Appendix for Towards Few-shot Inductive Link Prediction on Knowledge Graphs: A Relational Anonymous Walk-guided Neural Process Approach

[Submission ID: #252]

No Institute Given

1 Derivation of ELBO Loss

Given an unseen entity u and its context data (support set) $\mathcal{C}_u$, the objective of `RawNP` is to infer the distribution $P(z|\mathcal{C}_u)$ from the context data that minimizes the prediction loss on the target data (query set) $\mathcal{D}_u$. The training objective of the neural process can be written as

$$P(z, e_q|u, r_q, \mathcal{C}_u) = P(z|\mathcal{C}_u) \prod_{\{(u,r_q,?)\in\mathcal{D}_u\}} P(e_q|f_u(u, r_q, z)), \tag{1}$$

where $f_u(u, r_q, z)$ denotes the inductive neural process link predictor, and $(u, r_q, ?)$ denotes the query to be predicted in the target data.

Following Eq. (1), the prediction likelihood on target data $\log P\left(e_q|u, r_q, \mathcal{C}_u\right)$ can be written as

$$\log P\left(e_q|u, r_q, \mathcal{C}_u\right) = \log \frac{P\left(e_q, z|u, r_q, \mathcal{C}_u\right)}{P\left(z|\mathcal{C}_u\right)}, \tag{2}$$

$$= \log P\left(e_q, z|u, r_q, \mathcal{C}_u\right) - \log P\left(z|\mathcal{C}_u\right), \tag{3}$$

Assuming that $Q(z)$ is the true distribution of z, we can rewrite the Eq. (3) as

$$\log P\left(e_q|u, r_q, \mathcal{C}_u\right) = \log \frac{P\left(e_q, z|u, r_q, \mathcal{C}_u\right)}{Q(z)} - \log \frac{P\left(z|\mathcal{C}_u\right)}{Q(z)}, \tag{4}$$

We integrate both sides with $Q(z)$.

$$\int_z Q(z) \log P\left(e_q|u, r_q, \mathcal{C}_u\right)$$

$$= \int_z Q(z) \log \frac{P\left(e_q, z|u, r_q, \mathcal{C}_u\right)}{Q(z)} - \int_z Q(z) \log \frac{P\left(z|\mathcal{C}_u\right)}{Q(z)}, \tag{5}$$

(6)

And then we can rewrite the $\log P\left(e_q|u, r_q, \mathcal{C}_u\right)$ as

$$\begin{aligned} &\log P\left(e_q|u, r_q, \mathcal{C}_u\right) \\ &= \int_z Q(z) \log \frac{P\left(e_q, z|u, r_q, \mathcal{C}_u\right)}{Q(z)} + KL\left(Q(z) \| P\left(z|\mathcal{C}_u\right)\right), \end{aligned} \tag{7}$$

Since $KL\left(Q(z) \| P\left(z|\mathcal{C}_u\right)\right) \geq 0$, we can write Eq. (7) as

$$\log P\left(e_q|u, r_q, \mathcal{C}_u\right) \geq \int_z Q(z) \log \frac{P\left(e_q, z|u, r_q, \mathcal{C}_u\right)}{Q(z)}, \tag{8}$$

$$= \mathbb{E}_{Q(z)} \log \frac{P\left(e_q, z|u, r_q, \mathcal{C}_u\right)}{Q(z)}, \tag{9}$$

$$= \mathbb{E}_{Q(z)} \left[\log P\left(e_q|u, r_q, z\right) + \log \frac{P\left(z|\mathcal{C}_u\right)}{Q(z)}\right], \tag{10}$$

$$= \mathbb{E}_{Q(z)} \left[\log P\left(e_q|u, r_q, z\right)\right] - KL\left(Q(z) \| P\left(z|\mathcal{C}_u\right)\right), \tag{11}$$

where $Q(z)$ represents the true posterior distribution of z, which is intractable. To address this problem, we approximate it with $Q\left(z|\mathcal{C}_u, \mathcal{D}_u\right)$ calculated by the encoder aggregating data from both $\mathcal{C}_u$ and $\mathcal{D}_u$ during training. In Eq. (11), the first term is to improve the prediction accuracy by maximizing the prediction likelihood. By minimizing the KL divergence in the second term, we encourage the encoder to infer the target posterior distribution with limited context data.

2 Algorithms of Training and Testing Process

Training. We illustrate the training process shown in Algorithm 1. In the training phase, we first sample an unseen entity u together with its support set $\mathcal{C}_u$ and query set $\mathcal{D}_u$ (Line 2). Then, we generate the relational motif representation M'_e for each entity e in the support set using relational anonymous walk (RAW) (Line 3). Then, we generate the representation of unseen entity u' using I-RGNN (Line 4). Combining the representations of each triple in support set c_i, we define the prior distribution $P(z|\mathcal{C}_u)$ using a RAW-guided neural process encoder (Line 5). Since the true distribution of $Q(z)$ is intractable, in the training stage, we feed both $\mathcal{C}_u$ and $\mathcal{D}_u$ together into the encoder to generate the variational posterior distribution $Q\left(z|\mathcal{C}_u, \mathcal{D}_u\right)$ (Line 6) to approximate the true distribution of $Q(z)$. Last, we sample a z from the posterior distribution to realize the prediction function (Line 7). By optimizing the ELBO loss, we can not only maximize the prediction likelihood but also encourage the encoder to infer the distribution with limited context data (Line 8).

Testing. In the testing stage, given an unseen entity u, we first generate the unseen entity representation u' and the prior distribution $P(z|\mathcal{C}_u)$ from its support set. Since we only have the support set during the testing phase, we sample a z from the prior distribution $P(z|\mathcal{C}_u)$. Then, we obtain the representation of possible other entity e'_q and its corresponding relational motif representation M'_{e_q}. Finally, we adopt the inductive neural process link predictor to predict the possible other entity e_q for a query $q = (u, r_q, ?)$.

Algorithm 1: The training process of `RawNP`

Input: Knowledge graph $\mathcal{G}$; Training entities $\mathcal{E}_{train}$
Output: Model parameters Θ

while *not done* **do**
  Sample an unseen entity $\mathcal{T}_u = \{\mathcal{C}_u, \mathcal{D}_u\}$ from $\mathcal{E}_{train}$;
  Generate relational motif representation M_e' using RAW;
  Generate unseen entity representation u' using I-RGNN;
  Generate prior distribution $P(z|\mathcal{C}_u)$ using RAW-guided neural process encoder;
  Generate the variational posterior distribution $Q\left(z|\mathcal{C}_u, \mathcal{D}_u\right)$ by feeding $\mathcal{C}_u$, $\mathcal{D}_u$ into the encoder;
  Sample a z from the posterior distribution $Q\left(z|\mathcal{C}_u, \mathcal{D}_u\right)$;
  Optimize Θ using ELBO loss;
end

3 Baseline Models

We select a series of following baseline models for comparison, which can be divided into three categories: (1) **Traditional KGC methods**, including TransE [2], DistMult [10], ComplEx [7], RotatE [6]; (2) **GNN-based methods**, including R-GCN [5], MEAN [4], LAN [8]; (3) **Few-shot inductive methods**, including GMatching [9], MetaR [3], FSRL [11], GEN [1]. We describe these baseline models in detail as follows:

Traditional KGC methods. This group of methods contains the translation model and the semantic matching model. The translation model focuses on the use of relationships between entities and the semantic matching model adopts semantic similarity to mine the potential semantics. They are all transductive methods.

- TransE [2] is a typical translation embedding model, which represents both entities and relations as vectors in the same space.
- RotatE [6] extends the TransE with a complex operation, which defines each relationship as the rotation from the source entity to the target entity in the complex vector space.
- DistMult [10] represents the relationship between the head and tail entity in a bi-linear formulation to capture the semantic similarity.
- ComplEx [7] introduces embeddings on a complex space to handle asymmetric relations.

GNN-based methods. Thanks to the inductive ability of GNNs, this group of methods utilizes the graph structure to model relational data and predict links inductively.

- R-GCN [5] extends the graph convolutional network to model multi-relational data.

- MEAN [4] utilizes a GNN-based neighboring aggregation scheme to generate the embedding of entities.
- LAN [8] further applies the attention mechanisms to consider relations with neighboring information by extending the MEAN model.

Few-shot inductive methods. This group of methods is all under the meta-learning framework, which can predict links for an unseen entity with few-shot related triples.

- GMatching [9] introduces a local neighbor encoder to learn entity embeddings and an LSTM matching network to calculate the similarity.
- MetaR [3] adapts to unseen relations by a relation-meta learner and updates the parameter under the meta-learning framework.
- FSRL [11] utilizes an LSTM encoder to summarize the support set information.
- GEN [1] meta-learns the unseen node embedding for inductive inference and proposes a stochastic embedding layer to model the uncertainty.

4 More Detailed Cases of Motif Extraction

In this section, we illustrate three detailed examples of motif extraction for unseen entities in Fig. 1, 2, and 3, respectively. In each figure, we first illustrate the Top-3 distinctive motifs extracted by relational anonymous walk. For each motif, we illustrate two relational paths that can be anonymized to the same motif.

From the results, we can find that the relational paths in support sets, positive triples, and negative triples are quite distinct. Therefore, we cannot predict triples by matching the relational paths with ones from the support set. This is consistent with our motivation that relational paths are not general enough for inductive link prediction on knowledge graphs. On the other hand, the different relational paths can be mapped into a general motif. For example, as shown in Fig. 1, paths $award \rightarrow winner \rightarrow award\_won \rightarrow award\_won$ and $genre \rightarrow film \rightarrow award\_nomince \rightarrow award\_noiminee$ can be mapped into the same motif $1 \rightarrow 2 \rightarrow 3 \rightarrow 2$. This motif can also be found in the support set that can be mapped by two other distinct paths, e.g., $relase\_data\_s \rightarrow awards\_won \rightarrow award\_nominee \rightarrow award\_nominee$ and $release\_data\_s \rightarrow file\_release\_region \rightarrow combatants \rightarrow combatants$. In this way, by extracting the relational motifs, we can find the general semantic patterns that are more suitable for inductive link prediction on knowledge graphs.

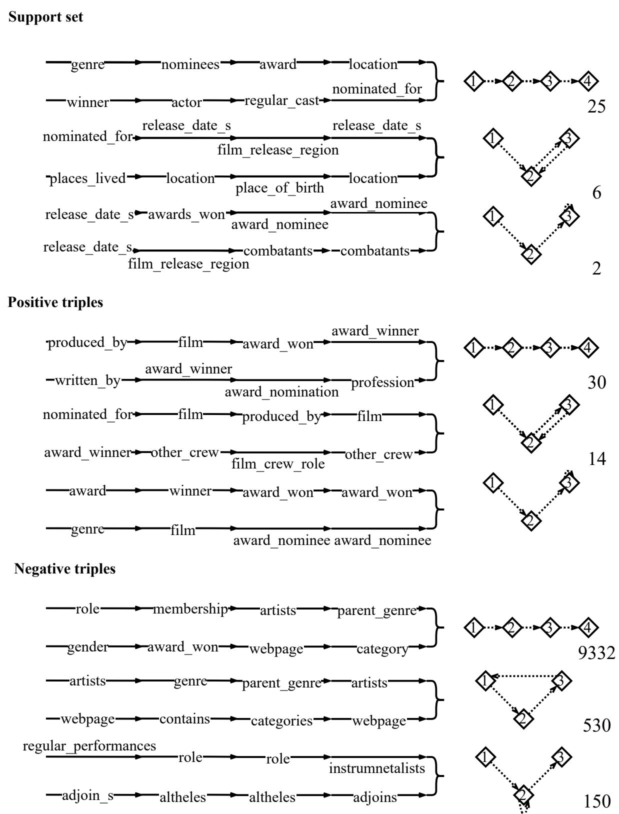

Fig. 1. Realistic relations for relational motifs in entity 4192

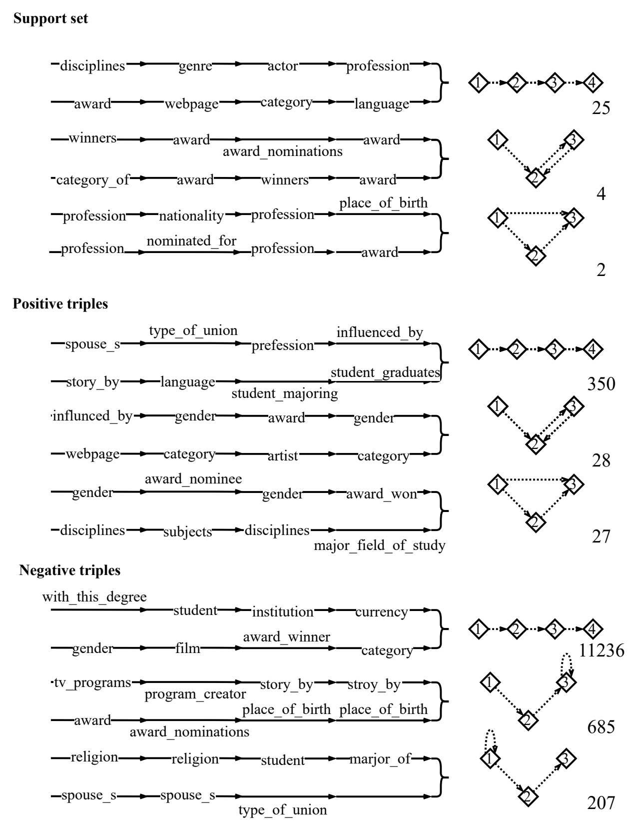

Fig. 2. Realistic relations for relational motifs in entity 3953

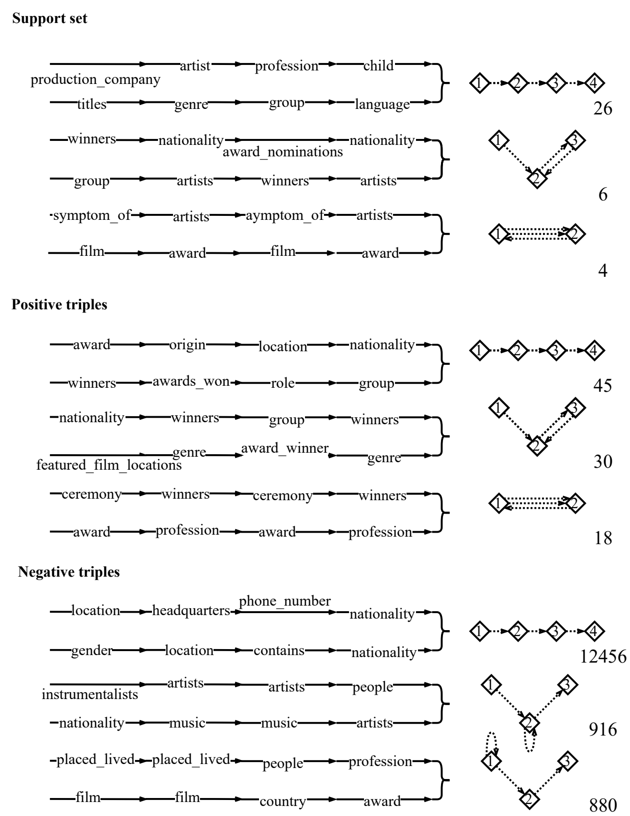

Fig. 3. Realistic relations for relational motifs in entity 7276

References

1. Baek, J., Lee, D.B., Hwang, S.J.: Learning to extrapolate knowledge: Transductive few-shot out-of-graph link prediction. Advances in Neural Information Processing Systems **33**, 546–560 (2020)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. Advances in Neural Information Processing Systems **26** (2013)
3. Chen, M., Zhang, W., Zhang, W., Chen, Q., Chen, H.: Meta relational learning for few-shot link prediction in knowledge graphs. arXiv preprint arXiv:1909.01515 (2019)
4. Hamaguchi, T., Oiwa, H., Shimbo, M., Matsumoto, Y.: Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. arXiv preprint arXiv:1706.05674 (2017)
5. Schlichtkrull, M., Kipf, T.N., Bloem, P., Berg, R.v.d., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European semantic web conference. pp. 593–607. Springer (2018)
6. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197 (2019)
7. 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)
8. Wang, P., Han, J., Li, C., Pan, R.: Logic attention based neighborhood aggregation for inductive knowledge graph embedding. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 7152–7159 (2019)
9. Xiong, W., Yu, M., Chang, S., Guo, X., Wang, W.Y.: One-shot relational learning for knowledge graphs. arXiv preprint arXiv:1808.09040 (2018)
10. 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)
11. Zhang, C., Yao, H., Huang, C., Jiang, M., Li, Z., Chawla, N.V.: Few-shot knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 3041–3048 (2020)