A Novel Integrating Approach Between Graph Neural Network and Complex Representation for Link Prediction in Knowledge Graph

Thanh Le[1,2](✉), Loc Tran[1,2], and Bac Le[1,2]

[1] Faculty of Information Technology, University of Science,
Ho Chi Minh City, Vietnam
{lnthanh,lhbac}@fit.hcmus.edu.vn, txloc18@clc.fitus.edu.vn
[2] Vietnam National University, Ho Chi Minh City, Vietnam

Abstract. Deep learning brings high results in many problems, including Link Prediction on Knowledge Graphs (KGs). Although there are many techniques to implement deep learning into KGs, Graph Neural Networks (GNNs) have recently emerged as a promising direction for representing the structure of KGs as input for a decoder. With this structural information, GNNs can help to retain more information from the original graph than conventional embeddings like TransE, TransH, RESCAL. As a result, the learning model achieves higher accuracy in predicting missing links between entities in the KG. Meanwhile, several studies have successfully demonstrated the intrinsic properties of the embedding process in complex space while keeping many binary relations (symmetric and asymmetric). Thus, this paper proposes deploying GNNs into complex space to increase the model's predictive capability. Another issue with GNNs is that they are susceptible to over-squashing when a large amount of information propagating between nodes is compressed down to a fixed representation space. As a result, we utilize a dynamic attention mechanism to minimize the adverse effects of these factors, and experiments on benchmark datasets have indicated that our proposal achieves a significant improvement compared to baseline models on almost all standard metrics.

Keywords: Knowledge graph embedding · Link prediction · Graph convolutional networks · Dynamic graph attention networks

1 Introduction

Knowledge Graphs (KGs) are becoming a widely used term in the field of artificial intelligence. Some of its outstanding applications are in a number of areas such as medicine [1] and e-commerce [11]. Among the KG-related challenges, we are putting our efforts into tackling the link prediction (LP) problem, whose objective is discovering the missed links in KG to accomplish itself. In fact, data

E. Szczerbicki et al. (Eds.): ACIIDS 2022, CCIS 1716, pp. 263–275, 2022.
https://doi.org/10.1007/978-981-19-8234-7_21

in the KG is regularly gathered from various sources, including manually. Moreover, identifying the relations between data aids in order to complete KGs. Furthermore, we could forecast potential relations between entities in the future. Although numerous rule-based and probability-based methods for addressing this problem were initially proposed, these models suffer the exploding computational cost when performing on large KGs. As a result, the embedding approaches emerged and gained widespread attention in recent years because of their above mentioned characteristics.

Currently, KGE is classified into two categories: Translational and Semantic matching models. At the Translational-based models, the models projecting entities and relations into vector or matrix representation with simplicity, scalable characteristics, TransE [3] and its extensions such as TransH [19], TransR [12], and TransD [8] are the top striking models of translational research. The semantic-matching models include bilinear models and neural network-based models. Bilinear methods such as the RESCAL model [14], DistMult [21], and ComplEx [17] can mine the intrinsic properties of KGs by using tensor decomposition, characterized by the less time-consuming and effective computation except for RESCAL. In neural network-based models, researchers apply the success of Convolutional neural networks (CNNs) in KGs and archive potential results such as ConvE [7], ConvKB [13], but it suffers to time-consuming, increases model complexity and other related CNN's problems. Another variant of CNNs is Graph Convolutional Networks [9], levering the same convolutional operation but performing on graph data while considering graph features during the embedding phase. Some recent models such as RGCN [15], GATv2 [4], VR-GCN [22], and TransGCN [5] - an associated model between GCN and Translational model, on the graph neural networks branches. Furthermore, incorporating additional information into the embedding process, such as literals, textual descriptions, entity, and relation types, has been shown to yield higher quality embedding and is especially effective in many downstream tasks such as triple classification, entity classification, and link prediction [20].

Although the Translational models are outstanding by its qualities, it has flaws when modeling the complex relations and TransE's extensions ignore two important characteristics including heterogeneity and imbalance [6]. In the semantic-matching models, neural networks-based models significantly contribute to and improve the performance of link prediction tasks, but the connectivity information between entities and neighbors is neglected. To solve this problem, GCNs emerged as the prisoner of applying graph structure information to its embedding space to perform more semantic calculations. Specifically, they create entity embeddings that aggregate local information in neighborhoods. However, GCNs mainly solve problems in real space, which is the main weakness leading to marginal performance growth in recent years. Thus, we propose implementing GCNs in complex space and conducting experiments to demonstrate the effectiveness of this approach in this paper. Moreover, the ComplEx model is employed as a decoder to utilize the graph features in the complex space because of its simplicity and scalability over large graphs. Specifically, ComplEx has low complexity while still representing interactions and entities in complex spaces because it uses only the Hermitian dot product. Moreover, the reason we integrate the GCNs and GATv2 [4] to tackle the challenge of GNNs, which is detail reported in Sect. 3.

In conclusion, our contributions are summarized:

- Proposing an end-to-end framework ComplEx-GNN for learning entity and relation embedding characteristics in complex space.
- Constructing an association between GCN and Dynamic Graph Attention (GATv2) layer as an encoder to mapping entities, exploiting graph information and alleviating the bottleneck problem.
- Leveraging ComplEx as a decoder to simulate the interacting between entities and relations in complex space.

The remaining paper includes five sections. In Sect. 2, we discuss some related works and the typical models in each branch of KGE. We focus on presenting the main idea of the ComplEx-GNN framework in detail in Sect. 3. Section 4 contains experimental results in detail on FB15K-237 and WN18RR. Section 5 summarises our contributions and recommends future research directions.

2 Related Work

Models for KGE problems can be categorized into two types: translational models and semantic-matching models. Translational-based models were regarded as the pioneers of KGC by their simplicity, clarity, and high interpretability. The scoring function and representation space distinguish these approaches. Specifically, they utilize relations as translation operators in translation-based models. In the embedded space, the head entity **h** is expected to approximate the tail entity **t** via the translation **r**, then the scoring function is computed based on the deviation of the predicted entities from the actual entities in the triple.

Semantic-based models include bilinear and neural network-based models, which leverage the similarity measures as the scoring functions for the triples. Bilinear models are based on matrix decomposition by using tensors to represent relational data $Y \in \{0,1\}^{N \times R \times R}$ where N is the number of entities, R is the number of relations, Y is state of the connection between entities (yes or no). Moreover, these approaches are distinguished based on structural conditions on tensors with the capability of modeling binary relations. Furthermore, Bilinear models consider link prediction tasks as tensor decomposition problems. Meanwhile, neural networks-based models consist of many different layers to mine the fundamental relations between entities and relations with self-feedback and fast convergence. In addition, current models also integrate some additional information such as context information, entity type, and path information to be exploited deep semantic information.

2.1 Translational Based Models

TransE model [3] was developed based on the idea of the energy-based model, which is the cornerstone for later translational-based approaches because of its efficient, less parameter, and scalable characteristics. TransE is considered the backbone of the embedding space. Moreover, Bordes et al. also mentioned that KGs have many hierarchical relations, and the translation is a natural transformation. Thus, they represent each relation as a translation in the embedding space such that the tail entity is close to the head entity in the embedding space ($\mathbf{h} + \mathbf{r} \approx \mathbf{t}$). Each entity or relation is represented by a single embedding vector. However, the model is only suitable for $1 - 1$ relations. And it has flaws when representing $1 - n$, $n - 1$, and $n - n$ relations. To address these issues, TransH [19], motivated by TransE, interprets the interaction by shifting operations into hyperplanes and introducing a relation-specific mechanism. In this way, it allows entities to take different roles in each relation type. Each relation, as a hyperplane, is determined by two vectors including the norm vector ($\mathbf{w_r}$) and the translation vector ($\mathbf{d_r}$). Although the latter comparative researches such as TransR [12], TransD [8] conducted in that direction gained considerable improvements compared to TransE, the complexity also increased quite a bit.

2.2 Semantic-Matching Models

Bilinear models such RESCAL [14], DistMult [21], and ComplEx [17] employ similarity measures as scoring functions to assess the plausibility of triplets. RESCAL also applies tensor decomposition to obtain the implicit semantic by integrating a two-way interaction of relations and entities to measure the probability of forming triplets. DistMult represents entities and relations as diagonal matrices and modifies the scoring function to Eq. 1. Although it has less parameters and reduces memory parameters from $O(Ne_d + N_r k^2)$ $(d = k)$ to $O(Ne_d + N_r k)$ $(d = k)$ [19] compared to RESCAL, it only handles symmetric relations [18]. Meanwhile, ComplEx increases the ability to cope with asymmetric relations by utilizing the complex value to embed interactions and entities into complex space $\mathbb{C}^d$. We realize that by combining tensor decomposition with graph features, the ComplEx model could extract deeper semantic information from the embedding phase. Besides that, some investigations, such as ComplEx-Literal [10] and DistMult-Literal [10], have focused on exploiting additional information to enhance the overall performance of downstream tasks as link prediction. Despite the fact that the additional information is applied during the embedding phase, this information has not been fully exploited, and it does not bring significant improvement results.

$$f_r(\mathbf{h}, \mathbf{r}) = \mathbf{h}^T diag(\mathbf{r})\mathbf{t} \mid \mathbf{r} \in \mathbf{R}^d \tag{1}$$

where d is the dimensionality of embedding space.

Neural networks-based models such as ConvE [7], ConvKB [13] are the representative models on this branch. ConvE uses a 2D reshaping operator to define the scoring function while using the convolutional and fully connected layers to model the interactions between entities and relations. ConvKB is inspired by a similar idea as ConvE by leveraging the success of a convolutional operator in computer vision but achieves a better result than ConvE. In particular, this approach uses the concatenate operator instead of the reshape operator to define the scoring function while keeping the transitional features and observing the global relationship among entities. Another variant of using the convolutional operator but performing on unstructured data is Graph Convolutional Networks. GCNs are applied in KGs such as RGCN [15], VR-GCN [22] and TransGCN [5], which aggregate the local informations of target entities by propagating the information between its neighbors. RGCN models the relational data by introducing relations-specific transformations for each relation type. Furthermore, RGCN uses parameter sharing techniques including basis and block diagonal-decomposition in regularization to tackle the sparsity problem and memory requirements of GCN on large KGs. Although RGCN leverages the benefit of different relation types in entity representation, it does not consider the representations of relations that contain rich semantic information. To address this problem, VR-GCN takes into account relations and entities to generate both embedding, allowing the relations to involve the multi-relational networks while keeping the translational properties. TransGCN operates on the same idea as VR-GCN because it considers entities and relations embedded in a unified framework, but TransGCN is supposed that each relation as a transformation operator transforms the head entity to the tail entity. When compared to R-GCN, TransGCN can transform a heterogeneous neighborhood in a KG into a homogeneous neighborhood with fewer parameters and directly perform link prediction without the support of external encoders such as DistMult in RGCN [2].

3 The Proposed Model

In this section, we demonstrate our model ComplEx-GNN. The encoder integrates a GCN layer and GATv2. It aims to aggregate the necessary information from a neighbor as specified by relations, then reproduce it into features representation for each entity including the real and the imaginary components. The decoder ComplEx focuses on representing entities and relations embedding in complex space, then calculates the possibility of forming triplets by leveraging the scoring function of the ComplEx model [17].

Our proposed framework including the encoder and decoder demonstrated in Fig. 1. For the encoder, entities and relations are embedded into the complex space to create two main components including real component and imaginary component for each entity and relation. The Graph Aggregation Layer takes the entities embedding as input, which consists GCN and GATv2 layers. We utilize GCN layer as the first layer to propagate information between neighbors because of its scalable on large KGs and can operate on local graph neighbors. Next, we expand the interaction between entities and assess neighbors contributions by

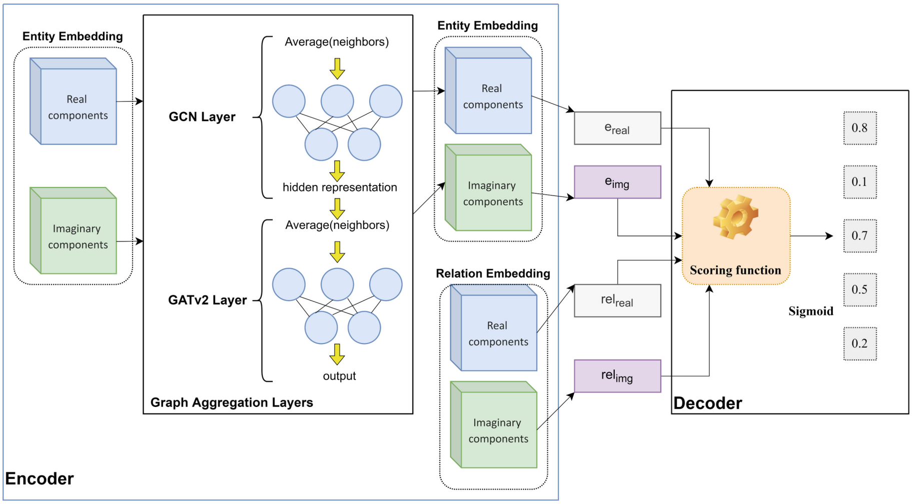

Fig. 1. An overview of our proposed model ComplEx-GNN

adding a Graph Attention Layer (GAL) to assign the different weight for the importance of neighbor's features in the entity's neighborhood. However, GAL could only compute the static attention layer for architecture including only one layer; if it exceeds ($n > 1$), the dynamic attention can not be computed, detail in [4]. Thus, we use its upgraded version - Dynamic Graph Attention Networks (GATv2) in our framework. GATv2 could also assign the weight for neighbor, while reducing the amount of information passed through an activation function, and alleviate the bottleneck problem of GNN because of exponential growing information into fixed-size vectors [2]. After obtaining the entities embedding from Graph Aggregation Layers, we take the entities and relations embedding matrices as input to the decoder. For the decoder, $\mathbf{e}_{real}$, $\mathbf{e}_{img}$, $\mathbf{rel}_{real}$, and $\mathbf{rel}_{img}$ are passed into the scoring function. We utilize the ComplEx scoring function as the scoring function, because it is the standard approach applying the complex representation without any additional information, while their representation space could describe the symmetric and asymmetric relations more accurately than other approaches. Next, a sigmoid function is applied to get the prediction results.

3.1 Graph Aggregation Layers

A knowledge graph $G(E, R)$, where $E = \{e_1, e_2, ..., e_M\}$ denotes the set of M entities, $R = \{r_1, r_2, ..., r_N\}$ denoting the set of N relations, $(i, j) \in R$ denotes a connection between two entities i and j , and (h, r, t) denoting a triplet with h representing the head entity, r representing relations, and t representing the tail entity.

We utilize the complex representation and GNN layer to perform aggregating graph features in our approach. First, entities and relations are embedded into the complex space. Thus each embedding consists of a real component and an imaginary component. We denote $\mathbf{H}^{(0)}_{real}$ and $\mathbf{H}^{(0)}_{img}$ are the set of real and imaginary components of entities E; $\mathbf{R}^{(0)}_{real}$ and $\mathbf{R}^{(0)}_{img}$ are the set of real and imaginary components of relations R as follows:

$$\mathbf{H}^{(0)}_{real} = \{\mathbf{h_{r1}}, \mathbf{h_{r2}}, ..., \mathbf{h_{rN}}\}; \qquad \mathbf{H}^{(0)}_{img} = \{\mathbf{h_{i1}}, \mathbf{h_{i2}}, ..., \mathbf{h_{iN}}\} \tag{2}$$

$$\mathbf{R}^{(0)}_{real} = \{\mathbf{r_{r1}}, \mathbf{r_{r2}}, ..., \mathbf{r_{rN}}\}; \qquad \mathbf{R}^{(0)}_{img} = \{\mathbf{r_{i1}}, \mathbf{r_{i2}}, ..., \mathbf{r_{iN}}\} \tag{3}$$

where $\mathbf{h_{rj}}$ and $\mathbf{h_{ij}}$ are the real and imaginary components of entity j^{th}, $\mathbf{r_{rj}}$ and $\mathbf{r_{ij}}$ are the real and imaginary components of relation j^{th}.

Then, we collect neighbors' information in $\mathbf{H}^{(0)}_{real}$ and $\mathbf{H}^{(0)}_{img}$ by using the GCN and GATv2 layers. GCN is operated on graph structure as the KGs structure, so we applied GCN to propagating information around one-hop neighbors, which helps to exploit the structural information of KGs at the first layer. In our work, the first layer takes the input matrix $\mathbf{H}^{(0)}$, and generates a feature representation for the second layer. The output matrix $\mathbf{H}^{(1)}$ is calculated as follows:

$$\mathbf{H}^{(j+1)} = \sigma(\mathbf{W}^{(j)}\mathbf{H}^{(j)}\mathbf{A}') \tag{4}$$

where σ is an activation function, $\mathbf{W}^{(0)}$ is weight matrix at the first layer, $\mathbf{A}'$ is the normalized adjacency matrix. To construct the normalized adjacency matrix $\mathbf{A}'$, we first construct the adjacency matrix $\mathbf{A}$ representing connections between entities. In addition, to consider the features of each entity itself, we create a matrix $\widehat{\mathbf{A}}$ by adding $\mathbf{A}$ with its identity matrix as follows:

$$\widehat{\mathbf{A}} = \mathbf{A} + \mathbf{I_A} \tag{5}$$

where $\mathbf{I_A}$ is the identity matrix of size $N \times N$. Next, $\widehat{\mathbf{A}}$ is normalized to form matrix $\mathbf{A}'$ as follows:

$$\mathbf{A}' = \mathbf{D}^{-1/2}\ \widehat{\mathbf{A}}\ \mathbf{D}^{-1/2} \tag{6}$$

where $\mathbf{D}$ is the degree matrix.

After obtaining two components of feature's representation for each entity $\mathbf{H}^{(1)}_{real}$ and $\mathbf{H}^{(1)}_{img}$. In the GCN layer, we continue to expand these representations by performing an additional one-hop aggregation to assess the importance coefficients for each neighbor in the Dynamic graph attention layer. To do that, the attention scores are calculated as follows:

$$e_{ij} = e(\mathbf{h_i'}, \mathbf{h_j'}) = a^T \text{LeakyRELU}(\mathbf{W}.[\mathbf{h_i'}||\mathbf{h_j'}]) \tag{7}$$

Next, these scores are normalized by Softmax function to make them easily compare as follows:

$$\alpha_{ij} = \text{Softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})} \tag{8}$$

where α_{ij} is the normalized attention score between entity i^{th} and j^{th}.

Finally, this attention score is leveraged to calculate the final output $\mathbf{H}_{real}^{(2)}$ and $\mathbf{H}_{img}^{(2)}$ as follows:

$$\mathbf{H}^{(2)} = \sigma(\sum_{j \in N_i} \alpha_{ij} \mathbf{W}_j) \tag{9}$$

3.2 The Scoring Function

To measure the plausibility of triplet, we employ the ComplEx's scoring function by Eq. 11, which can model more accurately the symmetric and symmetric in each triplet. By obtaining the entity representation from $\mathbf{H}_{real}^{(2)}$, $\mathbf{H}_{img}^{(2)}$, and the relation embedding in complex space, we can easily compute the score of triplets. Moreover, we also apply the Sigmoid function to the score before assessing the result with the truth labels. The probability of forming triplet (h, r, t) as follows:

$$P(Y_{(h,r,t)} = 1) = \sigma(\ \phi(h, r, t)\ ) \tag{10}$$

where σ is the Sigmoid function, ϕ is the scoring function can be computed as Eq. 11, $Y \in \{-1, 1\}$ represent a binary value of a relation between head and tail entity.

$$\begin{aligned}\phi(h, r, t) = & \left\langle Re(\mathbf{w_r}), Re(\mathbf{h_h''}), Re(\mathbf{h_t''})\right\rangle + \left\langle Re(\mathbf{w_r}), Im(\mathbf{h_h''}), Im(\mathbf{h_t''})\right\rangle \\ & + \left\langle Im(\mathbf{w_r}), Re(\mathbf{h_h''}), Im(\mathbf{h_t''})\right\rangle - \left\langle Im(\mathbf{w_r}), Im(\mathbf{h_h''}), Re(\mathbf{h_t''})\right\rangle\end{aligned} \tag{11}$$

where $\mathbf{h_h''}, \mathbf{h_t''} \in \mathbf{H}^{(2)}, \mathbf{w_r} \in \mathbf{W}$.

Generally, our proposed model (ComplEx-GNN) considers the graph connectivity, relation type, and complex representation into the complex space. The graph aggregation layers allow learning the graph features while retaining the learning model's essential features. Additionally, we refer to the ComplEx model as the encoder and decoder for our framework. It helps to encode the entities and relation into complex values and then estimate triplets' plausibility by the scoring function. We also observe that our model acquired remarkable improvement over the ComplEx model without additional information.

4 Experiments and Result Analysis

4.1 Datasets and Evaluation Protocol

We use two widely used datasets to evaluate the link prediction task, including **FB15-237** [16] and **WN18RR** [7]. Table 1 illustrates statistics about two datasets. We also use two common benchmarks including Hit@k, and mean reciprocal rank (MRR), to evaluate the performance of the link prediction task.

Table 1. Statistics of FB15k-237 and WN18RR datasets.

Dataset	Entities	Relations	Training triples	Validation triples
FB15k-237	14541	237	272115	17535
WN18RR	40943	11	86835	3034

4.2 Hyperparameters

The hyperparameters include {learning rate, embedding dim, hidden dim, dropout rate}. Through experiment, the following hyperparameters gave the best result on FB15k-237: {0.001, 300, 400, 0.3}; on WN18RR: {0.003, 200, 400, 0.4} and the other configuration based on the framework of ConvE. We use PyTorch version 1.9.1 and run on NVIDIA Tesla V100-DGXS-32 GB. The training time on WN18RR and FB15k-237 is about 1–2 min, respectively.

4.3 Results

Table 2. Link prediction results on FB15k-237 and WN18RR

Model	FB15k-237				WN18RR			
	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
TransE-GCN [5]	0.315	0.229	**0.388**	0.477	0.233	0.203	0.338	0.508
DistMult [19]	0.241	0.155	0.263	0.419	0.43	0.39	0.44	0.49
ComplEx [17]	0.247	0.158	0.275	0.428	0.44	0.41	0.46	0.51
RGCN [15]	0.248	0.153	0.258	0.417	0.402	0.345	0.437	0.494
VR-GCN [22]	0.248	0.159	0.272	0.432	–	–	–	–
ConvE-Literal [10]	0.303	0.219	0.33	0.471	-	-	-	-
Complex-Literal [10]	0.305	0.222	0.336	0.466	–	–	-	–
ComplEx-GNN (our)	**0.328**	**0.238**	0.361	**0.509**	**0.451**	**0.415**	**0.462**	**0.522**

To illustrate the efficiency of our model, we compare our works with some standard models on the bilinear group such as DistMult and Complex, while comparing with models using additional information like ComplEx-Literal and ConvE-Literal to compare the power of structural information with additional information. Furthermore, our result is also compared with some pioneer GCN models employing both embeddings of entities and relations such as VR-GCN, TransE-GCN, and R-GCN to demonstrate the benefits of complex values in complex space. Table 2 illustrates our comparisons on FB15k-237, and WN18RR datasets. Summarizes the results of eight different baseline models including our models on FB15k-237 and WN18RR in terms of MRR - Mean reciprocal rank, Hits@1, Hits@3, and Hits@10.

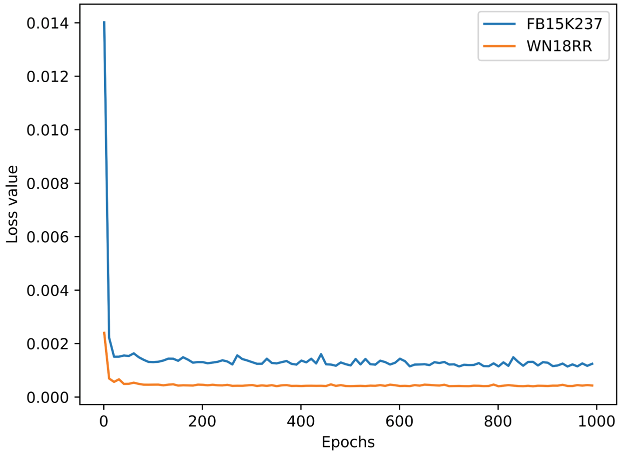

Fig. 2. The loss values of ComplEx-GNN models in FB15k-237 and WN18RR

In Table 2, our model achieved the best performance except for Hits@3. In detail, ComplEx-GNN model outperforms ComplEx Hits@10 by 5.9% and Complex Hits@3 by 7%. Furthermore, the Hits@1 in our model are 23.8%, which is 8% higher than the Complex model's 15.8%. Additionally, our approach improved the average rank of accurate triples by about 8.1% in MRR. The results confirmed the efficiency of the graph aggregation layer mentioned in Sect. 3. Meanwhile, when compared to the VR-GCN and RGCN models, our model achieved an improvement of more than 7% in all metrics. For the TransE-GCN model, ComplEx-GNN is higher around 1.3% on MRR, and 3.2% on Hits@10.

On WN18RR, the results of ConvE-Literal, ComplEx-Literal, and VR-GCN are empty due to the missing experiments on the original paper. In general, we obtain the best results on WN18RR. Compared with ComplEx, ComplEx-GNN enhances Hits@1 by 0.5% and MRR by 1.1%. For the Distmult, our model improves by around 2%-3% in all of metrics. Moreover, compared to the TransE-GCN model, ComplEx-GNN improves by 21.8% on MRR, 21.2% on Hits@1, 12.4% on Hit@3, and 1.4% on Hits@10. Meanwhile, our model achieves a marked increase of about 4.9% on MRR, 7% on Hits@1 and 2.5% on others metrics. This proves that our model retained the advantage property of the complex representations and enhancing the quality of embeddings to obtain improved results.

Next, we evaluate the convergence process of the model on both datasets. Figure 2 shows the loss proportion during the training phase of our model on FB15k-237 and WN18RR after every 10 epochs. Overall, the figure for WN18RR was consistently lower than that of FB15k-237. Additionally, after decreasing to 0.13% in FB15k-237 and 0.05% in WN18RR at the first 100 epochs, the WN18RR's loss percentage plateaued at 0.04%, and the figure for FB15k-237 witnessed a wild oscillation ranging from 0.13% to 0.16% in the remaining train-

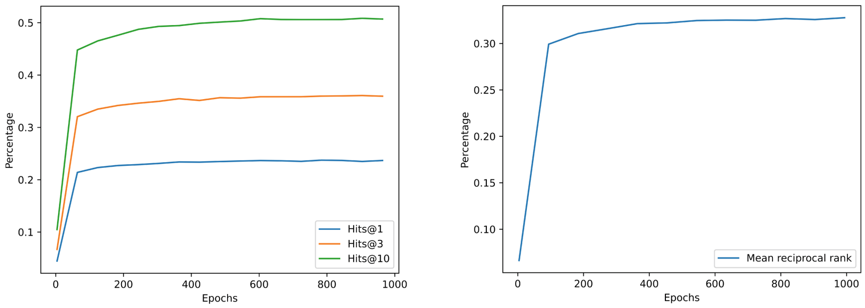

Fig. 3. The metric convergence research of ComplEx-GNN on FB15k-237

ing process, respectively. In addition, convergence proceeds quickly at the first 100 epochs on WN18RR, which is unstable on the remaining dataset. To save training time, we should use around the first 300 epochs on the WN18RR and about 1000 epochs for FB15k-237.

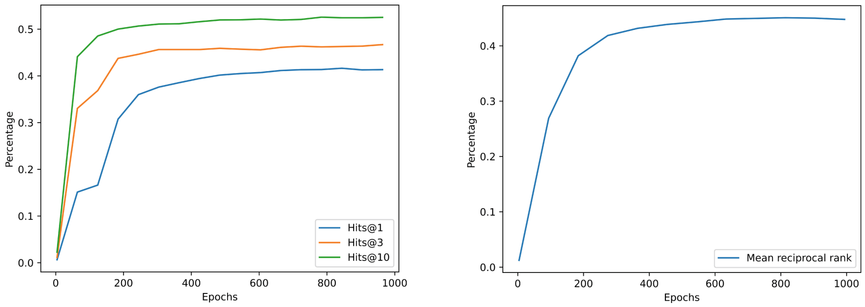

Fig. 4. The metric convergence research of ComplEx-GNN on WN18RR

Moreover, we also analyze the metrics' convergence to determine the possible epoch to train the model optimally. Figure 3 shows the Hit@n, mean rank, and mean reciprocal rank of ComplEx-GNN on FB15k-237 dataset. At first glance, on Hit@k's chart, the percentage saw an upward coverage tendency at the first 64 epochs, then having a mild fluctuation from 23.1% to 23.6% on Hits@1, 33.4% to 35.9% on Hits@3 and 49.4% to 50.6% on Hits@10, respectively. Meanwhile, a similar trend was experienced on MRR's chart ranging 32.1% to 32.8%, approximately. Figure 4 shows our experiments on WN18RR, the statistics for Hits@k(1, 3, 10) climbed substantially, reaching 37.6% on Hits@1, 45.6% on Hits@3, and 51.1% on Hits@10 at the first 304 epochs, before continuing to follow this tendency but less critically. Meanwhile, the MRR rapidly increased to 43.1% before fluctuating with a variation of around 1.6%. The charts shows that the convergence proceed of Hits@k is slightly slower than MRR on both two dataset.

In conclusion, the convergence of our model shows that it is fast and requires at least 300 epochs for WN18RR and 1000 epochs for FB15k-237. Although the stable process on WN18RR, the poor result on that dataset.

5 Conclusion

In this paper, we have proposed ComplEx-GNN model based on integrating graph neural networks and complex representation for the link prediction problem. First, we embedded entities and relations into complex space, then utilized the graph features by the Graph Aggregation Layers. By the way, the graph connectivity and relation types can be included in the learning process. The ComplEx's scoring function is applied to take advantage of the complex space, which could handle symmetric and asymmetric relations and enhance the model's expressivity. The gap between our model with ComplEx and LiteralE and GNN models proves the importance of structural information, indicating the power of complex space in representing entities and relations. Although obtaining the better result, the training time is significant and performs poorly on WN18RR.

In future work, we intend to embed graph features into the hypercomplex space to exploit more latent semantic information. Moreover, we also seek to simplify the Graph Aggregation Layer to be scalable in large KGs.

Acknowledgements. This research is funded by the University of Science, VNU-HCM, Vietnam under grant number CNTT 2022-02 and Advanced Program in Computer Science.

References

1. Abdelaziz, I., Fokoue, A., Hassanzadeh, O., Zhang, P., Sadoghi, M.: Large-scale structural and textual similarity-based mining of knowledge graph to predict drug-drug interactions. J. Web Semant. **44**, 104–117 (2017)
2. Alon, U., Yahav, E.: On the bottleneck of graph neural networks and its practical implications. arXiv preprint arXiv:2006.05205 (2020)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
4. Brody, S., Alon, U., Yahav, E.: How attentive are graph attention networks? arXiv preprint arXiv:2105.14491 (2021)
5. Cai, L., Yan, B., Mai, G., Janowicz, K., Zhu, R.: TransGCN: coupling transformation assumptions with graph convolutional networks for link prediction. In: Proceedings of the 10th International Conference on Knowledge Capture, pp. 131–138 (2019)
6. Dai, Y., Wang, S., Xiong, N.N., Guo, W.: A survey on knowledge graph embedding: approaches, applications and benchmarks. Electronics **9**(5), 750 (2020)
7. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

8. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 687–696 (2015)
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
10. Kristiadi, A., Khan, M.A., Lukovnikov, D., Lehmann, J., Fischer, A.: Incorporating literals into knowledge graph embeddings. In: Ghidini, C., et al. (eds.) ISWC 2019. LNCS, vol. 11778, pp. 347–363. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30793-6_20
11. Li, F.L., et al.: Alime assist: an intelligent assistant for creating an innovative e-commerce experience. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 2495–2498 (2017)
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
13. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.: A novel embedding model for knowledge base completion based on convolutional neural network. arXiv preprint arXiv:1712.02121 (2017)
14. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: ICML (2011)
15. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
16. Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., Gamon, M.: Representing text for joint embedding of text and knowledge bases. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1499–1509 (2015)
17. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning, pp. 2071–2080. PMLR (2016)
18. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017)
19. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)
20. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)
21. Yang, B., Yih, W.T., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)
22. Ye, R., Li, X., Fang, Y., Zang, H., Wang, M.: A vectorized relational graph convolutional network for multi-relational network alignment. In: IJCAI, pp. 4135–4141 (2019)