Deep Cognitive Reasoning Network for Multi-hop Question Answering over Knowledge Graphs

Jianyu Cai **Zhanqiu Zhang** **Feng Wu** **Jie Wang***
University of Science and Technology of China
Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{jycai,zqzhang}@mail.ustc.edu.cn
{fengwu, jiewangx}@ustc.edu.cn

Abstract

Knowledge Graphs (KGs) provide human knowledge with nodes and edges being entities and relations among them, respectively. Multi-hop question answering over KGs—which aims to find answer entities of given questions through reasoning paths in KGs—has attracted great attention from both academia and industry recently. However, this task remains challenging, as it requires to accurately identify answers in a large candidate entity set, of which the size grows exponentially with the number of reasoning hops. To tackle this problem, we propose a novel **D**eep **C**ognitive **R**easoning **N**etwork (DCRN), which is inspired by the *dual process theory* in cognitive science. Specifically, DCRN consists of two phases—the *unconscious* phase and the *conscious* phase. The unconscious phase first retrieves informative evidence from candidate entities by leveraging their semantic information. Then, the conscious phase accurately identifies answers by performing sequential reasoning according to the graph structure on the retrieved evidence. Experiments demonstrate that DCRN significantly outperforms state-of-the-art methods on benchmark datasets.

1 Introduction

Knowledge Graphs (KGs) store structured human knowledge, in which nodes represent entities and edges represent relations between pairs of entities. Multi-hop Question Answering over KGs (KGQA) aims to find answer entities by reasoning over paths in KGs. We illustrate this task with an example in Figure 1. Recently, multi-hop question answering over KGs has attracted great attention from both academia and industry (Li et al., 2017; Fu et al., 2020; Saxena et al., 2020). However, this task remains challenging, because the number of candidate entities grows exponentially with the number of reasoning hops (Sun et al., 2018, 2019a), making it difficult to accurately identify answers.

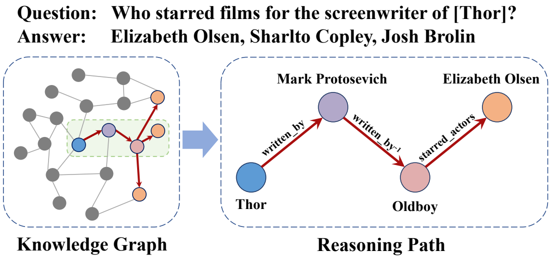
Question: Who starred films for the screenwriter of [Thor]?
Answer: Elizabeth Olsen, Sharlto Copley, Josh Brolin
Mark Protosevich
Elizabeth Olsen
written_by
written_by⁻¹
starred_actors
Thor
Oldboy
Knowledge Graph
Reasoning Path

Figure 1: Illustration of multi-hop question answering over KGs. Given a natural language question, we start from the topic entity in it, and reason along paths in KGs to find answers.

Previous works mitigate this problem by reducing the size of the candidate entity sets, but they often sacrifice the recall of answers. These methods including GRAFT-Net (Sun et al., 2018) and PullNet (Sun et al., 2019a) first extract question-specific subgraphs, and then perform multi-hop reasoning on the extracted subgraph via Graph Neural Networks (GNNs) to find answers. However, these approaches often sacrifice the recall of answers in exchange for small candidate entity sets. That is, the extracted subgraph may contain no answer at all. This trade-off between the recall of answer entities and the size of candidate entity sets limits their practical usage. Therefore, it is still desirable to find an approach that is capable of accurately identifying answers without sacrificing their recalls.

To tackle this problem, we take inspiration from the *dual process theory* (Evans, 1984, 2003, 2008) in cognitive science and propose a novel **D**eep **C**ognitive **R**easoning **N**etwork (DCRN). In cognitive science, researchers found that humans can reason over a large-capacity memory to find answers (Wang et al., 2003). Specifically, the *dual*

[*] Corresponding author.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 219–229
August 1–6, 2021. ©2021 Association for Computational Linguistics

process theory (Evans, 1984, 2003, 2008) suggests that humans accomplish cognitive tasks by first exploiting *fast intuition* to retrieve task-relevant evidence via an *unconscious process*, and then performing *sequential reasoning* based on the aforementioned evidence to derive answers via a *conscious process*. Similarly, the proposed DCRN consists of two phases. The first one is the unconscious phase, which can retrieve informative evidence by *softly* selecting candidate entities that are most likely to be correct answers. The second one is the conscious phase, which can accurately identify answers by performing sequential reasoning with Bayesian networks based the retrieved evidence from the first phase. Experiments demonstrate that DCRN significantly outperforms state-of-the-art methods on benchmark datasets.

2 Preliminaries

In this section, we first review the background of this paper and then introduce the notations used throughout this paper.

2.1 Background

In this part, we review the background of knowledge graph and milti-hop KGQA.

Knowledge Graph Given a set of entities $\mathcal{E}$, a set of relations $\mathcal{R}$, and a set of triplets $\mathcal{T} = \{(e_i, r_j, e_k)\} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, we define a knowledge graph $\mathcal{G}$ by $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$.

Multi-hop KGQA Given a knowledge graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$ and a natural language question q with its topic entity $e_{topic} \in \mathcal{E}$, the task of KGQA is to predict the answer e^* to question q by

$$e^* = \arg\max_{e_i \in \mathcal{E}} f(e_i),$$

where $f(e_i)$ is the score function that measures the plausibility of e_i being the correct answer. In multi-hop KGQA, the answers are not guaranteed to be direct neighbours of the topic entity in the given question. Therefore, it often requires multi-hop reasoning over KGs to find answers.

Bayesian Network A Bayesian network is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG). In a Bayesian network, the nodes represents random variables and the directed edges represent the conditional dependencies between random variables.

2.2 Notations

In this paper, we use lower-case letters e and r to represent an entity and a relation, respectively. The corresponding boldface letters **e** and **r** denotes the embeddings of e and r, respectively.

3 Related Work

In this section, we review related work for multi-hop KGQA and knowledge graph embeddings.

3.1 Multi-hop KBQA

Recent work in multi-hop KBQA can be divided into two categories: semantic parsing methods and information retrieval methods. Semantic parsing methods first parse the given question into an executable query, and then execute the query to locate answers. Information retrieval methods embeds questions and the knowledge graph into low-dimensional spaces, and then find answers based on question-answer semantic similarity. Our proposed DCRN belongs to information retrieval methods.

Key-Value Memory Network (KV-Mem) KV-Mem (Miller et al., 2016) is a variant of Memory Network (Weston et al., 2015), which performs reasoning based on a memory component, i.e., an array storing triplets in KGs. KV-Mem iteratively reads from the memory to update the question embedding, which is used to match correct answers.

Variational Reasoning Network (VRN) VRN (Zhang et al., 2018) proposes a variational framework for multi-hop KGQA. To identify answers, it computes the compatibility scores between the question type and the reasoning graph of each candidate. However, its performance is limited on the question that requires long reasoning paths to answer, due to the exponentially grown candidates.

GRAFT-Net GRAFT-Net (Sun et al., 2018) first extracts question-specific subgraph based on Personalized Page Rank (PPR), and then encode the subgraph with Graph Neural Networks (GNN) to identify answers. However, as described in Sun et al. (2019a), the extracted subgraphs are often too large and have a low recall for answer entities.

PullNet PullNet (Sun et al., 2019a) mitigates the problem of GraftNet with a trainable subgraph expansion strategy. It constructs question-specific subgraph starting from the list of entities mentioned in the question, and then iteratively "pulls" the relevant entities to expand the subgraph. However, it inevitably sacrifices the recall of answer entities

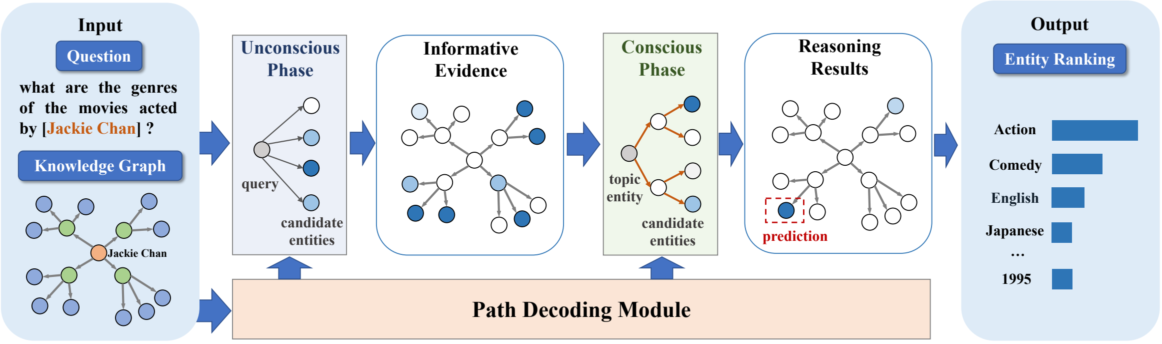

Figure 2: Overview of the proposed Deep Cognitive Reasoning Network (DCRN). DCRN consists of a Path Decoding Module and two phases—the unconscious phase and the conscious phase. Deeper color in coarse/fine-grained results denotes higher prediction score as the correct answer.

in exchange for small candidate entity sets, which limits its performance in practical usage.

EmbedKGQA EmbedKGQA (Saxena et al., 2020) models multi-hop KBQA as a link prediction task. It first embeds the given question into a latent relation embedding, and then exploits knowledge graph embedding techniques to identify answers.

3.2 Knowledge Graph Embedding in KGQA

Knowledge Graph Embedding (KGE) methods (Hitchcock, 1927; Trouillon et al., 2016; Sun et al., 2019b; Zhang et al., 2020a,b) aim to map entities and relations within KGs into distributed representations (vectors, matrices, etc.). These embeddings are often trained by the link prediction task, where the model is required to predict the missing head or tail entity of a triplet.

EmbedKGQA (Saxena et al., 2020) use ComplEx (Trouillon et al., 2016) to train knowledge graph embeddings, which represents entity and relation embeddings as vectors in complex spaces. For fair comparison with previous work including GRAFT-Net (Sun et al., 2018) and PullNet (Sun et al., 2019a), we use Canonical Polyadic (CP) decomposition (Hitchcock, 1927) to train knowledge graph embeddings in our proposed DCRN, which represents entity and relation embeddings as vectors in real spaces.

3.3 The Dual Process Theory

The *dual process theory* (Evans, 1984, 2003, 2008) is originally proposed in cognitive science. Inspired by this theory, researchers propose to mimic human cognition in various cognitive tasks. For example, Du et al. (2019) applies the theory to one-shot KG reasoning, and Ding et al. (2019) proposes a cognitive framework for multi-hop reasoning over documents. Different from these work, we focus on the task of multi-hop question answering over knowledge graphs.

4 Method

In this section, we introduce our proposed Deep Cognitive Reasoning Network (DCRN) for multi-hop KGQA. In section 4.1, we introduce the motivation and the overall architecture of DCRN. In Section 4.2, 4.3, and 4.4, we introduce the components of the proposed DCRN.

4.1 Motivation

For multi-hop questions, it is challenging to accurately identify answers from a large candidate set, of which the size grows exponentially with the number of reasoning steps. Existing approaches (Sun et al., 2018, 2019a) aim to reduce the size of candidate entity set by extracting question-specific subgraphs. However, these approaches often sacrifice the recall of answers in exchange for small candidate sets, which limits their performances in practical usage.

We take inspiration from the *dual process theory* (Evans, 1984, 2003, 2008) in cognitive science. Specifically, the theory suggests that humans accomplish cognitive tasks by first exploiting *fast intuition* to retrieve task-relevant evidence via an *unconscious process* (System 1), and then performing *sequential reasoning* based on the aforementioned evidence to derive answers via a *conscious process* (System 2).

Inspired by the *dual process theory* in cognitive science, we propose Deep Cognition Reasoning Network (DCRN) for multi-hop KGQA. The proposed DCRN consists of two phases. The first one is the *unconscious phase*, which can retrieve informative evidence from candidate entities by leveraging their semantic information. The second one is the *conscious phase*, which can accurately identify answers by performing sequential reasoning according to the graph structure on the retrieved evidence from the first phase.

The overall architecture of DCRN is shown in Figure 2. In DCRN, the basic module is the Path Decoding Module, based on which are the two phases—unconscious phase and conscious phase.

4.2 Path Decoding Module

, The Path Decoding Module is the basic component of DCRN. As multi-hop KGQA requires multi-hop reasoning to arrive at answer entities, we decode the reasoning path information from the question in this module.

Specifically, we adopt an RNN-based encoder-decoder structure, which first encodes the question into a hidden representation, and then decodes this representation to obtain the reasoning path information, i.e., the scores of each relation at each reasoning step. These scores will be used in the unconscious and conscious phase.

First, we encode the given question q with an RNN to obtain its latent representation $\mathbf{q} \in \mathbb{R}^d$.

$$\mathbf{q} = \text{RNN-Encoder}(q).$$

Then, we decode this representation $\mathbf{q}$ to obtain reasoning path information. We illustrate this process in Figure 3. At each step of decoding, the decoder predicts the scores of each relation. The predictions at step t is the input of the decoder at step $t + 1$.

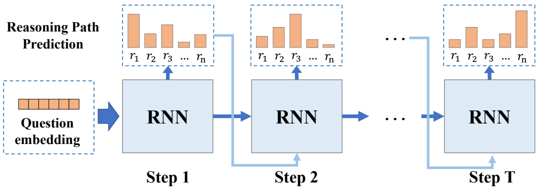
Reasoning Path Prediction
r_1 r_2 r_3 ... r_n
r_1 r_2 r_3 ... r_n
...
r_1 r_2 r_3 ... r_n
Question embedding
RNN
RNN
...
RNN
Step 1
Step 2
Step T

Figure 3: Illustration of the Path Decoding Module.

At step t, given the hidden state $\mathbf{h}^{(t-1)}$ of the previous iteration and the input $\mathbf{i}^{(t)}$, the RNN decoder outputs the updated hidden state $\mathbf{h}^{(t)}$ by

$$\mathbf{h}^{(t)} = \text{RNN-Decoder}(\mathbf{h}^{(t-1)}, \mathbf{i}^{(t)}),$$

where the initial hidden state $\mathbf{h}^{(0)}$ is initialized as the question embedding $\mathbf{q}$, and the initial input $i^{(0)}$ is a zero vector. Then, we compute the output of step t by

$$\mathbf{o}^{(t)} = \sum_i \alpha_i^{(t)} \mathbf{r}_i,$$

where the weights $\alpha_i^{(t)}$ is computed as

$$\alpha_i^{(t)} = \frac{\exp\left(f_{rel}^{(t)}(r_i)\right)}{\sum_j \exp\left(f_{rel}^{(t)}(r_j)\right)}.$$

Note that $f_{rel}^{(t)}(r_i)$ denotes the scores of each relation at step t, which is computed by

$$f_{rel}^{(t)}(r_i) = \mathbf{h}^{(t)} \mathbf{r}_i^\top.$$

Then, the output of step t will be the input of step $t + 1$. That is, $\mathbf{i}^{(t+1)} = \mathbf{o}^{(t)}$.

4.3 The Unconscious Phase

The unconscious phase corresponds to the *unconscious process* (System 1) in the *dual process theory* from cognitive science. In this phase, we retrieve informative evidence from candidate entities by leveraging their semantic information.

The *evidence* refers to sketched results that predicts which candidates are most likely to be correct answers. We expect the retrieved evidence to effectively filter out those candidate entities that are irrelevant to the given question.

To achieve this, we perform semantic matching between the given question and each candidate entity. The semantic matching scores $f_s(e)$ of candidate entity e is

$$f_s(e) = \overline{\mathbf{q}} \mathbf{e}^\top,$$

where $\overline{\mathbf{q}} \in \mathbb{R}^{1 \times d}$ is the query embedding obtained based on the given question, and $\mathbf{e} \in \mathbb{R}^{1 \times d}$ is the embedding of entity e.

In our model, the entity embedding $\mathbf{e} \in \mathbb{R}^d$ is pretrained by the CP (Hitchcock, 1927) model. Therefore, the key of the unconscious phase is to design informative query representation $\overline{\mathbf{q}} \in \mathbb{R}^d$.

To design informative query representations, we take inspiration from PTransE (Lin et al., 2015), which extends knowledge graph embedding to relation paths. In PTransE, if a relation path $e_1 \xrightarrow{r_1} e_2 \xrightarrow{r_2} \ldots \xrightarrow{r_{n-1}} e_n$ holds, then PTransE optimize the following objective:

$$\mathbf{e}_1 \circ \mathbf{r}_1 \circ \ldots \circ \mathbf{r}_{n-1} = \mathbf{e}_n,$$

where $\circ$ is an composition operation, and it can be addition, element-wise multiplication, RNN, etc. This objective can be considered as the semantic matching between the query $\mathbf{e}_1 \circ \mathbf{r}_1 \circ ... \circ \mathbf{r}_{n-1}$ and its target $\mathbf{e}_n$.

Similarly, we encode the query representation $\overline{\mathbf{q}}$ as follows. First, the start entity of the reasoning path is the topic e_{topic} in the given question. Second, recall that we decode reasoning path information in the Path Decoding Module, in which the output at step t (i.e., $\mathbf{o}^{(t)}$) represents the weighted sum of relation embeddings. Therefore, we represent the query embedding as

$$\overline{\mathbf{q}} = \mathbf{e}_{topic} \circ \mathbf{o}^{(1)} \circ ... \circ \mathbf{o}^{(T)},$$

where $\circ$ denotes element-wise multiplication in this formula, and T denotes the number of steps in the Path Decoding Module (i.e., the number of reasoning steps).

4.4 The Conscious Phase

The conscious phase corresponds to the *conscious process* (System 2) in the *dual process theory* from cognitive science. In this phase, we accurately identify answers by performing sequential reasoning according to the graph structure on the retrieved evidence from the unconscious phase.

To model the sequential reasoning, we take inspiration from the *consciousness prior* (Bengio, 2017). It suggests that the conscious process only refers to a few variables at a time, which can be modeled as *factor graphs*, a form of knowledge representation which is factored into pieces involving a few variables at a time.

In this work, we perform sequential reasoning based on Bayesian networks, which can be seen as a type of factor graphs. First, we build question-specific Bayesian networks from the given KG, in which we view the predictions of entities as random variables and relations as the relational dependencies between them. Second, we perform marginal inference on the Bayesian networks to predict the probability of each candidate entity as a correct answer.

4.4.1 Building Bayesian Networks

We build question-specific Bayesian networks from the given KG with the following two steps. First, we perform graph pruning on the KG to obtain a directed acyclic graph (DAG). Second, we transform the DAG into a Bayesian network.

Given a knowledge graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$ and a question q with a topic entity $e_{topic} \in \mathcal{E}$, we prune G to obtain a directed acyclic graph (DAG) by applying the breadth-first search (BFS) algorithm starting from e_{topic}. Specifically, we only keep the visited edges during searching, and remove the unvisited edges. We illustrate this process in Figure 4, in which we perform two-step BFS starting from the topic entity, and prunes the unvisited edges. Note that we add inverse relations r^{-1} for each relation r in KGs following previous work (Sun et al., 2018, 2019a). That is, if (e_i, r_j, e_k) is a valid triplet, then (e_k, r_j^{-1}, e_i) is also valid.

The reasons to perform the graph pruning is two-fold. First, the number of potential reasoning paths from e_{topic} to an arbitrary candidate entity e in the KG can be extremely large. Therefore, we apply graph pruning to reduce the search space, and only keep the shortest paths. Second, Bayesian Network is required to be a directed acyclic graph (DAG). Therefore, the graph pruning procedure only removes redundant edges, and the answer entities are guaranteed to be within the candidate set.

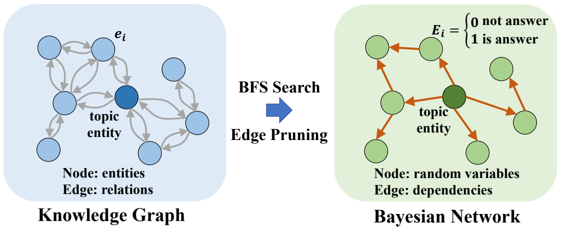

Figure 4: Illustration of question-specific Bayesian networks built from KGs. In KGs, we add an inverse relation r^{-1} for each relation r. That is, if a triplet (e_i, r, e_j) exists, then (e_j, r^{-1}, e_i) also exists.

We use $\hat{\mathcal{G}}(e_{topic})$ to denote the pruned graph given the topic entity e_{topic}. Then, we have the following proposition.

Proposition 1. *The pruned graph $\hat{\mathcal{G}}(e_{topic})$ is a directed acyclic graph (DAG).*

For detailed proof, please refer to the appendix. According to the properties of BFS, if $\mathcal{G}$ is connected, then there exists a path in the pruned graph $\hat{\mathcal{G}}(e_{topic})$ that starts from e_{topic} and ends with e. Furthermore, this path must be the shortest one among all paths in $\mathcal{G}$ that connects e_{topic} and e.

We then introduce how this DAG corresponds $\hat{\mathcal{G}}(e_{topic})$ to a Bayesian network. The transformed Bayesian Network $\mathcal{B}(_{topic})$ shares the same graph structure with $\hat{\mathcal{G}}(_{topic})$, but the definitions on nodes and edges are different. In Table 1, we illustrate the

Table 1: The relationship between the DAG and the corresponding Bayesian Network.

	DAG $\hat{\mathcal{G}}(e_{topic})$	Bayesian Network $\mathcal{B}(e_{topic})$
Nodes	entity e_i	random variable $X_{e_i} = \{0, 1\}$
Edges	relation r_j between e_i and e_k	conditional dependencies between X_{e_i} and X_{e_k}

relationship between the DAG and the corresponding Bayesian network.

Each entity e in $\hat{\mathcal{G}}(e_{topic})$ corresponds to a random variable $X_e = \{0, 1\}$ in $\mathcal{B}(e_{topic})$, where $X_e = \{0, 1\}$ represents the prediction of a candidate entity e. Given a question q, $X_e = 0$ denotes that e is an incorrect answer and $X_e = 1$ denotes that e is a correct answer. In $\hat{\mathcal{G}}(_{topic})$, each relation r connecting entity e_i and e_j corresponds to an directed edge connecting X_{e_i} and X_{e_j} in $\mathcal{B}(_{topic})$, which denotes the dependencies between them.

4.4.2 Bayesian Reasoning

Based on the Bayesian network $\mathcal{B}(_{topic})$, we can make marginal inferences to predict whether an entity e is a correct answer, which can be represented in a probabilistic way:

$$\mathbb{P}(X_e = 1 | \mathcal{G}, q, e_{topic}),$$

where $\mathcal{G}$ is the KG, q is the given question, and e_{topic} is the topic entity. To calculate this marginal probability, we have the following proposition.

Proposition 2. *The marginal probability $\mathbb{P}(X_e = 1 | \mathcal{G}, q, e_{topic})$ that predicts a candidate entity e can be calculated via variable elimination:*

$$\begin{aligned}&\mathbb{P}(X_e = 1 | \mathcal{G}, q, e_{topic})\\ &= \overline{\mathbb{P}}(X_e = 1) \prod_{e' \in pa(e)} \mathbb{P}(X_{e'} = 0 | \mathcal{G}, q, e_{topic}),\end{aligned}$$

where $pa(e)$ denotes the set of parent nodes of e, i.e., the nodes that have edges directing at e. The first component $\overline{\mathbb{P}}(X_e = 1)$ is the abbreviation of

$$\mathbb{P}(X_e = 1 | \mathcal{G}, q, e_{topic}, X_{pa(e)} = 0).$$

For detailed proofs, please refer to the appendix. The marginal probability $\mathbb{P}(X_e = 1 | \mathcal{G}, q, e_{topic})$ is the product of two components. The first component denotes the probability that entity e is an answer given that all e's parent entities $\text{pa}(e)$ are incorrect. The second component $\prod_{e' \in \text{pa}(X_e)} \mathbb{P}(X_{e'} = 0 | \mathcal{G}, q, e_{topic})$ denotes the product of the predictions of X_e's parent nodes. Note that we assume $\mathbb{P}(X_{e'} = 0 | \mathcal{G}, q, e_{topic}) = 1$ when computing $\mathbb{P}(X_e = 1 | \mathcal{G}, q, e_{topic}$ in our implementation for convenience of computation.

We model the first component as follows.

$$\begin{aligned}&\mathbb{P}(X_e = 1 | \mathcal{G}, q, e_{topic}, X_{\text{pa}(e)} = 0)\\ &= \text{sigmoid}(g(f_s(e), f_b(e))),\end{aligned}$$

where $f_s(e)$ is the evidence provided by the unconscious phase, $f_b(e)$ represents the score computed in the Bayesian network. $g(\cdot, \cdot)$ is a function for combining the two scores, and we choose $g(x, y) = x + y$ in this work. The score $f_b(e)$ is defined as follows:

$$f_b(e) = \sum_{\substack{e' \in \text{pa}(e),\\ (e', r, e) \in \mathcal{T}}} \alpha_r^{(t)} f_b(e') + f_{rel}^{(t)}(r),$$

where $f_{rel}^{(t)}(r)$ represents the prediction score of relation r at reasoning step t in the Path Decoding Module, $\alpha_r^{(t)}$ is defined in section **??**. Note that t also denotes the topological distance between e_{topic} and e, i.e., the required reasoning steps from e_{topic} to e. We initialize the score for the topic entity to zero. That is, $f_b(e_{topic}) = 0$.

The conscious phase is different from previous multi-hop reasoning approaches (Zhang et al., 2018; Sun et al., 2018, 2019a) in the following two aspects. First, we model the reasoning process in a probabilistic perspective with Bayesian networks, while previous works often apply GNN for reasoning. Second, the conscious phase propagates scalar scores along the paths for multi-hop reasoning, while previous works often propagates embeddings with GNNs.

4.5 Loss Function

We use the Binary Cross Entropy Loss for training. Specifically, given a question q, the loss $\mathcal{L}$ is computed as

$$\mathcal{L} = \frac{1}{|\mathcal{E}|}(\sum_{e \in \mathcal{A}} \log p(e) + \sum_{e' \in \mathcal{E}/\mathcal{A}} \log(1 - p(e'))),$$

where $\mathcal{E}$ is the set of entities, $\mathcal{A}$ is the set of correct answers, and $p(e) = \mathbb{P}(X_e = 1 | \mathcal{G}, q, e_{topic})$.

5 Experiments

This section is organized as follows. In Section 5.1, we introduce experimental settings in detail. In Section 5.2, we show the effectiveness of our model on benchmark datasets. In Section 5.3, we conduct ablation studies and analysis.

5.1 Experimental Settings

In this part, we introduce the benchmark datasets and the protocols for training and evaluation.

5.1.1 Datasets

We conduct experiments on two public datasets—WebQuestionSP (Yih et al., 2015) and MetaQA (Zhang et al., 2018), which have been divided into training, validation, and testing set by previous works. The statistics are shown in Table 2.

Table 2: The statistics of benchmark datasets. The second to fourth columns show the number of entities, relations, and triplets, respectively. The fifth to seventh columns show the size of training, validation, and testing dataset, respectively.

Dataset	Entity	Relation	Triplet	Train	Valid	Test
WebQuestionSP	601,445	567	1,261,849	2,848	250	1,639
MetaQA 1-hop	43,233	9	134,741	96,106	9,992	9,947
MetaQA 2-hop	43,233	9	134,741	118,980	14,872	14,872
MetaQA 3-hop	43,233	9	134,741	114,196	14,274	14,274

WebQuestionSP WebQuestionSP is a small dataset containing 4,737 questions. Those questions are 1-hop or 2-hop questions that can be answered with the Freebase (Bollacker et al., 2008) knowledge graph. Note that WebQuestionSP mainly consists of 1-hop questions, and only 0.5% of the questions are 2-hop.

MetaQA MetaQA is a large dataset containing over 400k questions in the movie domain. It is split into 1-hop, 2-hop, and 3-hop questions. Following previous work (Sun et al., 2018, 2019a; Saxena et al., 2020), we use the “vanilla” version of the dataset. On MetaQA, we evaluate our model under two settings: “full” setting and “half” setting. In the “full” setting, we use the vanilla knowledge graph for training. In the “half” setting, we follow previous work (Saxena et al., 2020) to randomly drop 50% of triplets in the knowledge graph.

5.1.2 Evaluation Protocols

Following previous work (Sun et al., 2018, 2019a), we use Hits at N (H@N) to evaluate model performance. For each given question, we rank the candidates in descending order according to their scores, and compute the percentage of correct answers that ranks at top N.

5.1.3 Training Protocols

We choose Adam (Kingma and Ba, 2015) as the optimizer, and use grid search to find the best hyperparameters based on the model performance on the validation datasets. For the details of hyperparameter selection, please refer to the appendix.

Following previous work (Sun et al., 2018, 2019a), we use GloVe (Pennington et al., 2014) as word embeddings, and use bidirectional LSTM as the encoder. We also use CP (Hitchcock, 1927) to train entity and relation embeddings.

5.1.4 Candidate Set Generation Protocol

In the ”full” setting, the candidate set of a n-hop question q consists of all entities that are within the n-hop of the topic entity of q. In the ’half’ setting, the candidate set of any question consists of all entities in the KG. Therefore, the answers are guaranteed to be included in the candidate sets, and the recall of answers is 1.0.

5.1.5 Hyperparameters

We use grid search to find the best hyperparameters. Specifically, we search the learning rate in $\{0.1, 0.01, 0.001\}$, and dropout rate in $\{0.1, 0.2, 0.3\}$. The optimal configurations of DCRN is that learning rate $= 0.01$ and dropout rate $= 0.2$. For fair comparison with previous work (Sun et al., 2018, 2019a; Saxena et al., 2020), we set the embedding size to 300. When training the knowledge graph embeddings with CP (Hitchcock, 1927), we search the learning rate in $\{0.1, 0.01, 0.001\}$, and the optimal configuration is learning rate $= 0.1$. We choose the values hyperparameter T as follows. On WebQuestionSP, we set $T = 2$. On MetaQA with ”full” setting, we set $T = t$ for t-hop questions. On MetaQA with ”half” setting, we set $T = 4$.

5.2 Main Results

In Table 3, we show the results of our proposed DCRN on WebQuestionSP and MetaQA datasets. Overall, our model significantly outperforms state-of-the-art models on benchmark datasets.

WebQuestionSP is a small dataset but it uses a large-scale KG, which is a subset of Freebase. This dataset follows an inductive setting—some entities in the test set have not appeared in the training set. Experiments demonstrate that our DCRN achieves 67.8 on H@1, which outperforms GraftNet and KV-Mem, and performs comparatively against previous state-of-the-art PullNet.

MetaQA is a large dataset consisting of 1 to 3-hop questions. Overall, our DCRN achieves state-of-the-art on all the three subdatasets. On MetaQA 1-hop and 2-hop, although some previous methods exhibits satisfying performance, they fail to achieve

Table 3: Evaluation results (H@1) of the proposed DCRN and previous state-of-the art methods on WebQuestionSP and MetaQA datasets.

Methods	**WebQSP**	**MetaQA 1-hop**	**MetaQA 2-hop**	**MetaQA 3-hop**
KV-Mem (Miller et al., 2016)	46.7	96.2	82.7	48.9
VRN (Zhang et al., 2018)	-	**97.5**	89.9	62.5
GraftNet (Sun et al., 2018)	66.4	97.0	94.8	77.7
PullNet (Sun et al., 2019a)	**68.1**	97.0	**99.9**	91.4
EmbedKGQA (Saxena et al., 2020)	66.6	**97.5**	98.8	94.8
DCRN (Ours)	67.8	**97.5**	**99.9**	**99.3**

consistent performances on both datasets. For example, PullNet only achieves 97.0 on MetaQA 1-hop, and EmbedKGQA only achieves 98.8 on MetaQA 2-hop. Different from previous methods, our DCRN achieves state-of-the-art on both MetaQA 1-hop and 2-hop.

The questions in the MetaQA 3-hop dataset are more difficult to answer compared to those in MetaQA 1-hop and 2-hop, as they require longer reasoning paths to find answers. However, experiments demonstrate that our model achieves 99.3 on H@1, which significantly outperforms previous state-of-the-arts. Specifically, it gains 7.9 against PullNet and 4.5 against EmbedKGQA. The results on MetaQA 3-hop illustrates the effectiveness of our model on answering questions that require long reasoning paths.

We also conduct experiments in "half" setting. In this setting, 50% of triplets are dropped, making it more challenging to accurately identify answers. The results are shown in Table 4. Experiments demonstrate our model achieves state-of-the art on all subsets of MetaQA.

Table 4: The evaluation results (H@1) of DCRN and previsou state-of-the art methods on MetaQA datasets under the "**half**" setting. In this setting, 50% triplets are randomly dropped.

Methods	**MetaQA 1-hop**	**MetaQA 2-hop**	**MetaQA 3-hop**
KV-Mem	63.6	41.8	37.6
GraftNet	64.0	52.6	59.2
PullNet	65.1	52.1	59.7
EmbedKGQA	83.9	91.8	70.3
DCRN (Ours)	**88.5**	**91.9**	**72.5**

5.3 Analysis

In this part, we conduct analysis on our model. In Section 5.3.1, we conduct ablation studies on the two phases in DCRN. In Section 5.3.2, we conduct a case study to illustrate the two-phase strategy of the proposed DCRN.

5.3.1 Ablation Studies on the Two Phases

In Table 5, we conduct ablations studies to show the performances of the two phases in DCRN.

Table 5: Ablation results (H@1) of the two modules on WebQuestionSP and MetaQA datasets. "Unconscious" and "Conscious" denote the unconscious phase and the conscious phase, respectively.

Methods	**WebQSP**	**MetaQA 1-hop**	**MetaQA 2-hop**	**MetaQA 3-hop**
DCRN	**67.8**	**97.5**	**99.9**	**99.3**
Unconscious Phase	60.8	96.9	92.1	68.4
Conscious Phase	47.2	97.4	93.8	37.2

Overall, the experiments show that both two phases are indispensable in our model. The reason is that the unconscious and the conscious phase are designed to better exploit node-level and path-level features, respectively. Both levels of features are critical to the accurate answer identification. Therefore, the cooperation of the two phases brings significant improvements to the performance, as shown in Table 5. On MetaQA 1-hop and 2-hop, both two phases achieves satisfying performances, as the number of candidate entities is relatively small. Furthermore, the conscious phase outperforms the unconscious phase. This is because the unconscious phase exploits the coarse-grained semantic of entities, while the conscious phase considers the fine-grained relational dependencies between entities. Therefore, on small candidate entity sets, the conscious phase could make more accurate predictions.

On MetaQA 3-hop, the unconscious phase outperforms conscious phase. This is because 3-hop questions usually have large candidate entity set, and the errors can propagate along reasoning paths. Therefore, to make accurate predictions, DCRN requires the unconscious phase to softly filter out irrelevant candidates. Experiments demonstrate that, by considering both phases, DCRN achieves 99.3 on H@1.

We further compare between the unconscious phase of DCRN and EmbedKGQA (Saxena et al., 2020). EmbedKGQA consists of two parts—knowledge graph embedding and relation matching. The former part use the question representation as latent relation embedding. Different from EmbedKGQA, the unconscious phase in DCRN decode a question into relation paths. To illustrate the effectiveness of the unconscious phase, we compare it with EmbedKGQA (w/o relation matching), and the results are shown in Table 6.

Table 6: Comparisons between EmbedKGQA (w/o relation matching) and the unconscious phase in the proposed DCRN on the MetaQA datasets.

Methods	**MetaQA 1-hop**	**MetaQA 2-hop**	**MetaQA 3-hop**
EmbedKGQA (w/o relation matching)	94.7	86.5	67.2
Unconscious Phase	**96.9**	**92.1**	**68.4**

Note that EmbedKGQA (Saxena et al., 2020) use RoBERTa (Liu et al., 2019) for word embeddings and ComplEx (Trouillon et al., 2016) for entity embeddings. For fair comparison with previous work including GRAFT-Net (Sun et al., 2018) and PullNet (Sun et al., 2019a), we use GloVe (Pennington et al., 2014) for word embeddings and CP (Hitchcock, 1927) for entity embeddings, and we reimplement EmbedKGQA (w/o relation matching) under our settings.

Experiments demonstrate that the unconscious phase outperforms EmbedKGQA (w/o relation matching) on all the three datasets of MetaQA, illustrating the effectiveness of our design on the query representation in the unconscious phase.

5.3.2 Case Study

In this part, we conduct a case study to illustrate the effectiveness of the two-phase strategy in DCRN. In Figure 5, we show the predictions made by DCRN on a 2-hop question *who is listed as screenwriter of John Derek acted films?*. This question is taken from the test set of MetaQA 2-hop.

The figure on the left shows the predictions of the unconscious phase. It shows that the unconscious phase successfully filters out the candidates that are unlikely to be correct answers. The predictions made by the unconscious phase provide informative evidence for the subsequent conscious phase. The figure on the right shows the predictions of the conscious phase. Based on the retrieved evidence, the conscious phase successfully ranks the correct answers in the first place.

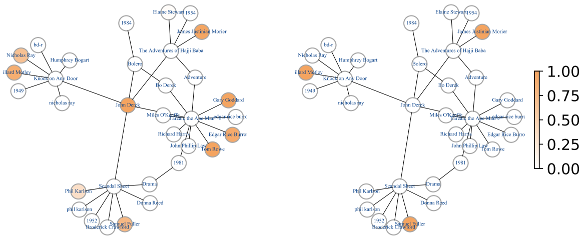

Figure 5: Illustration of the predictions made by the unconscious phase (**left**) and the conscious phase (**right**) to the question *who is listed as screenwriter of John Derek acted films?*. We exhibit the 2-hop subgraph of the topic entity *John Derek*. Deeper color for an entity indicates higher prediction score as a correct answer.

6 Conclusion

Multi-hop question answering over knowledge graphs aims to answer questions by multi-hop reasoning over knowledge graphs to find answers. In this work, we propose a novel Deep Cognitive Reasoning Network (DCRN), which is inspired by the dual process theory in cognitive science. DCRN can accurately identify answers with two phases—unconscious phase and conscious phase. Experiments demonstrate that our model outperforms state-of-the-art methods on benchmark datasets.

Acknowledgments

We would like to thank all the anonymous reviewers for their insightful comments. This work was supported in part by NSFC (61822604, 61836006, U19B2026), Fundamental Research Funds for the Central Universities No. WK3490000004, and National Key Research and Development Program of China under Grant 2017YFB1002203.

References

Yoshua Bengio. 2017. The consciousness prior. *arXiv e-prints*, 1709.08568.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. SIGMOD '08, page 1247–1250.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2694–2703.

Zhengxiao Du, Chang Zhou, Ming Ding, Hongxia Yang, and Jie Tang. 2019. Cognitive knowledge graph reasoning for one-shot relational learning. *arXiv preprint arXiv:1906.05489*.

Jonathan Evans. 2003. In two minds: Dual-process accounts of reasoning. *Trends in cognitive sciences*, 7:454–9.

Jonathan Evans. 2008. Dual-processing accounts of reasoning, judgment, and social cognition. *Annual review of psychology*, 59:255–78.

Jonathan St B. T. Evans. 1984. Heuristic and analytic processes in reasoning. *British Journal of Psychology*, 75(4):451–468.

Bin Fu, Yunqi Qiu, Chengguang Tang, Y. Li, H. Yu, and J. Sun. 2020. A survey on complex question answering over knowledge base: Recent advances and challenges. *ArXiv*, abs/2007.13069.

F. L. Hitchcock. 1927. The expression of a tensor or a polyadic as a sum of products. *J. Math. Phys*, 6(1):164–189.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Feng-Lin Li, Minghui Qiu, Haiqing Chen, Xiongwei Wang, Xing Gao, Jun Huang, Juwei Ren, Zhongzhou Zhao, Weipeng Zhao, Lei Wang, Guwei Jin, and Wei Chu. 2017. Alime assist: An intelligent assistant for creating an innovative e-commerce experience. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2495–2498.

Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling relation paths for representation learning of knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 705–714.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019a. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019b. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML*.

Yingxu Wang, Dong Liu, and Ying Wang. 2003. Discovering the capacity of human memory. *Brain and Mind*, 4(2):189–198.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1321–1331.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhanqiu Zhang, Jianyu Cai, and Jie Wang. 2020a. Duality-induced regularizer for tensor factorization based knowledge graph completion. In *Advances in Neural Information Processing Systems*, volume 33, pages 21604–21615.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020b. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 3065–3072. AAAI Press.

Appendix

A Proofs of the Propositions

Proposition 1. *The pruned graph $\hat{\mathcal{G}}(e_{topic})$ is a directed acyclic graph (DAG).*

Proof. *The pruned graph $\hat{\mathcal{G}}(e_{topic})$ is obtained by employing the Breadth-First Search (BFS) algorithm starting from e_{topic}. During the search process, we only keep the visited edges.*

First, it is clear that $\hat{\mathcal{G}}(e_{topic})$ is a directed graph. Second, we prove that $\hat{\mathcal{G}}(e_{topic})$ is acyclic. During the BFS search, the nodes are classified into several categories according to their topological distance to e_{topic}. After the pruning, each edge in $\hat{\mathcal{G}}(e_{topic})$ must connect a node with topological distance n to e_{topic} and a node with distance $(n+1)$ to e_{topic}. Therefore, there is no loop in $\hat{\mathcal{G}}(e_{topic})$.

Proposition 2. *The marginal probability $\mathbb{P}(X_e = 1|\mathcal{G}, q, e_{topic})$ that predicts a candidate entity e can be calculated via variable elimination:*

$$\begin{aligned}&\mathbb{P}(X_e = 1|\mathcal{G}, q, e_{topic})\\ &= \overline{\mathbb{P}}(X_e = 1) \prod_{e' \in pa(e)} \mathbb{P}(X_{e'} = 0|\mathcal{G}, q, e_{topic}),\end{aligned}$$

where $pa(e)$ denotes the set of parent nodes of e, i.e., the nodes that have edges directing at e. The first component $\overline{\mathbb{P}}(X_e = 1)$ is the abbreviation of

$$\mathbb{P}(X_e = 1|\mathcal{G}, q, e_{topic}, X_{pa(e)} = 0).$$

Proof. *First, the marginal probability is defined as*

$$\begin{aligned}&\mathbb{P}(X_e = 1|\mathcal{G}, q, e_{topic}) =\\ &\sum_{e_1, e_2, ... \in \mathcal{E}/e} \mathbb{P}(X_e = 1, X_{e_1}, X_{e_2}, ...|\mathcal{G}, q, e_{topic}),\end{aligned}$$

where $\mathbb{P}(X_e = 1, X_{e_1}, X_{e_2}, ...|\mathcal{G}, q, e_{topic})$ is the joint probability of variables $X_{e_1}, ...X_{e_{|\mathcal{E}|}}$.

By the definition of Bayesian networks, the joint probability is factorized into several components:

$$\begin{aligned}&\mathbb{P}(X_e = 1, X_{e_1}, X_{e_2}, ...|\mathcal{G}, q, e_{topic}) =\\ &\prod_{v \in \mathcal{E}} \mathbb{P}(X_v|X_{pa(v)}, \mathcal{G}, q, e_{topic}),\end{aligned}$$

Therefore, the marginal probability is represented as follows:

$$\begin{aligned}&\mathbb{P}(X_e = 1|\mathcal{G}, q, e_{topic}) =\\ &\sum_{v \in \mathcal{E}/e} \prod_{v \in \mathcal{E}} \mathbb{P}(X_v|X_{pa(v)}, \mathcal{G}, q, e_{topic}).\end{aligned}$$

We then perform variable elimination, which eliminates the variables other than X_e and $X_{pa(e)}$. The results are as follows:

$$\begin{aligned}&\mathbb{P}(X_e = 1|\mathcal{G}, q, e_{topic}) =\\ &\sum_{e' \in pa(e)} \hat{\mathbb{P}}(X_e = 1)\mathbb{P}(X_{e'}|\mathcal{G}, q, e_{topic}),\end{aligned}$$

where the notation $\hat{\mathbb{P}}(X_e = 1)$ denotes

$$\mathbb{P}(X_e = 1|\mathcal{G}, q, e_{topic}, X_{pa(e)}).$$

We then make the following assumption: if any parent of e is an answer, then e is not an answer. This assumption represents the fact that each question corresponds to a unique reasoning path.

Following this assumption, we have that

$$\mathbb{P}(X_e = 1|\mathcal{G}, q, e_{topic}, X_{pa(e)}) = 0,$$

if there exists $e' \in pa(e)$ such that $X_{e'} = 1$. Therefore, we have the following conclusion:

$$\begin{aligned}&\mathbb{P}(X_e = 1|\mathcal{G}, q, e_{topic}) =\\ &\overline{\mathbb{P}}(X_e = 1) \prod_{e' \in pa(e)} \mathbb{P}(X_{e'} = 0|\mathcal{G}, q, e_{topic}),\end{aligned}$$

where $pa(e)$ denotes the set of parent nodes of e, i.e., the nodes that have edges directing at e. The first component $\overline{\mathbb{P}}(X_e = 1)$ is the abbreviation of

$$\mathbb{P}(X_e = 1|\mathcal{G}, q, e_{topic}, X_{pa(e)} = 0).$$