Few-Shot Link Prediction using Variational Heterogeneous Attention Networks

1st Xi Tao
Shanghai University
Shanghai, China
20721546@shu.edu.cn

2nd Hao Wang
Shanghai University
Shanghai, China
wang-hao@shu.edu.cn

3rd Xiangfeng Luo
Shanghai University
Shanghai, China
luoxf@shu.edu.cn

4rd Pinpin Zhu
Shanghai University
Shanghai, China
zhupp@shu.edu.cn

***Abstract*—Few-shot link prediction is an important recent research direction in the field of knowledge graph. To learn better entity and relation representations, previous works utilize the neighborhood information of entities but they ignore the associated relations. More than this, the link invovles few-shot entities cannot be well linked by these methods. Therefore, this paper proposes a novel method that dynamically aggregates local neighbourhood information from entities and relations and introduces the information bottleneck principle to filter irrelevant features to better capture the feature of few-shot entities. Experimental results on three benchmark datasets demonstrate that our model outperforms other models and that our approach can effectively improve the model performance on those entities in the tail of the long-tailed distribution.**

***Index Terms*—Link Prediction, Variational Information Bottleneck, Heterogeneous Graph**

I. Introduction

Knowledge graphs use a structural and easily exploitable way to store the knowledge which describes the real world. This allows real-world knowledge to be well applied to many downstream tasks such as intelligent question and answer (QA) systems, financial anti-fraud domains, and search engines. These structural facts are recorded and described by KG in the form of (e_i, r_k, e_j), for example, $(BillGates, livein, Washington)$, describing the fact that Bill Gates lives in Washington. Although KG contains many entities, relations and triples, there are still incompleteness in KG, which has motivated researches on link prediction.

In literature, many sophisticated models have been designed to complement the facts in KG [1]. Among them, representation learning [2] is one of the mainstream approaches. It aims to learn better representation of nodes to accurately predict missing entities. Due to the graph structure of KG, it is natural to introduce graph convolutional networks (GCN) into link prediction. In addition, the emergence of graph attention networks (GAT) [3]–[5], which incorporate attention mechanisms into GCNs, achieve great improvement.

However, as shown in Figure 1, when we utilize the neighborhood information to learn the representation of entities, we observe the following phenomena: (i) The relation “cooperate“ links to different head entities, and it may be more important when linking to “Bill Gates“ instead of “David“, because the former is a influential figure, which illustrates that the contribution of the same relation may be different due to different head entities. (ii) The tail entity “ChatGPT“ may be more important when linking to “develop“ instead of “user_of“, because “develop“ needs more effort, which illustrates that the contribution of tail entity may be affected by the relation. (iii) The entities and relations presents a long-tailed distribution, which makes the model is easily disturbed by irrelevant features because of lack of sufficient data. Based on these observations, we summarize two drawbacks of the previous representation learning approaches:

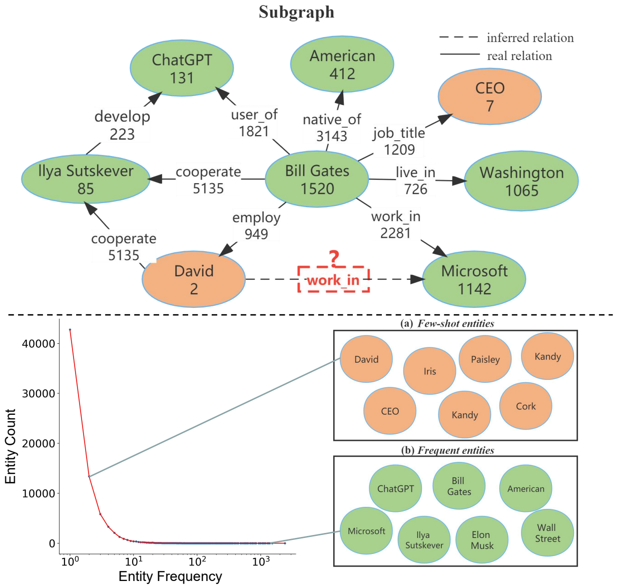

Fig. 1. The figure above is a subgraph of a KG containing real relations(solid lines) and inferred relations(dashed lines), and numbers indicate their frequency in the whole KG. The entity distribution in the figure below shows that most entities are few-shot entities, which increases the difficulty of inferring the relations between entities, and the green lines represent the distribution of the corresponding entities.

- When learning entity representations, various graph attention mechanisms are designed to measure the importance of neighboring entities, but ignore the fact that adjacency relations and neighboring entities can interact with each other and jointly determine the semantics of the current centric node.
- When extracting features of entities and relations, previous models take redundant features into account, though

they were not relevant to the link prediction task, which limited further improvements in model performance.

Therefore, to overcome these challenges, we propose a novel approach to simultaneously exploit neighboring entities and relations to obtain in-learning graph representations and add noise during the learning process to refine features useful for graph embedding while obtaining better robustness. Specifically, when learning graph embeddings, we design a novel type-aware graph attention mechanism to take neighbor entities and neighbor relations into account. Between the encoder and decoder, we introduce variational information bottleneck (VIB) [6], [7] to distill the latent representations output by the encoder, and add Gaussian noise in this process using the information bottleneck principle. Our contributions are as follows:

- We dynamically adjust the weights of relations and tail entities when aggregating information about neighboring nodes and let them optimize each other when updating node embeddings.
- We introduce the principle of information bottlenecks to control the information from the encoder to the decoder, which helps the model to better capture potentially useful features and improve its generalizability to few-shot entities.
- Experimental results on the FB15K-237, WN18RR and FB15K benchmark datasets show the strength of our model against another model in various metrics.

II. Related Work

A. Link Prediction

Massive efforts have been put into research on link prediction. Among all the link prediction methods, the most successful ones learn expressive representation for entities and relations. and predict missing entities in incomplete KG. These methods can be divided into three groups: translation-based models, neural network based model and GNN-based models. In this section, the three groups' models will be reviewed.

The translation-based models represented by TransE [1] and TransH [2] treat r_k in each triple (e_i, r_k, e_j) as a translation operation from e_i to e_j. Unfortunately, those models are not powerful enough to represent the complex facts in KG, so more and more sophisticated models have been designed for this challenge, such as TransD [8], TranSparse [9] and TransG [10]. In addition, RotatE [11] rotates entities through relations to achieve translation operations. These methods perform a series of subtle operations on entities and relations in combination with the actual situation, and effectively improve the accuracy of link prediction.

Due to the excellent performance of neural network models, neural networks are gradually being used in this task. ConvE [12] scores all candidate entities through convolution layer and dot layers, and ConvKB [13] considers global information on this basis. It is worth mentioning that ParamE [14] considers relations as parameters of neural networks to train models and complete link prediction tasks. However, none of these models were aware of the importance of graph structure information.

Therefore, GNN-based model gradually attracts the attention of more researchers. R-GCN [15] first applies the graph convolution operation to modeling relational data for link prediction. Latterly, KBGAT [16] introduced the attention mechanism to integrate neighborhood information to improve embedding quality. After that, researchers conducted a lot of research on this basis. Zhang, Zhuang et al. [4] transforms it into relation-level attention and entity-level attention. Zhao, Zhou et al. [17] adds additional global attention mechanism. Fang, Wang et al. [18] consider the impact of neighborhoods on representation learning of relations. However, those GNN-based methods cannot solve the few-shot entities problem, and they do not give enough consideration to neighborhood relations when aggregate information.

B. Information Bottleneck

The starting point of information bottleneck (IB) principle [19] is to transmit as much information as possible to the output y while compressing the input x in some way. The IB framework has been widely concerned in deep learning to get brief but comprehensive information. Recent works use VIB to improve the interpretability of the model [7] use it to reduce the overfitting on low-resource target tasks. Their research shows that the IB framework has significant advantages in capturing general features and can effectively enhance the robustness of the model in the case of few-shot learning.

Inspired by these ingenious algorithms, we propose the model ViHAN(**V**ariational **I**nformation Bottleneck for **H**eterogeneous **A**ttention **N**etwork) which can consider the impact of neighborhoods on the importance of relations and entities and can also filter out the useless features that the general-purpose feature extractors will inevitably extract.

III. Problem Statement

A large number of triples are stored in the KG, and every triple $\tau = (e_i, r_k, e_j)$ consists of a head entity e_i, a tail entity e_j and a relation r_k. Following the work of Bordes [1], we define link prediction as the task of analyzing useful information and predicting missing entity given the relation and its head entity. For example, for input $\tau = (France, capitalis, ?)$, the role of the link prediction model is to analyze KG and output the entity that the symbol "?" is most likely to refer to. During prediction, the model ranks the entities in the candidate set and selects the most likely correct entity [20] as the answer to what is the capital of France.

IV. Our Model

In this section, we will detail our ViHAN Model based on encoder-decoder architecture. Figure 2 shows the overall architecture of the model. Specifically, for each triple in KG, we dynamically calculate the importance of the relation and tail entity based on the inherent characteristics of the triple. According to the results, we aggregate the neighborhood information of the central entity to enhance the representation of entities. In order to eliminate useless features to improve robustness of model, we introduce VIB to filter features and help model learn better representation of few-shot entities. Afterwards, ConvKB analyze the global embedding properties of the triple with the help of score function to predict missing facts.

A. Joint Entity and Relation Encoder

In this section, we give detailed descriptions of our heterogeneous attention networks and the variational information bottleneck.

1) Heterogeneous Attention Networks: Intuitively, not all neighbors are equally important for representing the central entity, and the weights of neighbor entities and relations should be queried in combination with the graph structure information. Under the above consideration, we propose a new neighborhood-aware attention mechanism for link prediction. We add an additional query vector q to each node, which is used to query their importance in the current triple (e_i, r_k, e_j). The query method is as follows:

$$W_{(k,i)} = q_k^T q_i, W_{(j,k)} = q_j^T q_k \tag{1}$$

where $W_{(k,i)}$ refers to the importance of relation r_k queried from head entity e_i in current triple. Similarly, $W_{(j,k)}$ refers to the importance of tail entity e_j queried from relation r_k. $q \in \mathbb{R}^{1\times D}$ is the additional query vector used to calculate the relative attention score between two nodes and D is the dimension of our input embedding. After querying the importance of the relation and tail entity, to enrich the semantic representation, we integrate the neighborhood information in the current triple as Equation 2:

$$\alpha_i = W^H(e_i||\sigma\left(W_{(k,i)}r_k\right)||\sigma\left(W_{(j,k)}e_j\right)\ ) \tag{2}$$

where α_i is the semantically integrated representation of current triple, W^H is training parameter used to collect neighborhood information, σ is the activation function like tanh, and ”||” refers to the concatenating operation.

Then, we need to aggregate the information from different triples into the representation of each entity. Because different triples should be given different degrees of attention, we need to calculate the relative attention score η_i of each triple first. The process is shown in Equation 3:

$$\eta_i = softmax(\sigma(W^R\alpha_i)) = \frac{exp(GELU(W^R\alpha_i))}{exp(\sum_{\chi_i\in\chi}\sigma(W^R\alpha_i))} \tag{3}$$

where χ_i is the triples with the head entity e_i, χ denotes the triple set in KG. After using softmax function, activation function and weight matrix W^R, the neighborhood information from different triples should weight fused. Hence, we use the relative attention scores η and the triple information α to obtain the entity embedding $\hat{e}$:

$$\hat{e}_i = BN(\sum_{\chi_i\in\chi}\eta_i\alpha_i) \tag{4}$$

We calculate the importance weight of local neighborhood information and aggregate that information into the representation of entities. Then, we concatenate the embedding from different heads to extract diverse features as Equation 5:

$$\hat{e'}_i = \|_{m=1}^{M} BN(\sum_{\chi_i\in\chi}\eta_i^m\alpha_i^m) \tag{5}$$

Where M is the number of multiple heads and we use two heads in our code. Now $\hat{e'} \in \mathbb{R}^{2d}$ and $r \in \mathbb{R}^d$ are not consistent in dimension. In order to ensure dimensional consistency, we also perform a linear transformation on the embeddings of relations via a matrix $W^r \in \mathbb{R}^{D\times 2D}$ as follows:

$$\hat{r}^f = rW^r \tag{6}$$

Now entities already contain a large amount of information from the neighborhood but lack valuable information from their own. So we add their initial entity embedding information via a weight matrix $W^e \in \mathbb{R}^{D\times 2D}$ in the following way:

$$\hat{e}^f = eW^e + \hat{e'} \tag{7}$$

2) Variational Information Bottleneck: Such general-purpose feature extractors will inevitably extract features irrelevant to the target task, which is particularly obvious on few-shot entities and relations. Therefore, we introduce this component to control the representation from encoder, aiming to eliminate irrelevant features and redundant information. We compress the representation of entities and relations and extract the latent variables through two linear layers and two activation functions. After that, we use neural networks to fit the mean and variance of the input and add noise in the process of reconstructing the input:

$$E = \left\{\mu\left(\hat{e}^f\right) + \rho\Sigma\left(\hat{e}^f\right)\right\}||\hat{e}^f \tag{8}$$

$$R = \left\{\mu\left(\hat{r}^f\right) + \rho\Sigma\left(\hat{r}^f\right)\right\}||\hat{r}^f \tag{9}$$

where ρ represents the noise with Gaussian distribution. μ and Σ represent the mean and variance respectively, and their dimensions are one quarter of the input. E and R is the final representations learned by our encoder.

3) Training Objective: Inspired by the idea of TransE [1], the triple (e_i, r_k, e_j) are expected to satisfy $\overrightarrow{e_i} + \overrightarrow{r_k} \approx \overrightarrow{e_j}$, so for each triple, we measure its score as follows:

$$s_{\tau_{ijk}} = \|E_i + R_k - E_j\| \tag{10}$$

where $\tau = (e_i, r_k, e_j)$ is any triple including positive samples and negative samples. Absolutely, we want to score $s_{\tau_{ijk}}$ as low as possible for positive samples and as high as possible for negative samples, so we use hinge-loss during training:

$$L_G = max\left\{s_{\tau'_{ijk}} - s_{\tau_{ijk}} + \gamma, 0\right\} \tag{11}$$

where γ is a margin hyperparameter and τ'_{ijk} is the negative sample. The goal of training is to make $s_{\tau'_{ijk}}$ as close to 0 as possible, while $s_{\tau_{ijk}}$ as close to the threshold γ as possible.

At the same time, the VIB should also be limited during training. We hope it can remain the information related to

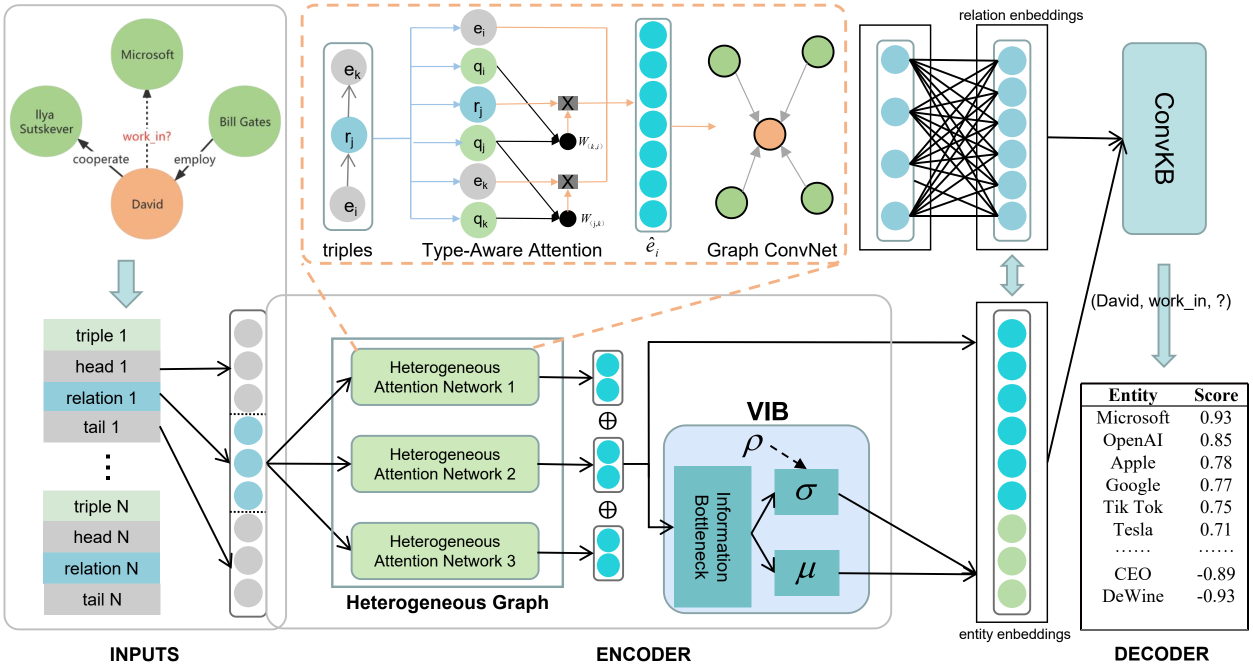

Fig. 2. An illustration of the ViHAN model. Given KG as input, ViHAN consists of the following three steps: (1) learns representations from KG by weighted fusion of neighborhood information. (2) masks irrelevant features and generates more generalized features. (3) performs link prediction through decoder.

link prediction task while compressing the input information, so the overall loss can be expressed by the following formula:

$$L_E = \beta KL\left[p_\theta\left(E|\hat{e}^f\right), r\left(E\right)\right] + \beta KL\left[p_\theta\left(R|\hat{r}^f\right), r\left(R\right)\right] + L_G \tag{12}$$

where β is a hyperparameter, which is used to ensure that the loss of prediction and the loss of compression is approximately equal. $p_\theta\left(\right)$ represents the estimate of the posterior probability of output when input is known. And $r\left(\right)$ is the estimate of the prior probability $p\left(\right)$.

B. Decoder

We tried different models as the decoder, such as TransE, ConvE and ConvKB [13], and finally selected the ConvKB model as our decoder with the best performance. Its convolutional layer uses multiple filters on triples to generate diverse feature maps. Since the samples consist of positive samples and negative samples, ConvKB defines an implausibility score:

$$F_D(\chi_i) = W \cdot \|_{m=1}^{M}(ReLU(E_i, R_k, E_j) * \zeta_m) \tag{13}$$

where ζ_m is the hyperparameter of the m^{th} filter. We concatenate the outputs of all filters, and then use the matrix $W \in \mathbb{R}^{1\times MD}$ to do the dot product (symbol "·") to calculate the scores of triples. Finally, the Adam optimizer is used in our decoder:

$$L_D = \sum_{\tau\in(\chi\bigcup\chi')} log(1 + exp(l(\tau)\cdot F_D(\tau))) + \frac{\lambda}{2}\|W\|_2^2 \tag{14}$$

Where $\chi^{'}$ is the negative triple set. When $\tau \in \chi$, $l(\tau) = 1$, and when $\tau \in \chi^{'}$, $l(\tau) = -1$. The purpose is to enable the model to effectively distinguish positive samples and negative samples.

V. Experiments

To demonstrate the superiority of the ViHAN model, we conduct extensive experiments on the link prediction task and further explore the generalizability of the model for rare entities in the long tail.

A. Datasets

We evaluate our model on three widely used benchmark datasets, FB15k-237, FB15k and WN18RR. FB15k is a subset of the KG Freebase, containing 14,951 entities and 1345 relations. FB15k-237 is the dataset which removed the reverse relations in FB15k and it contains 14541 entities and 237 relations. WN18RR is a subset of WordNet with 18 relations and 40,943 entities.

B. Evaluation Protocol

Our model scores each entity in the candidate entity set, ranks them according to their scores, and finally calculates various metrics such as mean rank(MR), mean reciprocal rank(MRR), the proportion of valid triples in the top-N ranks(hit@N).Therefore, we replace the original entities in triples with candidate entities, then score and rank them through the model.

C. Training Protocol

The training process of our model can be divided into two stages: (1) We train the encoder to obtain more expressive embeddings, and the hyperparameter β used to balance the

TABLE I
EXPERIMENTAL RESULTS. **BOLD** INDICATES THE BEST RESULTS IN EACH METRIC.

	FB15k-237				**WN18RR**				**FB15k**			
Methods	**MRR**	**H@1**	**H@3**	**H@10**	**MRR**	**H@1**	**H@3**	**H@10**	**MRR**	**H@1**	**H@3**	**H@10**
DistMult [21]	0.199	28.1	30.1	44.6	0.444	41.2	47.0	50.4	0.798	-	-	89.3
ComplEx [22]	0.194	2.8	29.7	45.0	0.449	40.9	46.9	53	0.692	59.9	75.9	84.0
ConvE [12]	0.225	31.2	34.1	49.7	0.456	41.9	47.0	53.1	0.657	55.8	72.3	83.1
RotatE [11]	0.338	24.1	37.5	53.3	0.476	42.8	49.2	57.1	0.699	58.5	78.8	87.2
KBGAT [16]	0.518	46.0	54.0	62.6	0.429	36.1	47.7	57.2	0.868	82.4	90.2	93.5
ParamE-Gate [14]	0.399	31.0	43.8	57.3	0.489	**46.2**	50.6	53.8	-	-	-	-
DualE [23]	0.365	26.8	40.0	55.9	**0.492**	44.4	51.3	58.4	0.813	76.6	85.0	89.6
HAN *w 1 head*	0.495	41.7	53.2	63.7	0.439	36.5	48.4	57.2	0.826	76.2	87.9	92.4
HAN *w 2 head*	0.508	42.9	54.7	65.8	0.434	35.2	47.7	58.7	0.886	85.9	90.6	92.7
HAN *w 3 head*	0.514	45.2	54.1	63.5	0.449	36.5	50.0	60.0	0.890	86.0	91.4	93.9
ViHAN *w 1 head*	0.528	45.2	56.6	66.5	0.459	37.9	50.4	59.9	0.853	80.3	85.2	92.8
ViHAN *w 2 heads*	**0.567**	**50.8**	**59.4**	**67.8**	0.474	39.8	**51.7**	**60.9**	0.889	85.9	91.1	93.7
ViHAN *w 3 heads*	0.548	48.4	57.6	66.8	0.462	38.4	50.8	59.9	**0.896**	**86.7**	**91.7**	**94.1**

loss of two components is set to $1e-4$; (2) On this basis, we train the decoder to perform the link prediction task. We use the Adam optimizer to update the parameters. The epoch number of the encoder and decoder is set to 3,600 and 200.

D. Results and Analysis

The experimental results of our models with different numbers of attention heads are shown in the table I. For FB15k-237 and FB15k, it can be concluded that our model has achieved the best results compared with the baseline model. This shows that our strategy, which dynamically assigns weights to entities and relations when aggregating neighborhood information and eliminates redundant feature, is correct.

Also, the result on WN18RR shows that our model outperforms other benchmark models on most metrics. At the same time, we notice that other models like DualE are about 20 points lower than our model on some metrics like Hit@3 on FB15K-237 but slightly improved over our model on WN18RR. Based on the difference between the two datasets, we believe that the idea of using neighbourhood information in our model enables us to perform well on FB15K-237 with many triples and few entities. However, it performs not good enough on WN18RR with few triples and many entities because of the relative lack of neighborhood information.

In order to further prove the effectiveness of our method, we select some few-shot relations that have relatively few occurrences in the training set and visualized the entities connected by them on the test set. Combined with Figure 3, it is not difficult to find that our model can better partition entities under the few-shot relations. For example, clubs such as "Watford F.C." and locations such as "Prague" are well distinguished by our model, while TransE is not. We think the reason why TransE can't distinguish these entities is that clubs can sometimes express the concept of geographical location. This shows that our method can learn high-quality representation of the entities under the few-shot relations.

E. Ablation Study

We conduct ablation experiments to explore whether the VIB can help the model capture deep latent features and improve its robustness. We counted the frequency of each entity in the training set and sorted them according to the number of occurrences. We name the 10% entity with the least number of occurrences as tail-10 entities, and use the same method to obtain tail-20 entities. Then we build four new test sets. The head entities and tail entities in those test sets are tail-10 entities and tail-20 entities respectively. Since irrelevant and redundant features are eliminated by the component, the model should have better performance than before. The experimental

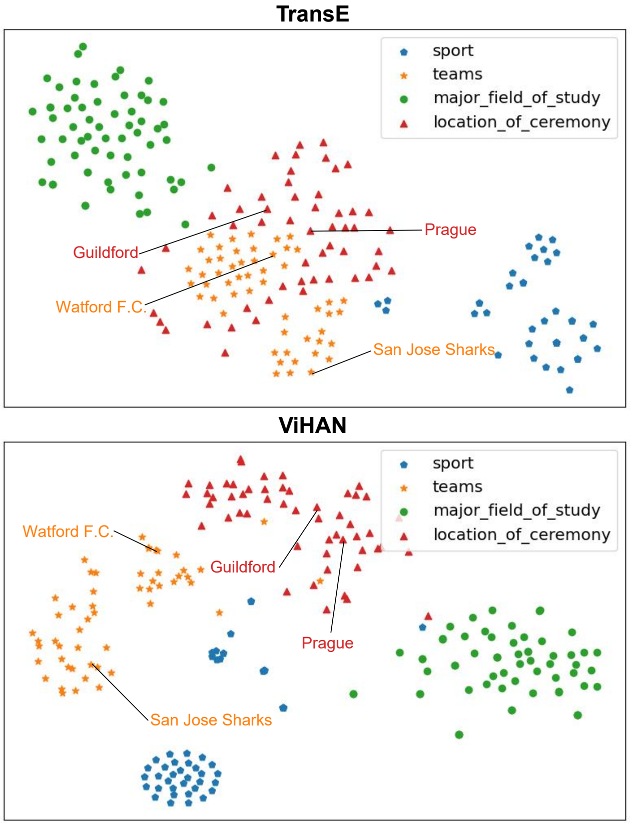

Fig. 3. T-SNE visualization of sampled relations and their corresponding tail entities in the test set using TransE (*above*) and ViHAN (*below*) on FB15k-237. Obviously, ViHAN can better distinguish the few-shot entities.

TABLE II

Ablation experiment results on few-shot entities, tail-10 entities and tail-20 entities represent the 10% and 20% tail entities with the lowest frequency respectively.

Dataset	Methods	Tail	Head	MR	MRR	H@1	H@3	H@10
FB15k-237	HAN	10	-	296	0.125	5.76	12.75	25.93
	ViHAN	10	-	**280**	**0.190**	**13.53**	**17.28**	**33.74**
	HAN	20	-	306	0.159	8.49	15.22	32.53
	ViHAN	20	-	**242**	**0.215**	**14.10**	**21.96**	**38.94**
	HAN	-	10	545	0.489	43.85	49.73	60.43
	ViHAN	-	10	**285**	**0.537**	**47.86**	**56.15**	**64.93**
	HAN	-	20	421	0.526	46.80	54.35	64.37
	ViHAN	-	20	**235**	**0.577**	**51.77**	**60.17**	**69.14**
WN18RR	HAN	10	-	3447	0.107	6.18	10.55	20.73
	ViHAN	10	-	**2907**	**0.147**	**9.81**	**15.27**	**24.36**
	HAN	20	-	2263	0.310	25.57	32.51	42.73
	ViHAN	20	-	**1958**	**0.345**	**28.21**	**37.46**	**45.21**
	HAN	-	10	2242	0.282	16.86	**36.77**	44.96
	ViHAN	-	10	**2130**	**0.340**	**26.23**	36.70	**47.78**
	HAN	-	20	2023	0.354	27.07	39.82	50.73
	ViHAN	-	20	**1886**	**0.417**	**35.35**	**44.11**	**53.94**

results prove this. The performance of the model with this component on all indicators has been significantly improved. Notably, the VIB component can compress and reconstruct the original representation. From this point of view, the lower the occurrence frequency of entities, the more important the role of the VIB. It is not difficult to see from the experimental results that our VIB significantly improves the generalization of few-shot entities. This shows that our hypothesis is completely correct.

VI. Conclusion

In this paper, we propose link prediction method for few-shot entities based on the information bottleneck principle and evaluate the representation capability of the model on link prediction task. Specifically, we design heterogeneous attention networks to model the importance of neighborhood information and better enhance the representation of the central entities. In addition, we introduce VIB to mine the deep features of nodes and improves the performance on few-shot entities. Extensive experimental results show that our approach outperforms other baseline models on mainstream datasets.

VII. Acknowledgement

The research reported in this paper was supported in part by the Shanghai Science and Technology Young Talents Sailing Program Grant 21YF1413900; the Outstanding Academic Leader Project of Shanghai under the grant 20XD1401700; and National Key R&D Plan 2021YFC3300602.

References

[1] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[2] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, 2014.

[3] Y. Han, Q. Fang, J. Hu, S. Qian, and C. Xu, "Gaeat: Graph auto-encoder attention networks for knowledge graph completion," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2053–2056.

[4] Z. Zhang, F. Zhuang, H. Zhu, Z. Shi, H. Xiong, and Q. He, "Relational graph neural network with hierarchical attention for knowledge graph completion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 9612–9619.

[5] Y. Zhao, H. Zhou, R. Xie, F. Zhuang, Q. Li, and J. Liu, "Incorporating global information in local attention for knowledge representation learning," in *Findings in ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, Aug. 2021, pp. 1341–1351.

[6] R. K. Mahabadi, Y. Belinkov, and J. Henderson, "Variational information bottleneck for effective low-resource fine-tuning," *arXiv preprint arXiv:2106.05469*, 2021.

[7] S. Bang, P. Xie, H. Lee, W. Wu, and E. Xing, "Explaining a black-box by using a deep variational information bottleneck approach," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 13, 2021, pp. 11 396–11 404.

[8] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of the 53rd ACL and the 7th IJCNLP*, 2015, pp. 687–696.

[9] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge graph completion with adaptive sparse transfer matrix," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[10] H. Xiao, M. Huang, Y. Hao, and X. Zhu, "Transg: A generative mixture model for knowledge graph embedding," *arXiv preprint arXiv:1509.05488*, 2015.

[11] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.

[12] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[13] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *Proceedings of the 16th NAACL-HLT*, 2018, pp. 327–333.

[14] F. Che, D. Zhang, J. Tao, M. Niu, and B. Zhao, "Parame: Regarding neural network parameters as relation embeddings for knowledge graph completion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 03, 2020, pp. 2774–2781.

[15] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European semantic web conference*. Springer, 2018, pp. 593–607.

[16] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning attention-based embeddings for relation prediction in knowledge graphs," *arXiv preprint arXiv:1906.01195*, 2019.

[17] Y. Zhao, H. Zhou, R. Xie, F. Zhuang, Q. Li, and J. Liu, "Incorporating global information in local attention for knowledge representation learning," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 1341–1351.

[18] H. Fang, Y. Wang, Z. Tian, and Y. Ye, "Learning knowledge graph embedding with a dual-attention embedding network," *Expert Systems with Applications*, vol. 212, p. 118806, 2023.

[19] A. Achille and S. Soatto, "Emergence of invariance and disentanglement in deep representations," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 1947–1980, 2018.

[20] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Machine Learning*, vol. 94, no. 2, pp. 233–259, 2014.

[21] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *arXiv preprint arXiv:1412.6575*, 2014.

[22] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.

[23] Z. Cao, Q. Xu, Z. Yang, X. Cao, and Q. Huang, "Dual quaternion knowledge graph embeddings," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6894–6902.