Distilling Causal Metaknowledge from Knowledge Graphs

Yuan Meng, Yancheng Dong, Shixuan Liu, Chaohao Yuan, Yue He, Jian Pei, Peng Cui

Abstract

In recent years, the explosive increase of information facilitates the massive knowledge graphs, which in turn increase burden of people to understand and leverage the regularity behind these superficial facts. Therefore, the metaknowledge, defined as the knowledge about knowledge, is proposed to identify complex processes of knowledge production and consumption. Unfortunately, even though the current correlation-based rule mining methods in knowledge graph distill the rule-formed metaknowledge, they can not explain the processes of knowledge production. In this paper, we focus on capturing the metaknowledge with causality which is generally regarded as one of the most promising techniques to reveal the interactions between components in the complex system. To the best of our knowledge, this is the first attempt to interpret the knowledge graph from the causal perspective.

For this purpose, we propose a causal metaknowledge method for link prediction, which achieves entity-level link prediction by discovering concept-level topological causality. Specifically, we first formalize causal metaknowledge as causal rule, following the form of logical rule. Then, we transform the relational data into propositional data to learn the causal relationships between topological structures. And an efficient algorithm for discovering local causal relationships is proposed using the d-seperation criterion. Eventually, the causal rules generated based on the mined relationships are used for link prediction. Both simulation-based and real data-based experiments demonstrate the effectiveness of the proposed approach, especially under the Out-of-Distribution(OoD) settings.

1 Introduction

In the era of information explosion, knowledge graph (KG) is a powerful representation for integrating billions of available relational facts, based on observational low-level knowledge in the world, to encapsulate the rich relationships of entities [17, 45]. Although the massive knowledge can benefit various downstream applications, *e.g.* query answering [42, 23, 4], recommendation systems [41, 40, 22], yet to better understand, exploit, and complete these underlying knowledge, it is necessary to explore the intrinsic principle of the emergence of this factual knowledge. For this purpose, the concept of meta-knowledge is proposed and defined as the *knowledge about knowledge* [6].

Current rule mining methods in the KG literature attempt to mine meta-knowledge, in the form of association rules, via correlation analysis represented by frequency analysis [9, 8, 27]. These association rules can be used for downstream tasks such as knowledge graph completion, and question answering. However, association does not imply causation [1]. Fortunato et al. points out that causality is necessary to identify the fundamental drivers

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

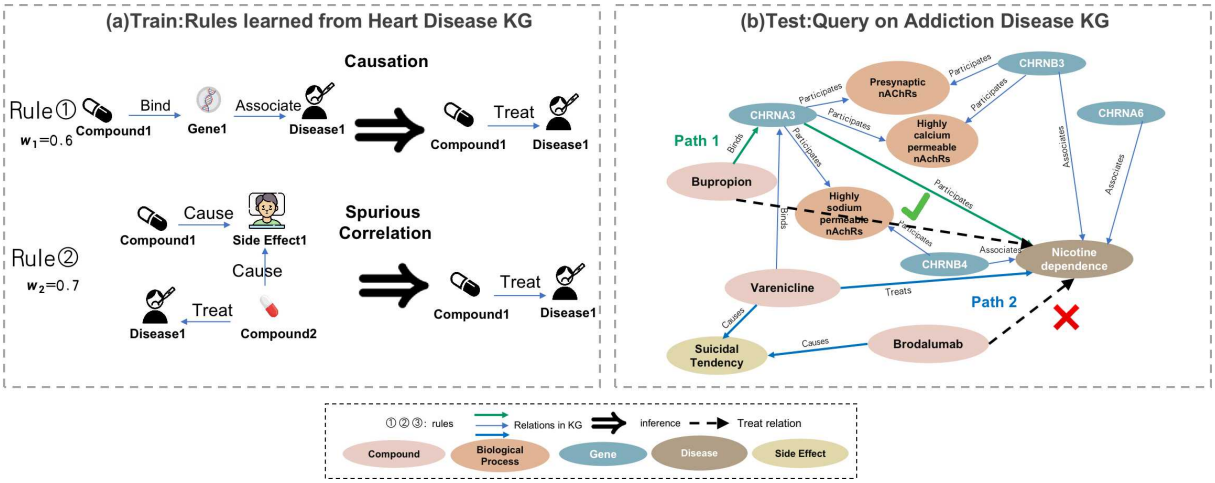

Figure 1: Motivation illustration. Consider a scenario to learn rules for inferring the `Treat` relation between compounds and diseases on a heart disease KG (top), and thereby discover novel drugs for treating nicotine addiction on an addiction disease KG (bottom). On the heart disease KG, drugs that treat the same disease often share the same side effects. The correlation-based approach establishes a strong spurious correlation between the shared side-effect information and the `treat` relation. In contrast, the underlying cause of the `treat` shows a weak association (Rule ①). Therefore, when the above rule is migrated to addiction disease KG (drugs that treat the same basic kind of disease often do not have shared side effects), false prediction could be resulted (e.g., Brodalumab is more likely to be prescribed for nicotine addiction, instead of Bupropion).

of knowledge [7]. The correlation-based method may lead to spurious correlations between the body and head of rule, which can not be generalized to new environments.

Here we take the KG-based drug repurposing task as an example (shown in Fig. 1). Given the heart disease KG as training data, traditional rule mining methods may rely on two rules ① and ② to predict the `Treat` relation. As heart disease drugs entail similar side effects, the confidence (weight) of Rule ②, calculated based on correlation, is greater than that of Rule ① (0.7 versus 0.6). However, the localization of the drug to the target protein produced by the disease gene, as indicated in Rule ①, is the recognized mechanism for physicians to prescribe drugs for the disease [15]. This phenomenon, typical of spurious correlations, is due to the scarcity of genetic information and the abundance of side-effect facts accompanying the data collection process. Therefore, these weighted rules could produce false KG completion results as the environment shifts. Fig. 1(b) visualizes such testing process where the learned rules from heart disease KG are used to answer queries from addiction disease KG. Both nicotine withdrawal drug (Varenicline) and psoriasis drug (Brodalumab) are known to cause the side effect suicidal tendencies. However, the available drug for nicotine withdrawal Bupropion (which binds the gene that participates in nicotine) is not. With the mined rules in Fig. 1(a), physicians could falsely prescribe Brodalumab (Path 2) as a new treatment for nicotine withdrawal, instead of Bupropion (Path 1). If we can learn stable relationships (such as causality) between predicted features and predicted targets, such effect of spurious correlations can be eliminated.

In this paper, we propose a method that learns rules from the causal perspective to ensure strong generalization ability whilst retaining decent interpretability. Specifically, we are concerned with understanding how KG links are generated, through causal discovery. There are two major challenges in this problem: 1) efficiency and 2) proper metrics. For the former, the complex topological structures between massive entity pairs could induce thousand-scale rule space with barely ten relations, posing challenges for both score-based and constraint-based causal discovery approaches. The complexity of the constraint-based technique increases exponentially with the number of nodes, whereas the score-based approach creates an NP-hard problem [19]. For the latter, rule-mining

algorithms generally require specific metrics, such as support rate, as the weights for inference. Therefore, we also need to design a metric to measure the strength of each causal relationship.

In this work, we first formulate causal meta-knowledge with the concept of *causal rule*, on which we further introduce several constraints to reduce the search space. Then, we propose the CMLP (Causal Metaknowledge-based Link Prediction) algorithm, which integrates efficient causal rule discovery approach and causation-based link prediction method. Specifically, we first introduce the concept of rule-induced variable, which uses relation paths to describe the topological structure of entities, and map the graphs into quantified samples with the designed assignment function. Further, we observe that the whole causal structure is not necessary for specific link prediction task, but only the part of the structure related to the predicted relation. Therefore, we design an efficient method based on d-seperation to achieve local causal discovery. Meanwhile, the causal strength based on conditional dependence can also be generated as the weights of learned causal rules. Finally, the predictions can be ranked from weighted causal rules.

Contributions. Our main contributions can be summarized as follows: *(i)* This is the first work that aims at improving link prediction in KG by causal inference to eliminate the effect of spurious correlation, as evidenced in traditional methods. *(ii)* This work introduces CMLP that learns a link predictor based on discovering causal meta-knowledge. *(iii)* CMLP outperforms other competitive baselines on link prediction tasks under Out-of-Distribution (OoD) setting. Furthermore, we analyze the learned meta-knowledge for insights on the mechanism of the applications.

2 Preliminaries and Problem Statement

2.1 Definitions and Notations

In this paper, we follow the definition of knowledge graph as in [17]:

Definition 2.1: A **Knowledge Graph (KG)** is defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{F})$, where $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{F}$ are sets of entities, relations and facts, respectively. Every fact is a triple $(e_h, R, e_t) \in \mathcal{F}$, where $e_h, e_t \in \mathcal{E}$ and $R \in \mathcal{R}$ are head entity, tail entity and the relation between entities, respectively. Without loss of generality, we simultaneously represent a fact as $R(e_h, e_t)$.

First-order logic (FOL) offers a pivotal way to represent real-world knowledge for reasoning. Horn rules, as a special and typical case of FOL rules, propose to represent a target relation by a body of conjunctive relations.

Definition 2.2: A **Horn Rule**, generally chain-like, is given as,

$$R_h(x, y) \leftarrow R_{b_1}(x, z_1) \circ \cdots \circ R_{b_l}(z_{l-1}, y)$$

where, $R_h(x, y)$ signifies the rule head (target relation) that we wishes to reason and $R_{b_1}(x, z_1) \circ \cdots \circ R_{b_l}(z_{l-1}, y)$ is the rule body (relation path). For simplicity, we denote a Horn rule as $R_h : \mathbf{R_b}$, where $\mathbf{R_b} = [R_{b_1}, \cdots, R_{b_l}]$. To reason R_h, the size of the rule space is $|\mathcal{B}_h|$. Every **closed path** of such Horn rule is required to: 1) connect (x, y) via the rule body, which is a sequence of relations $\mathbf{R_b}$, and 2) ensure (x, y) are accessible directly via the target relation R_h. Closed paths are also known as **rule instances**.

2.2 Problem Statement

The goal of this work is to learn the **causal rule** that is formalized as the horn rule. Specifically, the objective of traditional logical rule learning is to assign a plausibility score $\mathbf{S}(R_h|\mathbf{R_b})$ to each rule in the discovered rule space, which can be subsequently aggregated to answer queries about the KG. Currently, plausibility scores are defined over closed paths (e.g., the PCA confidence for AMIE [10]), which are correlational observations.

We have demonstrated that these scores are prone to spurious correlations and therefore result in inaccurate predictions under OoD settings, in Sec. 1. Therefore the other aim is to give a plausibility score based on causal strength.

In this paper, the causal rules are mined for link prediction in KG. We follow the commonly accepted problem definition of link prediction in KG [32, 39]: given an observed KG $\mathcal{G}$ with missing facts, our goal is to predict the correct entity for an given query $(e_h, R_h, ?)$ (or $(?, R_h, e_t)$).

3 Proposed Method: CMLP

In this section, we introduce the proposed approach CMLP which learns causal rules for KG link prediction. CMLP first transforms the relational data into the propositional data to conduct statistical analysis (Sec. 3.1). Then CMLP presents a local causality identification algorithm based on the d-seperation criterion to efficiently mines interpretable causal rules (Sec. 3.2). Finally, a specific causation-based score is applied in predictor to answer the queries with learned causal rules (Sec. 3.3). The pipeline of CMLP is illustrated in Fig. 2.

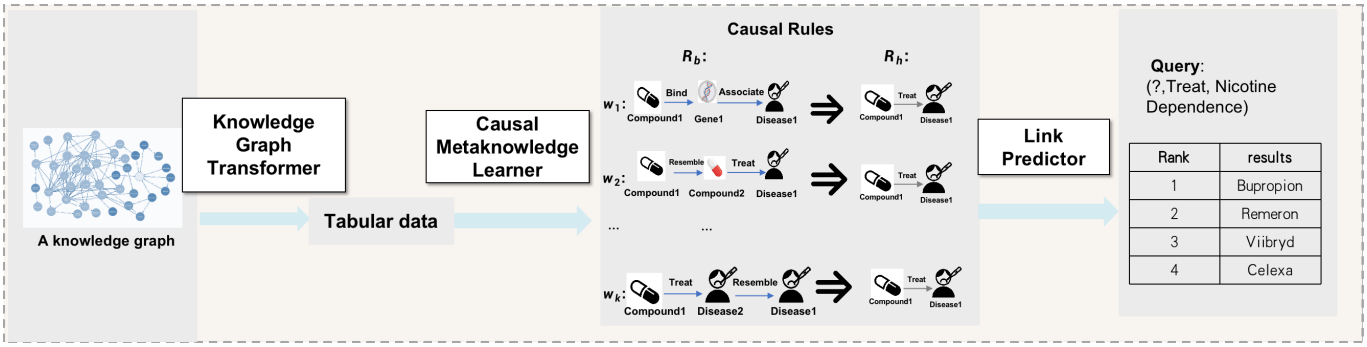

Figure 2: The framework of CMLP. Particularly, CFLP first transforms the relational data into propositional data for better statistical analysis. Then it mines interpretable causal rules, which can be interpreted as a kind of metaknowlege[6]. Finally, a plausibiliy score derived from the causality test is applied in predictor to rank the answers of the given query.

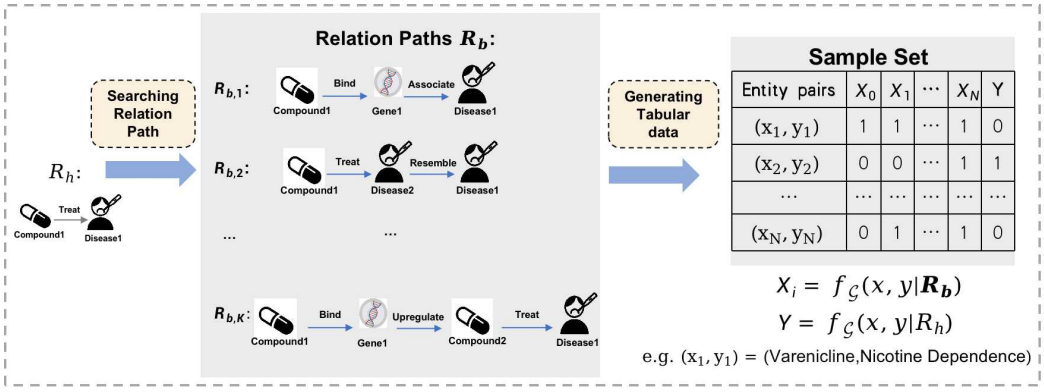

Figure 3: The process of knowledge graph transformer

3.1 Knowledge Graph Transformer

Traditional causal discovery algorithms are defined on propositional data, with well-defined variables and samples, which do not exist in relational data like KGs. Therefore, we give the definition and scope of the variables we study in the causal discovery phase by mapping the potential causes and queried relations into variables. Then we give the practical approach for transforming KG into tabular data, whose horizontal axis are the variables we defined. The process is shown schematically in Fig. 3

3.1.1 Causal variables in KG

The causal rule can be interpreted as a description of causal relationship between the body and the head. Naturally, we formulate variables based on the elements of rules.

Definition 3.1: For entity pair (x, y), its **Rule-induced Variable** $X_k = f_{\mathcal{G}}(x, y|{\mathbf{R_b}}^k)$, where ${\mathbf{R_b}}^k$ corresponds to the rule body in the k-th rule $R_h : {\mathbf{R_b}}^k (k \in \{1, ..., |\mathcal{B}_h|\})$. The assignment function $f_{\mathcal{G}}(\cdot|\mathbf{R_b})$ can be either connectivity feature or path count for $\mathbf{R_b}$ in KG $\mathcal{G}$.

The head of rule also induces a special variable $Y = f_{\mathcal{G}}(x, y|R_h)$. In the real link prediction task, the queries are normally on a specific relation, such as `Treat` in drug repurposing. Therefore, the causal rule mining problem is to discovery the causal relationship between variables $X_k = f_{\mathcal{G}}(x, y|{\mathbf{R_b}}^k), k \in \{1, ..., |\mathcal{B}_h|\}$ and variable $Y = f_{\mathcal{G}}(x, y|R_h)$. Then we introduce the practical approach that we transform the KG into tabular data for causality analysis.

3.1.2 Transforming knowledge graph into propositional data

(1) Step-1: Searching candidate causes X. According to definition 2.2, any rule-induced variable X, which is defined on entity pair(x, y) and seeks to help reasoning over R_h, is a valid candidate cause for $Y = f_{\mathcal{G}}(x, y|R_h)$. So we find all the candidate causes by searching all the paths between entity pairs (x, y), which have the relation R_h between them. There are many well-studied path finding algorithms, which can search the paths under different types of constraints, such as Dijkstra's algorithm [18], A* search [5], best-first search [14], etc. In the experiments, we adopt the best-first search algorithm. Since the number of candidate causes can be the power level of the number of relation types, we require that the length of the path is no more than ℓ, where ℓ is the hyper-parameters. In the experiments of this paper, we set ℓ as 3. (2) Step-2: generating samples. In this paper, we use the connectivity as the assignment function to get quantitative samples.

Definition 3.2: the **binary assignment function of rule-induced variable** is as following:

$$f_{\mathcal{G}}(e_h, e_t|{\mathbf{R_b}}^k) = \mathbb{1}_{\text{con}}(e_h, e_t|{\mathbf{R_b}}^k)$$

where $\mathbb{1}_{\text{con}}(e_h, e_t|{\mathbf{R_b}}^k) \in \{0, 1\}$ checks whether there exists a path instance of $\mathbf{R_b^k}$ between e_h and e_t in KG $\mathcal{G}$.

In this assignment function, we consider whether two entities can be connected via a relation path, instead of the entities or number of the connection paths. There are two main reasons for this design: (1) We expect that the mined causal relationship can be generalized to any dataset in this domain. Thus, if we want to distinguish different entities which instantiate the meta structure, we need to build a multinomial model for all possible entities. The multinomial would be infeasibly large. And our model can not be applied to any scenario which contain an unseen entity. (2) this function can be seen as an aggregation function to summary the connection information between entities. The aggregation function is very common in the causal relation model [25, 20, 21, 35]. With the aggregation function, we can build a concise and expressive model. Since the only thing we need is whether the entities are connected. Based on this assignment function, by sampling entity pairs in the training KG and querying the corresponding variable values, we can obtain tabular data for causal analysis.

3.2 Causal MetaKnowlege Discovery via d-seperation Criterion

The d-separation criteria [13] (see Definition 3.4) is a sufficient and necessary condition for the compatibility of a probability distribution with a causal model in the form of a directed acyclic graph (DAG). It states that a joint probability distribution of a set of random variables is compatible with the DAG (each node represents one of the given variables and each arrow represents the possibility of causal influence) if and only if the distribution satisfies a set of conditional independence relations encoded in the structure of the DAG. Therefore, d-separation is widely used in the algorithms in discovering causal structure[12, 36, 11].

Definition 3.3: d-seperation. A path p is blocked by a set of nodes Z if and only if:

1.p contains a chain of nodes $A \rightarrow B \rightarrow C$ or a fork $A \leftarrow B \rightarrow C$ such that the middle node B is in Z (i.e., B is conditioned on), or:

2. p contains a collider $A \rightarrow B \leftarrow C$ such that the collision node B is not in Z, and no descendant of B is in Z.

If Z blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on Z, and thus are independent conditional on Z.

Algorithm 1: Local causal metaknowledge discovery

Input: Y and $\{y_i\}, i = 1, \ldots, N$: variable and samples of queried variable $(C_h, C_t).M_q$;
$\mathcal{X}^{Ca} = \{X_k\}, k = 1, \ldots, K$ and $\{\{x_i\}_k\}, i = 1, \ldots, N$: variables and samples of candidate causes;

Output: causes $\mathcal{X}^C$ of Y

```
level d ← 0;
while d <= |X^Ca| − 1 do
    for each X_k ∈ X^Ca do
        for each subset Z ∈ X^Ca\{X_k} and |Z| = d do
            Test CI(X_k, Y|Z);
            if CI(X_k, Y|Z) then
                Test CI(Z, Y|X_k) (Reverse CI test.) ;
                if not CI(Z, Y|X_k) then
                    Remove X_k from X^Ca;
                    Break;
                end
            end
        end
    end
    d ← d + 1;
end
X^C = X^Ca
```

In this work, we design an efficient causal metaknowledge discovery algorithm based on d-separation. With d-separation, we can get the following conclusion: given any set of variables Z, where Z does not include X, X is not independent of its parent node (i.e. direct cause). Based on this conclusion, we can obtain a criterion for determining the direct cause of variable X. Furthermore, we design the following local causal metaknowledge discovery algorithm (Algo. 1) for the queried variable $Y = f_{\mathcal{G}}(x, y|R_h)$. We only mine the direct cause of Y, instead of the entire causal structure of variable set $\mathcal{X}^{Ca} \cup \{Y\}$. Particularly, given a queried variable $Y = f_{\mathcal{G}}(x, y|R_h)$, for each candidate cause in $\mathcal{X}^{Ca}$ (denoted as variable X_k), the proposed algorithm

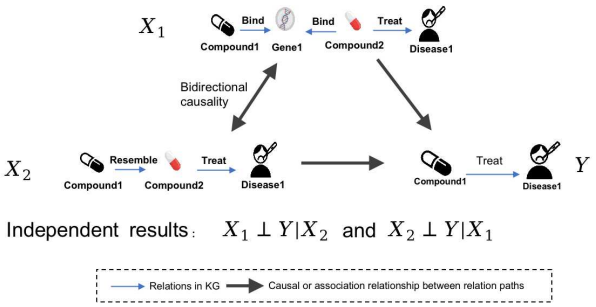

Figure 4: An example of bidirectional causal relationship, which may lead wrong results.

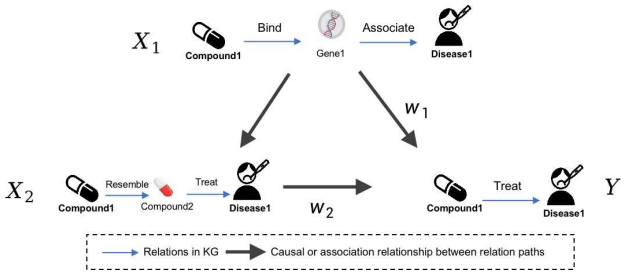

Figure 5: An example of non-independent rule-induced variables, which are both the causes of queried relation.

decides whether X_j should be retained in candidate causes set $\mathcal{X}^{Ca}$ by testing the independence of X_k and Y conditioning on a subset $\mathcal{Z}$ of $\mathcal{X}^{Ca} \backslash \{X_k\}$. The conditional independent(CI) tests are organised by levels (based on the size d of the conditioning sets). At the first level ($d = 0$), all pairs of variables are tested conditioning on the empty set. Some of the candidate causes would be removed and the algorithm only tests the remaining candidate causes in the next level ($d = 1$). The size of the conditioning set, d, is progressively increased (by one) at each new level until d is greater than $|\mathcal{X}^{Ca}| - 1$. Each corresponding relation path of $X \in \mathcal{X}^{C}$ construct a valid rule to predict the relation R_h in Y.

It is noteworthy that we add the reverse CI test in Algo. 1 (line 7) to avoid the impact of redundant relations in KGs. For example, $Compound1 \overset{Resembles}{\longrightarrow} Compound2$ and $Compound1 \overset{Binds}{\longrightarrow} Gene1 \overset{Binds}{\longleftarrow} Compound2$ express the similar message, which could lead the invalid independence test, as shown in Fig. 4. It will lead both X_1 and X_2 are removed from the candidate cause set of queried variable Y, even though they have very strong causal relationship with the drug treatment of diseases. Consequently, we use the reverse CI test to avoid this issue. In particular, if X_j and Y are judged to be independent conditioning on $\mathcal{Z}$, we will examine the independence between $\mathcal{Z}$ and Y conditioning on $\mathcal{X}_j$. When the result of the additional test is negative, X_j will be removed from $\mathcal{X}^{Ca}$. In this paper, we adopt SCI method [26] as the independent test method in the experiments, which works well on limited samples and discrete variables.

3.3 Link Prediction based on Explainable Causal Metaknowledge

The approach for link prediction based on interpretable rules tends to generate corresponding weights in the rule mining phase. By accumulating the weights of the rules satisfied by each predicted entity, a score of the predicted entities can be generated, and then the results are ranked based on this score. Here we first introduce how to generate rule weights under the causal model and then describe the approach for link prediction based on generated weights.

Weights of rules based on conditional dependency. In Algo. 1, we discover the direct causes by the non-independence relationship between the candidate meta structures and the queried meta structures. It is important

to note that the meta structures of $\mathcal{X}^C$ are not independent to each other. Fig. 5 gives an example for this case. Specifically, X_1 and X_2 are both causes of Y. Since X_1 is also a cause of X_2, if we directly calculate the causal strength between X_2 and Y, it is inevitable that w_2 will contain the causal effects that arise from X_1 along the path $X_1 \to X_2 \to Y$. Therefore, in order to better measure the importance of each causal rule and to avoid double-counted in the calculation of each proposed entity's score, we adopt the minimal conditional dependence as a measure of the importance of causal rules:

$$\begin{aligned} w_j = \min(\{dependence(X_j, Y|\mathcal{Z})\}) \\ \text{for any subset } \mathcal{Z} \in \mathcal{X}^{Ca} \backslash \{X_j\}, \end{aligned} \tag{44}$$

where w_j is the rule weight of the meta structure in X_j. In this paper, we use the $SCI_f(X, Y|Z)$ in SCI independence test [26] as the dependence score in Eq. 44, which can be get in the process of causal rules discovery. The higher of $SCI_f(X, Y|Z)$, the stronger the dependency.

Score function of entity results. Because of the incompleteness nature of KGs, open world assumption (OWA) [17] is often considered on real datasets. Under the OWA, the SUM function are usually adopted to calculate the ranking score of the predicted entity e_h in link prediction task $(?, R_h, e_t)$:

$$S_{R_q}^{sum} = \sum_{i=1}^{K} \tilde{w}_i Q_i, . \tag{45}$$

where K is the number of causal rules, $\tilde{w}_i$ is the normalized weight. $Q_i = 1$ when the body of the i-th causal rule holds for the entity pair (e_h, e_t), otherwise $Q_i = 0$. This approach focuses on the entities supported by multiple rules and does not use the non-existent relations between entity pairs, since the unreliable negative samples under OWA. In this paper, Eq. 45 is used in the link predictions on real data. For KG under closed world assumption(CWA) [17], the negative facts are also reliable, therefore we design a new function to apply the rules in the link prediction task. Particularly, given an query $(?, R_h, e_t)$, the score of the triple (e_h, R_h, e_t) is true can be formulated as:

$$S_{R_q}^{avg} = \sum_{i}^{K} \tilde{w}_i \big(Q_i \bar{Y}_{X_i=1} + (1 - Q_i)\bar{Y}_{X_i=0}\big), \tag{46}$$

where K is the number of causal rules for the queried relation, $\tilde{w}_i$ is the normalized weight for the i-th result rule. $\bar{Y}_{X_i=1}$ denotes the proportion of the queried relation to be true when the body of the i-th causal rule is true in the training data, and $\bar{Y}_{X_i=0}$ denotes the proportion of the queried relation to be true when the body of the i-th causal rule is false. $Q_i = 1$ when the body of the i-th causal rule holds for the entity pair (e_h, e_t), otherwise $Q_i = 0$. The results will be ranked by S_{R_q} of each valid e_t. In this paper, Eq. 46 is used in the link predictions on simulation data.

4 Experimental Study

4.1 Experimental Setup

In this section, we empirically evaluate the effectiveness and interpretability of the proposed CMLP on both simulation and real-world datasets. For interpretability, we focus on whether the algorithm can uncover the causal relationships inherent in the knowledge graph.

4.1.1 Baselines

To evaluate the interpretability of the algorithms, we select four rule-based methods that can conduct link prediction and generate explainable rules. To make a fair comparison, the inference rules, obtained from different

Table 20: Dataset statistics of all the experiments.

	#Triplets	#Relations	#Entities
Simulation	6,095	5	1,590
Douban Movie Rate	28,356	12	3,007
Hetionet	174,941	20	32,056

algorithms, are used to conduct the link prediction task based on the same prediction equations. This approach can also help us observe the impact of different rules on the link prediction task. For a complete evaluation of the effectiveness of the proposed approach, we also compute the LP performance of TuckER[2], one representation-based method which has the best overall performance among the representation-based methods across different datasets[32]. All baselines are listed in the following:

1) AMIE+[8], an efficient top-down method to discover the interpretable rules.

2) AnyBURL[27], a bottom-up approach to mine the logical rule.

3) Neural-LP[44], an end-to-end differentiable model to learn the first-order logical rule.

4) RNNLogic[30], an EM-based algorithm to learn the rule generator and the reasoning predictor iteratively.

5) TuckER[2], a linear model based on Tucker decomposition of the binary tensor representation of knowledge graph triples.

4.1.2 Datasets

To quantitatively evaluate the effectiveness of the algorithm in discovering causal knowledge, we construct a simulation dataset owing to a lack of groundtruth of real datasets. Douban and Hetionet [15] are selected as our real datasets on which we perform two link prediction tasks, movie rating prediction and drug repurposing, respectively. Here we provide more details for these datasets, and their statistics are shown in Table 20.

Simulation dataset. We generate simulated KGs based on a toy causal model specified in Fig. 6, which includes three concepts and five relations. In particular, we design the causal mechanisms in KG via a probabilistic model. The root nodes (X_1, X_4) in the causal graph are generated via Bernoulli distributions, whose probability mass function is $f_X(x) = p^x(1-p)^{1-x}$. Moreover, the non-root nodes (X_2, X_3) are generated via the conditional probability distributions, which are Bernoulli distributions, given the parent node (X_1). To maintain a stable causal mechanism, the parameters of conditional distributions are constant in training and testing, as shown in Table 21. In the out-of-distribution paragraph, we will introduce the parameters of root nodes in training and testing.

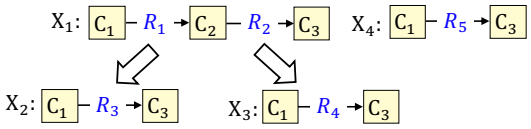

Figure 6: The causal graph of relation paths, based on which the simulated KGs are generated .

Douban movie rating. Douban is a famous Chinese website for movie reviews, where users can rate and comment on any movie. The rating range is from 1 to 5. A higher rating means that users like movies, while a lower rating means that users have negative feedback on movies. We collect the real-world data from Douban[1]

[1]https://www.douban.com/

Table 21: The parameters of conditional distributions.

Conditions	$X_2\|X_1=1$	$X_2\|X_1=0$	$X_3\|X_1=1$	$X_3\|X_1=0$
Parameters	p=0.9	p=0.1	p=0.9	p=0.1

and construct a dataset (this dataset will be released), whose statistics are shown in Table 20. Commonly, a movie with a score of 4 or 5 is identified as meeting the taste of users. So we transform the original 5-level rating to a 2-level rating with a threshold of 4. If the rating score is 4 or 5, the original relation *Rate* is replaced by *HighRate* We conduct the link prediction task on the relation *HighRate*. Because the raw data is too large and the relations between users and movies are very sparse, in this thesis, we first filter the 20 users who have made the most ratings and take the rating history of these users as the set of rating facts for our study. The facts unrelated to the rating are also included in our experimental data.

Hetionet[15] is a freely available knowledge database that integrates biomedical information from 29 prominent bioinformatics resources. Recently, Hetionet was successfully applied to drug repurposing tasks in terms of the link prediction task for relation `Treats`[15, 31].

Why do we choose those two real datasets instead of other commonly used datasets, such as WN18, FB15k? In this paper, we focus on link prediction with the help of causal relationships between knowledge graph relations. The core of causality lies in its asymmetry. The commonly used KGs for link prediction algorithms contain many symmetrical relationships, e,g., *hypernym* and *hyponym* in WN18. These symmetrical relationships may help with the link prediction task, but they go against the basic idea that causality is a one-way relationship. We, therefore, chose datasets with specific application scenarios and rich causal semantics.

4.1.3 Metrics.

For link prediction, we employ the commonly used metrics mean reciprocal rank (MRR) and Hits@k [32, 2, 3].

$$MRR = \frac{1}{|Q|}\sum_{q\in Q}\frac{1}{q} \tag{47}$$

$$Hit@K = \frac{|\{q \in Q : q \leq K\}|}{|Q|} \tag{48}$$

where Q is the rank results, for each $q_i \in Q$ is the ranking of the desired results in the i-th query $(?, R_h, e_t)$. In the case of ties in the calculation of Q, we use the mean rank to avoid misleading results[32, 33]. Both MRR and His@k are the higher, the better.

The interpretability of causal rules on the simulation dataset can be understood as the consistency between the mined causal rules and the actual causal structure. Therefore, we evaluate baselines and our approach on the simulation dataset using precision, recall, and structural Hamming distance (SHD) as the evaluation metrics, which are commonly used evaluation metrics in causal structure discovery studies [46, 47].

$$Precision = \frac{\#TFR}{\#FR}; Recall = \frac{\#TFR}{\#ATR} \tag{49}$$

Where $\#TFR$ is the number of right causal relationships discovered by an algorithm, $\#FR$ is the number of causal relationships discovered by an algorithm, and $\#ATR$ is the number of all causal relationships. Both precision and recall are the higher, the better. SHD calculates the difference between the learned graph and the ground truth graph by the number of edge insertions, deletions, or flips required to transform one graph into another. The lower the SHD, the better.

4.1.4 Out-of-Distribution Link Prediction

Traditional machine learning methods are designed based on the assumption of independent and identically distributed (I.I.D) data. This assumption means the training and test data come from the same distribution. However, the distribution of test data may alter due to changes in the test environment; such tasks are referred to as OoD tasks. Traditional algorithms perform poorly on generalization problems due to the violation of I.I.D assumption. Causality is seen as a stable inference mechanism in many research works on generalization problems. So, in this work, we provide an out-of-distribution generalization task for the knowledge link prediction task for the first time. On the one hand, the effect of causal metaknowledge on this task can be measured, and on the other, the performance of existing algorithms can be looked at. We also evaluate the performance of the algorithms on I.I.D link prediction tasks.

In this paper, we design two OoD experimental scenarios.

(1) *Simulation dataset*: We evaluate the link prediction performance under I.I.D and OoD settings, where the triples of root node X_1 are generated in testing under the same and different parameters with training. In the training and I.I.D testing datasets, $p_{X_1} = 0.5$. In the OoD testing datasets, $p_{X_1} \in \{0.2, 0.9\}$. For X_4, p_{X_4} maintains 0.9 in training and testing. The facts of the KGs are split into three parts:*train*, *test info*, and *test*. The facts in *train* are used to learn the rule. The effectiveness of the learned rule is assessed via the link prediction task on R_3. The *test info* part includes facts of new entities (did not appear in *train*) on R_1, R_2, R_4, R_5, and *test* part includes the queried facts on R_3.

(2)*Real datasets*: It is impossible to explicitly change the data distribution since real data distribution is inaccessible for real datasets. Recent research[38, 43] has suggested that graph models are biased towards nodes with larger degrees, which causes the bad performance of low-degree nodes in the test. Therefore we construct the OoD datasets based on degree shift. Specifically, given a query task $(?, R_h, e_t)$, we calculate the *median* of degree[2] of known entities e_t belonging to the triples (e_h, R_h, e_t) in the training. Then we bin the test queries by the degrees of the known entities e_t. The degree range in each bucket is decided based on the sample size balance. The test queries in the bucket, which the training median falls in, can be treated as the I.I.D test samples. Others are the OoD samples. The I.I.D bucket is labeled with $*$ in Table 23 and Table 24.

4.2 Performance of Link prediction task in Out-of-Distrition Settings

Results on simulations. For the simulation dataset, we construct a OoD setting named as covariance shift, by changing the probability distribution of the root nodes in the test phase. Table 22 presents the methods in performance and demonstrates the effectiveness of the proposed CMLP. In particular, CMLP outperforms the baseline models under all metrics except the Hits@10 under I.I.D setting. This shows our method can give a stable and high-quality result, especially in the OOD setting. Besides, compared with the baselines, the proposed CMLP perform significantly better at the Hits@1 metric (at least 25% absolute improvements that the second place under all settings), which suggests the our method are more suitable for scenarios with strict performance requirements, and this feature may achieved by the removal of association-based rules.

Results for Douban movie rating. Since we filtered the users of the Douban data, and the discrepancies of the degrees of experimental user nodes are close to each other. Therefore, in this experiment, to construct the OoD scenario, we adopt the head prediction (? , HighRate, Movie), predicting the set of users who gave high ratings to movies. Further, we bucketed the movie nodes in the test data according to their degree in the training data to observe the performance of the algorithm under different node prevalence. The MRR and Hits@5 results shown in Tab. 23 shows that the proposed CMLP get the best performance in the all OoD settings. Especially, at least 25.8% and 29.3 % relative improvements that the second place on the MRR and Hits@5, respectively. In the I.I.D setting, CMLP gets the second place on both MRR and Hits@5, lower than the representation-based method

[2]In this paper, we use the term "degree" to stand for the sum of in and out degrees

Table 22: The results of link prediction on simulation datasets.

Settings	p_{X_1}	Method	MRR	Hits		
				@10	@3	@1
I.I.D	0.5	AMIE+	0.87	**98.99**	94.95	78.79
		AnyBURL	0.87	**98.99**	94.95	78.79
		Neural-LP	0.80	**98.99**	92.42	66.16
		RNNLogic	0.87	**98.99**	94.95	78.79
		CMLP	**0.94**	98.48	**97.98**	**90.91**
OOD	0.2	AMIE+	0.875	96.91	93.81	81.44
		AnyBURL	0.875	96.91	93.81	81.44
		Neural-LP	0.68	98.97	79.38	50.51
		RNNLogic	0.875	96.91	93.81	81.44
		CMLP	**0.99**	**100**	**100**	**99.97**
OOD	0.9	AMIE+	0.91	**100**	96.34	85.67
		AnyBURL	0.91	**100**	96.34	85.67
		Neural-LP	0.88	**100**	96.95	79.57
		RNNLogic	0.91	**100**	96.34	85.67
		CMLP	**0.99**	**100**	**99.70**	**99.09**

TuckER. These results illustrate that for movie rating datasets, the rules learned by our method can capture more general user preferences and give relatively accurate rating predictions for movies that are in different popularity.
Results for drug repurposing on Hetionet. Consistent with the traditional setup of drug redirection, on Hetionet, we also use head predition (?, Treat, Disease), which is giving a Disease to predict new drugs. We also observe the performance of the algorithm under this task by bucketing for Disease node degrees. Tab. 24 reports the MRR and Hits@5 on this dataset, and we can find that: our CMLP performs significantly better than other baselines on low-degree diseases (0-17), while AMIE+ and AnyBURL get better results on low-degree diseases (17-100). These results indicate that our method can give more accurate drug discovery results for diseases with relatively low information. And for diseases with richer information, the correlation-based inference rules give more accurate drug prediction.

4.3 Quality and Interpretability of Causal Rules.

As stated in Sec. 1, the mined rules play a key role in our algorithm, and an important advantage of these rules is that they are well interpretable. In this section, we will evaluate the quality and interpretability of rules mined by CMLP .
Quality of causal rules from simulations. The ground-truth causal graph of KGSs shown in Fig 6, and Table 25 shows the accuracy of estimated rules of different methods. In particular, CMLP accurately discovers two causal rules in Fig 6 without any redundant rules. In contrast, correlation-based methods report some non-causal rules.
Interpretability of causal rules Moreover, for the simulation dataset, we analyze all methods' results, whose heads are R_3 and R_5, and the results are shown in Table 26 (We omit results of R_4, since R_3 and R_4 are symmetric in the causal graph). The rules follow the causal mechanism are in bold. There is only one causal rule for R_3, which is $R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)$. AMIE+, AnyBURL, RNNLogic and CMLP find this causal rule. Besides this causal rule, the results of AMIE+, RNNLogic and AnyBURL also include other rules, such as

Table 23: MRR (left) and Hits@5 (right) for Douban movie rating. The * marks columns that contain the I.I.D results. Other columns contain OoD results.

Methods	Degree Range			
	0-21*	**21-31**	**31-39**	**39-60**
AMIE+	0.120	0.205	0.261	0.395
AnyBURL	0.125	0.182	0.231	0.373
Neural-LP	0.078	0.097	0.126	0.217
RNNLogic	0.072	0.086	0.097	0.161
TuckER	**0.287**	0.186	0.186	0.149
CMLP	0.251	**0.343**	**0.392**	**0.497**

Methods	Degree Range			
	0-21*	**21-31**	**31-39**	**39-60**
AMIE+	13.7	30.7	39.4	68.3
AnyBURL	15.1	26.0	33.2	64.6
Neural-LP	0.1	0.6	3.4	60.0
RNNLogic	1.6	4.7	7.2	13.5
TuckER	**50.3**	31.9	28.2	21.5
CMLP	40.9	**60.0**	**68.1**	**88.3**

Table 24: MRR(left) and Hits@5(right) for drug repurposing on Hetionet. The * marks columns that contain the I.I.D results. Other columns contain OoD results.

Methods	Degree Range			
	0-8*	**8-17**	**17-31**	**31-100**
AMIE+	0.103	0.085	**0.132**	0.065
AnyBURL	0.116	0.189	0.090	**0.188**
Neural-LP	0.027	0.014	0.009	0.009
RNNLogic	0.07	0.021	0.029	0.012
TuckER	0.044	0.022	0.083	0.015
CMLP	**0.248**	**0.208**	0.095	0.093

Methods	Degree Range			
	0-8*	**8-17**	**17-31**	**31-100**
AMIE+	13.2	11.3	**22.5**	13.2
AnyBURL	15.8	**25.0**	9.6	**23.7**
Neural-LP	5.3	0	0	0
RNNLogic	7.9	0	3.2	0
TuckER	3.1	5.9	10.7	3.2
CMLP	**26.31**	**25.0**	16.1	13.1

Table 25: Experimental results on simulation data with $p_{X_1} = 0.5$, based on the metrics (precision, recall and SHD), which are commonly used to evaluate the estimated causal graph.

Method	Precision ↑	Recall ↑	SHD ↓
Neural-LP	0	0	10
AMIE+	0.22	**1.0**	7
RNNLogic	0.22	**1.0**	7
AnyBURL	0.25	**1.0**	6
CMLP	**1.0**	**1.0**	**0**

Table 26: All rules whose head are R_3 and R_5, obtained by each algorithm learned on simulated dataset. The strikethroughs indicate the wrong results (there is no entities satisfying the rule). The rules consistent with the generation process are in bold. The orange text denotes the weight of each rule with the form max-normalization(original weight)

.

Method	Rules of R_3 with $\mathbf{p_{X_1} = 0.5}$	Rules of R_3 with $\mathbf{p_{X_1} = 0.9}$	Rules of R_5
AMIE+	1.00 (0.908) $\mathbf{R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)}$ 0.91 (0.829) $R_3(C_1, C_3) \leftarrow R_4(C_1, C_3)$ 0.55 (0.500) $R_3(C_1, C_3) \leftarrow R_5(C_1, C_3)$	1.00 (0.900) $\mathbf{R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)}$ 0.99 (0.890) $R_3(C_1, C_3) \leftarrow R_4(C_1, C_3)$ 0.91 (0.820) $R_3(C_1, C_3) \leftarrow R_5(C_1, C_3)$	1.00 (0.898) $R_5(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)$ 0.99 (0.895) $R_5(C_1, C_3) \leftarrow R_3(C_1, C_3)$ 0.99 (0.894) $R_5(C_1, C_3) \leftarrow R_4(C_1, C_3)$
AnyBURL	1.00 (0.896) $\mathbf{R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)}$ 0.92 (0.823) $R_3(C_1, C_3) \leftarrow R_4(C_1, C_3)$ 0.56 (0.501) $R_3(C_1, C_3) \leftarrow R_5(C_1, C_3)$	1.00 (0.898) $R_3(C_1, C_3) \leftarrow R_4(C_1, C_3)$ 0.99 (0.893) $\mathbf{R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)}$ 0.91 (0.821) $R_3(C_1, C_3) \leftarrow R_5(C_1, C_3)$	1.00 (0.907) $R_5(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)$ 0.99 (0.902) $R_5(C_1, C_3) \leftarrow R_3(C_1, C_3)$ 0.99 (0.897) $R_5(C_1, C_3) \leftarrow R_4(C_1, C_3)$
Neural-LP	1.00 (0.318) ~~$R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_4(C_2, C_3)$~~ 0.99 (0.316) ~~$R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_5(C_2, C_3)$~~ 0.31 (0.100) ~~$R_3(C_1, C_3) \leftarrow R_5(C_1, C_2), R_4(C_2, C_3)$~~ 0.31 (0.099) ~~$R_3(C_1, C_3) \leftarrow R_5(C_1, C_2), R_5(C_2, C_3)$~~ 0.23 (0.073) $R_3(C_1, C_3) \leftarrow R_4(C_1, C_3)$ 0.09 (0.028) ~~$R_3(C_1, C_3) \leftarrow R_4(C_1, C_2), R_4(C_2, C_3)$~~ 0.07 (0.023) $R_3(C_1, C_3) \leftarrow R_5(C_1, C_3)$	1.00 (0.757) ~~$R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_1(C_2, C_3)$~~ 0.17 (0.128) ~~$R_3(C_1, C_3) \leftarrow R_1(C_1, C_3)$~~ 0.04 (0.056) ~~$R_3(C_1, C_3) \leftarrow R_5(C_2, C_3), R_1(C_1, C_2)$~~ 0.05 (0.035) ~~$R_3(C_1, C_3) \leftarrow R_4(C_2, C_1), R_1(C_1, C_2)$~~ 0.03 (0.025) $\mathbf{R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)}$	1.00 (0.125) ~~$R_5(C_1, C_3) \leftarrow R_4(C_1, C_2), R_3(C_2, C_3)$~~ 0.78 (0.097) ~~$R_5(C_1, C_3) \leftarrow R_3(C_1, C_2), R_3(C_2, C_3)$~~
RNNLogic	1.00 (0.076) $\mathbf{R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)}$ 0.58 (0.044) $R_3(C_1, C_3) \leftarrow R_4(C_1, C_3)$ 0.13 (0.010) $R_3(C_1, C_3) \leftarrow R_5(C_1, C_3)$	1.00 (0.071) $\mathbf{R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)}$ 0.49 (0.035) $R_3(C_1, C_3) \leftarrow R_4(C_1, C_3)$ 0.14 (0.010) $R_3(C_1, C_3) \leftarrow R_5(C_1, C_3)$	1.00 (0.220) $R_5(C_1, C_3) \leftarrow R_3(C_1, C_3)$ 0.28 (0.060) $R_5(C_1, C_3) \leftarrow R_4(C_1, C_3)$ 0.20 (0.045) $R_5(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)$
CMLP	1.00 (122.797) $\mathbf{R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)}$	1.00 (20.061) $\mathbf{R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)}$	-

$R_3(C_1, C_3) \leftarrow R_4(C_1, C_3)$ and $R_3(C_1, C_3) \leftarrow R_5(C_1, C_3)$. Especially, for those three methods, note that the weights of $R_3(C_1, C_3) \leftarrow R_4(C_1, C_3)$ are very close to the weights of $R_3(C_1, C_3) \leftarrow R_1(C_1, C_2), R_2(C_2, C_3)$. It means the algorithms think these two rules have the similar interpretability for the head relation R_3. With the change of root node X_1's distribution(from $p_{X_1} = 0.5$ to $p_{X_1} = 0.9$), AnyBURL even report the higher weight for the rule $R_3(C_1, C_3) \leftarrow R_5(C_1, C_3)$. According to the generation mechanism, the existence of R_3 between entities e_i and e_j is independent with whether there is R_4 and R_5 between e_i and e_j. AMIE+, AnyBURL and RNNLogic still return these association rules with high weights, because they only consider whether R_3 and R_4 co-occur frequently, but not the reason of the co-occurrence. The end-to-end completion-oriented method, Neural-LP, also return some wrong results, such as the top 1 rule $R_3(C_1, C_3) \leftarrow R_4(C_1, C_2), R_4(C_2, C_3)$, which can not be satisfied by any entities in KG. The results in [34] show the same phenomenon. The intermediate results of the completion-oriented method is incomprehensible sometimes.

Furthermore, We sort the rules generated by each algorithm based on their assigned weights and show the five top rules from Douban and Hetionet in Tab. 27 and Tab. 28, respectively. The results in Tab. 27 suggests that the ratings for the target movie are highly related to other movies which share the same staff, such as writer, actors, director, etc. According to the rating results, CMLP finds a strong causal relationship between the rating of the movie and its editor than other pairs. The top rules generated by AMIE+ and AnyBURL focus on other shared staffs, but the shared staff has different roles in the target movie and the movie in path. Those rules of AMIE+ and AnyBURL are hard to be satisfied for most queries. It is worth noting that RNNlogic report the 'fan' rules will impact the users' rating, but CMLP excludes this kind of rules. Our results suggest the working ability of the movie's stuff (*e.g.* actor or writer) should be the root cause of the users' rate, instead of the followers of the stuff. From the rules from Hetionet, we can see the learned rules are broadly divided into two classes, those in which the target drug and disease are connected by therapeutic information about the similar disease and drug, and those in which the target drug and disease are connected by commonly associated genes. Further, we find that the rules

Table 27: Top 5 Rules to infer HighRate(User, Movie) given by the methods. The strikethroughs indicate the wrong results (there is no entities satisfying the rule).

Method	Top rules to infer HighRate(User, Movie)
AMIE+	1.00 (0.565) HighRate(User,Movie) ← HighRate(User,Movie1), Writer(Person,Movie1),Director(Person,Movie) 0.98 (0.556) HighRate(User,Movie) ← HighRate(User,Movie1), Director(Person,Movie1), Writer(Person,Movie) 0.87 (0.489) HighRate(User,Movie) ← HighRate(User,Movie1), Writer(Person,Movie1), Actress(Person,Movie) 0.74 (0.417) HighRate(User,Movie) ← HighRate(User,Movie1), Director(Person,Movie1), Actor(Person,Movie) 0.72 (0.405) HighRate(User,Movie) ← HighRate(User,Movie1), Actress(Person,Movie1), Writer(Person,Movie)
AnyBURL	1.00 (0.400) HighRate(User,Movie) ← HighRate(User,Movie1), Composer(Person,Movie1), Actor(Person,Movie) 0.99(0.397) HighRate(User,Movie) ← HighRate(User,Movie1), Producer(Person,Movie1), Director(Person,Movie) 0.97 (0.386) HighRate(User,Movie) ← HighRate(User,Movie1), Director(Person,Movie1), Actress(Person,Movie) 0.89 (0.355) HighRate(User,Movie) ← HighRate(User,Movie1), Writer(Person,Movie1), Actress(Person,Movie) 0.85 (0.340) HighRate(User,Movie) ← HighRate(User,Movie1), Editor(Person,Movie1), Editor(Person,Movie)
Neural-LP	1.00 (0.120) HighRate(User,Movie) ← HighRate(User,Movie1), HighRate(User1,Movie1), HighRate(User1,Movie) 0.28 (0.034) HighRate(User,Movie) ← HighRate(User,Movie1), MovieType(Movie1,Type), MovieType(Movie,Type)
RNNLogic	1.00 (0.011) HighRate(User,Movie) ← Fan(User,Person),Editor(Person,Movie) 0.45 (0.005) HighRate(User,Movie) ← Fan(User,Person),Actor(Person,Movie) 0.36 (0.004) HighRate(User,Movie) ← Fan(User,Person),Director(Person,Movie) 0.36 (0.004) HighRate(User,Movie) ← Fan(User,Person),Writer(Person,Movie) 0.36 (0.004) HighRate(User,Movie) ← Fan(User,Person),Composer(Person,Movie)
CMLP	1.00 (0.034) HighRate(User,Movie) ← HighRate(User,Movie1), Editor(Person,Movie1), Editor(Person,Movie) 0.12 (0.004) HighRate(User,Movie1) ← HighRate(User,Movie1),Cinematographer(Person,Movie),Cinematographer(Person,Movie) 0.06 (0.002) HighRate(User,Movie1) ← HighRate(User,Movie1),Writer(Person,Movie),Writer(Person,Movie) 0.06 (0.002) HighRate(User,Movie1) ← HighRate(User,Movie1),Actress(Person,Movie),Actress(Person,Movie) 0.03 (0.001) HighRate(User,Movie1) ← HighRate(User,Movie1),Director(Person,Movie),Actor(Person,Movie)

Table 28: Top 5 Rules to infer Treats(Compound, Disease) given by the methods. For brevity, we use 'C' and 'D' for compound and disease, respectively.

Method	Top rules to infer Treats(Compound, Disease)
AMIE+	1.00 (0.393) Treats(C, D) ← Resembles(C,C1), Treats(C1, D) 0.82 (0.322) Treats(C, D) ← Resembles(C1,C),Treats(C1, D) 0.42 (0.167) Treats(C, D) ← Downregulates(C,Gene1), Associates(D,Gene1) 0.38 (0.151) Treats(C, D) ← Downregulates(C,Gene1), Upregulates(D,Gene1) 0.37 (0.144) Treats(C, D) ← Binds(C,Gene1), Upregulates(D,Gene1)
AnyBURL	1.00 (0.319) Treats(C, D) ← Includes(PharmacologicClass1,C),Includes(PharmacologicClass1,C1),Treats(C1, D) 0.60 (0.192) Treats(C, D) ← Resembles(C1,C),Treats(C1, D) 0.52 (0.166) Treats(C, D) ← Resembles(C1,C),Resembles(C1,C2),Treats(C2, D) 0.31 (0.098) Treats(C, D) ← Resembles(C1,C),Resembles(C2,C1),Treats(C2, D) 0.24 (0.077) Treats(C, D) ← Treats(C, D1), Resembles(D1,D)
Neural-LP	1.00 (0.659) Treats(C, D) ← Treats(C, D1), Treats(C1, D1),Treats(C1, D)
RNNLogic	1.00 (0.00007) Treats(C, D) ← Resembles(C, C1),Treats(C1, D) 1.00 (0.00007) Treats(C, D) ← Resembles(C, C1), Resembles(C1, C2),Treats(C2, D)
CMLP	1.00 (269.00)) Treats(C, D) ← Treats(C, D1),Resembles(D1, D2),Resembles(D, D2) 0.85 (229.32)) Treats(C, D) ← Includes(PharmacologicClass1, C),Includes(PharmacologicClass1, C1),Treats(C1, D) 0.83 (224.37)) Treats(C, D) ← Treats(C, D1),Resembles(D2, D1),Resembles(D2, D) 0.58 (155.18)) Treats(C, D) ← Treats(C, D1),Treats(C1, D1),Treats(C1, D) 0.10 (26.52)) Treats(C, D) ← Treats(C, D1), Resembles(D2,D1), Resembles(D,D1)

mined by AnyBURL also contain rules for reasoning through shared side effects.

5 Related Work

In this section, we first review the related studies in causal discovery for propositional domains and relational domains. Then we discuss and clarify the distinction between the proposed approach and the most relevant *rule mining* methods for relational data. We list the relevant research areas and our differences in the Tab. 29

Table 29: Comparison of our work and related work.

	Association	Causality
Propositional	Traditional machine learning	Traditioanal causal model
relational	Rule mining(e.g.logical rule), Graph representation learning	Relational causal model(with attribute), **Our(without attribute)**

Causal Discovery from Propositional Data. Rubin causal models [16] and structural causal models (SCMs) [29] are the two dominated frameworks for causal discovery from propositional data. Particularly, the former analyzes the causal effect between treatment and effect with partial structural information, while the later employs Bayesian

networks to identify causal structure. Our situation resembles causal discovery in SCM because we are primarily concerned with unearthing causal relationships from KG. Furthermore, there are two kinds of causal discovery algorithms, *constraint-based* and *score-based* [37], in SCMs. Contrary to the score-based approaches, which are based on the global score, the constraint-based approaches can employ the local conditional independence to determine the causal relationship between specific variables, which is critical when only partial causal relationship is interested. In addition, the constraint-based methods are non-parametric, which means that they do not depend on the specific functions to connect variables. Based on above advantages, we follow the constrained-based design to develop our method.

Relational Causal Model. To our best knowledge, the relational causal model [25, 20, 21, 35] is the only framework designed to extract causal information from relational data. The input of a relational causal model is a relational database containing entity and relation tables. Each entity table contains all entities corresponding to the same concept, and each relation table contains all facts corresponding to the same relationship between two entities. Consequently, a relational database is comparable to a KG. Relational causal model finds causal relationship between related entity attributes. For example, for the database with two relations: Develop(Employee, Product) and Funds(Company, Product), the relational causal relationship will give the results like [Employee, Develops, Products, Fundsby, Company].budgets $\rightarrow$ [Employee].Success. We can see that relational causal models emphasize relational models of attributes with a known entity-level connection graph [24], while our research focuses on generative process of the connection, which is the preceding step in the entire KG generation process.

Rule Mining from KG. Inductively knowledge reasoning involves generalising patterns from a given set of observed facts and then generating novel but potentially imprecise predictions. In addition to the incomprehensible techniques based on embedding, *rule mining* methods, which benefit from intuitive interpretation of the findings of link prediction, have maintained the popularity for decades. The rule mining studies in KG can be divided into two categories according to the main objectives: metrics-oriented and completion-oriented. Metrics-oriented methods [9, 8, 28] usually use predefined co-occurrence metrics, *confidence* and *support*, to find rules satisfying the given thresholds of the metrics, based on a top-down fashion. Recently, AnyBURL [27] designs a bottom-up technique for rule learning, which requires few computational resources. The other line of research is completion-oriented. Different from the predefined metrics-based methods, these studies are mainly based on end-to-end learning and target on the link completion task. The explainable rules are mainly the intermediate results, which are obtained via analyzing the parameters of the model. The trained models are used to predict the link directly. Neural-LP [44] adopts an attention mechanism to select a variable-length sequence as the body of rules for which confidence scores are learnt from the attention vectors. DRUM [34] uses bidirectional recurrent neural networks to learn the relations of sequences, which are the body of rules, and their confidence scores are estimated via the recurrent neural network. RNNLogic [30] utilizes logic rules as a latent variable and trains both a rule generator and a reasoning predictor with logic rules.

Our work can be seen as one of solutions for rule mining. Different from the past the association-based rule mining methods, our work aims to discover deeper relationship (*i.e.* causation) via a more rigorous statistical inference system. To the best of our knowledge, this is the fist attempt to study rule mining problem under causal perspective, as far as we know.

6 Conclusion and Future work

In this work, we propose a method, CMLP, for entity-level link prediction based on the causal relationships between topologies at the concept level. This method constructs complex connectivity between entities and predicts causal relationships between links at the conceptual level, eliminating spurious correlation that may be learned by traditional association-based methods. Extensive experiments have shown that the proposed CMLP achieves leading performance in a variety of OOD experimental settings. Note that previous representation learning based

models (generally learning from correlations) are hard to generalize to OOD setting, and our model demonstrates that causal-based learning is a promising solution for this setting.

Since the causal model itself is the method which modes the process of data generation, the mined causal rule can be used to understand the physical mechanism of knowledge graph generation. It opens up new possibilities for research in fields such as pharmaceutical economics. Besides, although this paper propose a rule-based model which can works under OOD setting, how to improve the generalization ability of the representation-based models is still a open problem and deserved further investigation.

References

[1] John Aldrich. Correlations Genuine and Spurious in Pearson and Yule. Statistical Science, 10(4):364 – 376, 1995.

[2] Ivana Balažević, Carl Allen, and Timothy M Hospedales. Tucker: Tensor factorization for knowledge graph completion. In Empirical Methods in Natural Language Processing, 2019.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. Advances in neural information processing systems, 26, 2013.

[4] Xuelu Chen, Ziniu Hu, and Yizhou Sun. Fuzzy logic based logical query answering on knowledge graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 3939–3948, 2022.

[5] Xiao Cui and Hao Shi. A*-based pathfinding in modern computer games. International Journal of Computer Science and Network Security, 2011.

[6] James A Evans and Jacob G Foster. Metaknowledge. Science, 2011.

[7] Santo Fortunato, Carl T Bergstrom, Katy Börner, James A Evans, Dirk Helbing, Staša Milojević, Alexander M Petersen, Filippo Radicchi, Roberta Sinatra, Brian Uzzi, et al. Science of science. Science, 2018.

[8] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. Fast rule mining in ontological knowledge bases with amie. The VLDB Journal, 24(6):707–730, 2015.

[9] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In Proceedings of the 22nd international conference on World Wide Web, pages 413–422, 2013.

[10] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In Proceedings of the 22nd international conference on World Wide Web - WWW '13, pages 413–422. ACM Press.

[11] Andreas Gerhardus and Jakob Runge. High-recall causal discovery for autocorrelated time series with latent confounders. Advances in Neural Information Processing Systems, 33:12615–12625, 2020.

[12] Enrico Giudice, Jack Kuipers, and Giusi Moffa. The dual pc algorithm for structure learning. In International Conference on Probabilistic Graphical Models, pages 301–312. PMLR, 2022.

[13] Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. Causal inference in statistics: A primer. John Wiley & Sons, 2016.

[14] Manuel Heusner, Thomas Keller, and Malte Helmert. Best-case and worst-case behavior of greedy best-first search. International Joint Conferences on Artificial Intelligence, 2018.

[15] Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio E Baranzini. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. Elife, 6:e26726, 2017.

[16] Guido W Imbens and Donald B Rubin. Rubin causal model. In Microeconometrics, pages 229–241. Springer, 2010.

[17] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems, 33(2):494–514, 2021.

[18] Daniel R Lanning, Gregory K Harrell, and Jin Wang. Dijkstra's algorithm and google maps. In Proceedings of the 2014 ACM Southeast Regional Conference, 2014.

[19] Thuc Duy Le, Tao Hoang, Jiuyong Li, Lin Liu, Huawen Liu, and Shu Hu. A fast pc algorithm for high dimensional causal discovery with multi-core pcs. IEEE/ACM transactions on computational biology and bioinformatics, 16(5):1483–1495, 2016.

[20] Sanghack Lee and Vasant Honavar. On learning causal models from relational data. In Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[21] Sanghack Lee and Vasant Honavar. Towards robust relational causal discovery. In Uncertainty in Artificial Intelligence. PMLR, 2020.

[22] Haotian Li, Yong Wang, Songheng Zhang, Yangqiu Song, and Huamin Qu. Kg4vis: A knowledge graph-based approach for visualization recommendation. IEEE Transactions on Visualization and Computer Graphics, 28(1):195–205, 2021.

[23] Zizheng Lin, Haowen Ke, Ngo-Yin Wong, Jiaxin Bai, Yangqiu Song, Huan Zhao, and Junpeng Ye. Multi-relational graph based heterogeneous multi-task learning in community question answering. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pages 1038–1047, 2021.

[24] Marc Maier, Katerina Marazopoulou, David Arbour, and David Jensen. A sound and complete algorithm for learning causal models from relational data. In Uncertainty in Artificial Intelligence, page 371. Citeseer, 2013.

[25] Marc Maier, Brian Taylor, Huseyin Oktay, and David Jensen. Learning causal models of relational domains. In Proceedings of the AAAI Conference on Artificial Intelligence, 2010.

[26] Alexander Marx and Jilles Vreeken. Testing conditional independence on discrete data using stochastic complexity. In The 22nd International Conference on Artificial Intelligence and Statistics, 2019.

[27] Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. Anytime bottom-up rule learning for knowledge graph completion. In IJCAI, pages 3137–3143, 2019.

[28] Pouya Ghiasnezhad Omran, Kewen Wang, and Zhe Wang. An embedding-based approach to rule learning in knowledge graphs. IEEE Transactions on Knowledge and Data Engineering, 2019.

[29] Judea Pearl. Causal inference. Causality: Objectives and Assessment, pages 39–58, 2010.

[30] Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In International Conference on Learning Representations, 2021.

[31] Florin Ratajczak, Mitchell Joblin, Martin Ringsquandl, and Marcel Hildebrandt. Task-driven knowledge graph filtering improves prioritizing drugs for repurposing. BMC bioinformatics, 23(1):1–19, 2022.

[32] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. Knowledge graph embedding for link prediction: A comparative analysis. ACM Transactions on Knowledge Discovery from Data (TKDD), 15(2):1–49, 2021.

[33] Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You can teach an old dog new tricks! on training knowledge graph embeddings. 2020.

[34] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. Drum: End-to-end differentiable rule mining on knowledge graphs. Advances in Neural Information Processing Systems, 32, 2019.

[35] Babak Salimi, Harsh Parikh, Moe Kayali, Lise Getoor, Sudeepa Roy, and Dan Suciu. Causal relational learning. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, 2020.

[36] Arjun Sondhi and Ali Shojaie. The reduced pc-algorithm: Improved causal structure learning in large random networks. J. Mach. Learn. Res., 20(164):1–31, 2019.

[37] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. Causation, prediction, and search. MIT press, 2000.

[38] Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Yiqi Wang, Jiliang Tang, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. Investigating and mitigating degree-related biases in graph convoltuional networks. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pages 1435–1444, 2020.

[39] Sudhanshu Tiwari, Iti Bansal, and Carlos R Rivero. Revisiting the evaluation protocol of knowledge graph completion methods for link prediction. In Proceedings of the Web Conference 2021, pages 809–820, 2021.

[40] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. Multi-task feature learning for knowledge graph enhanced recommendation. In The World Wide Web Conference, 2019.

[41] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019.

[42] Zihao Wang, Hang Yin, and Yangqiu Song. Benchmarking the combinatorial generalizability of complex query answering on knowledge graphs. In NeurIPS Datasets and Benchmarks Track, 2021.

[43] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, pages 726–735, 2021.

[44] Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. Advances in neural information processing systems, 30, 2017.

[45] Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. Aser: A large-scale eventuality knowledge graph. In Proceedings of the web conference 2020, pages 201–211, 2020.

[46] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. DAGs with NO TEARS: Continuous Optimization for Structure Learning. In Advances in Neural Information Processing Systems, 2018.

[47] Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Learning sparse nonparametric DAGs. In International Conference on Artificial Intelligence and Statistics, 2020.