Neighborhood Intervention Consistency: Measuring Confidence for Knowledge Graph Link Prediction

Kai Wang\textsuperscript{1,2*}, Yu Liu\textsuperscript{1} and Quan Z. Sheng\textsuperscript{2}

\textsuperscript{1}School of Software Technology, Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian University of Technology, Dalian, Liaoning, 116620, P.R. China
\textsuperscript{2}Department of Computing, Macquarie University, Sydney, NSW 2109, Australia
{kai\_wang, yuliu}@mail.dlut.edu.cn, michael.sheng@mq.edu.au

Abstract

Link prediction based on knowledge graph embeddings (KGE) has recently drawn a considerable momentum. However, existing KGE models suffer from insufficient accuracy and hardly evaluate the confidence probability of each predicted triple. To fill this critical gap, we propose a novel confidence measurement method based on causal intervention, called Neighborhood Intervention Consistency (NIC). Unlike previous confidence measurement methods that focus on the optimal score in a prediction, NIC actively intervenes in the input entity vector to measure the robustness of the prediction result. The experimental results on ten popular KGE models show that our NIC method can effectively estimate the confidence score of each predicted triple. The top 10% triples with high NIC confidence can achieve 30% higher accuracy in the state-of-the-art KGE models.

1 Introduction

Knowledge graphs (KGs), which record real-world factual triples in the form of (head entity, relation, tail entity), have been widely applied in various AI domains [Lin et al., 2020; Zhao et al., 2020]. To achieve automatic KG completion, link prediction based on the knowledge graph embedding (KGE) technologies have recently drawn considerable attention [Zhang et al., 2020; Ruffinelli et al., 2020]. Given an entity and a relation (we call it an e-r query), a typical KGE model scores all candidate entities and selects the entity with the optimal score to compose a new triple. However, the new predicted triples cannot be added into KGs directly because of the insufficient accuracy and unreliable confidence measurement [Safavi et al., 2020].

Researchers are recently devoted to improving the confidence measurement of KGE models using probability calibration methods [Platt, 1999; Guo et al., 2017]. Tabacof and Costabello [Tabacof and Costabello, 2020] find that KGE models are not well-calibrated, and the probability estimates for triple classification are unreliable. Safavi et al. [Safavi et al., 2020] demonstrate that calibration techniques can significantly reduce the calibration error of KGE models in the relation prediction task. These research efforts mainly work on classification tasks in the KGE domain, only predicting in a few categories. However, link prediction is a more challenging “learning to rank” problem, aiming to find the target entity from tens of thousands of candidate entities.

There are two main problems restricting the effectiveness of the probabilistic calibration methods for reliable link prediction in the KGE domain:

- **Unsuitable confidence measurement.** Previous measuring methods focus on the optimal score or compare it with other scores in the score sequence. However, as shown in Fig. 1(a), high optimal scores lead to high confidence but cannot achieve the same level accuracy. This is due to the fact that the score of each candidate only indicates its relative order among candidates in one prediction.

- **Unreliable calibration metrics.** Expected Calibration Error (ECE) [Niculescu-Mizil and Caruana, 2005] is commonly utilized to evaluate the calibration effect, but is not suitable for link prediction. Although ECE is greatly reduced after calibration in Fig. 1(b), most of the triples are compressed into the low confidence level, making it very hard to extract high-accuracy triples.

![Figure 1: Reliability diagrams of the RotH model [Chami et al., 2020] before and after calibration on the FB15k237 dataset, in which predicted triples are grouped into 10 bins according to their confidence scores. The blue bars refer to the average accuracy of triples in the same confidence level, while triple proportion (in marigold) is a percentage of the total triple quantity. The confidence score is the optimal score via a Sigmoid function.](image-url)
In this paper, we propose a novel confidence measurement method based on causal intervention, called \textit{Neighborhood Intervention Consistency (NIC)}. Different from previous methods focusing on the optimal scores, we evaluate the confidence score by verifying the robustness of prediction results. Benefiting from a \textit{causal inference analysis}, we propose a NIC framework actively intervening the scoring process of KGE models. Specifically, we generate a series of neighborhood vectors for an input entity by adjusting the entity vector’s value in different dimensions and observing whether the output sequences of the model changes or matches the original one. On this basis, we design several types of neighborhood intervention values and a dimension selection strategy for high-dimensional KGE models.

Furthermore, our link prediction experiments exploit new evaluation metrics to verify whether the predicted triples are valuable for KG completion, including the top 10% accuracy and confidence variance. Ten popular KGE models are selected for the evaluation, including six high-dimensional models and four low-dimensional ones. The experimental results show that NIC outperforms previous confidence measurement methods before and after calibration. The calibrated NIC score can effectively overcome the drawback of low confidence variance of the previous methods. Besides, the top 10% triples with high NIC confidence can achieve 30% higher accuracy in the state-of-the-art KGE models. Finally, we verify the optimal choices of the intervention value and prove that the dimension selection strategy can effectively balance the computational efficiency and prediction accuracy for high-dimensional KGE models.

2 Background

2.1 Knowledge Graph Embeddings

Let \( E \) and \( R \) denote the set of entities and relations, a knowledge graph (KG) \( \mathcal{G} \) is a collection of factual triples \((h, r, t)\), where \( h, t \in E \) and \( r \in R \). \(|E| \) and \(|R|\) refer to the number of entities and relations in \( \mathcal{G} \), respectively. Knowledge Graph Embeddings aim to represent each entity \( e \in E \) (or relation \( r \in R \)) as a \( d \)-dimensional continuous vector, and learn a scoring function \( f : E \times R \times E \rightarrow \mathbb{R} \) to score each triple. Most KGE models are trained by minimizing a negative sampling loss, to make the score of the qualified triple higher than those of negative samples [Wang et al., 2017].

Link Prediction. Generalized link prediction tasks include entity prediction and relation prediction. In this paper, we focus on the more challenging entity prediction task. Given an \( e \)-\( r \) query \((e, r)\), the typical link prediction aims to find the target entity \( m \in E \) satisfying that \((e, r, m)\) or \((m, r, e)\) belongs to knowledge graph \( \mathcal{G} \). As illustrated in Fig. 2(a), a KGE model needs to score all candidate triples and output a sorted score sequence. In the sequence, the entity with the optimal score is selected as \( m \) to construct the new triple.

Typical KGE Models. Various KGE models have been proposed, such as i) translation-based TransE [Bordes et al., 2013] and RotatE [Sun et al., 2019]; ii) factorization-based DistMult [Yang et al., 2015], ComplEx [Trouillon et al., 2016] and TuckER [Balazevic et al., 2019]; and iii) CNN-based ConvE [Dettmers et al., 2018] and ConvKB [Nguyen et al., 2017]. The major difference among them is the scoring function, because the negative sampling strategy or the loss function is generally universal.

Based on the hyperbolic space, Chami et al. [Chami et al., 2020] recently propose two KGE models, RotH and RefH, which perform well in low-dimensional KGE situations. To ensure the breadth of the low-dimensional KGE evaluation, we further extend two hyperbolic-based variants, TransH and DistH, inspired by TransE and DistMult. Table 1 displays the scoring functions of the ten KGE models used in this paper.

2.2 Probability Calibration

The calibration of KGE models focuses on predicting confidence scores representing the actual correctness probabilities of predicted triples. Under an ideal situation, if the model predicts that a triple is true with a 0.9 confidence score, it should be correct 90% of the time. Model calibration requires reliable confidence measurement and effective calibration methods to fix the calibration errors.

Confidence Measurement Methods. In the KG domain, two confidence measurement methods, SigmoidMax (SIG) and TopKSoftmax (TOP), have been applied in very recent literature [Tabacof and Costabile, 2020; Safavi et al., 2020]. Given the sorted score sequence \( S \) of a prediction, the two methods measure the confidence score by:

\[
p_{\text{sig}}(S) = 1/(1 + e^{-\max(S)})
\]

\[
p_{\text{top}}(S) = \max(e^{S[K]}/(\sum_{i=1}^{K} e^{S_i}))
\]

Both of them are computed by the single sequence \( S \) and focus on the optimal score.

Calibration Methods. There are also two typical calibration methods. Platt scaling [Platt, 1999] inputs prediction scores into a logistic regression, and learns scalar weights to output a confidence score for each sample. Isotonic regression [Guo et al., 2017] is a non-parametric calibration method, which fits a non-decreasing piece-wise constant function to the model output. Both methods can effectively improve calibration, but depend on reliable confidence measurement.

3 Neighborhood Intervention Consistency

We discuss a novel confidence measurement method, Neighborhood Intervention Consistency (NIC), in this section.

<table>
<thead>
<tr>
<th>Model</th>
<th>Score Function</th>
<th>Model</th>
<th>Score Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransE</td>
<td>(</td>
<td></td>
<td>h + r - t</td>
</tr>
<tr>
<td>DistMult</td>
<td>( h^2 \text{diag}(M_r) t )</td>
<td>DistH</td>
<td>( D_{\text{hyp}}(h \circ r, t) )</td>
</tr>
<tr>
<td>ComplEx</td>
<td>( \text{Re}(W^T \text{diag}(M_r^2) \text{diag}(W^T) t^*) )</td>
<td>RotH</td>
<td>( D_{\text{hyp}}(\text{Rot}(r), h, t) )</td>
</tr>
<tr>
<td>ConvE</td>
<td>( f(\text{vec}(f([r, \text{vec}(r) \ast e]^W) t) )</td>
<td>RefH</td>
<td>( D_{\text{hyp}}(\text{Ref}(r), h, t) )</td>
</tr>
<tr>
<td>RotatE</td>
<td>(</td>
<td></td>
<td>h \circ r - t</td>
</tr>
</tbody>
</table>

Table 1: The score functions of 10 different KGE models, where \( \gamma > 0 \) is the margin value, \( \circ \) denotes the element-wise Hadamard product, \( \ast \) denotes the convolution operator, \( \times \) denotes the tensor product, \( e^\alpha \) denotes a complex-number vector. \( N \) is the number of fact triples \( T \), and \( N_{T^*} \) is the number of negative samples \( T^* \). \( D_{\text{hyp}} \) is the hyperbolic distance, \( \oplus \) is Möbius addition operation, \( \text{Rot}() \) and \( \text{Ref}() \) refer to the specific transformation operation.
First, a causal graph is utilized to reveal the fundamental motivation of this method in Sec. 3.1, and then we introduce the basic framework of NIC in Sec. 3.2. After that, we detail two key technical components, neighborhood intervention and dimension selection, in Sec. 3.3 and 3.4, respectively.

3.1 Motivation and Causal Inference

For link prediction, a confidence score should reflect the possibility of whether the predicted triple is the optimal choice in the prediction. We argue that a high confidence contains two meanings: i) significance: the optimal score should obviously outperform the others; and ii) robustness: a disturbance of the input data should not affect the prediction results.

The two confidence measurement methods mentioned in Sec. 2.2 aim to measure the significance from the output score sequence. However, they fail in the link prediction task because the patterns among different scores vary unstably in different sequences. Therefore, we focus on the second aspect, the robustness of score sequences. Inspired by the causal intervention process, we propose the NIC framework in Sec. 3.2. After that, we detail two key technical components, neighborhood intervention and dimension selection, in Sec. 3.3 and 3.4, respectively.

3.2 The NIC Framework

Inspired by the causal intervention process, we propose the new confidence measurement which aims to evaluate the robustness of score sequences. The NIC framework is illustrated in Fig. 2(c) and contains three main steps:

Step 1. Neighborhood Intervention. To achieve causal intervention, we first modify each dimensional values of the input entity vector $e$ in turn and generate $d$ different neighborhood vectors. Specifically, the neighborhood vectors $N_e \in \mathbb{R}^{d \times d}$ is computed by:

$$N_e = (1 - \mathbf{I}_d) \times e + \mathbf{I}_d \times v(e),$$  

where $\mathbf{I}_d \in \mathbb{R}^{d \times d}$ is an identity matrix, and $v(e)$ is a function that outputs an intervention value used to replace the original value in $e$. We utilize multiple kinds of intervention values for different entity vectors, which will be detailed in Sec. 3.3.

Step 2. Sequence Generation. In this step, we first gather the original score sequence by inputting the original entity vector $e$ into the KGE model. As the size of entity set is huge, we extract the top $K$ scores $S$ and the corresponding entities $\{e_i \in E | 0 < i \leq K\}$ from the sorted sequence. Then, using the narrowed entity set and the neighborhood vectors $N_e$ (instead of $e$) as inputs, we can gather neighborhood score sequences $S^{N_e} = \{S^{N_e} | 0 < i \leq d\}$ from the model. The relative ranking of the $K$ candidate entities in each sequence, is denoted by $\varphi(S^{N_e}) = \text{Argsort}(S^{N_e})$, which contains the sorted candidate indexes in length $K$.

Step 3. Computing Consistency. The ranking $\varphi(S^{N_e})$ of each neighborhood sequence is expected to be consistent with $\varphi(S)$ of the original sequence, especially the part with higher scores. For this, we define the consistency value $\varsigma$ between the original sequence $S$ and each $S^{N_e}$, i.e.,

$$\varsigma(S, S^{N_e}) = \sum_{j=0}^{J} \sigma(S)_j(1 - \text{Sgn}(|\varphi(S)_j - \varphi(S^{N_e})|)),$$

where $\sigma(\cdot)$ and $\text{Sgn}(\cdot)$ refer to the Softmax function and the Sign function, respectively. The consistency value $\varsigma = 1$ when the two sequences having the same candidate index in $e'$ is determined by not only that of the same dimension in $e$, but also the other dimensional variables.

Benefiting from the causal graph, we can pinpoint the roles of different parts in the KGE model. To intervene in the score sequence, we use the causal intervention: $P(S | e, do(e_i))$. This process modifies one of the dimensions in the input entity vector and so as to exclude the effect of this variable on the score sequence. For $d$-dimensional embeddings, we can generate $d$ different neighborhood input vectors by modifying each dimensional variable, and formally we have:

$$P(S | do(e)) = \frac{1}{d} \sum_{i=0}^{d} P(S | e, do(e_i))$$  

It should be noted that we intervene the input vector $e$ rather than the direct cause $e'$, because before modifying an endogenous variable in $e'$, its exogenous variable has also affected the other dimensions of $e'$.
the top $J$ positions. Otherwise, a mismatching at the higher position would cause a larger decrease to $\varsigma$. Therefore, $\varsigma$ can indicates the robustness of prediction results before and after intervention. The hyperparameter $J$ determines the number of candidates when measuring consistency, it is usually lower than $K$ to ensure the candidate rankings having enough differentiation.

Finally, given the consistency value of each neighborhood sequence, the NIC score $p_{nic}$ is defined by:

$$p_{nic}(S) = \frac{1}{d} \sum_{i=0}^{d} (\varsigma(S, S^{N_i})).$$

(6)

The value domain of $p_{nic}(S)$ is $[0, 1]$, such that it can be used to measure the confidence score in KGE models.

### 3.3 Neighborhood Intervention

In the first step of the NIC framework, the $i$-th neighborhood vector is generated by modifying the $i$-th dimensional value of the original entity vector. There are still issues on how to define the intervention value $v$ to replace the original value. The most straightforward way is setting $v = 0$ for all entity vectors, such that the $i$-th dimension of the $i$-th neighborhood vector is equal to zero. However, we find that its performance is not ideal in our empirical studies.

Therefore, we further propose several intervention values by considering the statistical specificity of different entity vectors. Given the entity embedding vector $e$, there are four kinds of intervention values, including zero, mean, maximum, and minimum, i.e., $\{0, \text{avg}(e), \text{max}(e), \text{min}(e)\}$. There are more choices of intervention values, but we argue that the above four are representative. In Sec. 4, we will compare the effectiveness of different intervention values.

The intervention value cannot be selected randomly, because the NIC score should maintain uniqueness for the same input. In addition, as the first attempt of neighborhood intervention, this paper focuses on intervening one dimension in each neighborhood vector. It might be feasible to modify multiple values in the entity vector to generate more neighborhood vectors, which will be our future investigation.

### 3.4 Dimension Selection

We consider the computational complexity of NIC. Measuring the consistency by intervening each dimension costs much less than the original prediction, as long as setting a small value for the hyperparameter $K$. But the time complexity of NIC is still linearly dependent on the embedding dimension size $d$. Facing a high-dimensional KGE model (with more than 200 dimensions), NIC using all neighborhood vectors would cause high calculation cost.

Therefore, we design a dimension selection strategy to accelerate the NIC calculation. Based on a hypothesis that a dimension with lower variance hardly reflects the difference among entities, we measure the variance of each dimension. Then, we select a limited number of high-variance dimensions, which contribute more in the $d$-dimensional vector.

Specifically, given the entity embedding matrix $E \in \mathbb{R}^{[|E|] \times d}$ of a trained KGE model, $E_{i,j}$ refers to the $j$-th dimension of the $i$-th entity and the weight $w_j$ of the $j$-th dimension is defined as:

$$w_j = \frac{1}{|E|} \sum_{i=0}^{|E|} (E_{i,j} - \text{avg}(E_{:,j}))^2$$

(7)

Given the weight vector $w = [w_0, w_1, \ldots, w_d]$, we can maintain the top $D$ dimensions with the highest weights and set others to zero. Normalized by a Softmax function, the processed vector can be utilized to adjust the contribution of different dimensions.

### 4 Experiments

#### 4.1 Experimental Setup

Our experimental studies are conducted on two widely used datasets, WN18RR [Bordes et al., 2014] and FB15k237 [Toutanova and Chen, 2015]. The statistics of the datasets are given in Table 2. With $Train$, $Valid$, and $Test$, we refer to the number of triples in the training, validation and test sets.

| Dataset     | $|R|$   | $|E|$ | $#Train$ | $#Valid$ | $#Test$ |
|-------------|--------|------|----------|----------|---------|
| FB15k237    | 237    | 14,541 | 272,115  | 17,535   | 20,466  |
| WN18RR      | 11     | 40,943 | 86,845   | 3,034    | 3,134   |

Table 2: Statistics of the datasets.

Ten KGE models (see Table 1) are trained by following their original settings with the binary cross-entropy loss. For the six high-dimensional KGE models, such as TransE and TuckER, we set their embedding dimensions as 200, while the four low-dimensional models’ embedding dimension is 32. We select the hyper-parameters in the NIC framework via grid search. Specifically, we empirically select the number of remeasured entities $K$ among $\{3, 5, 10, 100\}$ and the position number $J$ for computing sequence consistency among $\{1, 3, 5, 10\}$. All experiments are performed on Intel Core i7-7700K CPU @ 4.20GHz and NVIDIA GeForce GTX1080 Ti GPU, and implemented in Python using the PyTorch framework.

#### 4.2 Evaluation Metrics

Considering ECE (Expected Calibration Error) [Niculescu-Mizil and Caruana, 2005] cannot adequately reflect the calibration effects in the link prediction task, we introduce three additional new evaluation metrics:

- **CVar**: the variance of confidence scores of all triples,
- **T10%MRR**: the average inverse rank of the top 10% high-confidence triples, and
- **T10%ACC**: the average accuracy (equal to Hits@1) of the top 10% high-confidence triples.

The CVar metric makes up for the drawback of ECE. The lower ECE and higher CVar jointly indicate that the confidence score can match the correctness probability better. T10%MRR and T10%ACC are designed for the demand of practical KG completion. Instead of manually screening the roughly-predicted triples, KG builders expect the high-confidence triples having high enough accuracy. Therefore, higher T10%MRR and T10%ACC scores indicate a better model performance for link prediction.
### 4.3 Experimental Results of the Ten Models

We first verify our NIC performance compared with existing confidence measurement methods, SigmoidMax (SIG) and TopKSoftmax (TOP), utilizing ten different KGE models on two datasets. NIC by default uses the intervention value “maximum” and no dimension weights. Given a trained KGE model and test triples, three confidence measurement methods are utilized respectively and then calibrated by two calibration methods, Platt scaling and Isotonic regression. To the best of our knowledge, this is the most comprehensive experimental study for the calibration of the link prediction task. We analyze the results in details in the rest of this section.

Table 3 shows the ECE and T10%ACC results of the ten KGE models after calibrated by isotonic regression. In terms of the ECE metric, all three confidence measurement methods can achieve low ECE values after calibration. NIC surpasses the other two methods on most of models, especially the 32-dimensional models. Comparing T10%ACC and the original accuracy ACC, all three methods can help distinguish high-accuracy triples. Our NIC outperforms the other two methods and significantly improves by more than 30% in some of the KGE models. Besides, TuckER and RotatE with 200 dimensions achieves the best ACC on two datasets, respectively. The RefH model outperforms the other low-dimensional models and achieves much higher T10%ACC.

In summary, fitting both low- and high-dimensional conditions, NIC can maintain the high-confidence triples having much higher accuracy. When only the most accurate triples are required in practical applications, we can use low-dimensional models with NIC measurements to achieve high-efficient link prediction.

### 4.4 Detailed Comparison in the RotH Model

We then concentrate on the confidence calibration in a single model, to better compare three confidence measurement methods using two calibration methods. Following the previous work, reliability diagrams bin all predicted triples by confidence scores into ten equally-sized regions of [0, 1]. To reflect the number distribution, we further add *triple proportion* (in marigold) to each confidence bin. The reliability diagrams for the RotH model using different confidence measurement methods before and after calibration are shown in Fig. 3. For other models, we observe the similar patterns.

The left three diagrams in each group illustrate the con-
Table 4: ECE and T10%MRR of four uncalibrated models using NIC with different intervention values on two datasets. The Avg refers to the mean value. The best scores are in **Bold**.

<table>
<thead>
<tr>
<th>Methods</th>
<th>FB15K237</th>
<th>WN18RR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>ECE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RotH</td>
<td><strong>0.063</strong></td>
<td>0.065</td>
</tr>
<tr>
<td>RefH</td>
<td>0.049</td>
<td>0.052</td>
</tr>
<tr>
<td>TransH</td>
<td>0.077</td>
<td>0.078</td>
</tr>
<tr>
<td>DistH</td>
<td>0.095</td>
<td><strong>0.087</strong></td>
</tr>
<tr>
<td>T10%MRR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RotH</td>
<td><strong>0.785</strong></td>
<td>0.778</td>
</tr>
<tr>
<td>RefH</td>
<td>0.830</td>
<td>0.840</td>
</tr>
<tr>
<td>TransH</td>
<td>0.740</td>
<td>0.727</td>
</tr>
<tr>
<td>DistH</td>
<td>0.642</td>
<td><strong>0.657</strong></td>
</tr>
</tbody>
</table>

4.5 Verification of Two Components

We deeply verify the effectiveness of two NIC components:

**Neighborhood Intervention.** We compare T10%MRR and ECE of NIC using four intervention values, i.e., zero, mean, maximum, and minimum. The results are shown in Table 4. For the four low-dimensional KGE models, using the maximum or minimum values achieve the best ECE in most models, and the zero and mean values perform slightly worse. Especially on FB15k237, their results are more than 50% lower than that of maximum and minimum. The similar trend can be found in the T10%MRR results, the maximum and minimum intervention precedes the others on RotH, TransH and DistH. The reason might be that the maximum and minimum values can provide better differentiation, while the others cannot change the original value significantly.

**Dimension Selection.** We select the RotH model with 256 dimensions to verify the effectiveness of dimension selection. Fig. 4 shows the T10%ACC and T10%MRR results on FB15K237. We can see that the T10%ACC of Weight(all) are slightly better than that of the original NIC (i.e., NIC-Max). As the improvement is not obvious, weighted summing is not needed when using all neighborhood vectors. However, the dimensional weight is valuable to reduce the number of neighborhood vectors for efficiency. When using only 32 neighborhood vectors, Weight(32) achieves slightly lower prediction accuracy than NIC-Max, but still outperforms significantly the other two methods (TopkSoftmax and Sigmod).

Figure 4: T10%ACC and T10%MRR of the RotH model using different dimension selection strategies on FB15K237. Weight(N) refers to using the neighborhood vectors whose dimension weight ranks in top N. Weight(all) means weighted summing all neighborhood vectors and NIC-Max is the original NIC method equally treating each vector.

5 Conclusion

Recent knowledge graph embedding (KGE) models can be rarely applied in the KG completion tasks in practice due to low prediction accuracy and unreliable confidence measurement. In this paper, we present a novel confidence measurement framework, namely Neighborhood Intervention Consistency (NIC). Based on the causal intervention, NIC actively intervenes the input entity vector to measure the prediction robustness. The experimental results show that our NIC method can effectively estimate the prediction accuracy while keeping an acceptable variance. Furthermore, NIC achieves 30% higher accuracy for the top 10% highest-confidence triples in the state-of-the-art KGE models. In the future, we will further improve our method by intervening multiple dimensions in one neighborhood vector, and by taking both statistical significance and prediction robustness into the consideration.

Acknowledgments

This research is supported by the National Natural Science Foundation in China (Grant: 61672128) and the Fundamental Research Fund for Central University (Grant: DUT20TD107). Quan Z. Sheng has been partially supported by Australian Research Council (ARC) Future Fellowship Grant FT140101247, and Discovery Project Grant DP200102298.
References


