Inductive Link Prediction in N-ary Knowledge Graphs

Jiyao Wei, Saiping Guan[*], Xiaolong Jin[*], Jiafeng Guo, Xueqi Cheng
School of Computer Science and Technology, University of Chinese Academy of Sciences;
Key Laboratory of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences.
{weijiyao20z, guansaiping, jinxiaolong, guojiafeng, cxq}@ict.ac.cn

Abstract

N-ary Knowledge Graphs (NKGs), where a fact can involve more than two entities, have gained increasing attention. Link Prediction in NKGs (LPN) aims to predict missing elements in facts to facilitate the completion of NKGs. Current LPN methods implicitly operate under a closed-world assumption, meaning that the sets of entities and roles are fixed. These methods focus on predicting missing elements within facts composed of entities and roles seen during training. However, in reality, new facts involving unseen entities and roles frequently emerge, requiring completing these facts. Thus, this paper proposes a new task, Inductive Link Prediction in NKGs (ILPN), which aims to predict missing elements in facts involving unseen entities and roles in emerging NKGs. To address this task, we propose a Meta-learning-based N-ary knowledge Inductive Reasoner (MetaNIR), which employs a graph neural network with meta-learning mechanisms to embed unseen entities and roles adaptively. The obtained embeddings are used to predict missing elements in facts involving unseen elements. Since no existing dataset supports this task, three datasets are constructed to evaluate the effectiveness of MetaNIR. Extensive experimental results demonstrate that MetaNIR consistently outperforms representative models across all datasets.

1 Introduction

Link prediction in KGs aims to predict missing links in KGs. It enriches KGs and enhances the performance of downstream applications such as recommendation systems (Kim et al., 2024), Web search (Peng et al., 2023), and question answering (Li and Ji, 2022; Qiao et al., 2022). Previous research has mainly focused on binary KGs, where facts are represented as (subject entity, relationship, object entity). However, real-world KGs are often N-ary KGs (NKGs), where a fact can involve more than two entities. N-ary facts, typically formalized as multiple role-entity pairs, are prevalent in real-world KGs. For example, in the well-known KG Freebase, over 1/3 of the entities involve n-ary facts (Wen et al., 2016), and more than 61% fact types are n-ary (Fatemi et al., 2021).

Link Prediction in NKGs (LPN) aims to predict missing elements in facts in NKGs. Current LPN methods can be divided into three categories: translation-based, tensor-based, and neural network-based approaches. Translation-based approaches encode n-ary facts using translation distances between entities (Wen et al., 2016; Zhang et al., 2018). Tensor-based approaches represent n-ary facts as a high-order tensor, where each element represents the validity of a fact (Liu et al., 2020; Di et al., 2021). Neural network-based approaches use various neural architectures to encode n-ary facts (Galkin et al., 2020; Luo et al., 2023). These approaches typically assume that NKGs are static, meaning that the sets of entities and roles are fixed. Specifically, they focus on predicting missing elements in facts composed of elements seen during training, known as the transductive setting.

However, real-world NKGs frequently emerge new facts involving unseen entities and roles. This requires methods that can complete facts containing unseen elements using some support facts, referred to as the inductive setting (see Figure 1). To the best of our knowledge, there is only one work (Ali et al., 2021) that completes n-ary facts involving unseen elements. However, in their setting, unseen elements can only be subject or object entities, and additional textual descriptions of entities are required. Therefore, we propose a more practical task, Inductive Link Prediction in NKGs (ILPN), which aims to predict missing elements in facts containing unseen entities and roles using some support facts in emerging NKGs.

The primary challenge of ILPN lies in generating accurate embeddings for unseen entities and roles.

[*]Corresponding Authors.

Proceedings of the 31st International Conference on Computational Linguistics, pages 8885–8896
January 19–24, 2025. ©2025 Association for Computational Linguistics

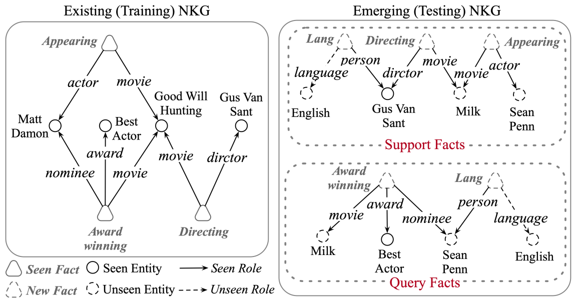

Figure 1: An example of NKGs in the inductive setting. Unseen entities and roles emerge during testing.

To address it, we design a Meta-learning-based N-ary knowledge Inductive Reasoner (MetaNIR), which leverages a Graph Neural Network (GNN) to adaptively generate embeddings for unseen elements. Drawing inspiration from the recent successful applications of meta-learning models (Finn et al., 2017; Wei et al., 2024), we sample multiple tasks comprising support facts and query facts from the existing NKG to simulate the inductive setting and let MetaNIR learn the ability to generate embeddings for unseen elements in each task. Since no existing benchmark supports ILPN, three new datasets are created to evaluate the performance of MetaNIR.

In summary, this paper makes the following contributions:

- We propose a new task, Inductive Link Prediction in NKGs (ILPN), which aims to predict missing elements in facts involving unseen entities and roles using some support facts in emerging NKGs.
- We propose a Meta-learning-based N-ary knowledge Inductive Reasoner (MetaNIR), which adaptively generates embeddings for unseen entities and roles.
- We construct three new datasets based on popular LPN benchmarks, offering valuable resources for this task and further research.
- Extensive experimental results on these datasets show that MetaNIR consistently outperforms existing representative models, validating its effectiveness.

2 Problem Definition

In this section, we provide the definitions of n-ary fact, NKG, and ILPN in turn.

N-ary fact is a fact involving two or more entities. In this paper, facts are formalized as $((r_s : e_s), (r_o : e_o), \{(r_i : e_i)\}_{i=1}^{n-2})$ following (Hou et al., 2023), where e_s and e_o denote its subject and object entities, respectively, and r_s and r_o denote the roles they play in the fact; e_i and r_i are its other entity (i.e., non-subject and non-object entity) and the corresponding role; n is the number of entities.

For example, the 4-ary fact "*Christopher Nolan won the MTV Movie Award for Memento in 2002*" is represented as $((winner : Christopher\ Nolan), (award : the\ MTV\ Movie\ Award), \{(work : Memento), (time : 2002)\})$.

NKG is a set of n-ary facts, where each can involve more than two entities.

ILPN aims to predict a missing element in facts involving entities and roles not seen during training in NKGs. The prediction is supported by some related facts within the emerging NKG, which are called support facts.

3 Related Work

The closest related research directions to this paper are LPN and Inductive Link Prediction in KGs (ILP).

3.1 Related Work of LPN

Existing LPN methods can be divided into three categories: translation-based methods, tensor-based methods, and neural network-based methods.

Translation-based methods learn relationships between entities based on the transfer distance between them. For example, m-TransH (Wen et al., 2016) projects entities into role spaces and evaluates the validity of facts based on the distance between projected entities. RAE (Zhang et al., 2018) enhances the performance of m-TransH by restricting the similarity of entities within the same facts to be higher.

Tensor-based methods represent the NKG as a high-order tensor, where each element indicates the validity of a fact. For example, GETD (Liu et al., 2020) extends Tucker (Balažević et al., 2019) to learn embeddings of entities and roles in n-ary facts by decomposing the tensor. However, this method is computationally intensive and cannot handle facts with varying numbers of entities simultaneously. Thus, S2S (Di et al., 2021) splits the embedding representation into multiple parts and uses different parts to encode facts with different numbers of entities.

Neural network-based methods employ various neural networks to encode n-ary facts. For example, NaLP (Guan et al., 2019) uses a fully connected layer to embed role-entity pairs, but it overlooks the differing importance of entities within a fact. Therefore, HINGE (Rosso et al., 2020) and NeuInfer (Guan et al., 2020) represent facts as a primary triple alongside auxiliary role-entity pairs, capturing entity importance more effectively. HINGE uses convolutional neural networks to encode the two components, while NeuInfer relies on fully connected layers. StarE (Galkin et al., 2020) applies GNNs to propagate information between auxiliary role-entity pairs and the primary triples, enhancing capturing the relationship between entities. GRAN (Wang et al., 2021) models an n-ary fact as a fully connected graph, capturing the various relationships between entities and roles within the fact. However, these methods only focus on the global or local structures of NKGs. HAHE (Luo et al., 2023) proposes to learn the embeddings of entities and roles from both global and local aspects with a two-stage GNN. Additionally, the above models overlook rich pattern features in NKGs, ShrinkE (Xiong et al., 2023) extends BoxE (Abboud et al., 2020) to capture these patterns. Recently, HyConvE (Wang et al., 2023) leverages 3D convolution to capture complex interactions of elements in n-ary facts. However, the above methods focus on the transductive setting and overlook unseen entities and roles that appear in emerging NKGs.

3.2 Related Work of ILP

ILP aims to predict missing elements in facts involving unseen elements in KGs. Existing ILP methods can be divided into two categories: GNN-based methods and text-based methods. GNN-based methods, like GraIL (Teru et al., 2020), INDIGO (Liu et al., 2021), and RMPI (Geng et al., 2023), use GNNs to encode nonlinear structures and generate representations for unseen elements from their neighbors. Text-based methods, such as KG-BERT (Yao et al., 2019) and SimKGC (Wang et al., 2022), encode textual descriptions to represent unseen elements. More recent approaches like MarKE (Chen et al., 2022) and INGRAM (Lee et al., 2023) employ GNNs with meta-learning mechanisms to produce embeddings for unseen elements and predict missing elements in facts involving them.

However, the above methods focus on triple facts and overlook n-ary facts. To our knowledge, only one work predicts missing elements in n-ary facts involving unseen elements (Ali et al., 2021). It utilizes neighbor information and additional textual descriptions of entities to generate embeddings for unseen elements. However, it restricts that unseen elements can only be subject or object entities, and it relies on external textual descriptions, which are often unavailable. Consequently, predicting missing elements in facts involving unseen roles and entities in NKGs remains a significant challenge.

4 Methodology

This section presents the proposed MetaNIR model. Before detailing its architecture, we first introduce the meta-learning process underlying MetaNIR.

4.1 Meta-learning Setting

Inspired by the concept of "learning to learn" in meta-learning (Santoro et al., 2016), we formulate a set of tasks comprising support facts and query facts on the training NKG with simulated unseen entities and roles to mimic the inductive setting. In this way, we can learn a model on these tasks to achieve "learning to generate unseen entity and role embeddings".

Each task $\mathcal{S}^i = (\mathcal{E}^i, \mathcal{R}^i, \mathcal{T}^i_{sup}, \mathcal{T}^i_{que})$ corresponds to a sub-NKG sampled from the training NKG $\mathcal{G}^{tr}$, where $\mathcal{E}^i$, $\mathcal{R}^i$, $\mathcal{T}^i_{sup}$, and $\mathcal{T}^i_{que}$ represent the entity set, role set, support facts and query facts for that task, respectively. Although $\mathcal{E}^i, \mathcal{R}^i$ are sampled from $\mathcal{E}^{tr}, \mathcal{R}^{tr}$, we relabel some entities and roles and treat them as unseen to simulate the inductive setting. Here, $\mathcal{E}^{tr}$ and $\mathcal{R}^{tr}$ denote the entity set and role set in $\mathcal{G}^{tr}$, respectively. Each task $\mathcal{S}_i$ is formally defined as follows:

$$\mathcal{S}^i = (\mathcal{E}^i = (\hat{\mathcal{E}}^i, \tilde{\mathcal{E}}^i), \mathcal{R}^i = (\hat{\mathcal{R}}^i, \tilde{\mathcal{R}}^i), \mathcal{T}^i_{sup}, \mathcal{T}^i_{que}), \tag{1}$$

where $\hat{\mathcal{E}}^i \in \mathcal{E}^{tr}$ are seen entities, and $\tilde{\mathcal{E}}^i \notin \mathcal{E}^{tr}$ are unseen entities; $\hat{\mathcal{R}}^i \in \mathcal{R}^{tr}$ are seen roles, and $\tilde{\mathcal{R}}^i \notin \mathcal{R}^{tr}$ are unseen roles. The meta-training goal is learning to embed both seen and unseen entities and roles using supporting facts to maximize the plausibility scores of query facts within meta-learning tasks $\mathcal{S}$ sampled from $\mathcal{G}^{tr}$ as follows:

$$\max_{\theta} \mathbb{E}_{\mathcal{S}^i \sim p(\mathcal{S})} \left[\sum_{t \in \mathcal{T}^i_{\text{que}}} \frac{1}{|\mathcal{T}^i_{\text{que}}|} \mathcal{M}_\theta \left(t \mid \mathcal{T}^i_{\text{sup}} \right) \right], \tag{2}$$

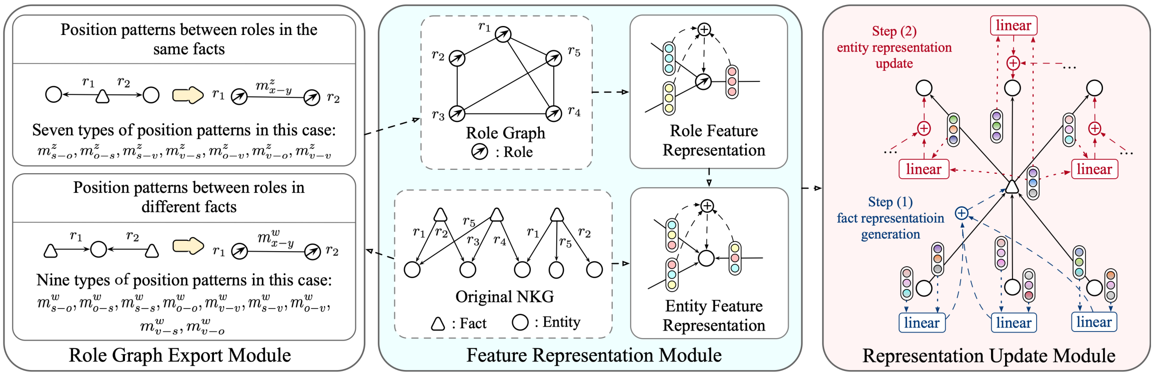

Figure 2: The overview of the proposed MetaNIR model.

where t denotes a query fact in $\mathcal{T}^i_{que}$; $\mathcal{M}$ is a model that calculates the plausibility score of the query fact based on the support facts.

4.2 The Architecture of MetaNIR

Since no existing model $\mathcal{M}$ is suitable for Equation 2, the MetaNIR model is proposed to embed unseen entities and roles using support facts from each sampled task.

MetaNIR is a GNN-based framework that utilizes neighbor information in support facts to generate representations for unseen elements. It comprises three modules: the role graph export module, the feature representation module, and the representation update module, as illustrated in Figure 2. The role graph export module exports a role graph, which expresses the patterns between roles. Then, the feature representation module leverages these patterns to generate feature representations for unseen roles and entities. Next, the representation update module employs a role-aware hierarchical GNN to update representations for both unseen and seen elements. These updated representations are subsequently used to predict missing elements in query facts. For clarity, in the following, we illustrate the entire process through a single task $\mathcal{S}_i$, as defined in the Subsection 4.1.

4.2.1 Role Graph Export Module

Inspired by the transferability of features such as node degree and subgraph structures in graph theory, we observe that there are relative position patterns between roles in NKGs. These position patterns can be represented by the set M=$\{m^j_k\}$, where $j \in \{z, w\}$, $k \in$$\{s-s,\ s-o,\ s-v,...\}$, as illustrated in Figure 2. The superscripts z or w indicate whether roles appear in the same fact or different facts, while the subscripts s, o, v represent the subject, object, and other entities, respectively; s, o, v are connected by $-$, representing the pattern between the two roles.

For instance, as the fact example in Section 2, roles '$time$' and '$winner$' link an other entity and the subject entity in the same fact, respectively. Thus, there is a position pattern m^z_{v-s} between them. More examples of position patterns are provided in Appendix A. These position patterns are universal, independent of specific entities, and transferable across different meta-learning tasks. To utilize them, we export a role graph from the original NKG. In this role graph, each role is represented as a node, and edges denote the position patterns between roles.

4.2.2 Feature Representation Module

This module initializes the feature representations of unseen roles and entities using the obtained position patterns. Specifically, unseen roles are initialized based on their adjacent position patterns as follows:

$$\mathbf{r} = \frac{1}{|N_p(r)|} \sum_{m \in N_p(r)} \mathbf{m}, \tag{3}$$

Where $\mathbf{r}$ represents the feature representation of the unseen role r; $N_p(r)$ is the set of in-going position patterns of r in the role graph; $\mathbf{m}$ is the embedding of position pattern m, which is randomly initialized and updated during the meta-learning process.

After obtaining the feature representations of unseen roles, the feature representations of unseen entities are initialized in a similar manner. Specifically, we generate the feature representations of unseen entities using their corresponding roles in $\mathcal{T}^i_{\text{sup}}$ as follows:

$$\mathbf{e} = \frac{1}{|N_r(e)|} \sum_{r \in N_r(e)} \mathbf{r}, \tag{4}$$

where $\mathbf{e}$ represents the feature representation of the unseen entity e; $N_r(e)$ denotes the set of roles that e plays in the associated facts in $\mathcal{T}^i_{\text{sup}}$.

In this process, the transferred position patterns enrich unseen role feature representations and unseen entity feature representations with reasonable semantic information.

4.2.3 Representation Update Module

After obtaining the feature representations of unseen entities and roles, a role-aware hierarchical GNN is proposed to update representations of both seen and unseen elements. This GNN operates in two steps: fact representation generation and entity representation update.

In the fact representation generation step, fact representations are generated by aggregating the representations of the entities involved as follows:

$$\mathbf{h}^l = \sigma\Big(\frac{1}{|\mathbb{E}(t)|}\sum_{\{r,e\}\in\mathbb{E}(h)} \mathbf{W}^l_h \phi(\mathbf{r}^l, \mathbf{e}^l)\Big), \quad (5)$$

where $\mathbf{h}^l$ is the representation of fact h in the l-th layer of the GNN; $\mathbb{E}(h)$ is the set of role-entity pairs in fact h; $\mathbf{W}^l_h$ is a weight matrix; ϕ is a subtraction function that combines the representations of entity e^l and role r^l in the l-th layer; σ is a non-linear activation function.

In the entity representation update step, entity representations are updated by aggregating the representations of the corresponding facts as follows:

$$\mathbf{e}^{l+1} = \sigma\Big(\frac{1}{|\mathbb{F}(e)|}\sum_{\{h,r\}\in\mathbb{F}(e)} \mathbf{W}^l_e \phi(\mathbf{h}^l, \mathbf{r}^l) + \mathbf{W}^l_{\text{self}}\mathbf{e}^l\Big), \quad (6)$$

where $\mathbf{e}^{l+1}$ is the representation of entity e in the $(l+1)$-th layer; $\mathbb{F}(e)$ is the set of fact-role pairs connected to entity e in $\mathcal{T}^i_{\text{sup}}$; $\mathbf{W}^l_e$ and $\mathbf{W}^l_{\text{self}}$ are weight matrixes. Role representations are updated at each layer as follows:

$$\mathbf{r}^{l+1} = \sigma(\mathbf{W}^l_r \mathbf{r}^l), \quad (7)$$

where $\mathbf{r}^{l+1}$ is the representation of role r at the $(l+1)$-th layer; $\mathbf{W}^l_r$ is a learnable parameter.

4.3 Model Training

For each task $\mathcal{S}_i$, after generating entity and role representations from $\mathcal{T}^i_{sup}$, a fact scorer is required to evaluate these representations and calculate the loss of $\mathcal{T}^i_{que}$. Various LPN models can serve as the fact scorer. Here, GRAN (Wang et al., 2021) is selected due to its demonstrated effectiveness in LPN. The scorer randomly masks one element in the query fact and predicts the missing element based on the remaining components as follows:

$$\hat{\mathbf{X}}_{\mathbf{p}} = f(t_{que}), \quad (8)$$

$$\mathbf{P} = Softmax(\hat{\mathbf{X}}_{\mathbf{p}}\mathbf{C}^T), \quad (9)$$

where f denotes the fact scorer; t_{que} represents a masked fact in $\mathcal{T}^i_{que}$; $\hat{\mathbf{X}}_{\mathbf{p}}$ is the predicted representation of the missing element in t_{que}; $\mathbf{C}$ is embeddings of the set of candidate entities or roles; $\mathbf{P}$ is the similarity probability of $\hat{X}_p$ with each candidate.

The final loss is calculated using the similarity between the target of prediction and all candidates as follows:

$$L = \sum_{c=1}^{|C|} y_c \log \mathbf{P}_c, \quad (10)$$

where y_c is the truth label of candidate c; C is the set of candidate entities or roles; $\mathbf{P}_c$ is the similarity probability of candidate c.

The model parameters are then updated to ensure that the ground-truth query facts receive higher scores than negative facts. The full training process of MetaNIR is summarized in Algorithm 1.

Algorithm 1 The training process of MetaNIR.

Input: Training NKG; initial parameters.

```
repeat
    Sample mini-batch tasks S from the training NKG.
    for each task S_i in S do
        Export a role graph from T^i_sup.
        Initial feature representations for unseen roles (Equation 3).
        Initial feature representations for unseen entities (Equation 4).
        Update role and entity representations using a role-aware hierarchical GNN (Equation 5∼Equation 7).
        Calculate the loss of T^i_que (Equation 8∼Equation 10).
    end for
    Update model parameters.
until convergence or maximum iterations reached.
```

5 Data Construction

To evaluate ILPN, three datasets are constructed based on widely used LPN benchmarks:

Dataset	Training NKG				Test NKG						Valid NKG					
	$\mathcal{E}^{tr}$	$\mathcal{R}^{tr}$	$\mathcal{F}^{tr}$	N^{tr}	$\mathcal{E}^{te}$	$\mathcal{R}^{te}$	$\mathcal{F}^{te}_{sup}$	N^{te}_{sup}	$\mathcal{F}^{te}_{que}$	N^{te}_{que}	$\mathcal{E}^{va}$	$\mathcal{R}^{va}$	$\mathcal{F}^{va}_{sup}$	N^{va}_{sup}	$\mathcal{F}^{va}_{que}$	N^{va}_{que}
JF-Ext	618	92	3305	54.0%	1854(1752)	359(296)	2708	29.0%	1283	21.0%	1791(1705)	305(257)	2304	27.0%	1061	30.0%
WIKI-Ext	835	91	3905	2.1%	1576(1489)	146(75)	2974	6.2%	4733	2.9%	1573(1460)	132(59)	1906	7.2%	6480	2.5%
WD-Ext	1042	178	5112	1.0%	1183(1100)	340(192)	1802	4.8%	3053	2.0%	1191(1076)	315(156)	1580	7.6%	2610	3.6%

Table 1: Statistics of the constructed datasets. The number in the bracket denotes the number of entities or roles that do not appear in corresponding training NKG (i.e., unseen entities or roles).

JF17K (Wen et al., 2016), WikiPeople (Guan et al., 2019), and WD50K (Galkin et al., 2020), naming them JF-Ext, WIKI-Ext, and WD-Ext, respectively. In these datasets, the unseen elements can be any entity (subject, object, or other entities of the fact) or role, and in some cases, unseen entities and roles may appear simultaneously.

For each dataset, a training NKG $\mathcal{G}^{tr}$, a test NKG $\mathcal{G}^{te}$, and a valid NKG $\mathcal{G}^{va}$ are created, all sampled from the original benchmark $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$, where $\mathcal{E}, \mathcal{R}, \mathcal{F}$ represent the sets of entities, roles, and facts therein, respectively. A part of entities and roles in $\mathcal{G}^{ts}$ and $\mathcal{G}^{va}$ are intentionally unseen in $\mathcal{G}^{tr}$. New ILPN datasets are constructed as follows:

- Sample a set of entities $\mathcal{E}_1$ from $\mathcal{E}$, and perform a random walk of length l_1 on $\mathcal{G}$ to obtain an extended entity set $\mathcal{E}_1'$.
- Extract facts consisting entities $\in \mathcal{E}_1'$ from $\mathcal{F}$ to form $\mathcal{G}^{te}$, and remove these facts from $\mathcal{F}$.
- Delete a subset of entities and roles from $\mathcal{E}$ and $\mathcal{R}$, respectively, based on a ratio of α to ensure that some elements are unseen in $\mathcal{G}^{tr}$.
- Repeat the above process to create $\mathcal{G}^{va}$ similar to $\mathcal{G}^{te}$.
- Sample a set of entities $\mathcal{E}_2$ from the remaining $\mathcal{E}$, and perform a random walk of length l_2 on $\mathcal{G}$ to obtain an extended entity set $\mathcal{E}_2'$.
- Extract facts consisting entity $\in \mathcal{E}_2'$ from $\mathcal{F}$ to form $\mathcal{G}^{tr}$.

Table ?? summarizes the statistics for the constructed datasets, with detailed sampling parameters provided in Appendix B. In Table ??, $\mathcal{F}^{tr}$ represents the number of facts in $\mathcal{G}^{tr}$, while N^{tr} indicates the proportion of n-ary facts. For the test NKG, $\mathcal{F}^{te}_{sup}$ and N^{te}_{sup} denote the number of support facts and the proportion of n-ary facts within those support facts, respectively, whereas $\mathcal{F}^{te}_{que}$ and N^{te}_{que} denote the number of query facts and the proportion of n-ary facts within those query facts. The valid NKG has similar statistics to the test NKG.

6 Experiments

6.1 Experimental Settings

Baselines. Since no existing model is specifically designed for ILPN, MetaNIR is primarily compared with LPN and ILP models:

1) LPN models: Several representative or well-performing LPN models are selected, including NeuInfer (Guan et al., 2020), StarE (Galkin et al., 2020), GRAN (Wang et al., 2021), ShrinkE (Shomer et al., 2023), HyConvE (Wang et al., 2023), and HAHE (Luo et al., 2023). Since these models are designed for the transductive setting, we adapt them by using our feature representation module to generate embeddings for unseen entities and roles.

2) ILP models: Since textual descriptions are often unavailable, and for fairness, text-based models are not used as baselines. Instead, we compare with the latest GNN-based models, MarKE (Chen et al., 2022) and INGRAM (Lee et al., 2023), which can predict missing elements in facts involving unseen entities and roles simultaneously. To adapt these models to NKGs, virtual entities are introduced to convert n-ary facts into multiple binary facts. The embeddings for virtual entities are generated by projecting the embeddings of the original entities and roles through a fully connected layer.

Implementation Details. Hyper-parameters of MetaNIR are selected within the following ranges: The dimension of element embeddings $\in \{128, 256, 512, 1024\}$, learning rate $\in \{5e-3, 1e-3, 5e-4, 1e-4\}$, batch size $\in \{128, 256, 512, 1024\}$, number of GNN layers $\in \{1, 2, 3, 4\}$. The ratio of the unseen elements during training is 30%∼80%. To ensure a fair comparison, the hyper-parameters of all baselines are finetuned on each experimental dataset. Before meta-training our model, we sampled 10,000 tasks on the training NKG for each dataset. Details on task sampling are provided in Appendix C. The datasets and source code are available at https://github.com/JiyaoWei/MetaNIR.

Evaluation Metrics. We evaluate model perfor-

Model	JF-Ext				WIKI-Ext				WD-Ext			
	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1
NeuInfer	0.088	19.7	9.6	1.8	0.182	22.9	20.1	12.3	0.072	17.3	6.3	0.3
HINGE	0.193	31.6	21.2	12.2	0.071	15.8	6.3	1.1	0.076	15.9	6.7	0.8
StarE	0.181	26.0	19.8	11.4	0.204	27.5	21.4	12.6	0.079	13.1	8.2	2.1
GRAN	0.215	33.6	27.2	12.7	0.177	22.2	18.6	12.4	0.066	12.7	4.6	0.6
HAHE	0.238	34.1	26.2	16.7	0.197	30.4	20.9	12.7	0.087	21.7	8.9	1.3
ShrinkE	0.210	32.8	22.7	13.8	0.249	41.0	28.8	15.8	0.146	28.9	19.2	9.6
HyConvE	0.282	48.8	38.1	16.5	0.156	34.2	20.0	5.9	0.088	21.3	9.0	1.5
MaKEr	0.168	29.2	24.6	7.9	0.170	25.1	20.3	10.5	0.097	27.2	10.3	2.5
INGRAM	0.474	74.0	63.2	33.5	0.354	52.0	43.0	26.1	0.265	50.8	32.8	16.3
MetaNIR	**0.716**	**90.3**	**80.5**	**63.4**	**0.591**	**90.4**	**74.3**	**45.5**	**0.582**	**90.1**	**77.6**	**43.3**

Table 2: The experimental results on all datasets in terms of MRR and Hits@k (%).

Model	u_ent				u_rel				u_both			
	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1
ShrinkE	0.162	32.1	21.8	5.8	0.144	22.0	20.0	6.0	0.094	18.9	11.3	1.8
HyConvE	0.092	23.3	10.8	1.4	0.066	30.0	0.0	0.0	0.092	24.5	11.4	1.27
INGRAM	0.270	50.0	32.5	17.3	0.167	32.6	13.0	8.7	0.264	55.9	34.9	14.2
MetaNIR	**0.633**	**95.6**	**81.4**	**49.3**	**0.283**	**65.2**	**34.7**	**17.3**	**0.549**	**89.5**	**73.2**	**40.2**

Table 3: The experimental Results on u_ent, u_rel, and u_both of WD-Ext in terms of MRR and Hits@k (%).

mance using Mean Reciprocal Rank (MRR) and Hits@k, where $k \in \{1, 5, 10\}$, following Ali et al. (2021). Hits@k is the proportion of correct answers ranked within the top k, while MRR is the average of the reciprocal rank of the correct answers. Higher MRR and Hits@k values indicate better performance. For a fair comparison with baselines, following (Chen et al., 2022; Liu et al., 2021), each query fact is ranked among 50 randomly sampled negative candidates, and the results are averaged over five runs. Since entity prediction is more critical and challenging than role prediction, our experiments primarily focus on entity prediction. MetaNIR also excels in role prediction, as detailed in Appendix D.

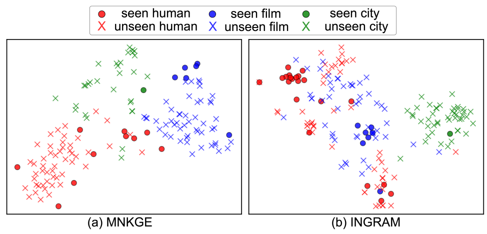

Figure 3: Visualization for t-SNE embeddings (Van der Maaten and Hinton, 2008) of MetaNIR and INGRAM.

6.2 Experimental Results and Analysis

Table 2 displays the experimental results across all datasets, illustrating that MetaNIR consistently surpasses all existing baselines. For example, MetaNIR shows improvements in MRR of 24.2%, 23.7%, and 31.7% over the best baseline on JF-Ext (51.0% relative improvement), WIKI-Ext (93.3%), and WD-Ext (119.6%), respectively. These results highlight MetaNIR's effectiveness in ILPN. Existing LPN methods often fall short because they initialize embeddings for unseen elements in a fixed, simplistic manner and do not refine them through training. ILP methods, which convert n-ary facts into multiple binary facts using virtual entities, suffer from sparsity in KGs, limiting their performance. In contrast, MetaNIR uses meta-learning mechanisms to adaptively generate embeddings for unseen elements and directly encode n-ary facts without relying on virtual entities, enhancing its ability to handle ILPN.

Furthermore, MetaNIR is evaluated on different query types: those involving only unseen entities (u_ent), only unseen roles (u_rel), and both unseen entities and roles (u_both). Table **??** presents the results for MetaNIR and the optimal baselines on the largest dataset, WD-Ext. MetaNIR outperforms all baselines across these query types, demonstrating its robustness in diverse query scenarios.

6.3 Ablation Studies

The key components of MetaNIR include the meta-learning mechanism, feature representation module, and representation update module. The feature representation module contains role feature representation and entity feature representation. To assess the effectiveness of each component, ablation studies are conducted, with results shown in Table 4. Each component contributes positively to the performance of MetaNIR, highlighting its effec-

Model	JF-Ext				WIKI-Ext				WD-Ext			
	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1
MetaNIR	**0.716**	**90.3**	**80.5**	**63.4**	**0.591**	**90.4**	**74.3**	**45.5**	**0.582**	**90.1**	**77.6**	**43.3**
-Meta-learning	0.556	74.8	64.8	45.51	0.433	73.0	57.1	30.7	0.449	74.2	58.1	32.5
-Role feature	0.545	74.4	66.2	43.2	0.514	82.3	65.5	38.5	0.456	76.1	57.7	33.7
-Entity feature	0.535	84.2	70.7	38.7	0.557	88.4	71.6	41.5	0.512	87.0	69.4	36.1
-Update	0.151	38.9	21.1	5.1	0.121	33.4	16.6	3.3	0.166	32.9	19.2	8.8

Table 4: The ablation experimental results on all datasets in terms of MRR and Hits@k (%).

Seen roles	Unseen roles	Similarities
cast member	performer	0.417
	tributary	-0.164
country	location	0.455
	drug or therapy used for treatment	-0.107

Table 5: Cosine similarities between representations of seen and unseen roles generated by MetaNIR.

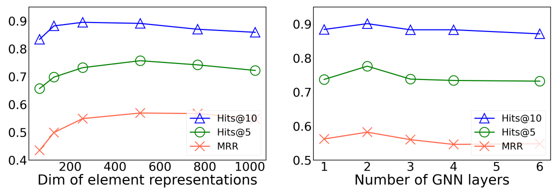

Figure 4: Impact of the dim of element representations and number of GNN layers on WD-Ext.

tiveness. Notably, performance drops significantly when the representation update module is removed, as it enables full interaction between unseen and seen elements. The feature representation module is also crucial, as position patterns play a key role in initializing unseen elements. Furthermore, omitting the meta-learning mechanism markedly reduces performance, demonstrating its importance in enabling the model to "learn to learn" and enhance overall performance.

6.4 Case Study

We analyze the embeddings of unseen entities and roles generated by MetaNIR to demonstrate its effectiveness in ILPN.

For unseen entities, we visualized the embeddings generated by MetaNIR and the best baseline INGRAM on WD-Ext in Figure 3. In this figure, three types of entities are distinguished by different colors, with $\circ$ representing seen entities and $\times$ representing unseen entities. Compared to INGRAM, the embeddings of MetaNIR align more consistently with their respective type. INGRAM's embeddings for file and human entities are mixed, while MetaNIR separates them into different clusters more clearly. Additionally, MetaNIR clusters the embeddings of unseen entities with the same type of seen entities, indicating that MetaNIR captures meaningful semantics and knowledge in its embeddings for unseen entities.

For unseen roles, we randomly selected two seen roles from WD-Ext and used cosine similarity to identify their most similar and dissimilar unseen roles, highlighted in red and blue in Table ??, respectively. The most similar unseen roles align semantically with the seen roles, and they can connect to entities of the same type, while the most dissimilar roles have no semantic relationship with seen roles. These results confirm that MetaNIR's role embeddings are both semantically accurate and reliable.

6.5 Analyses on Key Hyper-parameters

Figure 4 illustrates the impact of two key hyper-parameters—element representation dimension and GNN layer number—on MetaNIR's performance. The performance of the model first increases and then decreases as the embedding dimension increases, and the best performance is achieved under 512-dimensional embeddings. This is because when the dimension is too low, the model's expressive power is limited, and when the dimension is too high, it may cause the model to overfit. For the GNN layers, the best performance is achieved with a 2-layer network, demonstrating that considering interactions between closer neighbors effectively generates accurate element representations.

7 Conclusion

In this paper, we proposed a new task, ILPN, which aims to predict missing elements in facts involving unseen entities and roles using some support facts in emerging NKGs. To address this task, we proposed the MetaNIR model, which adaptively generates embeddings for unseen elements with meta-learning mechanisms. Since no dataset supports ILPN, we carefully constructed three datasets for evaluation. Extensive experimental results demonstrated the effectiveness of MetaNIR for ILPN.

8 Limitations

This paper focuses on NKGs that incorporate new facts with unseen entities and roles. Real-world NKGs not only emerge new facts but also remove incorrect facts. Additionally, the proposed model cannot utilize multi-modal information to enhance model performance. Future work will address these limitations.

Acknowledgments

The work is supported by the Strategic Priority Research Program of the CAS under Grants No. XDB0680102, the SMP-IDATA Open Youth Fund, and the Lenovo-CAS Joint Lab Youth Scientist Project under Grants No. JCKY2022130C039. We thank anonymous reviewers for their insightful comments and suggestions.

References

Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. Boxe: A box embedding model for knowledge base completion. *Advances in Neural Information Processing Systems*, 33:9649–9661.

Mehdi Ali, Max Berrendorf, Mikhail Galkin, Veronika Thost, Tengfei Ma, Volker Tresp, and Jens Lehmann. 2021. Improving inductive link prediction using hyper-relational facts. In *The Semantic Web–ISWC 2021: 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24–28, 2021, Proceedings 20*, pages 74–92. Springer.

Ivana Balažević, Carl Allen, and Timothy Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5185–5194.

Mingyang Chen, Wen Zhang, Zhen Yao, Xiangnan Chen, Mengxiao Ding, Fei Huang, and Huajun Chen. 2022. Meta-learning based knowledge extrapolation for knowledge graphs in the federated setting. *arXiv preprint arXiv:2205.04692*.

Shimin Di, Quanming Yao, and Lei Chen. 2021. Searching to sparsify tensor decomposition for n-ary relational data. In *Proceedings of the Web Conference 2021*, pages 4043–4054.

Bahare Fatemi, Perouz Taslakian, David Vazquez, and David Poole. 2021. Knowledge hypergraphs: prediction beyond binary relations. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 2191–2197.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. 2020. Message passing for hyper-relational knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7346–7359.

Yuxia Geng, Jiaoyan Chen, Jeff Z Pan, Mingyang Chen, Song Jiang, Wen Zhang, and Huajun Chen. 2023. Relational message passing for fully inductive knowledge graph completion. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1221–1233. IEEE.

Saiping Guan, Xiaolong Jin, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. 2020. Neuinfer: Knowledge inference on n-ary facts. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 6141–6151.

Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2019. Link prediction on n-ary relational data. In *The world wide web conference*, pages 583–593.

Zhongni Hou, Xiaolong Jin, Zixuan Li, Long Bai, Saiping Guan, Yutao Zeng, Jiafeng Guo, and Xueqi Cheng. 2023. Temporal knowledge graph reasoning based on n-tuple modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1090–1100.

Yejin Kim, Scott Rome, Kevin Foley, Mayur Nankani, Rimon Melamed, Javier Morales, Abhay K Yadav, Maria Peifer, Sardar Hamidian, and H Howie Huang. 2024. Improving content recommendation: Knowledge graph-based semantic contrastive learning for diversity and cold-start users. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8743–8755.

Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang. 2023. Ingram: Inductive knowledge graph embedding via relation graphs. In *International Conference on Machine Learning*, pages 18796–18809. PMLR.

Mingchen Li and Shihao Ji. 2022. Semantic structure based query graph prediction for question answering over knowledge graph. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1569–1579.

Shuwen Liu, Bernardo Grau, Ian Horrocks, and Egor Kostylev. 2021. Indigo: Gnn-based inductive knowledge graph completion using pair-wise encoding. *Advances in Neural Information Processing Systems*, 34:2034–2045.

Yu Liu, Quanming Yao, and Yong Li. 2020. Generalizing tensor decomposition for n-ary relational knowledge bases. In *Proceedings of the web conference 2020*, pages 1104–1114.

Haoran Luo, E Haihong, Yuhao Yang, Yikai Guo, Mingzhi Sun, Tianyu Yao, Zichen Tang, Kaiyang Wan, Meina Song, and Wei Lin. 2023. Hahe: Hierarchical attention for hyper-relational knowledge graphs in global and local level. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8095–8107.

Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. 2023. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, 56(11):13071–13102.

Zile Qiao, Wei Ye, Tong Zhang, Tong Mo, Weiping Li, and Shikun Zhang. 2022. Exploiting hybrid semantics of relation paths for multi-hop question answering over knowledge graphs. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1813–1822.

Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. 2020. Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In *Proceedings of the web conference 2020*, pages 1885–1896.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR.

Harry Shomer, Wei Jin, Juanhui Li, Yao Ma, and Hui Liu. 2023. Learning representations for hyper-relational knowledge graphs. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*, pages 253–257.

Komal Teru, Etienne Denis, and Will Hamilton. 2020. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*, pages 9448–9457. PMLR.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Chenxu Wang, Xin Wang, Zhao Li, Zirui Chen, and Jianxin Li. 2023. Hyconve: A novel embedding model for knowledge hypergraph link prediction with convolutional neural networks. In *Proceedings of the ACM Web Conference 2023*, pages 188–198.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4281–4294.

Quan Wang, Haifeng Wang, Yajuan Lyu, and Yong Zhu. 2021. Link prediction on n-ary relational facts: A graph-based approach. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 396–407.

Jiyao Wei, Saiping Guan, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2024. Few-shot link prediction on hyper-relational facts. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7196–7207.

Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. 2016. On the representation and embedding of knowledge bases beyond binary relations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 1300–1307.

Bo Xiong, Mojtaba Nayyeri, Shirui Pan, and Steffen Staab. 2023. Shrinking embeddings for hyper-relational knowledge graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13306–13320.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Donghan Yu and Yiming Yang. 2021. Improving hyper-relational knowledge graph completion. *arXiv preprint arXiv:2104.08167*.

Richong Zhang, Junpeng Li, Jiajie Mei, and Yongyi Mao. 2018. Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In *Proceedings of the 2018 world wide web conference*, pages 1185–1194.

A Position Pattern Examples

Figure 5 shows some examples of mining position patterns from the original NKG.

B Details of Dataset Construction

The detailed sampling parameters during the dataset construction process are shown in Table 6.

Parameters	JF-Ext	WIKI-Ext	WD-Ext
$\|\mathcal{E}_1\|$	200	100	100
$\|\mathcal{E}_2\|$	100	100	100
l_1	15	20	15
l_2	15	10	15
α	0.1	0.1	0.1

Table 6: Parameters of dataset sampling for three datasets.

Additionally, we provide an example to illustrate the dataset construction process. Consider an NKG $\mathcal{G}$ with the following components:

- Entities ($\mathcal{E}$): $\{Matt\ Damon$, $Best\ Actor$, $Good\ Will\ Hunting$, $Gus\ Van\ Sant$, $Matt\ Damon$, $Milk$, $SeanPenn\}$
- Roles ($\mathcal{R}$): $\{winner$, $award$, $movie$, $nominee$, $actor$, $director\}$
- Facts ($\mathcal{F}$): {Fact 1: $((nominee : Matt\ Damon)$, $(award : Best\ Actor)$, $\{(movie : Good\ Will\ Hunting)\})$, Fact 2: $((movie : Good\ Will\ Hunting)$,$(actor : Matt\ Damon))$, Fact 3: $((movie$: $Good\ Will\ Hunting)$,$(director : Gus\ Van\ Sant))$, Fact 4: $((nominee : Sean\ Penn)$, $(award : Best\ Actor)$, $\{(movie : Milk)\})\}$

To construct the test set $\mathcal{G}^{te}$, we first randomly sample a set of entities $\mathcal{E}_1 = \{Milk\}$, from $\mathcal{E}$. A random walk of length $l_1 = 1$ on the graph $\mathcal{G}$ generates the expanded entity set $\mathcal{E}_1'$ = $Best\ Actor$, $Sean\ Penn$, $Milk$. The corresponding facts from $\mathcal{F}$, consisting of entities $\in \mathcal{E}_1'$, are extracted to form the test set $\mathcal{G}^{te}$ = {Fact 4: $((nominee : Sean\ Penn)$, $(award : Best\ Actor)$, $\{(movie : Milk)\})\}$. These facts are then removed from $\mathcal{F}$. To ensure some elements are unseen during training, we delete a subset of entities and roles from $\mathcal{E}$ and $\mathcal{R}$, respectively. In this example, the entity $Matt\ Damon$ and the role $nominee$ are deleted.

The validation set $\mathcal{G}^{va}$ is constructed similarly to the test set.

For the training set $\mathcal{G}^{tr}$, we randomly sample a set of entities $\mathcal{E}_2 = \{Good\ Will\ Hunting\}$, from the remaining $\mathcal{E}$. A random walk of length $l_2 = 1$ on the graph $\mathcal{G}$ results in the expanded entity set $\mathcal{E}_2' = \{Good\ Will\ Hunting, Gus\ Van\ Sant\}$. Notably, $Matt\ Damon$ cannot be included, as it was deleted in the previous step, and $Best\ Actor$ cannot be included due to to the deletion of the nominee role in Fact 1. Finally, the facts containing entities in $\mathcal{E}_2'$ are extracted from $\mathcal{F}$ to form the training set $\mathcal{G}^{tr}$ = {Fact 3: $((movie$: $Good\ Will\ Hunting)$,$(director : Gus\ Van\ Sant))\}$.

C Sampling Meta-Learning Tasks

The sampling process of the meta-learning task is as follows: 1) First, a mini-batch of facts is randomly selected from the training NKG as query facts $\mathcal{T}_{que}^i$; 2) Then, we build the entity set and role set of the query facts; 3) We sample d instances for each entity and role from the training NKG as support facts $\mathcal{T}_{sup}^i$, where $d \in \{5, 10, 15, 30\}$. Note that the query facts are not allowed to appear in the support facts.

D Role Prediction Experiments

Table 7 presents the role prediction results on all datasets. MetaNIR achieves the best results on all datasets, demonstrating the effectiveness of MetaNIR in role prediction. Additionally, baseline methods are not stable in role prediction, while MetaNIR performs well in all datasets. The ablation results are shown in Table 8. These results further demonstrate the effectiveness of each module in MetaNIR.

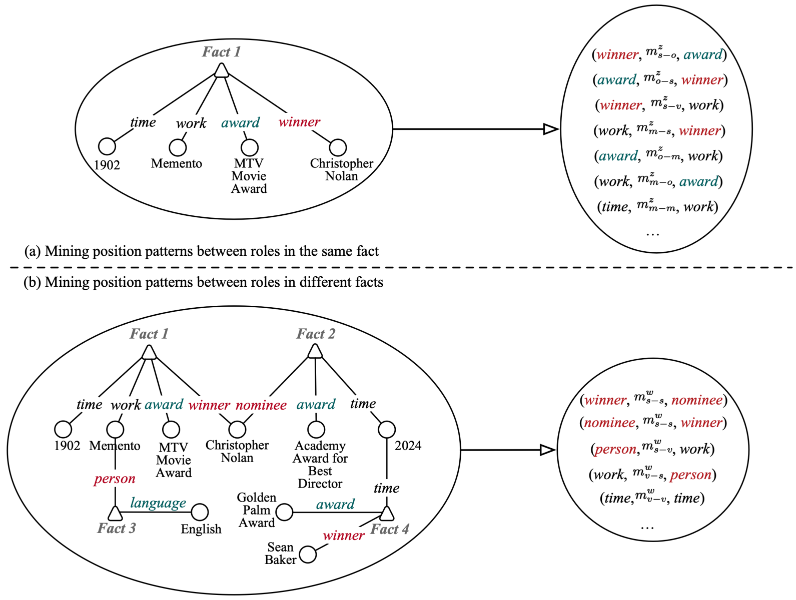

Figure 5: Examples of obtaining position patterns from the original NKG. To distinguish different entities, we use red to mark the role corresponding to the subject entity, green to mark the role corresponding to the object entity, and black to mark the role corresponding to other entities.

Model	JF-Ext				WIKI-Ext				WD-Ext			
	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1
NeuInfer	0.149	24.3	15.4	8.2	0.346	35.0	33.5	31.7	0.130	34.8	17.3	3.5
HINGE	0.180	38.0	29.0	7.0	0.123	25.4	16.4	5.0	0.116	33.8	19.0	1.9
StarE	0.274	63.3	41.7	13.0	0.096	24.5	11.7	2.4	0.191	43.5	25.5	8.6
GRAN	0.542	70.1	60.6	47.0	0.508	71.3	58.9	42.1	0.103	25.7	13.2	2.4
HAHE	0.140	33.1	17.0	4.7	0.502	70.1	61.6	39.8	0.082	20.6	10.5	1.2
ShrinkE	0.218	20.5	20.3	19.6	0.346	35.0	33.5	31.7	0.064	6.7	2.6	1.5
HyConvE	0.462	70.5	58.9	34.0	0.191	42.4	26.4	7.82	0.079	20.4	8.8	0.8
MaKEr	0.126	13.0	9.8	8.3	0.045	3.3	1.0	0.3	0.071	7.5	4.7	1.8
INGRAM	0.487	74.5	64.4	35.2	0.370	59.3	47.5	25.9	0.277	55.8	39.4	15.1
MetaNIR	**0.597**	**89.0**	**70.4**	**49.5**	**0.857**	**97.3**	**94.7**	**78.0**	**0.650**	**86.8**	**73.9**	**55.9**

Table 7: Role prediction results on all datasets in terms of MRR and Hits@k (%).

Model	JF-Ext				WIKI-Ext				WD-Ext			
	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1	MRR	Hits@10	Hits@5	Hits@1
MetaNIR	**0.597**	**89.0**	**70.4**	**49.5**	**0.857**	**97.3**	**94.7**	**78.0**	**0.650**	**86.8**	**73.9**	**55.9**
-Meta-learning	0.505	62.6	54.4	44.4	0.308	73.1	40.5	16.7	0.506	68.5	57.6	41.8
-Role feature	0.517	.62.9	55.6	45.6	0.777	87.6	85.2	70.1	0.507	62.4	55.0	44.1
-Entity feature	0.522	79.1	60.5	43.0	0.764	94.2	89.5	66.0	0.532	82.7	66.2	41.3
-Update	0.165	47.1	26.6	4.0	0.113	28.1	13.9	3.14	0.165	47.1	26.6	4.0

Table 8: The ablation role prediction results on all datasets in terms of MRR and Hits@k (%).