACM Int. Conf. Inf. Knowl. Manage.*, 2023, pp. 803–812.

[28] C. Ying et al., "Do transformers really perform badly for graph representation?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 28877–28888.

[29] B. Liu, D. Li, J. Wang, Z. Wang, B. Li, and C. Zeng, "Integrating user short-term intentions and long-term preferences in heterogeneous hypergraph networks for sequential recommendation," *Inf. Process. Manage.*, vol. 61, no. 3, 2024, Art. no. 103680.

[30] W. Zhao et al., "DADL: Double asymmetric distribution learning for head pose estimation in wisdom museum," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. 1, 2024, Art. no. 101869.

[31] K. Lin, D. Li, Y. Li, S. Chen, and X. Wu, "FHCPL: An intelligent fixed-horizon constrained policy learning system for risk-sensitive industrial scenario," *IEEE Trans. Ind. Informat.*, vol. 20, no. 4, pp. 5794–5804, Apr. 2024.

[32] B. Yang, W. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–12. [Online]. Available: https://arxiv.org/pdf/1412.6575

[33] I. Balažević, C. Allen, and T. M. Hospedales, "Hypernetwork knowledge graph embeddings," in *Proc. Int. Conf. Artif. Neural Netw.*, 2019, pp. 553–565.

[34] X. Changyi, H. Xiangnan, and C. Yixin, "Knowledge graph embedding by normalizing flows," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 4756–4764.

[35] S. Liang, J. Shao, D. Zhang, J. Zhang, and B. Cui, "DRGI: Deep relational graph infomax for knowledge graph completion," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2486–2499, Mar. 2023.

[36] Z. Zhang, J. Wang, J. Ye, and F. Wu, "Rethinking graph convolutional networks in knowledge graph completion," in *Proc. ACM Web Conf.*, 2022, pp. 798–807.

[37] Q. Wang et al., "CoKE: Contextualized knowledge graph embedding," 2019, *arXiv: 1911.02168*.

[38] Z. Xu, P. Ye, H. Chen, M. Zhao, H. Chen, and W. Zhang, "Ruleformer: Context-aware rule mining over knowledge graph," in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 2551–2560.

[39] Y. Liu, Z. Sun, G. Li, and W. Hu, "I know what you do not know: Knowledge graph embedding via co-distillation learning," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 1329–1338.

[40] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[42] P. Baghershahi, R. Hosseini, and H. Moradi, "Self-attention presents low-dimensional knowledge graph embeddings for link prediction," *Knowl.-Based Syst.*, vol. 260, 2023, Art. no. 110124.

[43] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–21. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[44] R. Xiong et al., "On layer normalization in the transformer architecture," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, Art. no. 975.

[45] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *Proc. 3rd Workshop Continuous Vector Space Models Their Compositionality*, 2015, pp. 57–66.

[46] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda, "Learning systems of concepts with an infinite relational model," in *Proc. 21st Nat. Conf. Artif. Intell.*, 2006, pp. 381–388.

[47] A. García-Durán, S. Dumančić, and M. Niepert, "Learning sequence encoders for temporal knowledge graph completion," in *Proc. 2018 Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 4816–4821.

[48] S. S. Dasgupta, S. N. Ray, and P. Talukdar, "HyTE: Hyperplane-based temporally aware knowledge graph embedding," in *Proc. 2018 Conf. Empir. Methods Natural Lang. Process.*, Brussels, Belgium, 2018, pp. 2001–2011.

[49] Z. Sun, S. Vashishth, S. Sanyal, P. Talukdar, and Y. Yang, "A re-evaluation of knowledge graph completion methods," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5516–5522.

[50] K. Hayashi, K. Kishimoto, and M. Shimbo, "Binarized embeddings for fast, space-efficient knowledge graph completion," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 141–153, Jan. 2023.

[51] Y. Fang, W. Lu, X. Liu, W. Pedrycz, Q. Lang, and J. Yang, "CircularE: A complex space circular correlation relational model for link prediction in knowledge graph embedding," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 3162–3175, 2023.

[52] H. Wang et al., "Simple and effective relation-based embedding propagation for knowledge representation learning," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, 2022, pp. 2755–2761.

[53] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, and Y. Chang, "Structure-augmented text representation learning for efficient knowledge graph completion," in *Proc. Web Conf.*, 2021, pp. 1737–1748.

[54] G. Li, Z. Sun, W. Hu, G. Cheng, and Y. Qu, "Position-aware relational transformer for knowledge graph embedding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 8, pp. 11580–11594, Aug. 2024.

[55] D. Li et al., "SDFormer: A shallow-to-deep feature interaction for knowledge graph embedding," *Knowl.-Based Syst.*, vol. 284, 2024, Art. no. 111253.

[56] T. Ebisu and R. Ichise, "TorusE: Knowledge graph embedding on a lie group," in *Proc. AAAI Conf. Artif. Intell.*, 2018, Art. no. 222.

[57] S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion knowledge graph embeddings," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, Art. no. 246.

[58] Z. Cao, Q. Xu, Z. Yang, X. Cao, and Q. Huang, "Dual quaternion knowledge graph embeddings," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 6894–6902.

[59] J. Wu, M. Cao, J. C. K. Cheung, and W. L. Hamilton, "TeMP: Temporal message passing for temporal knowledge graph completion," in *Proc. 2020 Conf. Empir. Methods Natural Lang. Process.*, 2020, pp. 5730–5746.

[60] T. Lacroix, G. Obozinski, and N. Usunier, "Tensor decompositions for temporal knowledge base completion," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=rke2P1BFwS

[61] J. Jung, J. Jung, and U. Kang, "Learning to walk across time for interpretable temporal knowledge graph completion," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2021, pp. 786–795.

[62] J. Messner, R. Abboud, and I. I. Ceylan, "Temporal knowledge graph completion using box embeddings," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 7779–7787.

[63] K. Chen, Y. Wang, Y. Li, and A. Li, "RotateQVS: Representing temporal information as rotations in quaternion vector space for temporal knowledge graph completion," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 5843–5857.

[64] J. Li, X. Su, and G. Gao, "TeAST: Temporal knowledge graph embedding via archimedean spiral timeline," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 15460–15474.

[65] C. Xu, M. Nayyeri, Y.-Y. Chen, and J. Lehmann, "Geometric algebra based embeddings for static and temporal knowledge graph completion," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4838–4851, May 2023.

[66] J. Wang, B. Wang, J. Gao, X. Li, Y. Hu, and B. Yin, "QDN: A quadruplet distributor network for temporal knowledge graph completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 10, pp. 14018–14030, Oct. 2024.

[67] Y. Li, X. Zhang, Y. Cui, and S. Ma, "Hyperbolic graph neural network for temporal knowledge graph completion," in *Proc. 2024 Joint Int. Conf. Comput. Linguistics Lang. Resour. Eval.*, 2024, pp. 8474–8486.

[68] K. Chen, Y. Wang, Y. Li, A. Li, H. Yu, and X. Song, "A unified temporal knowledge graph reasoning model towards interpolation and extrapolation," in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics*, 2024, pp. 117–132.

[69] C. Xu, M. Nayyeri, F. Alkhoury, H. Shariat Yazdi, and J. Lehmann, "TeRo: A time-aware knowledge graph embedding via temporal rotation," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 1583–1593.

[70] C. Xu, M. Nayyeri, F. Alkhoury, H. Yazdi, and J. Lehmann, "Temporal knowledge graph completion based on time series Gaussian embedding," in *Proc. Int. Semantic Web Conf.*, 2020, pp. 654–671.

[71] C. Xu, Y.-Y. Chen, M. Nayyeri, and J. Lehmann, "Temporal knowledge graph completion using a linear temporal regularizer and multivector embeddings," in *Proc. 2021 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2021, pp. 2569–2578.

[72] S. Zhang, X. Liang, Z. Li, J. Feng, X. Zheng, and B. Wu, "BiQCap: A biquaternion and capsule network-based embedding model for temporal knowledge graph completion," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2023, pp. 673–688.

[73] S. Zhang, X. Liang, H. Tang, X. Zheng, A. X. Zhang, and Y. Ma, "DuCape: Dual quaternion and capsule network–based temporal knowledge graph embedding," *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 7, 2023, Art. no. 104.

[74] Z. Fang, J. Qin, X. Zhu, C. Yang, and X.-C. Yin, "Arbitrary time information modeling via polynomial approximation for temporal knowledge graph embedding," in *Proc. 2024 Joint Int. Conf. Comput. Linguistics Lang. Resour. Eval.*, 2024, pp. 1455–1465.

[75] R. Li et al., "HousE: Knowledge graph embedding with householder parameterization," in *Proc. 39th Int. Conf. Mach. Learn.*, 2022, pp. 13209–13224.

[76] Y. Cao, X. Ji, X. Lv, J. Li, Y. Wen, and H. Zhang, "Are missing links predictable? An inferential benchmark for knowledge graph completion," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 6855–6865.

Fobo Shi received the MS degree from Central China Normal University in Wuhan, China, in 2024. He is currently working toward the PhD degree with the School of Information Management, Wuhan University, Wuhan, China. His primary research interests lie in large language models and knowledge graphs. He is exploring the interconnectivity of knowledge within these models and graphs, with the aim of developing more robust algorithms for knowledge embedding, enhancing large language model reasoning, and organizing knowledge more effectively.

Duantengchuan Li received the MS degree from Central China Normal University, Wuhan, China, in 2021, and the PhD degree in software engineering with the School of Computer Science, Wuhan University, Wuhan, China. He has frequently been a reviewer for serveral international journals, including *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Industrial Informatics*, *Information Processing & Management*, *Knowledge Based Systems*, *Engineering Applications of Artificial Intelligence*, *Neurocomputing*, etc. His research interests include computer vision, recommendation system, representation learning, knowledge graph, natural language processing, intention recognition etc.

Xiaoguang Wang received the MS degree in informatics from Wuhan University, in 2004, and the PhD degree in management science and engineering from Wuhan University, in 2007. He is currently a professor with the School of Information Management, Wuhan University, and he is also executive vice dean of the Big Data Institute, Wuhan University. His research interests include pattern recognition, knowledge graph, knowledge organization, semantic publishing, and digital humanities.

Bing Li received the PhD degree from the Huazhong University of Science and Technology, Wuhan, China, in 2003. He is currently a professor with the School of Computer Science, Wuhan University. His main research areas are services computing, software engineering, artificial intelligence, and cloud computing. He is now a distinguished member of the China Computer Federation (CCF).

Xindong Wu (Fellow, IEEE) received the bachelor's and master's degrees in computer science from the Hefei University of Technology, Hefei, China, and the PhD degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K., in 1993. He is a senior research scientist with the Research Center for Knowledge Engineering, Zhejiang Lab, China. His research interests include Big Data analytics, data mining, knowledge engineering, and knowledge-based systems. He was an editor-in-chief of *IEEE Transactions on Knowledge and Data Engineering* (2005–2008) and *Knowledge and Information Systems*, and is the Steering Committee chair of the IEEE International Conference on Data Mining. He is a Foreign member of the Russian Academy of Engineering and a fellow of American Association for the Advancement of Science.