Entity-Relation Guided Random Walk for Link Prediction in Knowledge Graphs

Weisheng Li, Hao Zhong, Ronghua Lin, Chao Chang, Zhihong Pan, and Yong Tang

***Abstract*—Knowledge graphs (KGs) are structured knowledge bases that represent information as a collection of interconnected entities and relations. Link prediction in KGs aims to infer missing or potential links between entities based on triple facts. Among different link prediction methods, knowledge graph embedding (KGE) has gained widespread popularity, with the goal of learning low-dimensional representations for KGs. However, most present KGE methods struggle to capture both local and global neighborhood information efficiently. Additionally, many hybrid methods have limitations in modeling and capturing interactions between triples. In this article, we propose an entity-relation-guided random walk (ERGRW) method for link prediction in KGs. Unlike conventional approaches that solely focus on entity-based walks, ERGRW creatively introduces relations as objects to walk as well. Inspired by distance-based methods, we design novel random walk rules based on the translation principle within triples. Thus, the ERGRW not only captures local and global neighborhood information but also discovers potential semantic relationships and interactions in the KGs. Furthermore, the encoder–decoder framework of ERGRW is able to learn comprehensive representation and improve link prediction performance. Extensive experiments conducted on four standard datasets demonstrate the superiority of ERGRW for link prediction.**

***Index Terms*—Graph neural networks, knowledge graph embedding (KGE), link prediction, random walk.**

I. Introduction

KNOWLEDGE graphs (KGs) are knowledge bases that integrate data using graph-structured representation models or topologies, such as Freebase [1], WordNet [2], and YAGO [3], and include a multitude of triple facts (h, r, t), where h, r, and t denote head entities, relations, and tail entities, respectively. In recent years, KGs have received substantial attention and are extensively utilized in diverse domains, such as knowledge completion [4], [5], [6], information retrieval [7], [8], [9], and recommendation systems [10], [11], [12]. Although the considerable scale triples in most existing KGs, KGs still suffer from significant incompleteness due to numerous missing links. The completion of missing links is crucial for improving the quality and usability of KGs. For instance, given the triple $(h, r, ?)$, the goal is to correctly predict the target entity t. Link prediction is a downstream task in KG completion, with the purpose of predicting the missing links between triples [i.e., $(h, r, ?)$ or $(?, r, t)$].

Knowledge graph embedding (KGE) is one of the most popular link prediction methods due to its efficacy and versatility. KGE is able to capture complex interactions and semantic structure by learning low-dimensional representations for entities and relations within KGs. Early works related to KGE primarily focused on distance-based methods (DBMs). These methods intend to evaluate the plausibility of facts by calculating the distance between two entities following specific translations. For instance, TransE [13], a representative method, enforces the equation $h + r \approx t$ into vector space by translating the embedding of the head entity using relation embedding to approximate the embedding of the tail entity. Many other translation-based methods, like TransH [14] and TransR [15], extend the TransE and address its shortcomings when dealing with 1-to-N, N-to-1, and N-to-N relations in KGs.

While DBMs provide intuitive and effective techniques for KGE, most of them face performance degradation when tasked with modeling complex relations, and they are not suitable for complex scenarios involving the same entities with different relations. The advent of random walk [16] has led to numerous advancements in the field of KGE, such as GraphRNA [17], CGAT [18], and CNARW [19]. Random walk refers to traversing a graph by moving from one entity to another, randomly selecting the next entity based on certain probabilities determined by the edges in the graph. To enhance capabilities in modeling complex relations and higher order interactions, semantic matching-based methods, also known as bilinear methods, exploit the semantic relevance between entities and relations to better model complex interactions in KGs. These methods include but are not limited to DistMult [20], ComplEx [21], and

Manuscript received 24 August 2023; revised 29 November 2023 and 24 January 2024; accepted 19 March 2024. This work was supported in part by the National Key Research and Development Program of China (Research and Demonstration Application of Key Technologies for Personalized Learning Driven by Educational Big Data) under Grant 2023YFC3341200, the National Natural Science Foundation of China under Grant 62377015, and the Research Cultivation Fund for The Youth Teachers of South China Normal University under Grant 23KJ29. *(Corresponding author: Yong Tang.)*

Weisheng Li, Hao Zhong, Ronghua Lin, and Yong Tang are with the School of Computer Science, South China Normal University, Guangzhou, Guangdong 510632, China and Pazhou Lab, Guangzhou, China (e-mail: liws@m.scnu.edu.cn; hzhong@m.scnu.edu.cn; rhlin@m.scnu.edu.cn; ytang@scnu.edu.cn).

Chao Chang is with the School of Information Engineering, Guangzhou Panyu Polytechnic, Guangzhou, Guangdong 511483, China and Pazhou Lab, Guangzhou, China (e-mail: changchao@m.scnu.edu.cn).

Zhihong Pan is with the School of Artificial Intelligence, South China Normal University, Foshan, Guangdong 528225, China (e-mail: zhpan@m.scnu.edu.cn).

Digital Object Identifier 10.1109/TCSS.2024.3382263

HolE [22]. Semantic matching typically adopts the multiplication formulation $\mathbf{h}^\top \mathbf{M}_r \approx \mathbf{t}^\top$ to transform the embedding of head entities in the representation space, aiming to approximate the embedding of tail entities. Meanwhile, matrix factorization is also a crucial technique for KGE. Taking RESCAL [23] as an example, it enhances matrix factorization by employing a three-way tensor factorization to model the KG structure. However, semantic matching-based methods (SMBMs) still struggle to capture fine-grained semantics and handle ambiguous relations or entities, and they often rely on pretrained language models or external semantic resources.

Recently, neural networks have exhibited substantial capabilities when constructing encoding models based on semantic matching or translation, including SLM [24], MLP [25], and NAM [26]. Neural network-based methods (NNBMs) take entity and relation embedding as inputs, employing nonlinear activation functions and complex network architectures to predict the probability of factual triples. As one of the representative methods in convolutional neural networks (CNNs), ConvE [27] employs two-dimensional convolutional layers to learn deep representations of KGs, where the interactions between the input of entities and relations are captured through the convolutional and fully connected layers. InteractE [28] reshapes feature permutation and employs circular convolution to enhance the interaction between entity and relation embeddings. Unlike ConvE, which only captures local relations, ConvKB [29] extends TransE by considering global relations between entries of embedded triples in the same dimension, thereby preserving transitional features. R-GCN [30] applies graph convolution operations to the neighbors of each entity to learn KG representations. KBGAT [31] incorporates graph attention networks (GATs) [32] in order to obtain attention-based representations of entities and relations. Despite the successes achieved by NNBMs, most of them may struggle to effectively capture complex interactions between entities and relations when the KG is sparse or incomplete. Moreover, they still struggle to effectively balance the capture of local and global information.

To address the aforementioned issues, we propose a novel method of entity-relation-guided random walk (ERGRW) for link prediction in KGs. Unlike the traditional strategy of random walk, ERGRW introduces random walk with designed rules to traverse the graph and collect more comprehensive and interactive sequences information, thus capturing more comprehensive local and global information in KGs. This enables ERGRW to capture potential correlations within neighborhoods and discover long-range dependencies between entities, effectively digging into the deep structure of KGs. In addition, the model employs an encoder–decoder framework for learning KG representation, which effectively models the interactions between entities and relations, leading to comprehensive representations and improved link prediction performance.

In summary, the main contributions of ERGRW we propose are summarized as follows:

1) We propose a model called ERGRW that introduces a novel random walk procedure guided by both entities and relations in the KGs. ERGRW effectively captures local and global neighborhood information in the KGs, thereby enhancing its ability to detect meaningful associations between entities.
2) We design an end-to-end encoder–decoder framework, which promotes collaboration and interaction among multiple components, facilitating the model to effectively capture the complex long-range dependencies and learn the comprehensive representation of the KGs.
3) Evaluation for link prediction tasks on four standard datasets demonstrates the superiority of ERGRW, indicating the capabilities of ERGRW and highlighting its potential real-world applications in KG scenarios.

The subsequent sections of this article are structured as follows. There is a brief summary of related work in Section II. Section III describes the proposed model ERGRW in detail. Extensive experiments are reported in Section IV. In Section V, we draw a conclusion to our work and discuss potential future works.

II. Related Work

In this section, the methods introduced to address link prediction tasks in KGs are mainly classified into three categories: DBMs, SMBMs, and NNBMs. The score functions, embedding, and time complexity of these methods are shown in Table I.

A. DBMs

DBMs are among the earliest KGE techniques, employing distance-based score functions to evaluate fact plausibility as the distance between two entities. TransE [13] is a widely recognized method that represents entities and relations as vectors within the same vector space. Suppose a fact (h, r, t), TransE interprets r as a translation vector, allowing the embedded h and t to be connected by r with low error. In other words, if the fact (h, r, t) is true, it satisfies the equation $h + r \approx t$. Due to the inability of TransE to model complex relations (1-N, N-1, and N-N relations) in KGs, several extensions have emerged, including TransH [14], TransR [15], TransD [33], and others. STransE [34] builds on TransR by representing each entity and relation as a low-dimensional representation through the utilization of two matrices and a translation vector. To address the constraints posed by regularization in TransE, TorusE [35] introduces a modification by replacing the mapping space from a conventional vector space to a compact Lie group. MuRE [36] and MuRMP [37] are variants of TransE that incorporate multirelational information or diagonal relational matrix. CrossE [38] goes a step further by explicitly accounting for the bidirectional influence between entities and relations through the explicit modeling of cross-interactions. Drawing inspiration from Euler's formula, RotatE [39] employs complex vector spaces to map entities and relations, characterizing each relation as a rotation from the head entity to the tail entity.

B. SMBMs

SMBMs, also known as bilinear methods, employ similarity-based score functions to evaluate the plausibility of facts.

TABLE I
SCORE FUNCTIONS, EMBEDDING, AND TIME COMPLEXITY OF THE USED STATE-OF-THE-ART KG EMBEDDING METHODS

Category	Method	Score function	Embedding	$\mathcal{O}_{time}$
DBMs	TransE	$-\Vert\mathbf{h}+\mathbf{r}-\mathbf{t}\Vert_{L_1/L_2}$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^d$	$\mathcal{O}(d)$
	TransH	$-\left\Vert\left(\mathbf{h}-\mathbf{w}_r^\top\mathbf{h}\mathbf{w}_r\right)+\mathbf{r}-\left(\mathbf{t}-\mathbf{w}_r^\top\mathbf{t}\mathbf{w}_r\right)\right\Vert_2^2$	$\mathbf{h},\mathbf{r},\mathbf{t},\mathbf{w}_r\in\mathbb{R}^d$	$\mathcal{O}(d)$
	TransR	$-\Vert\mathbf{M}_r\mathbf{h}+\mathbf{r}-\mathbf{M}_r\mathbf{t}\Vert_2^2$	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d,\mathbf{r}\in\mathbb{R}^k,\mathbf{M}_r\in\mathbb{R}^{k\times d}$	$\mathcal{O}(d^2)$
	TransD	$-\left\Vert\mathbf{h}\left(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I}\right)+\mathbf{r}-\left(\mathbf{w}_r\mathbf{w}_r^\top+\mathbf{I}\right)\right\Vert_2^2$	$\mathbf{h},\mathbf{r},\mathbf{w}_h,\mathbf{w}_t\in\mathbb{R}^d$	$\mathcal{O}(d)$
	STransE	$\left\Vert\mathbf{M}_{r,1}\mathbf{h}+\mathbf{r}-\mathbf{M}_{r,2}\mathbf{t}\right\Vert_{1/2}$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^d,\mathbf{M}_{r,1},\mathbf{M}_{r,2}\in\mathbb{R}^{d\times d}$	$\mathcal{O}(d^2)$
	TorusE	$\min_{(\mathbf{x},\mathbf{y})\in([\mathbf{h}]+[\mathbf{r}])\times[\mathbf{t}]}\Vert\mathbf{x}-\mathbf{y}\Vert_i$	$[\mathbf{h}],[\mathbf{r}],[\mathbf{t}]\in\mathbb{T}^n$	$\mathcal{O}(d)$
	MuRS	$-d^{(\mathbf{r})}_{\mathbb{S}^d_K}(\mathbf{h},\mathbf{t})^2+b_h+b_t$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{S}^d_K$	$\mathcal{O}(d)$
	MuRE	$-d^{(\mathbf{r})}_{\mathbb{E}^d_K}(\mathbf{h},\mathbf{t})^2+b_h+b_t$	$\mathbf{h},\mathbf{t}\in\mathbb{E}^d_K,\mathbf{r}\in\mathbb{R}^{d\times d}$	$\mathcal{O}(d^2)$
	RotatE	$\Vert\mathbf{h}\circ\mathbf{r}-\mathbf{t}\Vert$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{C}^d$	$\mathcal{O}(d)$
	HakE	$\Vert\mathbf{h}_m\circ\mathbf{r}_m-\mathbf{t}_m\Vert_2+\lambda\Vert\sin\left(\left(\mathbf{h}_p+\mathbf{r}_p-\mathbf{t}_p\right)/2\right)\Vert_1$	$\mathbf{h}_m,\mathbf{r}_m,\mathbf{t}_m\in\mathbb{R}^d$	$\mathcal{O}(d)$
	QuatE	$\mathbf{h}\otimes\frac{\mathbf{r}}{\vert\mathbf{r}\vert}\cdot\mathbf{t}$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{H}^d$	$\mathcal{O}(d)$
	CrossE	$\sigma\left(\tanh\left(\mathbf{c}_r\circ\mathbf{h}+\mathbf{c}_r\circ\mathbf{h}\circ\mathbf{r}+\mathbf{b}\right)\mathbf{t}^\top\right)$	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d,\mathbf{r}\in\mathbb{R}^d$	$\mathcal{O}(d)$
	MuRMP	$-d^{\mathbf{r}}_{\mathbb{P}^{\{d_o,d_h,d_s\}}_{\{K_o,K_h,K_s\}}}(\mathbf{h},\mathbf{t})^2+b_h+b_{\mathbf{t}}$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{P}^d_K$	$\mathcal{O}(d^2)$
SMBMs	RESCAL	$\mathbf{h}^\top\mathbf{M}_r\mathbf{t}$	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d,\mathbf{M}_r\in\mathbb{R}^{d\times d}$	$\mathcal{O}(d^2)$
	DistMult	$\mathbf{h}^\top\mathrm{diag}(\mathbf{r})\,\mathbf{t}$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^d$	$\mathcal{O}(d)$
	ComplEx	$\mathrm{Re}\left(\mathbf{h}^\top\mathrm{diag}(\mathbf{r})\,\bar{\mathbf{t}}\right)$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{C}^d$	$\mathcal{O}(d)$
	HolE	$\mathbf{r}^\top(\mathbf{h}\star\mathbf{t})$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^d$	$\mathcal{O}(d\log(d))$
	ANALOGY	$\mathbf{h}^\top\mathbf{M}_r\mathbf{t}$	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d,\mathbf{M}_r\in\mathbb{R}^{d\times d}$	$\mathcal{O}(d)$
	SimplE	$\frac{1}{2}(\mathbf{h}\circ\mathbf{r}\mathbf{t}+\mathbf{t}\circ\mathbf{r}'\mathbf{t})$	$\mathbf{h},\mathbf{r},\mathbf{t},\mathbf{r}'\in\mathbb{R}^d$	$\mathcal{O}(d)$
	TuckER	$\mathbf{w}\times_1\mathbf{h}\times_2\mathbf{r}\times_3\mathbf{t}$	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d_e,\mathbf{r}\in\mathbb{R}^d_r$	$\mathcal{O}(d^3)$
NNBMs	R-GCN	$\boldsymbol{e}_s^T\mathbf{R}_r\boldsymbol{e}_o$	$\boldsymbol{e}_s,\boldsymbol{e}_o\in\mathbb{R}^d,\mathbf{M}\in\mathbb{R}^{d\times d}$	$\mathcal{O}(d^2)$
	ConvE	$\sigma\left(\mathrm{vec}\left(\sigma\left([\mathbf{M}_h;\mathbf{M}_r]*\omega\right)\right)\mathbf{W}\right)\mathbf{t}$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^d$	$\mathcal{O}(k^2d)$
	ConvKB	$\mathrm{concat}(g([\mathbf{h},\mathbf{r},\mathbf{t}]*\omega))\mathbf{w}$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^d$	$\mathcal{O}(kd)$
	HypER	$\sigma\left(\tanh\left(\mathbf{c}_r\circ\mathbf{h}+\mathbf{c}_r\circ\mathbf{h}\circ\mathbf{r}+\mathbf{b}\right)\mathbf{t}^\top\right)$	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d,\mathbf{r}\in\mathbb{R}^d$	$\mathcal{O}(d^2)$
	CapsE	$\Vert\mathrm{capsnet}(g([\mathbf{h},\mathbf{r},\mathbf{t}]*\omega))\Vert$	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d,\mathbf{r}\in\mathbb{R}^d$	$\mathcal{O}(d)$
	A2N	$g\left(\mathbf{h}^\top\right)\mathrm{diag}\left(g\left(\mathbf{r}\right)\right)g\left(\mathbf{t}\right)$	$\mathbf{h},\mathbf{t},\mathbf{r}\in\mathbb{R}^{K\times 2K}$	$\mathcal{O}(d^2)$
	ConvR	$f\left(\mathbf{W}f\left(\mathbf{h}*\omega_r\right)+\mathbf{b}\right)^T\mathbf{t}$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^d$	$\mathcal{O}(kd)$
	KBGAT	$g\left(\mathrm{Re}\left(\mathbf{h},\mathbf{r},\mathbf{t}\right)*\omega^m\right)\mathbf{w}$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^d$	$\mathcal{O}(d^2)$
	InteractE	$g(\mathrm{vec}(f(\phi([\mathbf{h},\mathbf{r}])\circledast w))\mathbf{W})\mathbf{h}$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^d,\mathbf{w}_r\in\mathbb{R}^{k\times k}$	$\mathcal{O}(k^2d)$
	NodePiece	$\Vert\mathbf{h}\circ\mathbf{r}-\mathbf{t}\Vert$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{C}^d$	$\mathcal{O}(d)$
	RMCNN	$\mathrm{vec}(f(\mathbf{B}*\Omega))\times\mathbf{W}\cdot\mathbf{w}$	$\mathbf{h},\mathbf{r},\mathbf{t}\in\mathbb{R}^d$	$\mathcal{O}(d^3)$

Note: $\mathbb{R}^d$: d-dimensional real-valued space; $\mathbb{H}^d$: d-dimensional quaternion space; $\mathbb{T}^d$: d-dimensional torus space; $\mathbb{C}^d$: d-dimensional complex space; d: vector dimension; k: number of convolutional kernels; $\circ$: Hadamard product; $\otimes$: Hamilton product; $*$: convolution operation; $\star$: circular correlation; ω: convolution filter; $[\mathbf{M}_r]_{ij}$: ijth element of matrix $\mathbf{M}_r$; $\sigma(\cdot), g(\cdot)$: nonlinear activation functions; $\mathbf{B}$: feature matrix; $\mathbf{W}$: weight matrix; $\mathbf{w}$: weight vector; $concat\,()$, $[h,r]$: vector or matrix Concatenation.

RESCAL [23], as one of the earliest SMBMs, leverages tensor decomposition on KGs to obtain entity embedding and relation matrices. Similarly, SimplE [40] employs tensor factorization techniques to solve the issue of independence between an entity's two embedding vectors. Based on RESCAL, ANALOGY [41] models the analogical structure of relational data through multirelational reasoning. DistMult [20] captures tripartite interactions among relations, head entities, and tail entities. Due to DistMult's limitations in effectively modeling antisymmetric relations, ComplEx [21] introduces complex-valued embedding, so that both symmetric and antisymmetric relations can be captured. As a special instance of ComplEx, HolE [22] exploits a specific scenario where head and tail entity embedding exhibit circular correlation, contributing to its ability to obtain comprehensive representations of compositional vector space in KGs. TuckER [42] learns embedding for entities and relations by generating a core tensor alongside the corresponding embedding vectors.

C. NNBMs

NNBMs have attracted considerable interest in the field of KGE in recent years. ConvE [27] employs two-dimensional convolutional layers to model the interactions among input from entities and relations, subsequently followed through fully connected layers for additional encoding. ConvKB [29] extends ConvE by considering global relationships among entities through modeling the embedding of a triple's head, relation, and tail. On the other hand, R-GCN [30] employs relation-specific weights in graph convolutional networks to aggregate and propagate information effectively through different relations. HypER [43] extends ConvE by incorporating a hypernetwork that yields convolutional filters with relational specificity. CapsE [44] incorporates two capsule layers to achieve KGE after ConvKB's feature mapping using convolution. Meanwhile, A2N [45] introduces attention mechanisms to identify and weigh different aspects of entities and relations. KBGAT [31] extends

GAT [32] by using GATs to aggregate neighborhood information. ConvR [46] extends ConvE by introducing adaptive convolution with relation-specific filters. NodePiece [47] adopts an anchor-based method to model a fixed-size entity vocabulary. RMCNN [48] combines relational memory networks and CNNs for KGE. In addition, InteractE [28] improves convolution-based KGE by increasing entity–relation feature interactions.

Despite their effectiveness, the existing methods in each category have their limitations. Although DBMs are intuitive, they encounter difficulties in managing complex relations, especially those involving multiple entities and intricate patterns. Moreover, they still struggle to provide accurate embeddings when dealing with incomplete graphs or sparse information, as they heavily rely on pairwise distances between entities. SBMSs face challenges in effectively capturing intricate relations and interactions within KGs. NNBMs struggle to effectively capture semantic relationships and interactions when dealing with entities or relations with limited connections in the KGs. To tackle these challenges, we propose our model ERGRW, which leverages ERGRW to capture both local and global neighborhood information. ERGRW also effectively embeds KGs using an encoder–decoder framework, enabling better modeling of entity–relation interactions and achieving improved performance in link prediction tasks.

III. The Proposed Model

We present a thorough explanation of our proposed ERGRW in this section, including preliminaries, framework overview, encoder, decoder, training ERGRW, and complexity analysis.

A. Preliminaries

Definition 1 (KG): KG can be defined as a directed graph $G=(E,\mathcal{E},R)$, where $E=\{e_1,e_2,\ldots,e_m\}$ denotes the set of entities, $\mathcal{E}=\{\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_n\}$ denotes the set of directed edges, $\varepsilon=\{(e_i,e_j)|e_i,e_j\in E\}$ denotes a directed relation from entity e_i to entity e_j, and $R=\{r_1,r_2,\ldots,r_k\}$ denotes relation types set associated with the edges.

Definition 2 (KGE): The purpose of KGE is to employ mapping functions that convert relations and entities into continuous and low-dimensional vector representations. Given a graph $G=(E,\mathcal{E},R)$, the task of KGE aims to learn mapping functions $\phi: E\rightarrow\mathbb{R}^d$ and $\psi:\mathcal{E}\rightarrow\mathbb{R}^d$ that encode entities and relations of KGs respectively into d-dimensional vector representations.

Definition 3 (Link Prediction): Link prediction in a KG refers to the task of predicting missing or potential relationships between entities. In KGs, link prediction is the downstream task of identifying missing or potential relations between entities. Given a knowledge graph $G=(E,\mathcal{E},R)$, the purpose of link prediction is to discover missing or potential edges $\hat{\mathcal{E}}$ such that $\hat{\mathcal{E}}\subset E\times E\setminus\mathcal{E}$. The predicted edges denote potential relations that could exist in the KGs.

B. Framework Overview

The framework of ERGRW is illustrated in Fig. 1. Our proposed ERGRW is composed of two main components: an encoder and a decoder. Specifically, the encoder is in charge of capturing the semantic information from the KG through a series of steps, while the decoder utilizes the encoded information to perform link prediction. First, a graph with both entities and relations subject to random walk is transformed from the KG. By incorporating the ERGRW, it effectively captures semantic information and discovers the interactions and dependencies within the KG. Subsequently, the generated sequences are input to a BiLSTM layer, which allows for the encoding of sequential information by taking into account both the forward and backward context of the sequences. The BiLSTM's output is subsequently subjected to a mean pooling operation, facilitating the fusion of vector representations. On the other hand, the decoder utilizes the pooled representation from the encoder as the input and applies the InteractE model to predict links between triples of the KG.

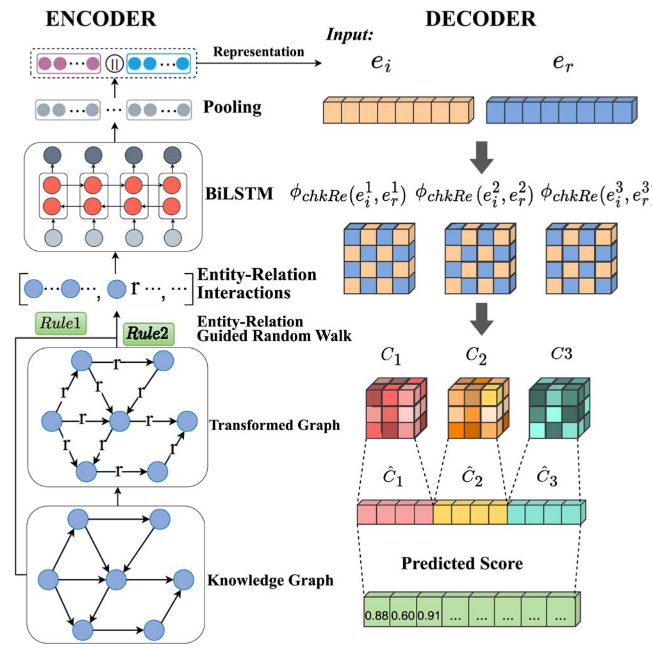

Fig. 1. Overall framework of ERGRW is composed of an encoder and a decoder. In order to capture sequential information, the encoder utilizes two specific rules to perform random walk over a KG, generating context sequences that are then merged and processed by a bidirectional long short-term memory (BiLSTM) model. Then, mean pooling is employed to obtain a fixed-length representation, which is subsequently input into the InteractE model for link prediction.

C. Encoder

The encoder plays an essential role in capturing semantic information and discovering interactions and dependencies within the KG. In this part, we provide a detailed description of the encoder of ERGRW, including rule guidance, ERGRW, recurrent neural networks, and representation fusion.

1) Rule Guidance: The goal of rule guidance is to generate interactive sequences by random walk. These interactive sequences provide a valuable information source, enabling the model to detect and capture the local and global graph information and interactions in the KG. Traditional methods of KGs

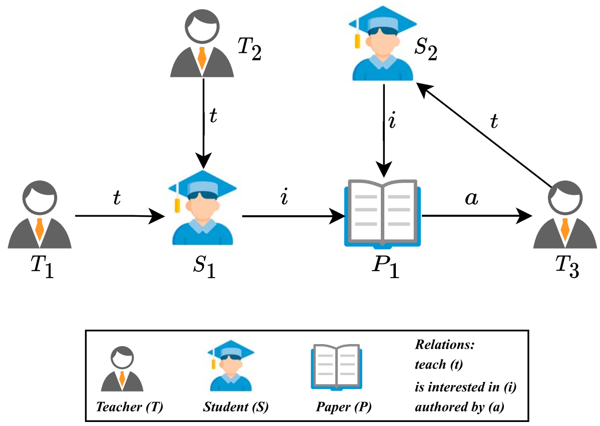

Fig. 2. KG taking an academic social network as an example.

often apply random walk on entities based solely on the edges connecting them, neglecting the rich variety of relation types between them. Therefore, we design and combine two rules for random walks, namely $Rule1$ and $Rule2$.

As illustrated in Fig. 2, an academic social network serves as an example of two rules. $Rule1$ conducts random walk on entities in the same manner as traditional methods, traversing from one entity to another without considering their relationships. For example, starting from S_1, the sequence set could be: $\{T_1, S_1, P_1, T_3, S_2\}$. For $Rule2$, Fig. 2 demonstrates that both teacher 1 and teacher 2 teach student 1. From this, we infer a high correlation and interaction between teacher 1 and teacher 2. Specifically, starting from T_1 based on the relation t to S_1, considering the triples (T_1, t, S_2) and (T_2, t, S_2), t and T_2 are incorporated into the sequence set, resulting in (T_1, t, T_2). Next, from S_1 based on the relation i to P_1, similarly, considering triples (S_1, i, P_1) and (S_2, i, P_1), it can be observed that both student 1 and student 2 are interested in paper 1. This suggests a potential connection between student 1 and student 2. Subsequently, we include i and S_2 in the sequence set, resulting in $\{T_1, t, T_2, i, S_2\}$.

Comparing the sequences obtained under $Rule1$ and $Rule2$, the random walk process involving only entities and directed edges leads to S_2 through $Rule1$, resulting in a sequence set with topological structural characteristics. On the other hand, the random walk process with $Rule2$ not only finds S_2 with potential associations but also identifies T_2 with high interactivity. Additionally, the relations of S_2 and T_2 are included in the sequence set, enhancing their interactions. It indicates that even in a sparse graph, $Rule2$ can capture entities with high correlation, not limited to whether the structural closeness between entities is sufficient. Therefore, by leveraging the strengths of both $Rule1$ and $Rule2$, we employ ERGRW to capture more interactions. The detailed theoretical description will be described in the next subsection.

2) ERGRW: As shown in Fig. 3, we obtain sequences ζ_1 and ζ_2 generated by the two rules with a ratio of α and $1-\alpha$, respectively. From the KG G, the adjacency matrices G_1 (with entities as rows and columns), G_2 (with entities as rows and relations as columns), and G_3 (with relations as rows and entities as columns) can be obtained then normalized to $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_3$. In the case of $Rule1$, sequences consisting of entities can be obtained, such as $\{e_1, e_2, e_5, e_7, e_6\}$. The probability of an entity walking to the next entity based on directed edges is denoted as

$$P(e_i \to e_j) = \frac{\phi_{ij}}{\sum_{k=1}^{n} \phi_{ik}} \tag{1}$$

where ϕ_{ij} represents the existence of the edge from entity e_i to entity e_j in the adjacency matrix, and $\sum_{k=1}^{n} \phi_{ik}$ represents the total number of neighboring entities of entity e_i. With a sequence set ratio of α, the corresponding transition probability matrix could be written as

$$\mathbf{P} = [\alpha \mathcal{G}_1] \in \mathbb{R}_+^{n \times n}. \tag{2}$$

To incorporate relations into the random walk, $Rule2$ adopts a strategy of treating relations as entities based on the transformed KG. Similarly, the probability that an entity random walk to a relation can be represented as

$$P(e_i \to r_j) = \frac{\lambda_{ij}}{\sum_{k=1}^{m} \lambda_{ik}}. \tag{3}$$

With a sequence set ratio of $1-\alpha$, the corresponding transition probability matrix could be written as

$$\mathbf{P} = [(1-\alpha)\mathcal{G}_2] \in \mathbb{R}_+^{n \times m}. \tag{4}$$

Similarly, the probability that a relation random walk to an entity can be represented as

$$P(r_i \to e_j) = \frac{\delta_{ij}}{\sum_{k=1}^{n} \delta_{ik}}. \tag{5}$$

With a sequence sets ratio of $1-\alpha$, the corresponding transition probability matrix could be written as

$$\mathbf{P} = [(1-\alpha)\mathcal{G}_3] \in \mathbb{R}_+^{m \times n}. \tag{6}$$

Therefore, based on (3) and (5), the probability of transitioning from e_i to e_j under $Rule2$ can be defined as follows:

$$P(e_i \to e_j) = \frac{\lambda_{ij} \cdot \delta_{ij}}{\sum_{k=1}^{m} \lambda_{ik} \cdot \sum_{k=1}^{n} \delta_{ik}}. \tag{7}$$

Because KGs contain relations and their inverse relations, it is common to observe duplicate entities and relations in the sequences obtained by the random walk process, and the sequences can be walked to a specified length. These sequences also play a crucial role in capturing semantic information that characterizes graph features. A detailed parameter analysis of the walk length will be provided in the experimental section.

The final sequence ζ is generated by applying the weighted combination of sequences ζ_1 and ζ_2 according to $Rule1$ and $Rule2$. By assigning a weight of α to ζ_1 and $1-\alpha$ to ζ_2, we effectively incorporate both the entity-to-entity random walk rule and the rule utilizing entity–relation connections. The merged sequence ζ represents a consolidated representation that incorporates both the entity-based random walks and the innovative rule utilizing entities and relations. The resulting sequences ζ serve as a comprehensive representation for the subsequent recurrent neural network (RNN) step.

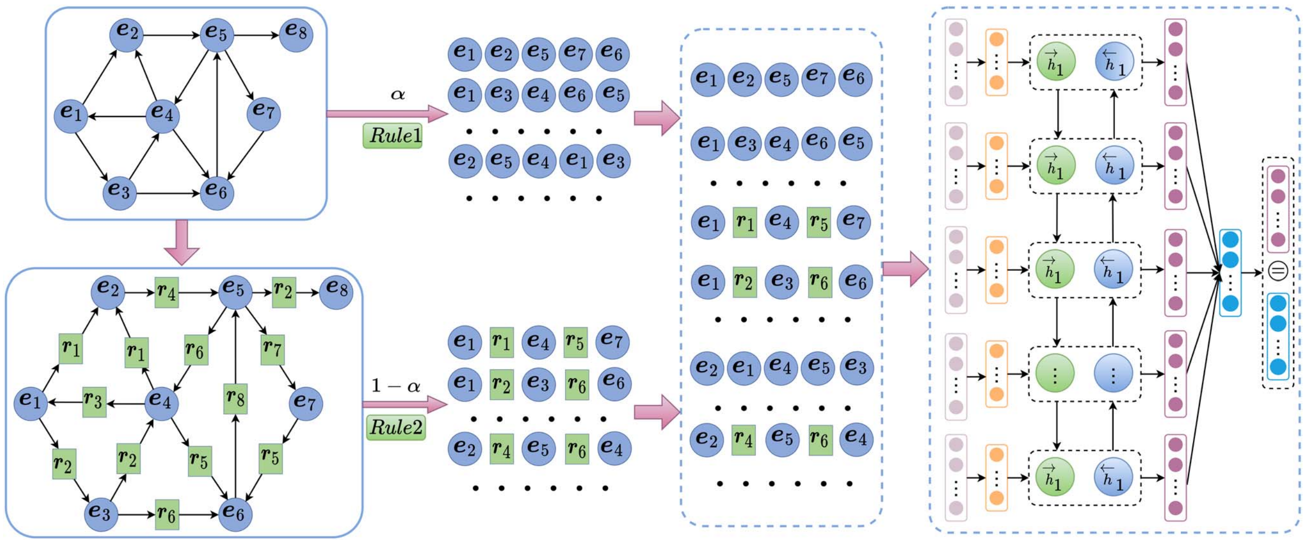

Fig. 3. KG is transformed into a new graph where random walk can be performed on both entities and relations. Then, random walks are performed using two rules. $Rule1$ focuses on entity-to-entity walks without relation information and generating sequences of entities. $Rule2$ combines both entity and relation information. Starting from an entity, if its outgoing edges lead to the same target entity and have the same relation type, a direct jump to the target entity is allowed. By applying these two rules, the sequences are obtained from random walks. Finally, these sequences are fed into BiLSTM for capturing sequential information and contextual relevance.

3) RNNs: The proposed model incorporates BiLSTM to capture sequence information and encode it into vector representations. BiLSTM is a sort of RNN that can capture dependencies and patterns from both past and future context information. By encoding the sequential information in vector representations, BiLSTM can enhance the semantic information and graph representations.

The initial step is to apply a fully connected layer to reduce the dimensionality of each element in the sequence

$$\boldsymbol{x}_t = \sigma\left(\boldsymbol{e}_i \boldsymbol{W}_a + \boldsymbol{b}_a\right) \text{ or } \boldsymbol{x}_t = \sigma\left(\boldsymbol{r}_i \boldsymbol{W}_a + \boldsymbol{b}_a\right) \tag{8}$$

where $\boldsymbol{e}_i$, $\boldsymbol{r}_i$, σ, $\boldsymbol{W}_a$, and $\boldsymbol{b}_a$ represent entities, relations, sigmoid function, weight matrix, and bias vector, respectively.

During forward propagation of BiLSTM, the input gate, forget gate, output gate, candidate cell state, cell state, and hidden state can each be computed, respectively, as

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{ix}\boldsymbol{x}_t + \boldsymbol{W}_{ih}\overrightarrow{\boldsymbol{h}_{t-1}} + \boldsymbol{b}_i) \tag{9}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_{fx}\boldsymbol{x}_t + \boldsymbol{W}_{fh}\overrightarrow{\boldsymbol{h}_{t-1}} + \boldsymbol{b}_f) \tag{10}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{ox}\boldsymbol{x}_t + \boldsymbol{W}_{oh}\overrightarrow{\boldsymbol{h}_{t-1}} + \boldsymbol{b}_o) \tag{11}$$

$$\tilde{\boldsymbol{c}}_t = \tanh(\boldsymbol{W}_{cx}\boldsymbol{x}_t + \boldsymbol{W}_{ch}\overrightarrow{\boldsymbol{h}_{t-1}} + \boldsymbol{b}_c) \tag{12}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t \tag{13}$$

$$\overrightarrow{\boldsymbol{h}_t} = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t) \tag{14}$$

where $\boldsymbol{W}$ represents weight matrix, $\boldsymbol{b}$ represents bias vector, $\odot$ represents the Hadamard product, and $\tanh(\cdot)$ denotes a hyperbolic tangent function. Likewise, the hidden state of backward propagation can be computed as

$$\overleftarrow{\boldsymbol{h}_t} = \boldsymbol{o}'_t \odot \tanh(\boldsymbol{c}'_t). \tag{15}$$

The final representation $\boldsymbol{h}_t$ is obtained by concatenating the forward hidden state $\overrightarrow{\boldsymbol{h}_t}$ and the backward hidden state $\overleftarrow{\boldsymbol{h}_t}$ at each time step which is computed as follows:

$$\boldsymbol{h}_t = \left[\overrightarrow{\boldsymbol{h}_t}, \overleftarrow{\boldsymbol{h}_t}\right]. \tag{16}$$

4) Representation Fusion: Mean pooling is further adapted to obtain the final representation of entire sequences. Mean pooling is a pooling operation that calculates the average value of each feature across the temporal dimension of the BiLSTM outputs. With the pooling method, information from the entire input sequence can be aggregated into a fixed-length representation.

Given a set $\{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_\ell\}$ of length ℓ, by applying mean pooling to $\{\boldsymbol{h}_2, \boldsymbol{h}_3, ..., \boldsymbol{h}_\ell\}$, a vector representation denoted as $\hat{\boldsymbol{h}}_i$ is obtained. The final representation of the entire sequence is obtained using the mean pooling method. Furthermore, the vector representation $\boldsymbol{r}_i$ of entity e_i can be generated by concatenating the pooled representation $\hat{\boldsymbol{h}}_i$ with the representation $\boldsymbol{h}_1$ of entity e_i

$$\boldsymbol{r}_i = \left[\hat{\boldsymbol{h}}_i, \boldsymbol{h}_1\right]. \tag{17}$$

The fusion of representations captures both the contextual information from the sequence and the individual semantics of the entity, providing a comprehensive representation $\boldsymbol{r}_i$ for the entity e_i. The representation fusion enables ERGRW to effectively incorporate essential features and relationships of the entities within the sequences.

D. Decoder

The decoder is in charge of processing the encoded representations further and utilizing them to accomplish the link prediction task. In this part, the InteractE model is employed

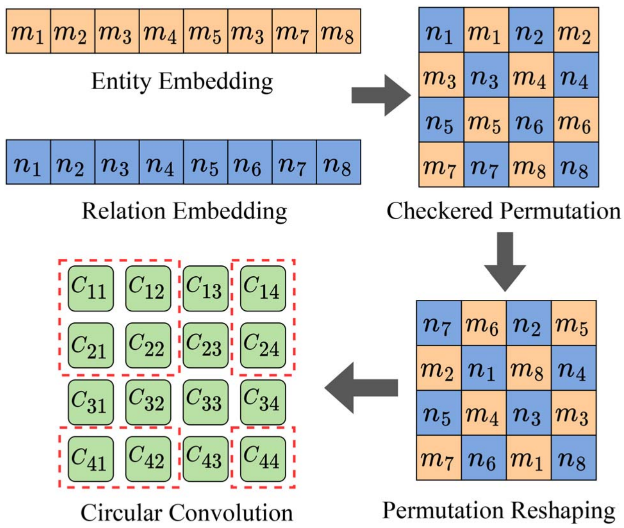

Fig. 4. Key steps of the decoder process. Note that the features of both entity embedding and relation embedding experience a checkered permutation, transforming into a matrix resembling a chequer. Subsequently, through permutation reshaping, a matrix with randomly arranged features is obtained. Finally, circular convolution is applied to convolve each reshaped permutation, with the red dashed lines indicating the positions where filters are applied.

because it offers several advantages and improvements over traditional methods.

InteractE leverages interactions of the KG to enhance the efficacy of the representation learning, allowing for the modeling of higher order interactions and the incorporation of context-specific information for link prediction tasks.

First, the vector representation $\boldsymbol{r}_i$ is obtained by the encoder is further linearly transformed into $\boldsymbol{\tau}_i$ using the following equation:

$$\boldsymbol{\tau}_i = \tanh(\boldsymbol{W}_l \boldsymbol{r}_i) + \boldsymbol{x}_i \tag{18}$$

where $\boldsymbol{W}_l$ is the weight matrix used for the linear transformation, $\boldsymbol{x}_i$ is obtained from (8), and $\tanh(\cdot)$ denotes an activation function.

As depicted in Fig. 4, let entity embedding and relation embedding be denoted as $\{m_1, \ldots, m_8\}$ and $\{n_1, \ldots, n_8\}$, respectively. Their features are arranged into a checkered matrix, enhancing the interaction between entity and relation embeddings, as opposed to a simple stacking configuration. Subsequently, the checkered matrix undergoes permutation reshaping, further increasing the interaction between entity and relation embeddings. Finally, circular convolution is applied to facilitate cross-channel sharing of filters. A set of kernel weights is trained across multiple input instances, capturing more complex and intricate interactions.

Specifically, in the decoder of our model, we adopted the setting of three-random permutations to implement feature permutation. The formula is as follows:

$$\boldsymbol{\mathcal{P}} = \left[\left(\boldsymbol{e}_i^1, \boldsymbol{e}_r^1\right); \left(\boldsymbol{e}_i^2, \boldsymbol{e}_r^2\right); \left(\boldsymbol{e}_i^3, \boldsymbol{e}_r^3\right)\right] \tag{19}$$

where the embedding of entities and relations are taken as inputs, denoted by $\boldsymbol{e}_i$ and $\boldsymbol{e}_r$ respectively, both with a dimension of d.

Subsequently, we applied the checkered reshaping operation $\left(\phi_{chkRe}\left(\boldsymbol{e}_i^t, \boldsymbol{e}_r^t\right), t \in \{1,2,3\}\right)$ to reshape the checkered permutations, aiming to capture more heterogeneous feature interactions

$$\phi_{chkRe}\left(\boldsymbol{\mathcal{P}}\right) = \left[\phi_{chkRe}\left(\boldsymbol{e}_i^t, \boldsymbol{e}_r^t\right) \mid t \in \{1,2,3\}\right]. \tag{20}$$

Following that, each reshaped permutation is convolved by circular convolution. Let i represent the subject entity, r represent the relation, and o represent the object entity. The score function for measuring the validity of the triple $\psi(i, r, o)$ can be calculated as follows:

$$\psi(s, r, o) = g\left(\operatorname{vec}\left(f\left(\phi_{chkRe}\left(\boldsymbol{\mathcal{P}}\right) \otimes \boldsymbol{w}\right)\right) \boldsymbol{W}\right) \boldsymbol{e}_o \tag{21}$$

where $\operatorname{vec}(\cdot)$ represents the vector concatenation operation, $\otimes$ represents the depthwise circular convolution operation, $\boldsymbol{w} \in \mathbb{R}^{k \times k}$ denotes a convolutional kernel with a size k, $\boldsymbol{W}$ serves as a learnable weight matrix, $\boldsymbol{e}_o$ denotes the matrix of object entity embedding, g represents sigmoid activation functions, and f represents ReLU functions.

E. Training ERGRW

The binary cross-entropy function is employed as the loss function in our proposed ERGRW

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{|E|} \left(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right) \tag{22}$$

where $|E|$ represents the entities set for which binary cross-entropy loss is calculated, entity i has a true label of y_i and a projected probability of p_i indicating whether entity i is part of the positive class. Through the comparison of predicted probabilities and true labels, the loss function gauges the prediction performance of ERGRW, which is then fine-tuned during training to minimize the loss.

For clarity, Algorithm 1 provides a concise overview of the whole ERGRW algorithm.

F. Complexity Analysis

Table I presents the time complexity ($\mathcal{O}_{time}$) of state-of-the-art KG embedding methods under the assumption of a single triple. We consistently employ the embedding dimension d to represent time complexity. For methods like TransE [13] and TransH [14], the time complexity is expressed as $\mathcal{O}(d)$ since they operate linear transformations of entity and relation embeddings in a manner dependent on d. In the case of CNN methods ConvE [27] and InteractE [28], their time complexity is expressed as $\mathcal{O}\left(k^2 d\right)$. The convolutional layer of these two methods involves k filters (or kernels), each of size $h \times w \times d$ where h and w are the height and width of the filter. The d in $k^2 d$ comes from the fact that each filter has d parameters in the convolutional operation. The k^2 term arises because there are k filters, and each filter has k output channels. Compared to traditional linear transformations, ConvE and InteractE better capture intricate and nonlinear relationships in the KGs, leveraging the expressive power of CNNs to model complex relationships beyond linear transformations. Unlike ConvE and InteractE, the time complexity of ConvKB [29] is $\mathcal{O}(kd)$ due

Algorithm 1 ERGRW

Input: Knowledge graph G; length of random walk: l; number of random walk: η; entities number: N_e; relations number: N_r; embedding dimension d; parameters α.

Output: The score $\psi(e_i, r, e_j)$ that e_i and e_j are connected.

for $i = 1$ to N_r **do**
    for $p = 1$ to η **do**
        Random walk starting from entity e_i via Eq. (3).
        Obtain sequences ζ_1 of length l via Eq. (4).
    end for
end for
for $i = 1$ to $N_e + N_r$ **do**
    for $q = 1$ to η **do**
        Random walk starting from entity e_i via Eq. (7).
        Obtain sequences ζ_2 via Eq. (4) and Eq. (6).
    end for
end for
Obtain sequences ζ by merging sequences ζ_1 and ζ_2 with the weights α and $1-\alpha$, respectively.
Concatenate forward hidden state $\overrightarrow{\boldsymbol{h}}_t$ and forward hidden state $\overleftarrow{\boldsymbol{h}}_t$ to get $\boldsymbol{h}_t$ via Eq. (16).
Apply mean pooling to $\{\boldsymbol{h}_2, \boldsymbol{h}_3, ..., \boldsymbol{h}_\ell\}$ yields $\hat{\boldsymbol{h}}_i$.
Concatenate $\hat{\boldsymbol{h}}_i$ and $\boldsymbol{h}_1$ to get vector representation $\boldsymbol{r}_i$ of e_i via Eq. (17).
$\boldsymbol{r}_i$ is further linearly transformed into $\boldsymbol{\tau}_i$ via Eq. (18).
Compute score $\psi(s, r, o)$ of the triples via Eq. (21).

to its use of concatenation in the convolutional operations. In a large-scale KG, the computation of KGE models is primarily determined by the number of entities and relations, embedding dimensions, and the scale of the training data.

Algorithm 1 illustrates the complete process of our proposed ERGRW. It can be observed that the time complexity of our proposed ERGRW is composed of two parts: encoder and decoder. Initially, the time complexity of the random walk component arises from two parts: $Rule1$ and $Rule2$, denoted as $\mathcal{O}(N_e l w)$ and $\mathcal{O}((N_e + N_r) l w)$, respectively. Assuming that the hidden unit is h, the time complexity of BiLSTM can be expressed as $\mathcal{O}(lh^2 + lhd)$. The time cost of the decoder section is primarily attributed to InteractE. Assuming that the size of the convolutional kernel is k, the time complexity of the decoder is $\mathcal{O}((N_e + N_r)k^2 d)$. It can be seen that the time cost of ERGRW is largely dominated by the decoder. Similarly, the time complexity of ERGRW can be expressed as $\mathcal{O}(k^2 d)$ under the assumption of a single triple.

IV. Experiments

In this section, extensive experiments are conducted on four standard datasets to validate the performance of ERGRW by comparing with baseline methods in the task of link prediction. First, the experimental settings are introduced, including datasets, compared methods, evaluation metrics, and parameter setting. Subsequently, comparison results and parameter analysis are reported and analyzed. Finally, we perform an ablation study to confirm the superior performance of ERGRW.

TABLE II
Statistics of the Used Four Datasets in the Experiments

Datasets	#Entity	#Relation	#Training set	#Validation set	#Testing set
FB15K	14 941	1345	483 142	50 000	59 071
FB15K-237	14 541	237	272 115	17 535	20 466
WN18RR	40 943	11	86 835	3034	3134
YAGO3-10	123 182	37	1 079 040	5000	5000

A. Experimental Settings

1) Datasets: We conduct experiments on four real-world datasets, including FB15k, FB15k-237, WN18RR, and YAGO3-10. The statistics of datasets are presented in Table II.

- *FB15k* [49] is a KG dataset extracted from the Freebase knowledge base, encompassing around 14 951 distinct entities and 592 213 triples. The dataset contains 1345 distinct relations, making it suitable for link prediction tasks and evaluating the performance of KGE methods.
- *FB15k-237* [50] is a variant of FB15k that focuses on evaluating the capabilities of KGE models. FB15k-237 achieves its reduction in relation count by removing redundant and inverse relations from the original FB15k dataset.
- *WN18RR* [51] is a dataset derived from WordNet, a comprehensive English lexical database. It is a reduced and reverse version of the original WN18 dataset, enriched with reverse relations to create a comprehensive and challenging link prediction dataset.
- *YAGO3-10* [52] is a large-scale KG dataset that combines information from multiple sources, including Wikipedia, WordNet, and GeoNames. These datasets serve as valuable resources for training and evaluating KGE models.

2) Compared Methods: In our experiments, the baseline methods of KGE are categorized into three types: DBMs, SMBMs, and NNBMs. DBMs, including *TransE* [13], *TransH* [14], *TransR* [15], *TorusE* [35], *RotatE* [39], *HakE* [53], *MuRE* [36], *MuRS* [36], *QuatE* [39], *CrossE* [38], and *MuRMP* [37], focus on modeling the relationships between entities by measuring the distances or transformations. SMBMs, including *RESCAL* [23], *DistMult* [20], *ComplEx* [21], *HolE* [22], *ANALOGY* [41], and *SimplE* [40], leverage bilinear product or complex-valued tensor products to better capture semantic similarity and compatibility between entities and relations. NNBMs, including *ConvE* [27], *ConvKB* [29], *R-GCN* [30], *HypER* [43], *CapsE* [44], *A2N* [45], *KBGAT* [31], *ConvR* [46], InteractE [28], *NodePiece* [47], and *RMCNN* [48], leverage neural network architectures to learn expressive embedding and exploit the structural information of KGs. As shown in Table I, the score functions, embedding, and time complexity of the introduced methods have been exhibited.

3) Evaluation Metrics: Mean reciprocal rank (MRR) is a commonly used evaluation metric in information retrieval and ranking tasks. It is commonly used in information retrieval and recommendation systems. The formula for calculating the MRR is

$$\mathrm{MRR} = \frac{1}{Q} \sum_{i=1}^{|Q|} \frac{1}{\mathrm{rank}_i} \tag{23}$$

TABLE III
EXPERIMENTAL RESULTS ON FB15K DATASET

Categories	Methods	FB15k			
		MRR	Hits@1	Hits@3	Hits@10
DBMs	TransE	0.463	0.297	0.578	0.749
	TransH	0.556	0.436	0.638	0.758
	TransR	0.346	0.218	0.404	0.582
	TorusE	0.746	0.685	-	0.839
	RotatE	0.699	0.585	0.788	0.872
	HakE	0.714	0.639	0.768	0.834
	QuatE	0.770	0.700	0.821	0.878
	CrossE	0.728	0.634	0.802	0.875
SMBMs	RESCAL	0.354	0.235	0.409	0.587
	DistMult	0.379	0.306	0.423	0.515
	ComplEx	0.692	0.599	0.759	0.840
	HolE	0.542	0.402	0.613	0.739
	ANALOGY	0.725	0.646	0.784	0.837
	SimplE	0.727	0.600	0.773	0.838
NNBMs	R-GCN	0.651	0.541	0.736	0.825
	ConvE	0.745	0.670	0.801	0.873
	ConvKB	0.211	0.114	-	0.408
	NodePiece	0.148	0.078	0.158	0.288
	HypER	0.767	0.701	0.817	0.880
	CapsE	0.087	0.019	-	0.217
	ConvR	0.782	0.720	0.826	0.887
	ERGRW (ours)	**0.792**	**0.742**	**0.838**	**0.890**

Note: The best results are indicated in bold.

where Q represents the overall count of test cases and rank_i signifies the position of the correct answer within the ranking for the ith test case.

Hits@k is another evaluation metric used in KG completion tasks. It measures the proportion of test cases where the right answer is included among the top-k predictions. The formula for calculating Hits@k is

$$\text{Hits@}k = \frac{1}{|N|} \sum_{i \in N} I_k\left(\text{rank}_i\right) \tag{24}$$

where N represents the total number of test cases and I_k is an indicator function that yields a value of 1 if the rank of the right answer is less than or equal to k, and 0 otherwise.

4) Parameter Setting: In our implementation, the parameters of every baseline are set to be their optimal values for fair comparisons. For our proposed ERGRW, we set the embedding dimensions d to be 200. The hidden size and dropout rate of BiLSTM are set as 128 and 0.3, respectively. We set the batch size as 64 on FB15k, FB15k-237, and WN18RR, 128 on YAGO3-10. The Adam optimizer is adopted with a learning rate of 0.001. The number of feature permutations is set as 3, and most parameters of the decoder are adopted from the InteractE [28] model. For walk length l, we set $l = 6$ on FB15k, FB15k-237, and YAGO3-10, and $l = 4$ on WN18RR. Regarding walk number η, we set $\eta = 5$ on FB15k and WN18RR, and $\eta = 6$ on FB15k-237 and YAGO3-10. As for ratio α, it is set to 0.5 on FB15k and YAGO3-10, and 0.6 on FB15k-237 and YAGO3-10. A detailed analysis of the parameters l, η, and α will be reported in Section IV-D.

TABLE IV
EXPERIMENTAL RESULTS ON FB15K-237 DATASET

Categories	Methods	FB15k-237			
		MRR	Hits@1	Hits@3	Hits@10
DBMs	TransE	0.223	0.147	0.263	0.398
	TransH	0.302	0.203	0.330	0.483
	TransR	0.352	0.261	0.389	0.537
	TorusE	0.281	0.196	-	0.447
	RotatE	0.338	0.205	0.328	0.480
	MuRE	0.336	0.245	0.370	0.521
	MuRS	0.338	0.249	0.373	0.525
	HakE	0.346	0.250	0.381	0.542
	QuatE	0.311	0.221	0.342	0.495
	CrossE	0.299	0.212	0.331	0.474
	MuRMP	0.345	0.258	0.385	0.542
SMBMs	RESCAL	0.356	0.263	0.393	0.541
	DistMult	0.241	0.155	0.263	0.419
	ComplEx	0.247	0.158	0.275	0.428
	HolE	0.222	0.133	0.253	0.391
	ANALOGY	0.202	0.126	-	0.353
	SimplE	0.162	0.090	0.170	0.317
NNBMs	R-GCN	0.249	0.151	0.264	0.417
	ConvE	0.325	0.237	0.356	0.501
	ConvKB	0.243	0.155	0.371	0.421
	NodePiece	0.256	-	-	0.420
	HypER	0.341	0.252	0.376	0.520
	CapsE	0.160	0.073	-	0.356
	A2N	0.317	0.232	0.348	0.486
	KBGAT	0.205	0.114	0.228	0.396
	ConvR	0.350	0.261	0.385	0.528
	InteractE	0.353	0.260	0.390	0.541
	RMCNN	0.358	0.255	0.394	0.535
	ERGRW (ours)	**0.362**	**0.268**	**0.396**	**0.549**

Note: The best results are indicated in bold.

B. Comparison Results

The results of the comparison, presented in Tables III, IV, V, and VI, showcase the performance of various baseline methods across different categories in terms of MRR and Hits@k (where $k = 1, 3, 10$) on the four datasets. The baseline methods are categorized into three types: DBMs, SMDMs, and NNBMs. Among the three categories of methods, all of them exhibit high performance, with NNBMs generally outperforming SMBMs and DBMs, and the NNBMs predominantly demonstrate the best performance across various evaluation metrics within the baseline methods.

The results presented in the tables indicate that ERGRW demonstrates superior performance in terms of MRR, Hits@1, Hits@3, and Hits@10 on the FB15k, FB15k-237, and YAGO3-10 datasets. However, it can be observed that ERGRW's performance is marginally below that of QuatE on WN18RR, specifically in terms of Hits@3. The results show that ERGRW exhibits the largest performance improvement on the FB15k dataset, surpassing the best baseline method ConvR. Specifically, compared to ConvR, ERGRW shows an improvement of 1.25% in MRR, 3.05% in Hits@1, 1.45% in Hits@3, and 0.3% in Hits@10 on the FB15k dataset. Due to the substantial scale and intricate complexity of the FB15k dataset, ERGRW is able to effectively leverage the graph structure and exploit the interactions between entities and relations, thereby capturing complex dependencies and latent associations in the KG.

TABLE V
EXPERIMENTAL RESULTS ON WN18RR DATASET

Categories	Methods	WN18RR			
		MRR	Hits@1	Hits@3	Hits@10
DBMs	TransE	0.226	0.021	0.374	0.501
	TransH	0.257	0.073	0.423	0.515
	TransR	0.412	0.398	0.425	0.451
	TorusE	0.463	0.427	-	0.534
	MuRS	0.454	0.432	0.482	0.550
	MuRE	0.475	0.436	0.487	0.554
	RotatE	0.451	0.405	0.470	0.546
	QuatE	0.481	0.436	**0.500**	0.564
	CrossE	0.405	0.381	-	0.450
	MuRMP	0.473	0.435	0.485	0.552
SMBMs	RESCAL	0.467	0.439	0.480	0.517
	DistMult	0.430	0.390	0.440	0.490
	ComplEx	0.440	0.410	0.460	0.510
	HolE	0.432	0.402	-	0.487
	ANALOGY	0.366	0.358	-	0.380
	SimplE	0.398	0.382	-	0.427
NNBMs	R-GCN	0.123	0.137	0.180	0.207
	ConvE	0.430	0.400	0.440	0.520
	ConvKB	0.249	0.057	0.417	0.524
	HypER	0.465	0.436	0.477	0.522
	CapsE	0.415	0.337	-	0.560
	A2N	0.430	0.410	0.440	0.510
	ConvR	0.475	0.433	0.489	0.537
	KBGAT	0.404	0.322	0.448	0.554
	InteractE	0.467	0.435	0.482	0.529
	NodePiece	0.403	-	-	0.515
	RMCNN	0.473	0.440	0.479	0.540
	ERGRW (ours)	**0.483**	**0.441**	0.495	**0.567**

Note: The best results are indicated in bold.

TABLE VI
EXPERIMENTAL RESULTS ON YAGO3-10 DATASET

Categories	Methods	YAGO3-10			
		MRR	Hits@1	Hits@3	Hits@10
DBMs	TransE	0.501	0.392	0.405	0.673
	TransR	0.256	0.223	0.356	0.478
	TorusE	0.342	0.274	-	0.474
	MuRS	0.351	0.244	0.382	0.562
	MuRE	0.532	0.444	0.584	0.694
	RotatE	0.495	0.402	0.550	0.670
	QuatE	0.564	0.260	0.075	0.385
	CrossE	0.446	0.331	-	0.655
	MuRMP	0.358	0.248	0.389	0.566
SMBMs	DistMult	0.340	0.240	0.380	0.540
	ComplEx	0.360	0.260	0.400	0.550
	HolE	0.502	0.418	-	0.651
	ANALOGY	0.283	0.192	-	0.457
	SimplE	0.453	0.358	-	0.632
NNBMs	ConvE	0.440	0.350	0.490	0.620
	ConvKB	0.420	0.322	-	0.605
	HypER	0.533	0.455	0.580	0.678
	A2N	0.445	0.349	0.482	0.501
	ConvR	0.527	0.446	-	0.673
	KBGAT	0.166	0.097	0.183	0.302
	InteractE	0.549	0.472	0.595	0.685
	NodePiece	0.247	-	-	0.488
	RMCNN	0.557	0.483	0.607	0.698
	ERGRW (ours)	**0.561**	**0.492**	**0.610**	**0.701**

Note: The best results are indicated in bold.

TABLE VII
PERFORMANCE COMPARISON OF ABLATION STUDY

Datasets	Metrics	$Rule1$ + R	$Rule2$ only	ERGRW
FB15K	MRR	0.633	0.455	**0.792**
	Hits@1	0.598	0.368	**0.742**
	Hits@3	0.706	0.576	**0.838**
	Hits@10	0.799	0.679	**0.890**
FB15K-237	MRR	0.212	0.162	**0.362**
	Hits@1	0.162	0.098	**0.268**
	Hits@3	0.233	0.186	**0.396**
	Hits@10	0.402	0.312	**0.549**
WN18RR	MRR	0.356	0.133	**0.483**
	Hits@1	0.324	0.154	**0.441**
	Hits@3	0.401	0.236	**0.495**
	Hits@10	0.467	0.375	**0.567**
YAGO3-10	MRR	0.361	0.333	**0.561**
	Hits@1	0.325	0.334	**0.492**
	Hits@3	0.442	0.427	**0.610**
	Hits@10	0.583	0.555	**0.701**

Note: The best results are indicated in bold.

This observation suggests that ERGRW is adept at capturing complex dependencies and latent associations within the KG, leading to improved performance in link prediction tasks. Meanwhile, ERGRW also achieves significant performance improvements on the FB15k-237 dataset, which has a simplified graph structure as it is a subset of FB15k. In particular, compared to the best results across various metrics, ERGRW demonstrates improvements of 1.11% in MRR, 2.68% in Hits@1, 0.5% in Hits@3, and 1.29% in Hits@10 on the FB15k-237 dataset. This indicates that the benefits of ERGRW extend to datasets with varying graph structures, highlighting its versatility and effectiveness in diverse settings. Additionally, the large-scale YAGO3-10 dataset provides ERGRW with increased opportunities to discover and leverage the underlying structure and dependencies in the knowledge graph. As a result, ERGRW showcases modest improvements across various metrics on this dataset as well. This highlights ERGRW's ability to enhance link prediction performance on complex and extensive KGs. The results demonstrate that ERGRW's capability to explore semantic relationships between entities, capture contextually rich information, and discover latent associations within the KG further contributes to its effectiveness in link prediction tasks. However, on the WN18RR dataset, ERGRW falls short of achieving the optimal performance in terms of Hits@3, and the improvements in other metrics are relatively modest as well. This may be related to the WN18RR dataset's small number of relations, which limits the presence of complex relationship patterns, and ERGRW faces limitations in effectively leveraging entities and relations to capture the underlying structure and potential long-range dependencies within the dataset. Compared to the InteractE model, ERGRW exhibits improvements across all metrics as shown in Tables IV, V, and VI. It suggests that the random walk based on $Rule1$ and $Rule2$ in the encoder enables the collection of more comprehensive sequences, which are subsequently utilized for richer representations in the subsequent BiLSTM. Due to the sparsity of connection between entities

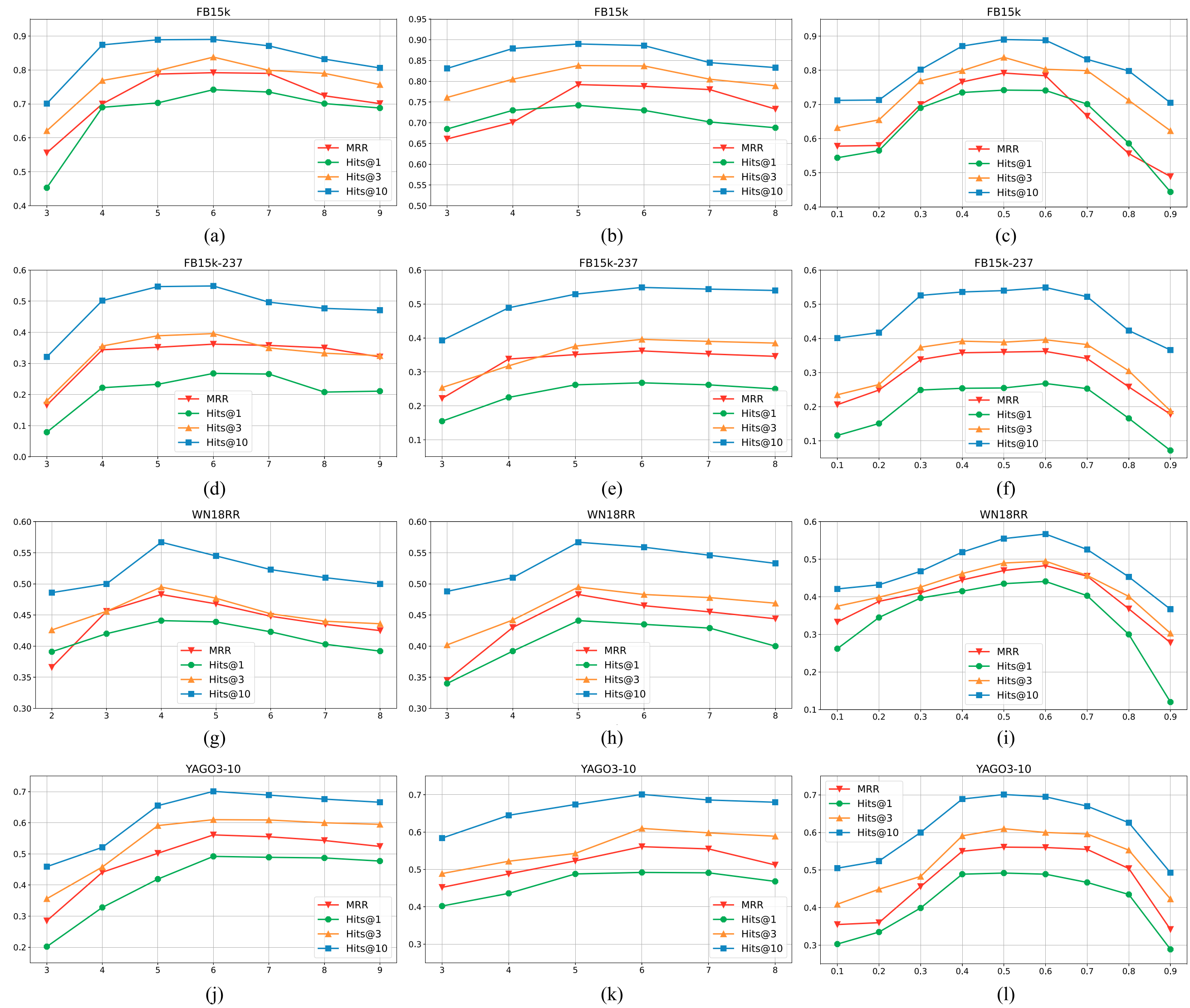

Fig. 5. Results of MRR and Hits@k (k = 1, 3, 10) when changing walk length (l), walk number (η), and ratio (α) on FB15k, FB15k-237, WN18RR, and YAGO3-10 datasets, respectively, where walk length (l) denotes the length of the sequences obtained through random walks, walk number (η) represent the number of random walks conducted (i.e., the sequences number), and ratio (α) indicates the percentage of sequences following $Rule1$ (i.e., $1-\alpha$ represents the percentage of sequences following $Rule2$).

in KGs, it means that only a small fraction of entities have direct associations. Therefore, random walk traverse the graph according to certain rules, collecting sequences information that cover different parts of the graph. This process helps explore potential associations and dependencies, mitigating the issue of information loss caused by the sparsity of the graph.

Based on the above analysis, the results clearly demonstrate that ERGRW enables the exploration of semantic relationships between entities, allowing it to capture contextually rich information. Through this capability, ERGRW is able to discover latent associations and semantic relationships within the KG, thereby enhancing its understanding of entity interactions. Furthermore, ERGRW effectively captures the underlying structure and dependencies within the KG. This ability enhances its capacity to detect meaningful associations between entities, leading to improved performance in link prediction tasks.

C. Ablation Study

In order to evaluate the impact of $Rule1$ and $Rule2$ toward enhancing the performance of our proposed ERGRW, ablation experiments are conducted, including $Rule1 + R$ and $Rule2$. In the scenario of $Rule1 + R$, we exclusively employed $Rule1$ (when $\alpha = 1$), where the sequences are composed of entities obtained under $Rule1$. Since the ERGRW model takes both entities and relations as inputs, we incorporated the relations between entities in the sequence as part of the input. In the scenario of $Rule2$, we solely utilized $Rule2$ (when $\alpha = 0$), where the sequences are composed of entities and relations under $Rule2$.

As shown in Table VII, our proposed ERGRW model (combining $Rule1 + R$ and $Rule2$) achieved the best performance, and $Rule1 + R$ outperforms $Rule2$. When only $Rule2$ is used, the sequences are obtained through random walks under $Rule2$. Although these sequences also include entities and relations, and capture some long-range dependencies and global structural information between entities, they fail to provide comprehensive representations of entities' adjacency information and the underlying graph structure. On the other hand, when only $Rule1$ is used and the relationships between entities in the sequences are incorporated as part of the input, it yields relatively favorable results. This scenario effectively incorporates the adjacency information of entities and the structure of the graph but struggles to capture long-range dependencies and the underlying structure from the graph, limiting its ability to explore potential associations between entities and relations.

In contrast, ERGRW successfully combines the advantages of $Rule1 + R$ and $Rule2$ by integrating the strengths of both rules. By jointly leveraging entities and relations to guide the random walks, ERGRW generates sequences with rich semantic information. Consequently, ERGRW captures both local information and global information from the KG, enabling the discovery of long-range dependency relationships among entities and the exploration of latent associations between entities. The integration of $Rule1 + R$ and $Rule2$ allows ERGRW to achieve a more comprehensive representation of entities and relations, resulting in improved link prediction performance.

D. Parameter Analysis

To investigate the impact of parameters on ERGRW's performance, we conduct experiments to analyze the parameter sensitivity of walk length l, walk number η, and ratio α on four datasets, respectively. The results of the parameter analysis are shown in Fig. 5.

In terms of the walk length l, the FB15k, FB15k-237, and YAGO3-10 datasets performed best at lengths of 6, while the WN8RR dataset achieved its best performance at a length of 4. In terms of the walk number η, the FB15k and WN18RR datasets achieved their best results with a value of 5, while the FB15k-237 and YAGO3-10 datasets achieved optimal performance with a number of 6. The results indicate a similar trend in the effect of walk length and walk number on the performance of ERGRW. As the length or number increases from a smaller value, the performance gradually improves. However, after reaching a certain threshold, the model's performance starts to slightly decline. This observation suggests that when the length or number is too small, the resulting sequence information might not adequately capture the structural characteristics of the graph. Conversely, when the length or number becomes too large, it introduces noise that leads to poorer representations of the graph. Regarding the ratio parameter α, both FB15k and YAGO3-10 datasets achieved the best performance at a ratio of 0.5, while WN8RR and FB15k-237 datasets achieved optimal results at a ratio of 0.6. The results indicate that if the ratio is too small, approaching 0, ERGRW is effectively reduced to using only $Rule2$. Without the graph structure information provided by $Rule1$, the model fails to capture sufficient local information, resulting in poor performance. On the other hand, with an increasing ratio, ERGRW tends to rely more on $Rule1$, leading to insufficient relationship information in the generated sequences and consequently resulting in inferior performance.

V. Conclusion

In this article, we propose ERGRW, a KGE method that utilizes ERGRW and an encoder–decoder framework. By creatively incorporating relations as part of the random walk, ERGRW effectively explores potential correlations and dependencies between entities, improving its capacity to obtain both local and global neighborhood information. Moreover, the model designs an encoder–decoder framework for link prediction tasks, allowing ERGRW to learn meaningful representations by capturing the interactions among triples. As a result, ERGRW is able to gain a more comprehensive understanding of the underlying semantic structure of the KG, leading to improved link prediction performance. Experimental results on four standard datasets verify the improvements of ERGRW compared with baseline methods. In our future work, further research will be conducted to explore potential enhancements to ERGRW. One avenue for improvement could be integrating attention mechanisms to more efficiently capture the significance of different entities and relations by random walk. Additionally, integrating external semantic information or domain-specific knowledge into the ERGRW framework may further improve its performance.

References

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

[2] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[3] F. M. Suchanek, G. Kasneci, and G. Weikum, "YAGO: A core of semantic knowledge," in *Proc. 16th Int. Conf. World Wide Web (WWW)*, 2007, pp. 697–706.

[4] C. Wang, X. Zhou, S. Pan, L. Dong, Z. Song, and Y. Sha, "Exploring relational semantics for inductive knowledge graph completion," in *Proc. 36th AAAI Conf. Artif. Intell. (AAAI)*, 2022, pp. 4184–4192.

[5] J. Ju, D. Yang, and J. Liu, "Commonsense knowledge base completion with relational graph attention network and pre-trained language model," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2022, pp. 4104–4108.

[6] H. Chang, J. Cai, and J. Li, "Knowledge graph completion with counterfactual augmentation," in *Proc. ACM Web Conf. (WWW)*, 2023, pp. 2611–2620.

[7] P. Li, G. Ren, A. L. Gentile, C. DeLuca, D. Tan, and S. Gopisetty, "Long-form information retrieval for enterprise matchmaking," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval (SIGIR)*, 2023, pp. 3260–3264.

[8] M. Aldwairi, M. H. A. Jarrah, N. Mahasneh, and B. Al-khateeb, "Graph-based data management system for efficient information storage, retrieval and processing," *Inf. Process. Manage.*, vol. 60, no. 2, 2023, Art. no. 103165.

[9] G. Goel, R. P. Leme, J. Schneider, D. Thompson, and H. Zhang, "Eligibility mechanisms: Auctions meet information retrieval," in *Proc. ACM Web Conf. (WWW)*, 2023, pp. 3541–3549.

[10] H. Liu et al., "Lion: A GPU-accelerated online serving system for web-scale recommendation at Baidu," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining (KDD)*, 2022, pp. 3388–3397.

[11] P. Wu et al., "On the opportunity of causal learning in recommendation systems: Foundation, estimation, prediction and challenges," in *Proc. 31st Int. Joint Conf. Artif. Intell. (IJCAI)*, 2022, pp. 5646–5653.

[12] H. Tang, G. Zhao, Y. He, Y. Wu, and X. Qian, "Ranking-based contrastive loss for recommendation systems," *Knowl.-Based Syst.*, vol. 261, 2023, Art. no. 110180.

[13] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst. 26 (NeurIPS)*, 2013, pp. 2787–2795.
[14] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. 28th AAAI Conf. Artif. Intell. (AAAI)*, 2014, pp. 1112–1119.
[15] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. 29th AAAI Conf. Artif. Intell. (AAAI)*, 2015, pp. 2181–2187.
[16] K. Pearson, "The problem of the random walk," *Nature*, vol. 72, no. 1865, pp. 294–294, 1905.
[17] X. Huang, Q. Song, Y. Li, and X. Hu, "Graph recurrent networks with attributed random walks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2019, pp. 732–740.
[18] Y. Liu, S. Yang, Y. Xu, C. Miao, M. Wu, and J. Zhang, "Contextualized graph attention network for recommendation with item knowledge graph," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 181–195, Jan. 2023.
[19] R. Wang et al., "Common neighbors matter: Fast random walk sampling with common neighbor awareness," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4570–4584, May 2023.
[20] 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 (ICLR)*, 2015.
[21] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proc. 33nd Int. Conf. Mach. Learn. (ICML)*, vol. 48, 2016, pp. 2071–2080.
[22] M. Nickel, L. Rosasco, and T. A. Poggio, "Holographic embeddings of knowledge graphs," in *Proc. 30th AAAI Conf. Artif. Intell. (AAAI)*, 2016, pp. 1955–1961.
[23] M. Nickel, V. Tresp, and H. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 809–816.
[24] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Proc. Adv. Neural Inf. Process. Syst. 26 (NeurIPS)*, 2013, pp. 926–934.
[25] X. Dong et al., "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 601–610.
[26] Q. Liu, H. Jiang, Z. Ling, S. Wei, and Y. Hu, "Probabilistic reasoning via deep learning: Neural association models," 2016, *arXiv:1603.07704*.
[27] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 1811–1818.
[28] S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal, and P. P. Talukdar, "InteractE: Improving convolution-based knowledge graph embeddings by increasing feature interactions," in *Proc. 34th AAAI Conf. Artif. Intell. (AAAI)*, 2020, pp. 3009–3016.
[29] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Q. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL-HLT)*, 2018, pp. 327–333.
[30] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proc. 15th Int. Conf. Semantic Web (ESWC)*, vol. 10843, 2018, pp. 593–607.
[31] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning attention-based embeddings for relation prediction in knowledge graphs," in *Proc. 57th Conf. Assoc. Comput. Linguistics (ACL)*, 2019, pp. 4710–4723.
[32] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Representations (ICLR)*, 2018.
[33] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics/7th Int. Joint Conf. Natural Lang. Process. Asian Federation Natural Lang. Process. (ACL)*, 2015, pp. 687–696.
[34] D. Q. Nguyen, K. Sirts, L. Qu, and M. Johnson, "STransE: A novel embedding model of entities and relationships in knowledge bases," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL-HLT)*, 2016, pp. 460–466.
[35] T. Ebisu and R. Ichise, "TorusE: Knowledge graph embedding on a lie group," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 1819–1826.
[36] I. Balazevic, C. Allen, and T. M. Hospedales, "Multi-relational poincaré graph embeddings," in *Proc. Adv. Neural Inf. Process. Syst. 32 (NeurIPS)*, 2019, pp. 4465–4475.
[37] S. Wang et al., "Mixed-curvature multi-relational graph neural network for knowledge graph completion," in *Proc. ACM Web Conf. (WWW)*, 2021, pp. 1761–1771.
[38] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, and H. Chen, "Interaction embeddings for prediction and explanation in knowledge graphs," in *Proc. 12th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2019, pp. 96–104.
[39] 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 (ICLR)*, 2019.
[40] S. M. Kazemi and D. Poole, "Simple embedding for link prediction in knowledge graphs," in *Proc. Adv. Neural Inf. Process. Syst. 31 (NeurIPS)*, 2018, pp. 4289–4300.
[41] H. Liu, Y. Wu, and Y. Yang, "Analogical inference for multi-relational embeddings," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 70, 2017, pp. 2168–2178.
[42] I. Balazevic, C. Allen, and T. M. Hospedales, "TuckER: Tensor factorization for knowledge graph completion," in *Proc. Conf. Empirical Methods Natural Lang. Process./9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 5184–5193.
[43] I. Balažević, C. Allen, and T. M. Hospedales, "Hypernetwork knowledge graph embeddings," in *Proc. Artif. Neural Netw. Mach. Learn. (ICANN)*, vol. 11731, New York, NY, USA: Springer-Verlag, 2019, pp. 553–565.
[44] D. Q. Nguyen, T. Vu, T. D. Nguyen, D. Q. Nguyen, and D. Q. Phung, "A capsule network-based embedding model for knowledge graph completion and search personalization," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL-HLT)*, 2019, pp. 2180–2189.
[45] T. Bansal, D. Juan, S. Ravi, and A. McCallum, "A2N: Attending to neighbors for knowledge graph inference," in *Proc. 57th Conf. Assoc. Comput. Linguistics (ACL)*, 2019, pp. 4387–4392.
[46] X. Jiang, Q. Wang, and B. Wang, "Adaptive convolution for multi-relational learning," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL-HLT)*, 2019, pp. 978–987.
[47] M. Galkin, E. G. Denis, J. Wu, and W. L. Hamilton, "NodePiece: Compositional and parameter-efficient representations of large knowledge graphs," in *Proc. 10th Int. Conf. Learn. Representations (ICLR)*, 2022.
[48] M. Shi et al., "Convolutional neural network knowledge graph link prediction model based on relational memory," *Comput. Intell. Neurosci.*, vol. 2023, pp. 3909697–3909699, 2023.
[49] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst. 26 (NeurIPS)*, 2013, pp. 2787–2795.
[50] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *Proc. 3rd Workshop Continuous Vector Space Models Compositionality (CVSC)*, 2015, pp. 57–66.
[51] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 1811–1818.
[52] F. Mahdisoltani, J. Biega, and F. M. Suchanek, "YAGO3: A knowledge base from multilingual Wikipedias," in *Proc. 7th Biennial Conf. Innovative Data Syst. Res. (CIDR)*, 2015.
[53] Z. Zhang, J. Cai, Y. Zhang, and J. Wang, "Learning hierarchy-aware knowledge graph embeddings for link prediction," in *Proc. 34th AAAI Conf. Artif. Intell./32nd Innovative Appl. Artif. Intell. Conf. (AAAI)*, 2020, pp. 3065–3072.

Weisheng Li received the B.S. degree in Internet of Things engineering from Zhongkai University of Agriculture and Engineering, Guangzhou, China, in 2019. He is currently working toward his Ph.D. degree in software engineering with South China Normal University (SCNU), Guangzhou, China.

His research interests include knowledge graph embedding, recommender systems, and link prediction.

Hao Zhong received the Ph.D. degree in software engineering from South China Normal University, Guangzhou, China, in 2022.

He is currently a post-doctoral researcher in South China Normal University, Guangzhou, China. His research interests include combinatorial optimization, algorithm, and social network.

Ronghua Lin received the Ph.D. degree in software engineering from South China Normal University (SCNU), Guangzhou, China, in 2023.

He is currently a Postdoctoral Researcher at SCNU. His research interests include recommender systems, system optimization based on machine learning, and social networks.

Chao Chang received the Ph.D. degree in software engineering from South China Normal University (SCNU), Guangzhou, China, in 2023.

She is currently a Lecturer at the School of Information Engineering, Guangzhou Panyu Polytechnic, Guangzhou, China. Her research interests include KGs, social network services, and big data analysis for education.

Zhihong Pan received the B.S. degree in electronic information engineering from Dongguan University of Technology, Dongguan, China, in 2008, and the M.Eng. degree in communication and information system from Jinan University, Guangzhou, China, in 2011. He is currently working toward the Ph.D. degree in software engineering with the School of Artificial Intelligence, South China Normal University (SCNU), Foshan, China.

He is also an Associate Professor at the School of Information and Intelligent Engineering, Guangzhou Xinhua University, Guangzhou, China. His main research interests include deep learning and graph data mining.

Yong Tang received the B.S. and M.Sc. degrees from Wuhan University in 1985 and 1990, respectively, and Ph.D. degree, all in computer science, from University of Science and Technology of China in 2001.

He is the founder of SCHOLAT, a kind of scholar social network. He is currently a Professor with the School of Computer Science at South China Normal University (SCNU), Guangzhou, China. He was the Dean of the School of Computer Science at SCNU and the Vice Dean of the School of Information of Science and Technology at Sun Yat-Sen University. He has published more than 400 papers and books. He has supervised more than 60 Ph.D. students and post-doctors since 2003 and more than 200 Master's students since 1996. His research interests include data and knowledge engineering, social networking, and collaborative computing. He currently serves as the director of the technical committee on collaborative computing of the China Computer Federation (CCF) and the executive vice president of Guangdong Computer Academy.