Paths to Causality: Finding Informative Subgraphs Within Knowledge Graphs for Knowledge-Based Causal Discovery

Yuni Susanti
Fujitsu Limited
Kawasaki, Japan
susanti.yuni@fujitsu.com

Michael Färber
ScaDS.AI, TU Dresden
Dresden, Germany
michael.faerber@tu-dresden.de

Abstract

Inferring causal relationships between variable pairs is crucial for understanding multivariate interactions in complex systems. *Knowledge-based causal discovery*—which involves inferring causal relationships by reasoning over the *metadata* of variables (e.g., names or textual context)—offers a compelling alternative to traditional methods that rely on observational data. However, existing methods using Large Language Models (LLMs) often produce unstable and inconsistent results, compromising their reliability for causal inference. To address this, we introduce a novel approach that integrates Knowledge Graphs (KGs) with LLMs to enhance knowledge-based causal discovery. Our approach identifies informative *metapath*-based subgraphs within KGs and further refines the selection of these subgraphs using *Learning-to-Rank*-based models. The top-ranked subgraphs are then incorporated into *zero-shot* prompts, improving the effectiveness of LLMs in inferring the causal relationship. Extensive experiments on biomedical and open-domain datasets demonstrate that our method outperforms most baselines by up to 44.4 points in F1 scores, evaluated across diverse LLMs and KGs. Our code and datasets are available on GitHub.[1]

CCS Concepts

• **Computing methodologies** → **Natural language processing**; **Causal reasoning and diagnostics**.

Keywords

causal discovery; large language models; knowledge graphs

ACM Reference Format:
Yuni Susanti and Michael Färber. 2025. Paths to Causality: Finding Informative Subgraphs Within Knowledge Graphs for Knowledge-Based Causal Discovery. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737076

1 Introduction

Many of the most critical research questions involve inferring causal relationships [12]. For example, a physician must understand the potential side effects of a drug on a patient, as well as how it affects disease progression, before recommending a specific drug or treatment. Thus, inferring causal relationships between pairs of variables, such as `(smoking, lung cancer)`, is a fundamental step in causal discovery, as it enhances our understanding of the complex interactions among interconnected variables. Consequently, uncovering causal information through the analysis of observational data—a process known as *causal discovery*—has recently garnered significant attention [46].

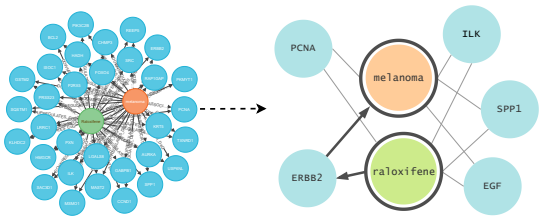

Figure 1: 1-hop connections between `(raloxifene, melanoma)` in Hetionet [18]. Relation types are omitted for readability.

Conventionally, causal discovery involves learning causal relations from observational data by using statistical methods and algorithms to infer causal structures and dependencies between variables, such as with PC and FCI algorithms [45]. Recently, Large Language Models (LLMs) offer a new perspective to tackle the causal discovery problem by reasoning on the *metadata* associated with variables (e.g., variable names) instead of their actual data values–an approach referred to as *knowledge-based causal discovery* [25, 50]. In addition to being less dependent on large observational data, LLMs offer extensive prior knowledge drawn from diverse sources, such as scientific literature, databases, and domain-specific texts. This provides a rich context that observational data alone cannot offer. Thus, we argue that LLMs possess the capability to reveal complex causal patterns that traditional statistical methods might overlook, particularly in high-dimensional observational data.

However, despite their strengths, LLMs often produce unstable results and may produce *hallucinations* during reasoning, leading to incorrect conclusions and reducing their performance and reliability [19, 58]. These issues significantly compromise the reliability of LLMs, especially for causal reasoning in sensitive domains such as healthcare and biomedicine. To address these issues, knowledge graphs (KGs) with their rich and diverse structured information have been integrated to enhance LLMs' reasoning, especially for question answering (QA), which needs both textual understanding and extensive real-world knowledge [34]. KGs such as Wikidata [56] (cross-domain) and `Hetionet` [18] (domain-specific) serve as reliable, interpretable sources for causal reasoning by capturing interconnections between information.

[1] https://github.com/susantiyuni/path-to-causality

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737076

The information in a KG that is relevant for causal reasoning is often complex, involving non-linear dependencies or associations that can be indirectly found through multi-hop paths. For example, in Figure 1, the *subgraph* `[raloxifene-ERBB2-melanoma]` provides valuable insights for inferring the causal relationship between `raloxifene` and `melanoma`. This subgraph indicates that `raloxifene`, a type of drug, upregulates gene `ERBB2`, which has been associated with disease `melanoma`. This suggests the potential of `melanoma` as a side effect of taking `raloxifene`. A challenge lies in identifying such informative subgraphs within KGs for a given variable pair and using them with LLMs. As shown in Figure 1, there are a total of 32 1-hop connections between `raloxifene` and `melanoma` in the KG, which increases to 1,031 subgraphs when extended to 2-hop connections. Naively processing all of these subgraphs would lead to an excessive number of LLM queries. While related works often select contextually relevant information from KGs based on vector similarity [2, 23, 43], we argue that relying solely on similarity-based approach for identifying informative subgraphs is not optimal for the more complex task of causal discovery.

In this paper, we present a novel approach for integrating informative subgraphs from KGs with LLMs to enhance knowledge-based causal discovery. Our method focuses on identifying and leveraging informative *metapath*-based subgraphs within KGs. First, an LLM estimates the relevance of the subgraphs in KGs with respect to determining causal relationships between variable pairs. Then, we develop specialized subgraph ranking models to refine the subgraph selection process, exploring various Learning-to-Rank approaches (*pairwise, pointwise, listwise*) and algorithms (*neural network* and classical *gradient boosting* methods). This marks the first dedicated approach for identifying informative subgraphs within KGs for this task. Finally, we integrate the top-ranked subgraphs into a comprehensive framework for knowledge-based causal discovery. The framework incorporates the target variable pair, instructions, textual context, and top-ranked informative subgraphs into the prompt, enabling zero-shot reasoning by leveraging the pretrained knowledge of LLMs—without requiring large observational datasets or extensive supervised training. Our experiments across multiple benchmark datasets from different domains show significant performance improvements of LLMs in inferring causal relationships between variable pairs, as evaluated across diverse language models and knowledge graphs.

Overall, we make the following contributions:

- We propose a new approach to enhance LLMs' causal reasoning by integrating them with KGs for zero-shot knowledge-based causal discovery, eliminating reliance on observational data.
- We introduce specialized subgraph ranking models based on various Learning-To-Rank methods to refine subgraph selection from KGs, enhancing the performance of LLMs for inferring causal relationships by up to a 44.8-point increase in F1 score.
- We conduct extensive experiments on publicly available biomedical and open-domain datasets, demonstrating the effectiveness of our approach. Our method surpasses most baselines and shows generalizability across different LLMs and KGs.
- We compared our approach to conventional statistical causal discovery methods using observational data, achieving up to a 17-point reduction in Hamming distance and a 25.17-point F1 score improvement. This shows that LLMs, when integrated with KGs, effectively leverage semantic and contextual information of the variable, outperforming purely statistical methods.

2 Related Work

Knowledge-Based Causal Discovery. Unlike *statistical*-based causal discovery which uses purely observational data, *knowledge-based causal discovery* focuses on the metadata associated with variables instead of their data values [25, 50]. This metadata may include variable names or any textual description related to the variables. Traditionally, such metadata-based causal reasoning relied on Subject Matter Experts (SMEs), but LLMs are now capable of providing knowledge that previously required SME expertise. Recent works [13, 25, 55, 60, 64] show that LLMs effectively provide background knowledge for causal discovery, notably, [25] explores causal capabilities of LLMs by experimenting on cause-effect pairs. Their finding suggests that LLM-based prompting methods achieved superior performance than non-LLM approaches. Other recent studies have explored the use of LLMs to extract prior knowledge to be used for statistical-based causal discovery methods [1, 4, 52], however, it mainly relies on observational data. Other works focus on evaluating LLMs' ability to identify causal relations described in text [8, 24, 50, 51], focusing on identifying causal relationships rather than discovering new causal relations, which aligns with our work. Unlike previous research that relies on prompt engineering [8, 13, 22, 25, 55, 64], we enhance LLMs' capabilities for causal discovery by integrating *informative* knowledge from KGs.

KGs and LLMs. Knowledge graphs (KGs) provide a reliable foundation for grounding LLMs' reasoning by organizing relational information and contextual knowledge in a structured manner. This enables retrieval-augmented generation (RAG) techniques, which effectively incorporate external knowledge, and is widely used in knowledge-intensive NLP tasks [27] such as dialogue generation and question answering [2, 3, 21, 23, 30, 43, 59].

Research on integrating KGs with LLMs can be broadly categorized by the granularity of the retrieved information: *entity* [38, 43], *triple* [2, 50], and *subgraph* [23] levels. For instance, in the dialogue generation task, SURGE [23] augments LLMs by incorporating relevant subgraphs from KGs, which are extracted using a similarity-based approach combined with contrastive learning. Conversely, MK-TOD [43] operates at the entity level by identifying relevant KG entities based on vector similarities, also for dialogue generation. Similarly, KAPING [2] utilize embedding similarities to retrieve relevant triples from the KG for zero-shot QA. [50] integrates KGs' structures (triples, subgraphs) into prompt-based learning for predicting causal relations; but it randomly selects information from KGs and relies on supervised learning to fine-tune the LLMs. Given that KGs contain a vast amount of information, we focus on identifying relevant information to ensure that the model uses the most useful knowledge needed for knowledge-based causal discovery.

3 Task Formulation

In this work, we focus on *pairwise* knowledge-based causal discovery: given a pair of entities a and b, i.e., variable or node pairs such as `(raloxifene, melanoma)`, the task is to predict if a causal relationship can be inferred between the pair. We formulate the

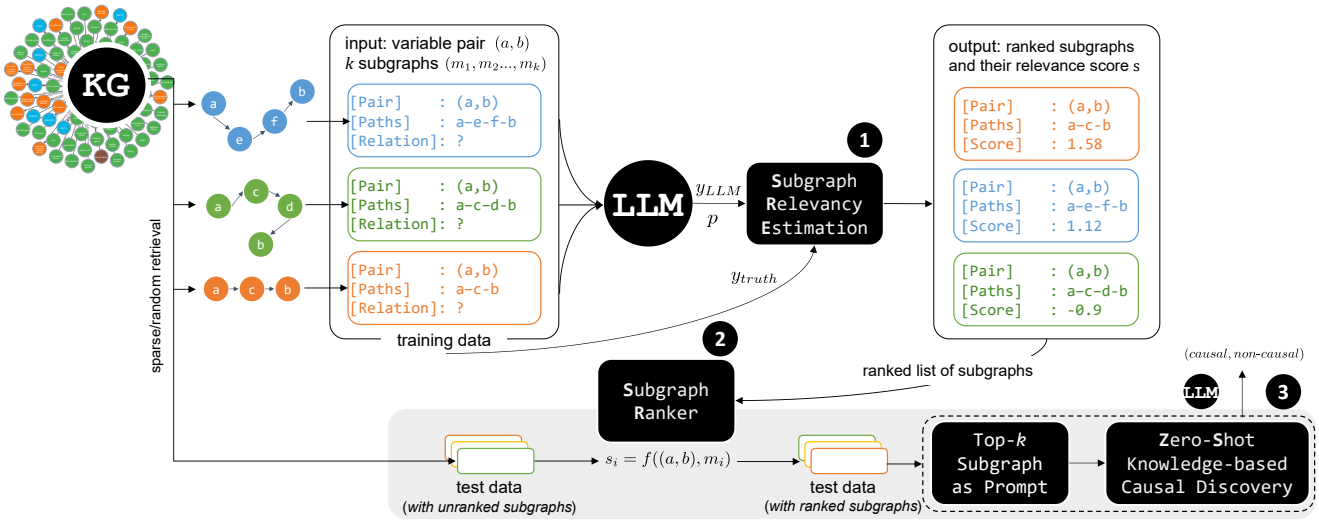

Figure 2: Overview of the proposed approach, composed of three main modules: (1) `Subgraph Relevancy Estimation` (2) `Subgraph Ranker` and (3) `Knowledge-Based Causal Discovery with Subgraphs as Prompt` (zero-shot).

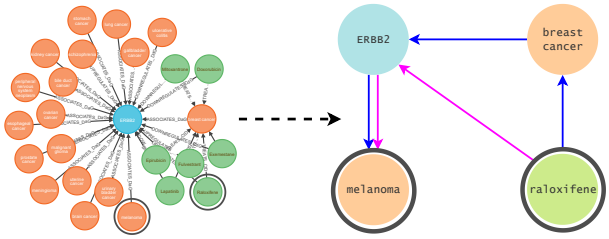

Figure 3: Metapaths connecting `raloxifene` and `melanoma`: (1) `[raloxifene-ERBB2-melanoma]` with `[drug-gene-disease]` node sequence, and (2) `[raloxifene-breast cancer-ERBB2-melanoma]` with `[drug-disease-gene-disease]` node type sequence.

task as *classification task*, classifying the relation as *causal* or *non-causal*. We evaluate our approach on dataset $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$, where $\mathcal{X}$ is a set of data instance and $\mathcal{Y} = \{causal, non\text{-}causal\}$ is a set of relation labels. Each instance $x \in \mathcal{X}$ consists of a token sequence $x = \{w_1, w_2, \ldots, w_{|n|}\}$ and the spans of a marked variable pair, and is annotated with a label $y_x \in \mathcal{Y}$.

4 Proposed Approach

In a heterogeneous network like a KG, two nodes can be connected through many different paths. These paths–each being a *subgraph* with a specific sequence of node types and relationships from an origin node to a destination node–are referred to as *metapaths* [49]. Figure 3 illustrates two metapaths with different node type sequences connecting `raloxifene` and `melanoma`.

Such metapaths can capture various contextual and relational nuances, suggesting different underlying meanings, semantic interpretations, and potential causal implications. As a result, they are frequently used in explaining machine learning models [7, 65] and for network similarity analysis, especially in the biomedical domain [40, 57, 62]. In this domain, the term "metapath" itself refers to specific node type combinations thought to be *informative* or meaningful [33]. Taking an example from Figure 3, to determine the likelihood of a drug causing a specific disease, the metapath `[raloxifene-ERBB2-melanoma]` with `[drug-gene-disease]` node type sequence is more informative than `[drug-disease-gene-disease]`. The former suggests a direct biological mechanism where a drug (`raloxifene`) influences (`upregulates`) a gene (`ERBB2`), which has been associated with a disease (`melanoma`), making it highly informative for inferring causality between the drug and the disease.

Inspired by this, we utilize the concept of metapaths to identify *informative* subgraphs within KGs. Given that many ***metapath-based subgraphs*** may exist for a variable pair, the challenge is to pinpoint the most informative ones for determining causality. By utilizing these subgraphs, we aim to improve the accuracy and reliability of LLMs for knowledge-based causal discovery.

Figure 2 illustrates the overview of our proposed approach, composed of the following three modules:

1. `Subgraph Relevancy Estimation` utilizes LLMs to estimate the relevance of subgraphs, with the primary objective of generating a ranked list of subgraphs based on their informativeness for determining causal relations between variable pairs.
2. `Subgraph Ranker` distills the LLMs' capability in inferring causal relationship by training specialized subgraph ranker models using the generated ranked list of subgraphs. This module further refines the subgraph selection, ultimately selecting the most informative subgraph for inferring causality.
3. `Knowledge-Based Causal Discovery with Subgraphs as Prompt` performs *zero-shot* knowledge-based causal discovery on variable pairs given the top-ranked subgraphs ranked by `Subgraph Ranker` model, as a prompt for LLMs.

We elaborate the proposed approach in the following subsections. We start with preliminaries (§4.1), followed by the detailed explanation of `Subgraph Relevancy Estimation` (§4.2) and `Subgraph`

Ranker (§4.3) modules. Lastly, we explain the integration of informative subgraphs from KGs with LLMs in **K**nowledge-Based **C**ausal **D**iscovery with Subgraphs as Prompt (§4.4).

4.1 Preliminaries

Definition 4.1.1. Subgraph. Let $\mathcal{G} = (V, E)$ be a graph with a vertex set V and an edge set E. A subgraph $\mathcal{G}' = (V', E')$ of a graph $\mathcal{G}$ is then another graph whose vertex set and edge set are subsets of those of $\mathcal{G}$, i.e., where $V' \subseteq V$ and $E' \subseteq E$.

Definition 4.1.2. Knowledge Graph. A knowledge graph is a specific type of graph representing a network of entities and the relationships between them. Formally, we define a knowledge graph as a directed labeled graph $\mathcal{KG} = (N, E, R, \mathcal{F})$, where N is a set of nodes (entities), $E \subseteq N \times N$ is a set of edges (relations), R is a set of relation labels, and $\mathcal{F} : E \rightarrow R$, is a function assigning edges to relation labels. For instance, assignment label r to an edge $e = (x, y)$ can be viewed as a triple (x, r, y), e.g., (Tokyo, IsCapitalOf, Japan).

Definition 4.1.3. Metapath-based Subgraph. In this work, we focus on *metapath*-based subgraphs within a KG. Formally, a metapath-based subgraph $\mathcal{M}$ can be defined as a path that is denoted in the form of $T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \ldots \xrightarrow{R_n} T_{n+1}$ describing a composite relation $R = R_1 \circ R_2 \circ \ldots \circ R_n$ between node types T_1 and T_{n+1}[49]. This illustrates how two nodes are interconnected through a series of relationships and sequences of node types, providing context, i.e., semantics, and potential causal meaning within a KG. Note that when we refer to a **subgraph** in this paper, we specifically mean a **metapath-based subgraph**. We also use the terms **variable** and **entity** pairs interchangeably throughout the paper.

4.2 Subgraph Relevancy Estimation

In this module, we use LLMs to estimate how *informative* or *relevant* a subgraph is for inferring causality. The goal is to generate a ranked list of subgraphs based on their relevance for accurately determining causal connections between variable pairs. Formally, given a pair (a, b) and k subgraphs $(m_1, m_2, \ldots, m_k)$ extracted from a KG, **S**ubgraph **R**elevancy **E**stimation produces a ranking $(r_1, r_2, \ldots, r_k)$, where $r_i \in \{1, 2, \ldots, k\}$ denotes the rank of subgraph m_i.

As illustrated in Figure 2 (1), given the variable pair and a set of k subgraphs[2] as input prompt to the LLM, we instruct the LLM to predict the causal relationship between the variable pair by outputting the relation, e.g., $(causal, non\text{-}causal)$. A subgraph is considered as relevant if the LLM predicts the relation correctly, according to the human-labeled ground truth relation. The relevance score s_i for each subgraph is then calculated based on the probability p of the LLM generating the prediction, as follows:

$$s_i = \begin{cases} 1 + p('causal/non\text{-}causal') & if \quad y_{LLM} = y_{truth} \\ 1 - p('causal/non\text{-}causal') & if \quad y_{LLM} \neq y_{truth} \end{cases} \tag{1}$$

where $p('causal/non\text{-}causal')$ denotes the log probability of LLMs generating the prediction, y_{LLM} denotes the output of an LLM, and y_{truth} denotes the ground truth of the relation. Finally, we finalize the ranking of the subgraphs: given the set of subgraphs $M = \{m_1, m_2, \ldots, m_k\}$ with scores $\{s_1, s_2, \ldots, s_k\}$, we assign the ranking $r_i = \text{argsort}_i(s_1, s_2 ..., s_n)$ for the subgraphs. The final output is a dataset $\mathcal{R}$ containing variable pairs with subgraphs ranked based on their *informativeness* in inferring the causal relation. We provide examples of prompts and ranked lists of subgraphs in Appendix C.

[2]We employ *sparse/random retrieval* to filter the subgraphs (see Appendix B.

4.3 Subgraph Ranker

Ranking plays a crucial role in our approach, as KGs often include multiple subgraph connections between a variable pair. In our approach, we introduce specialized models for subgraph ranking to further refine the subgraph selection process, utilizing additional *meta* information from the metapath-based subgraphs, e.g., node/edge type sequences. As illustrated in Figure 2 (2), we train a dedicated subgraph ranker using the ranked list of subgraphs produced by the **S**ubgraph **R**elevancy **E**stimation module, thereby *distilling* the LLM's capability to infer causal relationships.

Ranking problems are prevalent in Information Retrieval field, including tasks like web search ranking and text retrieval. In these tasks, given a query q and a relevant list of documents D, *Learning-to-Rank* (LTR) methods learn a function to predict the relevance scores of the documents based on the given query, arranging the documents in an ordered list [29]. Inspired by this, we formulate the subgraph ranking as an Information Retrieval task. Our proposed **S**ubgraph **R**anker takes a variable pair (a, b) and the set of subgraphs $(m_1, m_2 \ldots, m_k)$ to train an LTR-based ranker model. This model ranks the subgraphs based on their *informativeness* in inferring the causal relationship between the variables (a, b).

4.3.1 **Training Objective**. Formally, **S**ubgraph **R**anker is a ranking model $s_i = f_\theta((a, b), m_i)$ with parameter θ which computes a relevance score s_i for input pair $((a, b), m_i)$. Let $M = \{m_1, m_2, \ldots, m_k\}$ be a set of subgraphs, with their corresponding predicted scores $\{s_1, s_2, \ldots, s_k\}$ and ground truth relevance scores $\{y_1, y_2, \ldots, y_k\}$. Using the ranked dataset $\mathcal{R}$ generated by **S**ubgraph **R**elevancy **E**stimation (§4.2), we consider the following ***pointwise***, ***pairwise*** and ***listwise*** objective functions to optimize the ranker model.

Pointwise. In the pointwise approach, the subgraph ranking is formulated as a *regression* task. That is, each subgraph is scored independently and the loss function minimizes the difference between the predicted and ground truth relevance scores. We consider a loss function based on *Root Mean Square Error* (RMSE):

$$\mathcal{L}_{\text{RMSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{k}(s_i - y_i)^2} \tag{2}$$

Pairwise. In the pairwise approach, we consider a loss function based on RankNet [5], defined as follows:

$$\mathcal{L}_{\text{RankNet}} = \sum_{i=1}^{k}\sum_{j=1}^{k}\mathbb{I}_{r_i<r_j}\log(1 + \exp(s_i - s_j)) \tag{3}$$

RankNet is a pairwise loss that measures the correctness of relative subgraph orders, thus, for a variable pair with k subgraphs, we can construct in total $k(k-1)/2$ pairs.

Listwise. In the listwise approach, we consider the ListNet [6] objective function. We use ListNet softmax loss as defined in Eq. 4,

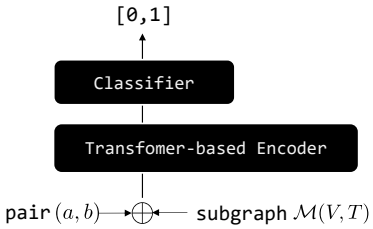

Figure 4: Subgraph Ranker cross encoder model architecture with Transformer-based encoder.

implemented in [36]. Instead of modeling the probability of a pairwise comparison using scoring difference, ListNet models the probability of the entire ranking result.

$$\mathcal{L}_{\text{ListNet}} = -\sum_{i=1}^{k} \text{softmax}(y)_i \times \log(\text{softmax}(s)_i) \tag{4}$$

4.3.2 **Model Architecture**. As shown in Figure 4, our **S**ubgraph **R**anker is built on a Transformer-based cross-encoder, a model widely used for text ranking [14, 32]. A cross-encoder processes two text inputs simultaneously, encoding them into a single representation and generating a relevance score. We adapt this approach by concatenating variable pair (a, b) with subgraph $\mathcal{M}$, as follows:

$$(a, b) \oplus \mathcal{M}\{V, T\} = \texttt{[CLS]}a, b\texttt{[SEP]}(t_1v_1, t_2v_2, \ldots, t_nv_n) \tag{5}$$

where $\mathcal{M}\{V, T\}$ is a subgraph $\mathcal{M}$ with path length n containing a set of nodes $V = (v_1, v_2, \ldots v_{|n|})$ with their corresponding node types $T = (t_1, t_2, \ldots t_{|n|})$. We use the *meta* information *node type* for training the ranker, as it is considered informative [49]. The node v_1 corresponds to a, and the last node v_n corresponds to b. We insert special tokens [CLS] and [SEP] as shown in Eq. 5 following the practice of the Transfomer-based encoder [11]. Lastly, we use the representation of [CLS] to estimate the relevance score with a classification layer, as illustrated in Figure 4.

Studies have shown that classical Learning-to-Rank methods, especially Gradient Boosting Decision Trees (GBDT), often outperform neural-based approaches, including cross encoder [37]. Thus, we train a GDBT-based subgraph ranker as an alternative. We concatenate the pair with the subgraph as in Eq. 5 and train an *n-gram* model on these combined sequences. This model is then used to extract features for training the gradient boosting-based subgraph ranker. Details of this approach are provided in Appendix D.2.

4.4 Knowledge-Based Causal Discovery with Subgraphs as Prompt

In this module, we perform the knowledge-based causal discovery task. For a given variable pair (a, b), we prompt an LLM to predict if a causal relationship can be inferred between them. The prompt also includes textual context and the top-k subgraphs, as ranked by the **S**ubgraph **R**anker model. The following describes the details:

4.4.1. Selecting Top-k Subgraphs. Given the ranked subgraphs produced by **S**ubgraph **R**anker , we select the top-k most relevant subgraphs. Let $\{(m_1, s_1), (m_2, s_2), \ldots, (m_n, s_n)\}$ be the set of subgraphs and their scores for a pair (a, b). The top-k subgraphs are selected as follows:

$$\text{top-k}((a, b)) = \{m_i \mid i \in \text{argsort}(s_1, s_2, \ldots, s_n)[:k]\} \tag{6}$$

where $\text{argsort}(s_1, s_2, \ldots, s_n)$ is a function that returns the indices of the sorted scores and $[:k]$ selects the top-k indices from the list.

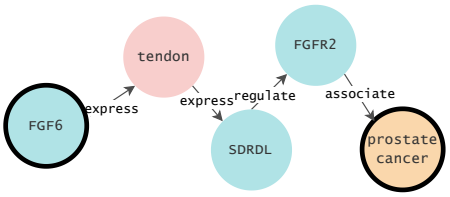

Figure 5: Subgraph example. Different node colors represent different node types: gene, anatomy, disease

4.4.2. Subgraphs as Prompt. We further transform the top-k subgraphs into prompts by converting the structures into a sequence of elements that can be processed sequentially. We *verbalize* the subgraphs by converting the sequential relationships between nodes into natural language text, as follows:

$$\begin{aligned}&\mathcal{M}\{V, E, T\} = \text{“Relation paths between the pair: ”}\\ &\{(t_1v_1, e_1, t_2v_2), (t_2v_2, e_2, t_3v_3), \ldots, (t_{n-1}v_{n-1}, e_n, t_nv_n)\}\end{aligned} \tag{7}$$

where $\mathcal{M}\{V, E, T\}$ is a subgraph $\mathcal{M}$ with path length n containing a set of nodes $V = (v_1, v_2, \ldots v_{|n|})$ with their corresponding node types $T = (t_1, t_2, \ldots t_{|n|})$ Each edge e_i connects node v_i to its immediate neighbor v_{i+1} within the subgraph. The string “Relation paths...” are optional and can be replaced by other tokens.

Consider a subgraph as illustrated in Figure 5. With Eq. 7, the subgraph can be linearized into a prompt, as follows:

```
Gene FGF6 --express--> anatomy tendon, anatomy tendon --express-->
gene SDRDL, gene SDRDL --regulate--> gene FGFR2, gene FGFR2
--associate--> disease prostate cancer
```

This representation captures the sequential relationships between entities and their corresponding relations in a linear format suitable for use in prompts. As it is technically a *metapath*, it includes the *meta* information (node types T), as well. Studies have shown that giving more textual context does not always lead to better performance, and sometimes a simpler prompt can be more beneficial than a complex one [35, 41]. Therefore, we can also opt to exclude the node types T and relation labels E from the prompt. In such case, the prompts would be as follows:

```
FGF6 --express--> tendon --express--> SDRDL --regulate--> FGFR2 --associate-->
prostate cancer
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
FGF6→ tendon→ SDRDL→ FGFR2→ prostate cancer
```

These representations capture the subgraph structure in its simplest form, while still suitable for processing by LLMs designed for sequential inputs. The arrow symbols “→” are optional and can be replaced by other fitting symbols or tokens (e.g., hyphen “–”).

4.4.3. Zero-Shot Knowledge-Based Causal Discovery. In this final step, the LLM is tasked with inferring a causal relationship

between a variable pair based on the provided information in the prompt. We perform knowledge-based causal discovery with no training examples or parameter updates to the LLMs, i.e., a *zero-shot* approach. A zero-shot approach enables the model to predict causal relationships between unseen variable pairs without any training/fine-tuning to the model. In our work, this method is preferable given the scarcity of causal datasets.

Formally, given $\mathcal{I}$ as the instruction, x as the textual context, and $\mathcal{M}$ as the subgraph prompt, the final full prompt x' as the input to the LLMs for the pair (a, b) can be formulated as follows:

$$x' = [\mathcal{I}][x]\ [\mathcal{M}]\ \texttt{The relation between}\ [a]\ \texttt{and}\ [b]\ \texttt{is} \quad (8)$$

Additional tokens "`The relation between...`" are optional and can be replaced by other fitting templates. Additional information on this step is provided in Appendix E.

5 Evaluation

5.1 Experiment Settings

We evaluate the proposed approach in a knowledge-based causal discovery task: given a pair of entities a and b, the task is to predict if a causal relationship can be inferred between the pair. We specifically focus on integrating informative subgraphs from KGs to enhance the internal knowledge of the LLMs in inferring causality.

To identify informative subgraphs in the KGs, we run **S**ubgraph **R**elevancy **E**stimation (§4.2) using the reliable and cost-efficient `mistral-7b` [20] model, generating an up to k=10 ranked subgraphs dataset per variable pair (example in Appendix C). Next, to further refine the subgraph selection, we train **S**ubgraph **R**anker (§4.3) model using this dataset, experimenting with several cross encoders and `XGBoost` [9]-based models. Due to comparable performance across cross encoders, (provided in Appendix D), we selected (1) `roberta-base` cross encoder and (2) `XGBoost`-based models for ranking the subgraphs that would be used for the subsequent zero-shot knowledge-based causal discovery (§4.4) experiments.

Implementation details for each module of the proposed approach are in Appendices C, D, E, with baseline details in Appendix F. Computational complexity and hardware specifications are in Appendix G. Code and resources are available on GitHub.

5.2 Choice of LLMs and KGs

Recent studies have shown that 7B LLMs deliver relatively strong performance across various tasks [28, 54]. Thus, we experiment with the following widely-used 7–8B LLMs:

- `mistral-7b-instruct`: a highly efficient model with high-level performance, surpassing even 16-34B models [20].
- `llama-3-8b-instruct`: the 8B model from Meta's Llama series offering improved reasoning capabilities [31].
- `gemma-7b-it`: a lightweight LLM based on Google's Gemini [53].

For the knowledge graphs, we select the following:

- `Wikidata` [56], as the open-domain KG.
- `Hetionet` [18], a domain-specific KG assembled from 29 different databases, covering genes, compounds, and diseases.

We select `Wikidata` for its wide coverage of subjects and `Hetionet` since we primarily evaluate on biomedical domain datasets. Details on querying the KGs and subgraphs extraction are in Appendix B.

5.3 Datasets

Since causality is often studied in the biomedical domain, we primarily evaluate our approach in this field using `GENEC` [50] for gene-gene relationships, `ADE` [16] for drug-side effect relationships, and `COMAGC` [26] for gene-disease relationships. Additionally, we evaluate it on the open-domain dataset `SEMEVAL` [17]. The dataset includes the ground truth causal relationships labeled by human experts, allowing us to assess the models' prediction performance in the evaluation. The dataset details and examples are in Appendix A.

5.4 Method Comparison

We compare the following methods: Models 1–4 serve as baselines, including a model without any knowledge from KGs and models enhanced with subgraphs ranked using baseline methods. Models 5–8, labeled "`Subgraph Ranker`", represent our proposed approach, where the model is provided with subgraphs ranked by variations of our proposed **S**ubgraph **R**anker .

(1) **`no-subgraph`**: only the variable pair, without any subgraphs.
(2) **`random subgraph selection`**: providing the models with randomly selected subgraphs.
(3) **`similarity-based ranker`**: text similarity-based method [2, 23, 43] adapted to subgraph ranking, detailed in Appendix F.1.
(4) **`GPT-based ranker`**: instructing LLMs (`GPT`) to directly rank subgraphs, adapted from [48] and detailed in Appendix F.2.
(5) **`Subgraph Ranker (rmse-cross encoder)`**: Our proposed **S**ubgraph **R**anker based on cross encoder model with *pointwise* objective function `RMSE`, detailed in Appendix D.1.
(6) **`Subgraph Ranker (ranknet-cross encoder)`**: Our proposed **S**ubgraph **R**anker based on cross encoder model with *pairwise* objective function `RankNet`, detailed in Appendix D.1.
(7) **`Subgraph Ranker (listnet-cross encoder)`**: Our proposed **S**ubgraph **R**anker based on cross encoder model with *listwise* objective function `ListNet`, detailed in Appendix D.1.
(8) **`Subgraph Ranker (XGBoost)`**: Our proposed **S**ubgraph **R**anker based on gradient boosting `XGBoost`[9], see Appendix D.2.

6 Results and Discussion

Table 1 & 2 summarize the evaluation results for the biomedical and open-domain datasets. To summarize, methods incorporating subgraphs from KGs consistently outperformed the no-subgraph baseline across all experiments and datasets, with improvements of up to 44.4 points in F1 scores, evaluated using three different LLMs. For subgraph ranking, our proposed **S**ubgraph **R**anker emerged as the top-performing model in nearly all experiments, except in one case where the GPT-based ranker outperforms it. We provide detailed analysis and discussions of the results in the following.

1. Subgraph vs. No-Subgraph: Does the Subgraph Help LLMs Infer Causality? Methods incorporating subgraphs consistently outperform no-subgraph baselines across experiments and datasets. This evaluation, conducted with three different LLMs, shows significant performance improvements in knowledge-based causal discovery when enhanced with knowledge from KGs. The most notable result was observed with the `llama-3-8b` model on the `COMAGC` dataset, where our proposed approach achieved an F1 score of 77.78 compared to 33.33 for the no-subgraph baseline–a difference of 44.4

	COMAGC [26]			GENEC [50]			ADE [16]		
	P	R	F1	P	R	F1	P	R	F1
				mistral-7b-instruct [20]					
(baseline) no-subgraph	66.67	47.06	55.17	27.27	16.67	20.69	100.00	36.84	53.85
(baseline) random subgraph selection	90.00	52.94	66.67	100.00	16.67	28.57	100.00	52.63	68.97
(baseline) similarity-based ranker	76.92	58.82	66.67	100.00	16.67	28.57	100.00	55.26	71.19
(baseline) GPT-based ranker	93.33	82.35	**87.50**	60.00	16.67	26.09	100.00	57.89	**73.33**
(ours) Subgraph Ranker (rmse-cross encoder)	86.67	76.47	81.25	83.33	27.78	**41.67**	100.00	55.26	71.19
(ours) Subgraph Ranker (ranknet-cross encoder)	85.71	70.59	77.42	83.33	27.78	**41.67**	100.00	63.16	**77.42**
(ours) Subgraph Ranker (listnet-cross encoder)	81.25	76.47	78.79	100.00	27.78	**43.48**	100.00	57.89	**73.33**
(ours) Subgraph Ranker (XGBoost)	92.86	76.47	**83.87**	80.00	22.22	34.78	100.00	57.89	**73.33**
				llama-3-8b-instruct [31]					
(baseline) no-subgraph	57.14	23.53	33.33	66.67	44.44	53.33	100.00	50.00	66.67
(baseline) random subgraph selection	61.11	64.71	62.86	43.75	38.89	41.18	100.00	71.05	83.08
(baseline) similarity-based ranker	60.00	70.59	64.86	46.67	38.89	42.42	100.00	76.32	86.57
(baseline) GPT-based ranker	68.42	76.47	**72.22**	55.56	55.56	**55.56**	100.00	78.95	88.24
(ours) Subgraph Ranker (rmse-cross encoder)	63.64	82.35	71.79	50.00	38.89	43.75	100.00	76.32	86.57
(ours) Subgraph Ranker (ranknet-cross encoder)	73.68	82.35	**77.78**	58.82	55.56	**57.14**	100.00	81.58	**89.86**
(ours) Subgraph Ranker (listnet-cross encoder)	65.00	76.47	70.27	61.54	44.44	51.61	100.00	81.58	**89.86**
(ours) Subgraph Ranker (XGBoost)	60.00	70.59	64.86	46.15	33.33	38.71	100.00	86.84	**92.96**
				gemma-7b-it [53]					
(baseline) no-subgraph	50.00	50.00	50.00	75.00	16.67	27.27	100.00	65.79	79.37
(baseline) random subgraph selection	60.71	100.00	75.56	52.94	50.00	51.43	100.00	76.32	86.57
(baseline) similarity-based ranker	62.96	100.00	77.27	50.00	61.11	**55.00**	100.00	78.95	88.24
(baseline) GPT-based ranker	62.96	100.00	77.27	50.00	55.56	52.63	100.00	81.58	**89.86**
(ours) Subgraph Ranker (rmse-cross encoder)	62.96	100.00	77.27	47.37	50.00	48.65	100.00	86.84	**92.96**
(ours) Subgraph Ranker (ranknet-cross encoder)	65.38	100.00	**79.07**	57.89	61.11	**59.46**	100.00	81.58	**89.86**
(ours) Subgraph Ranker (listnet-cross encoder)	62.96	100.00	**78.27**	52.63	55.56	54.05	100.00	81.58	**89.86**
(ours) Subgraph Ranker (XGBoost)	60.71	100.00	75.56	56.25	50.00	52.94	100.00	73.68	84.85

Table 1: Precision (P), Recall (R), and F1 scores on biomedical datasets. Values marked with underline and bold represent the top and second-best F1 scores per dataset under different LLMs. The subgraphs were extracted from the Hetionet [18] KG.

	P	R	F1	P	R	F1	P	R	F1
	mistral-7b-instruct			llama-3-8b-instruct			gemma-7b-it		
(baseline) no-subgraph	60.00	25.00	35.29	71.43	41.67	52.63	50.00	91.67	64.71
(baseline) random subgraph selection	50.00	33.33	40.00	55.00	91.67	68.75	55.00	91.67	68.75
(baseline) similarity-based ranker	71.43	41.67	**52.63**	57.14	100.00	**72.73**	55.00	91.67	68.75
(baseline) GPT-based ranker	71.43	41.67	**52.63**	50.00	91.67	64.71	57.89	91.67	**70.97**
(ours) Subgraph Ranker (rmse-cross encoder)	66.67	50.00	**57.14**	60.00	100.00	**75.00**	57.89	91.67	**70.97**
(ours) Subgraph Ranker (ranknet-cross encoder)	66.67	33.33	44.44	57.89	91.67	70.97	57.89	91.67	**70.97**
(ours) Subgraph Ranker (listnet-cross encoder)	57.14	33.33	42.11	57.89	91.67	70.97	57.89	91.67	**70.97**
(ours) Subgraph Ranker (XGBoost)	55.56	41.67	47.62	57.14	100.00	**72.73**	61.11	91.67	**73.33**

Table 2: Precision (P), Recall (R), and F1 scores on open-domain dataset SEMEVAL[17], with subgraphs from Wikidata [56].

points in F1 scores. In addition, even randomly selected subgraphs, i.e., the random subgraph selection method, achieved better scores than the no-subgraph approach, with an average improvement of 12.5 points in F1 scores. This shows that metadata alone in prompts is not sufficient for accurate causal inference, and that integrating KGs significantly aids LLMs in inferring causality between variable pairs, leading to more accurate and reliable causal discovery.

2. Our Subgraph Ranker vs. LLM (GPT)-Based Ranker: How to Best Select Informative Subgraphs? In almost all experiments, at least one variation of our Subgraph Ranker is the best-performing model, except in one case where GPT-based ranker outperforms it (83.87 *vs*. 87.50 on COMAGC dataset with mistral-7b). This shows that our Subgraph Ranker excels in subgraph selection due to its specialized training on LLM-generated subgraph ranking data produced by Subgraph Relevancy Estimation method. Additionally, incorporating *meta* information from metapath-based subgraphs (e.g., node types) into the specialized model further optimized it for the ranking task. On the other hand, GPT-based ranker adapts the *permutation generation* [48] method for subgraph ranking by instructing GPT to rank subgraphs for a given variable pair. While easy to implement, it is less effective than our approach, as LLMs like GPT are not optimized for specific ranking tasks.

	P ↑	R ↑	F1 ↑	HD ↓	NHD ↓
(statistical-based) PC Algorithm [47]	38.00	52.25	44.00	24	0.198
(statistical-based) Exact Search (A*) [63]	18.00	31.84	23.00	31	0.256
(statistical-based) DirectLingam [44]	28.00	50.40	36.00	29	0.240
(LLM-based-baseline) no-subgraph	100.00	15.79	27.27	16	0.132
(ours, LLM+KG) Subgraph Ranker (rmse-cross encoder)	100.00	42.11	59.20	12	0.100
(ours, LLM+KG) Subgraph Ranker (ranknet-cross encoder)	100.00	31.58	48.00	13	0.107
(ours, LLM+KG) Subgraph Ranker (listnet-cross encoder)	100.00	63.16	**77.42**	**7**	**0.057**
(ours, LLM+KG) Subgraph Ranker (XGBoost)	100.00	42.11	**59.26**	**11**	**0.091**

Table 3: Additional experiments on SACHS [42] observational data.

To conclude, our proposed **S**ubgraph **R**anker offers an effective solution for selecting informative subgraphs from KGs to guide LLMs in knowledge-based causal discovery: it combines the strengths of LLMs in assessing subgraph *informativeness* with a specialized trained model to produce an optimal ranking.

3. Our Subgraph Ranker vs. Similarity-Based Ranker. In tasks such as QA and dialogue generation, similarity-based methods (e.g., cosine similarity of embeddings) is often employed to rank relevant information from KGs to ground the LLMs' responses [2, 23, 43]. However, our experimental results show that this method (`similarity-based ranker`) is less effective than our approach for ranking subgraphs in KGs for knowledge-based causal discovery, with an average difference of 7.74 points in F1. This discrepancy likely arises because inferring causal relationship involves understanding deeper, often non-linear relationships between nodes, which similarity-based methods may fail to capture adequately.

4. Cross Encoder vs. Gradient Boosting: Neural or Classical Subgraph Ranker? Our cross encoder-based **S**ubgraph **R**anker uses neural networks to rank the subgraphs, relying on contextual embeddings generated by the encoder model. In contrast, our gradient boosting-based **S**ubgraph **R**anker ranks subgraphs using decision trees that learn from manually defined features, such as *n-grams*. Overall, while cross encoder-based subgraphs rankers generally deliver superior performance (9 out of 12 cases), the gradient boosting-based ranker performs relatively well in our task, emerging as the top model in 2 out of 12 cases (`SEMEVAL` with `gemma-7b` and `ADE` with `llama-3-8b`). According to [15], gradient-boosting excels on structured data, and although the subgraphs are represented as token sequences, they retain their structured nature from KGs, ensuring effective ranking. Meanwhile, within the variation of the LTR-based cross encoders, the pairwise approach (`ranknet`) consistently performs best, followed by pointwise (`rmse`) and listwise (`listnet`). The pairwise approach allows the model to learn subtle differences between subgraphs, leading to better ranking prediction.

5. Wikidata vs. Hetionet. For the biomedical datasets, using subgraphs from the `Hetionet` KG improved F1 scores by up to 44.4 points over the no-subgraph baseline. For open-domain dataset using `Wikidata`, the improvement was as high as 22.37 points, indicating its flexibility regarding the choice of KGs. The smaller improvement with `Wikidata` suggests that the LLMs already possess partial knowledge of certain relationships, likely learned from similar sources such as Wikipedia. However, our approach enhances this by guiding the model to focus on relevant causal paths, resulting in improved performance.

6.1 Comparison to Statistical-Based Causal Discovery

Conventional causal discovery uses statistical methods on observational data, while our approach leverages LLMs and knowledge graphs to infer causal relationships from metadata. Thus, to comprehensively evaluate our approach, we conducted additional experiments comparing it to conventional statistical-based methods. Specifically, we evaluated on the causal benchmark `SACHS` [42] protein causality dataset, comparing our method against conventional statistical-based causal discovery methods: `PC Algorithm` [47], `Exact Search/A*` [63], and `DirectLingam` [44]. The dataset was selected for its suitability in this evaluation scenario, as it contains real-world observational data needed for statistical-based analysis and a ground truth causal graph created by human experts, allowing empirical evaluation. The results are summarized in Table 3. Some of the statistical-based scores are obtained from [52].

To sum up, our approach consistently outperforms statistical-based methods, achieving up to a 33.42-point improvement in F1 scores (77.42 *vs.* 44.00) and a 17-point reduction in Hamming Distance (HD; the number of mismatches in the inferred adjacency matrix) (7 *vs.* 24). This demonstrates that LLMs, when integrated with structured information from KGs, effectively leverage semantic and contextual information embedded within the KGs, which allows them to infer causal relationship more accurately than traditional statistical methods. Unlike purely statistical approaches that rely heavily on observational datasets, our method does not require such data, making it more versatile and adaptable in real-world applications where observational data may be scarce or unavailable.

7 Conclusion

In this paper, we introduced a novel approach integrating KGs with LLMs to enhance knowledge-based causal discovery. Specifically, we utilized LLMs to identify informative subgraphs within KGs relevant to causal relationships and explored Learning-to-Rank-based specialized subgraph ranker models to further refine the subgraph selection. Our experiments revealed that our approach outperforms most baselines and performs consistently well across various LLMs and KGs on zero-shot knowledge-based causal discovery.

Our research has focused on pairwise causal relationship; in future work, we aim to explore complex scenarios involving multiple interconnected variables, i.e., *full* causal graph discovery. We also plan to co-optimize the integration of LLMs and KGs, while incorporating causality-related evaluation metrics to gain deeper insights into the underlying causal structures.

The following sections constitute the appendix.

A Dataset Details

The datasets are summarized in Table A1. Each instance in the dataset includes textual context where a variable pair co-occurs in a text, and is annotated by human experts to determine if there is a causal relation between the variable pair, as shown in the example.

Example: ***FGF6*** *contributes to the growth of* ***prostate cancer***
Ground truth: `(FGF6, causal, prostate cancer)`

dataset	domain/type	total instances
GENEC [50]	biomedical/gene-gene	789
ADE [16]	biomedical/drug-side effect	6,821
COMAGC [26]	biomedical/gene-disease	820
SEMEVAL [17]	open-domain causality	10,717

Table A1: Dataset sizes and types.

In the experiments, we filter the datasets (available in Github) to include only the instances that contain subgraph in the KGs. In addition, for the experiment in Table 3, we used SACHS [42] protein causality dataset, which contains 11 continuous variables.

B KG Querying & Subgraph Extraction

We accessed the **Hetionet** KG through its official public Neo4j API[3], using `neo4j.v1` Python library. Neo4j is a third-party graph database for querying and visualizing knowledge graphs. We sampled the subgraphs from KGs based on **hop distance**; we queried up to 4 hops for extracting the subgraphs from Hetionet, using the built-in functions from Neo4J Cypher `allShortestPaths`.

For **Wikidata**, we accessed the knowledge graph through its official public SPARQL endpoint[4] using the `SPARQLWrapper` Python library. We employed the official Wikidata API (e.g., `wbsearchentities` and `wbgetentities` functions) to extract Wikidata IDs for all variable pairs. To extract subgraphs from `Wikidata`, we used the **shortest-path** method by [10]. We queried Wikidata to retrieve the subgraphs using the `CLOCQ`[5] API, setting the maximum hops to 2.

Since we will further rank these subgraphs, we performed ***sparse retrieval* extraction** methods for each pair in the initial step before ranking, i.e., using **keyword-based/pattern-matching** querying. For instance, in Hetionet, to identify *drugs* that target *genes* related to specific *diseases*, we query the KG using pattern such as: `(Drug)-[:TARGETS]-> (Gene)-[:ASSOCIATED_WITH]->(Disease)`. For Wikidata, we performed random sparse retrieval in this initial step.

C Subgraph Relevancy Estimation : Details

We conducted **S**ubgraph **R**elevancy **E**stimation with `mistral-7b` from `huggingface` [61] models library. To calculate subgraph relevance, we derive the log probability of tokens using beam transition scores, which are based on the log probabilities of tokens conditioned on the log softmax of previously generated tokens within the same beam. Example of the prompt and the generated subgraph ranking dataset are provided in the following:

```
Given the following information, classify the relation
between the drug and side effect. If there is a cause-effect
relationship, state causal; otherwise, state non-causal.

[Pair]: dihydrotachysterol and hypercalcemia
[Textual context]: Severe hypercalcemia in a patient
treated for hypoparathyroidism with dihydrotachysterol.
[Relation Paths]: FGF6 - tendon - SDRDL - FGFR2 - prostate
cancer
[Relation]:
```

```
"qid": "1414", "e1": "carbamazepine", "e2": "systemic
lupus erythematosus", "groundtruth": "1", "metapaths":

["pathid": 1, "relscore": 1.41565, "probscore": -0.7063895,
"relevant": "1", "stops": "Carbamazepine - Conjunctivitis -
Dasatinib - Systemic lupus erythematosus rash", "reltypes":
"CAUSESCcSE - CAUSESCcSE - CAUSESCcSE", "nodelabels":
"Compound - SideEffect - Compound - SideEffect",

"pathid": 2, "relscore": -1.53165, "probscore": -0.6528874,
"relevant": "0", "stops": "Carbamazepine - Renal failure -
Dasatinib - Systemic lupus erythematosus rash", "reltypes":
"CAUSESCcSE - CAUSESCcSE - CAUSESCcSE", "nodelabels":
"Compound - SideEffect - Compound - SideEffect"]
```

D Subgraph Ranker : Details

D.1 Cross Encoder-Based Subgraph Ranker

We experiment with the following models for training the cross encoders-based subgraph rankers: (1) `roberta-base`, (2) `deberta-base`, (3) `biomed_roberta_base`. We use the models obtained from the `huggingface` [61] models library, and use their default hyperparamater settings to train the cross encoder. We use the implementation of the loss functions from `allRank` [36][6]. All experiments were implemented in Python with `Pytorch` and `Transfomer` library.

We provided up to 10 subgraphs per pair for subgraph ranker training, excluding pairs without any metapath-based subgraphs in our selected KGs. We concatenated the pair and its subgraph as the input to train the ranker, as shown in Eq. 5. We added the `CLS` to mark the beginning of the sequence and `SEP` to separate the variable pair and the subgraph. When the *node types* are not available, we used the *relation labels* as the *meta* information. The following shows the example of an input, where we included the *node/variable name* and *node types* as features:

```
CLS FGF6 prostate cancer SEP Gene FGF6 - anatomy tendon -
gene SDRDL - gene FGFR2 - disease prostate cancer
```

Since the performance of the cross-encoder with different encoder models did not vary significantly, we selected `roberta-base` to train the cross encoder-based subgraph rankers. We provide the evaluation results of the cross encoder-based subgraph rankers (with `roberta-base` as the encoder) in Table D2. Note that we used

[3] bolt://neo4j.het.io
[4] https://query.wikidata.org/sparql
[5] https://clocq.mpi-inf.mpg.de/documentation
[6] https://github.com/allegro/allRank/tree/master/allrank/models/losses

the ranking estimation of the subgraphs produced by **S**ubgraph **R**elevancy **E**stimation as ground truth to calculate the scores.

data	metric	rmse	ranknet	listnet
COMAGC	NDCG@5	56.60	57.12	55.38
	Recall@5	71.43	71.43	71.43
GENEC	NDCG@5	56.79	55.89	55.52
	Recall@5	71.77	71.77	71.77
ADE	NDCG@5	57.61	55.54	56.64
	Recall@5	72.18	72.18	72.18
SEMEVAL	NDCG@5	91.92	89.42	88.99
	Recall@5	100.00	100.00	100.00

Table D2: Subgraph Ranker experiment results.

D.2 XGBoost-Based Subgraph Ranker

We train our xgboost-based subgraph ranker as follows:

D.2.1. N-gram language model. This step trains an n-gram language model to generate embeddings as features for xgboost-based subgraph ranker training. During preprocessing, each variable pair in the dataset is concatenated with its corresponding subgraphs, combining all their connecting subgraphs into a single text sequence, as shown below.

```
carbamazepine - systemic lupus erythematosus: carbamazepine
- renal failure - dasatinib - systemic lupus erythematosus
```

All concatenated sequences are then combined into a single corpus of text, which is used to generate n-grams. Using these n-grams, a simple neural network-based language model is built and trained to predict the next token in a sequence, learning the context and relationships between tokens. We further extract the embeddings from the n-gram model as features to train an xgboost-based subgraph ranker. We used n=2-gram and an embedding dimension of 128 to train the language model throughout the experiments.

D.2.2. XGBoost-based subgraph ranker training. We implemented a gradient boosting-based subgraph ranker using the XGBoost Python package.[7] For each sample in the dataset, we concatenated the variable pair with its corresponding subgraphs and extracted their embeddings using the trained n-gram language model as features to train the ranker models. Table D3 summarizes the hyperparameter values for XGBoost-based subgraph ranker training.

hyperparameter	value
tree method	hist
lambdarank num pair per sample	8
lambdarank pair method	topk
objective	rank:ndcg

Table D3: XGBoost-based ranker hyperparameter values.

[7] https://github.com/dmlc/xgboost/

E Knowledge-Based Causal Discovery with Subgraphs as Prompt : Details

The previous module (**S**ubgraph **R**anker) produces ranked subgraphs; this step selects the top-k most *informative* subgraphs and incorporates them into the zero-shot prompt (details in §4.4). In our experiments, we found that including **up to one** (k=1) subgraph in the prompt yielded the best average results across all LLMs. We provide examples of the prompt in our Github.

F Baseline Implementation Details

F.1 Similarity-Based Subgraph Ranker

In the similarity-based ranker method, we obtained embeddings for each pair and subgraph using the Sentence Transformer [39] pretrained model all-MiniLM-L6-v2, and then calculated the cosine similarity scores between them. The subgraphs were subsequently ranked based on these scores.

F.2 GPT-Based Subgraph Ranker

RankGPT [48] proposes a *permutation generation* approach for passage re-ranking. This method involves passing a list of documents directly to an LLM and instructing it to rank them based on their relevance to a search query. We adapted RankGPT's original implementation[8] to rank subgraphs using a GPT model, providing up to 10 subgraphs per pair. We used the OpenAI API with the gpt-3.5-turbo-instruct engine for the GPT-based ranker. Generally, we assume only query access to the LLMs. Note that RankGPT also introduced a specialized passage ranker using permutation generation outputs; in our adaptation, we directly used the ranking output from permutation generation to rank the subgraphs. The following example shows the prompt for subgraph ranking:

```
role: system, content: You are an intelligent assistant
that rank paths based on relevancy to entity pair.
role: user, content: I will provide you with 3 paths, each
indicated by number identifier []. Rank the paths based on
their usefulness in inferring causal relationships between
the pair: (TRPV6, prostate cancer).
role: assistant, content: Okay, please provide the paths.
role: user, content: [1] TRPV6 - vagina - SQRDL -
Lenalidomide - Prostate cancer metastatic
role: user, content: [2] TRPV6 - TRAF1 - prostate cancer
role: user, content: [3] TRPV6 - seminal vesicle - prostate
cancer
role: user, content: pair: (TRPV6, prostate cancer). Rank
the 3 paths above based on their usefulness in inferring
causal relationships between entity pair...
```

G Computational Complexity and Hardware

Our method comprises 3 modules with the following complexities:

- **Module 1 (Subgraph Relevancy Estimation):** Complexity: $O(n \cdot p)$, where n = subgraphs, p = LLM parameters. Each subgraph requires a separate inference pass, 5–15s on 7–8B LLMs.

[8] https://github.com/sunnweiwei/RankGPT

- **Module 2 (`Subgraph Ranker`):** Subgraph ranking via (i) Transformer cross-encoder: $O(n \cdot m \cdot L^2)$, or (ii) XGBoost: $O(k \cdot d \cdot n \log(n))$, with m = layers, L = input length, k = trees, d = depth.
- **Module 3:** Zero-shot causal discovery. Complexity: $O(p)$. Each variable pair requires 5–15s on 7–8B LLMs.

Hardware specification:

- 2× Tesla P100-PCIE (16GB each), Driver: 440.33.01, CUDA: 10.2

References

[1] Ahmed Abdulaal, adamos hadjivasiliou, Nina Montana-Brown, Tiantian He, Ayodeji Ijishakin, Ivana Drobnjak, Daniel C. Castro, and Daniel C. Alexander. 2024. Causal Modelling Agents: Causal Graph Discovery through Synergising Metadata- and Data-driven Reasoning. In *ICLR 2024 poster*. International Conference on Learning Representations, Kigali, Rwanda, 123–130.

[2] Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering. In *Proceedings of MATCHING 2023*, Estevam Hruschka, Tom Mitchell, Sajjadur Rahman, Dunja Mladenić, and Marko Grobelnik (Eds.). Association for Computational Linguistics, Toronto, ON, Canada, 70–98. doi:10.18653/v1/2023.matching-1.7

[3] Jinheon Baek, Soyeong Jeong, Minki Kang, Jong Park, and Sung Hwang. 2023. Knowledge-Augmented Language Model Verification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 1720–1736. doi:10.18653/v1/2023.emnlp-main.107

[4] Taiyu Ban, Lyvzhou Chen, Xiangyu Wang, and Huanhuan Chen. 2023. From Query Tools to Causal Architects: Harnessing Large Language Models for Advanced Causal Discovery from Data. arXiv:2306.16902 [cs.AI] https://arxiv.org/abs/2306.16902

[5] Christopher J. C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview*. Technical Report MSR-TR-2010-82. Microsoft Research.

[6] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. *Learning to Rank: From Pairwise Approach to Listwise Approach*. Technical Report MSR-TR-2007-40. Microsoft Research. 9 pages.

[7] Heng Chang, Jiangnan Ye, Alejo Lopez-Avila, Jinhua Du, and Jia Li. 2024. Path-based Explanation for Knowledge Graph Completion. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*. ACM, 231–242. doi:10.1145/3637528.3671683

[8] Pulkit Chatwal, Amit Agarwal, and Ankush Mittal. 2025. Enhancing Causal Relationship Detection Using Prompt Engineering and Large Language Models. In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, Chung-Chi Chen, Antonio Moreno-Sandoval, Jimin Huang, Qianqian Xie, Sophia Ananiadou, and Hsin-Hsi Chen (Eds.). Association for Computational Linguistics, Abu Dhabi, UAE, 248–252. https://aclanthology.org/2025.finnlp-1.26/

[9] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 785–794. doi:10.1145/2939672.2939785

[10] Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. 2022. Beyond NED: Fast and Effective Search Space Reduction for Complex Question Answering over Knowledge Bases. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Virtual Event, AZ, USA) *(WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 172–180. doi:10.1145/3488560.3498488

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[12] Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. 2022. Causal Inference in Natural Language Processing: Estimation, Prediction, Interpretation and Beyond. *Transactions of the Association for Computational Linguistics* 10 (10 2022), 1138–1158. https://doi.org/10.1162/tacl_a_00511

[13] Jinglong Gao, Xiao Ding, Bing Qin, and Ting Liu. 2023. Is ChatGPT a Good Causal Reasoner? A Comprehensive Evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 11111–11126. doi:10.18653/v1/2023.findings-emnlp.743

[14] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink Training of BERT Rerankers in Multi-stage Retrieval Pipeline. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 – April 1, 2021, Proceedings, Part II*. Springer-Verlag, Berlin, Heidelberg, 280–286. doi:10.1007/978-3-030-72240-1_26

[15] Andrea Treviño Gavito, Diego Klabjan, and Jean Utke. 2023. Gradient-Boosted Based Structured and Unstructured Learning. arXiv:2302.14299 [cs.LG] https://arxiv.org/abs/2302.14299

[16] Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics* 45, 5 (2012), 885 – 892. doi:10.1016/j.jbi.2012.04.008 Text Mining and Natural Language Processing in Pharmacogenomics.

[17] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Uppsala, Sweden, 33–38. https://aclanthology.org/S10-1006

[18] Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio E Baranzini. 2017. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife* 6 (sep 2017), e26726. doi:10.7554/eLife.26726

[19] Ruixin Hong, Hongming Zhang, Hong Zhao, Dong Yu, and Changshui Zhang. 2023. Faithful Question Answering with Monte-Carlo Planning. arXiv:2305.02556 [cs.CL] https://arxiv.org/abs/2305.02556

[20] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[21] Xinke Jiang, Ruizhe Zhang, Yongxin Xu, Rihong Qiu, Yue Fang, Zhiyuan Wang, Jinyi Tang, Hongxin Ding, Xu Chu, Junfeng Zhao, and Yasha Wang. 2024. HyKGE: A Hypothesis Knowledge Graph Enhanced Framework for Accurate and Reliable Medical LLMs Responses. arXiv:2312.15883 https://arxiv.org/abs/2312.15883

[22] Thomas Jiralerspong, Xiaoyin Chen, Yash More, Vedant Shah, and Yoshua Bengio. 2024. Efficient Causal Graph Discovery Using Large Language Models. arXiv:2402.01207 [cs.LG] https://arxiv.org/abs/2402.01207

[23] Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. 2023. Knowledge Graph-Augmented Language Models for Knowledge-Grounded Dialogue Generation. arXiv:2305.18846 [cs.CL] https://arxiv.org/abs/2305.18846

[24] Vivek Khetan, Md Imbesat Rizvi, Jessica Huber, Paige Bartusiak, Bogdan Sacaleanu, and Andrew Fano. 2022. MIMICause: Representation and automatic extraction of causal relation types from clinical notes. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, Dublin, Ireland, 764–773. doi:10.18653/v1/2022.findings-acl.63

[25] Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal Reasoning and Large Language Models: Opening a New Frontier for Causality.

[26] Hee-Jin Lee, Sang-Hyung Shim, Mi-Ryoung Song, Hyunju Lee, and Jong C. Park. 2013. CoMAGC: a corpus with multi-faceted annotations of gene-cancer relations. *BMC Bioinformatics* 14, 1 (Nov 2013), 323. https://doi.org/10.1186/1471-2105-14-323

[27] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 793, 16 pages.

[28] Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. 2024. Common 7B Language Models Already Possess Strong Math Capabilities. arXiv:2403.04706 https://arxiv.org/abs/2403.04706

[29] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.* 3, 3 (mar 2009), 225–331. doi:10.1561/1500000016

[30] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In *International Conference on Learning Representations (ICLR)*.

[31] Meta. [n. d.]. Introducing Meta Llama 3: The most capable openly available LLM to date. https://ai.meta.com/blog/meta-llama-3/. [Accessed 29-07-2024].

[32] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *ArXiv* abs/1901.04085 (2019).

[33] Ayush Noori, Michelle M Li, Amelia L M Tan, and Marinka Zitnik. 2023. Metapaths: similarity search in heterogeneous knowledge graphs via meta-paths. *Bioinformatics* 39, 5 (05 2023), btad297. https://doi.org/10.1093/bioinformatics/btad297

[34] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Transactions on Knowledge and Data Engineering* 36, 7 (July 2024), 3580–3599. doi:10.1109/tkde.2024.3352100

[35] Fabio Petroni, Patrick S. H. Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How Context Affects

Language Models' Factual Predictions.. In *AKBC*. http://dblp.uni-trier.de/db/conf/akbc/akbc2020.html#PetroniLPRWM020

[36] Przemyslaw Pobrotyn, Tomasz Bartczak, Mikolaj Synowiec, Radoslaw Bialobrzeski, and Jaroslaw Bojar. 2020. Context-Aware Learning to Rank with Self-Attention. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGI ReCom* (China (Virtual Event)).

[37] Zhen Qin, Le Yan, Honglei Zhuang, Yi Tay, Rama Kumar Pasumarthi, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2021. Are Neural Rankers still Outperformed by Gradient Boosted Decision Trees?. In *International Conference on Learning Representations (ICLR)*.

[38] Priyanka Ranade and Anupam Joshi. 2024. FABULA: Intelligence Report Generation Using Retrieval-Augmented Narrative Construction. In *Proceedings of the 2023 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (Kusadasi, Turkiye) *(ASONAM '23)*. Association for Computing Machinery, New York, NY, USA, 603–610. doi:10.1145/3625007.3627505

[39] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084

[40] Zhong-Hao Ren, Zhu-Hong You, Quan Zou, Chang-Qing Yu, Yan-Fang Ma, Yong-Jian Guan, Hai-Ru You, Xin-Fei Wang, and Jie Pan. 2023. DeepMPF: deep learning framework for predicting drug–target interactions based on multi-modal representation with meta-path semantic analysis. *Journal of Translational Medicine* 21, 1 (25 Jan 2023), 48. https://doi.org/10.1186/s12967-023-03876-3

[41] Laria Reynolds and Kyle McDonell. 2021. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI EA '21)*. Association for Computing Machinery, New York, NY, USA, Article 314, 7 pages. doi:10.1145/3411763.3451760

[42] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. 2005. Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data. *Science* 308, 5721 (2005), 523–529. doi:10.1126/science.1105809 arXiv:https://www.science.org/doi/pdf/10.1126/science.1105809

[43] Weizhou Shen, Yingqi Gao, Canbin Huang, Fanqi Wan, Xiaojun Quan, and Wei Bi. 2023. Retrieval-Generation Alignment for End-to-End Task-Oriented Dialogue System. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 8261–8275. doi:10.18653/v1/2023.emnlp-main.514

[44] Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvärinen, Yoshinobu Kawahara, Takashi Washio, Patrik O. Hoyer, and Kenneth Bollen. 2011. DirectLiNGAM: A Direct Method for Learning a Linear Non-Gaussian Structural Equation Model. *J. Mach. Learn. Res.* 12, null (July 2011), 1225–1248.

[45] Peter Spirtes, Clark Glymour, and Scheines R. 2000. Constructing bayesian networks models of gene expression networks from microarray data. In *Proceedings of the Atlantic Symposium on Computational Biology* (North Carolina).

[46] P. Spirtes, C. Glymour, and R. Scheines. 2001. *Causation, Prediction, and Search* (2nd ed.). MIT press.

[47] Peter Spirtes, Clark Glymour, and Richard Scheines. 2001. *Causation, Prediction, and Search*. The MIT Press. doi:10.7551/mitpress/1754.001.0001

[48] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 14918–14937. doi:10.18653/v1/2023.emnlp-main.923

[49] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. PathSim: meta path-based top-K similarity search in heterogeneous information networks. *Proc. VLDB Endow.* 4, 11 (aug 2011), 992–1003. doi:10.14778/3402707.3402736

[50] Yuni Susanti and Michael Färber. 2024. Knowledge Graph Structure as Prompt: Improving Small Language Models Capabilities for Knowledge-based Causal Discovery. In *Proceedings of the ISWC'24* (Baltimore, USA).

[51] Yuni Susanti and Kanji Uchino. 2024. Causal-Evidence Graph for Causal Relation Classification. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing* (Avila, Spain) *(SAC '24)*. Association for Computing Machinery, New York, NY, USA, 714–722. doi:10.1145/3605098.3635894

[52] Masayuki Takayama, Tadahisa Okuda, Thong Pham, Tatsuyoshi Ikenoue, Shingo Fukuma, Shohei Shimizu, and Akiyoshi Sannai. 2024. Integrating Large Language Models in Causal Discovery: A Statistical Causal Approach. arXiv:2402.01454 [cs.LG] https://arxiv.org/abs/2402.01454

[53] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open Models Based on Gemini Research and Technology. arXiv:2403.08295 [cs.CL] https://arxiv.org/abs/2403.08295

[54] Hiren Thakkar and A Manimaran. 2023. Comprehensive Examination of Instruction-Based Language Models: A Comparative Analysis of Mistral-7B and Llama-2-7B. In *2023 International Conference on Emerging Research in Computational Science (ICERCS)*. 1–6. doi:10.1109/ICERCS57948.2023.10434081

[55] Ruibo Tu, Chao Ma, and Cheng Zhang. 2023. Causal-Discovery Performance of ChatGPT in the context of Neuropathic Pain Diagnosis. arXiv:2301.13819 [cs.CL]

[56] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (sep 2014), 78–85. https://doi.org/10.1145/2629489

[57] Hong Wang, Xiaoqi Wang, Wenjuan Liu, Xiaolan Xie, and Shaoliang Peng. 2022. deepDGA: Biomedical Heterogeneous Network-based Deep Learning Framework for Disease-Gene Association Predictions. In *IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2022, Las Vegas, NV, USA, December 6-8, 2022*, Donald A. Adjeroh, Qi Long, Xinghua Mindy Shi, Fei Guo, Xiaohua Hu, Srinivas Aluru, Giri Narasimhan, Jianxin Wang, Mingon Kang, Ananda Mondal, and Jin Liu (Eds.). IEEE, 601–606. https://doi.org/10.1109/BIBM55620.2022.9995651

[58] Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. 2023. Knowledge-Driven CoT: Exploring Faithful Reasoning in LLMs for Knowledge-intensive Question Answering. arXiv:2308.13259 [cs.CL] https://arxiv.org/abs/2308.13259

[59] Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. 2023. KnowledGPT: Enhancing Large Language Models with Retrieval and Storage Access on Knowledge Bases. arXiv:2308.11761 [cs.CL] https://arxiv.org/abs/2308.11761

[60] Moritz Willig, Matej Zečević, Devendra Singh Dhami, and Kristian Kersting. 2022. Can Foundation Models Talk Causality? arXiv:2206.10591 [cs.AI]

[61] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Qun Liu and David Schlangen (Eds.). Association for Computational Linguistics, Online, 38–45. https://aclanthology.org/2020.emnlp-demos.6

[62] Wenjie Yao, Weizhong Zhao, Xingpeng Jiang, Xianjun Shen, and Tingting He. 2022. MPGNN-DSA: A Meta-path-based Graph Neural Network for drug-side effect association prediction. In *IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2022, Las Vegas, NV, USA, December 6-8, 2022*, Donald A. Adjeroh, Qi Long, Xinghua Mindy Shi, Fei Guo, Xiaohua Hu, Srinivas Aluru, Giri Narasimhan, Jianxin Wang, Mingon Kang, Ananda Mondal, and Jin Liu (Eds.). IEEE, 627–632. https://doi.org/10.1109/BIBM55620.2022.9995486

[63] Changhe Yuan and Brandon Malone. 2013. Learning optimal bayesian networks: a shortest path perspective. *J. Artif. Int. Res.* 48, 1 (Oct. 2013), 23–65.

[64] Cheng Zhang, Stefan Bauer, Paul Bennett, Jiangfeng Gao, Wenbo Gong, Agrin Hilmkil, Joel Jennings, Chao Ma, Tom Minka, Nick Pawlowski, and James Vaughan. 2023. Understanding Causality with Large Language Models: Feasibility and Opportunities. arXiv:2304.05524 [cs.LG]

[65] Shichang Zhang, Jiani Zhang, Xiang Song, Soji Adeshina, Da Zheng, Christos Faloutsos, and Yizhou Sun. 2023. PaGE-Link: Path-based Graph Neural Network Explanation for Heterogeneous Link Prediction. arXiv:2302.12465 [cs.LG] https://arxiv.org/abs/2302.12465