RCENR: A Reinforced and Contrastive Heterogeneous Network Reasoning Model for Explainable News Recommendation

Hao Jiang
Communication University of China,
Beijing, China
jianghaocuc@foxmail.com

Chuanzhen Li
Communication University of China,
Beijing, China
lichuanzhen@cuc.edu.cn

Juanjuan Cai
Communication University of China,
Beijing, China
caijuanjuan@cuc.edu.cn

Jingling Wang
Communication University of China,
Beijing, China
wjl@cuc.edu.cn

ABSTRACT

Existing news recommendation methods suffer from sparse and weak interaction data, leading to reduced effectiveness and explainability. Knowledge reasoning, which explores inferential trajectories in the knowledge graph, can alleviate data sparsity and provide explicitly recommended explanations. However, brute-force preprocessing approaches used in conventional methods are not suitable for fast-changing news recommendation. Therefore, we propose an explainable news recommendation model: the **Reinforced and Contrastive Heterogeneous Network Reasoning Model for Explainable News Recommendation** (RCENR), consisting of NHN-R^2 and MR&CO frameworks. The NHN-R^2 framework generates user/news subgraphs to enhance recommendation and extend the dimensions and diversity of reasoning. The MR&CO framework incorporates contrastive learning with a reinforcement-based strategy for self-supervised and efficient model training. Experiments on the MIND dataset show that RCENR is able to improve recommendation accuracy and provide diverse and credible explanations.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Natural language processing**.

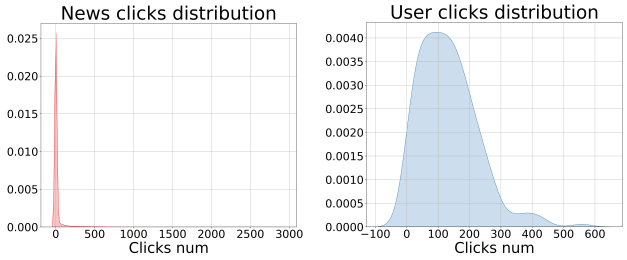
KEYWORDS

News Recommendation; Explainable Recommendation; Knowledge Reasoning; Contrastive Learning; Markov Decision Process

ACM Reference Format:
Hao Jiang, Chuanzhen Li, Juanjuan Cai, and Jingling Wang. 2023. RCENR: A Reinforced and Contrastive Heterogeneous Network Reasoning Model for Explainable News Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3539618.3591753

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591753



(a) Distribution of news clicks (b) Distribution of user clicks

Figure 1: Distribution of news clicks and user clicks.

1 INTRODUCTION

Nowadays, personalized news recommendations are essential for alleviating information overload and helping users quickly find content of interest. However, two critical and pending problems persist: (1) *sparse and weak interaction data* and (2) *poor interpretability*.

Sparse and Weak Interaction Data (P1). News recommendation faces specialized challenges compared to traditional recommendation scenarios due to the vast number of rapidly published news [8, 32] and limited user interaction data with weak feedback, i.e., "Clicks". As our analysis of the real-world MIND dataset from MSN News[1], shown in Fig. 1, A staggering **87.7%** of news articles are never clicked by anyone, while user click behaviors exhibit a long-tailed distribution and are concentrated in roughly **20.14%** of the news articles. These statistics highlight the need for news recommendation algorithms to recommend news articles that have not been previously observed by the model to users whose interests are uncertain and ambiguous. As a result, accurately capturing user interests for specific news from sparse and weak user behaviors remains a major challenge in news recommendation. Therefore, collaborative filtering (CF)-based methods [7, 36] are constrained by the limited quality and quantity of interaction data, resulting in poor recommendation performance. While content-based methods [3, 12, 13, 51, 52, 56] alleviate some issues, they struggle to capture users' personalized preferences for complex and diversity news content due to static news representation, leading to biased recommendations [67]. Moreover, insufficient click behaviors can lead to imprecise user interest modeling [65], and click noise can easily introduce uncertainty into user interest representation.

[1]https://www.msn.com/en-us/news

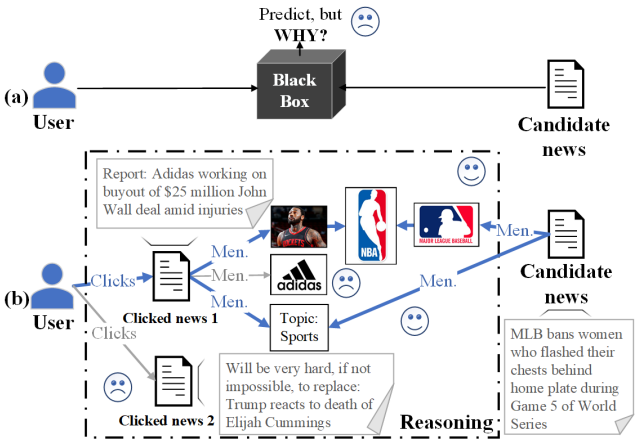


Figure 2: Two pipelines of news recommendation.

Poor Interpretability (P2). News recommendations typically use the collaborative filtering framework to infer users' preferences based on their historical behaviors. However, interpretability is often poor due to sparse interaction data, which makes it difficult to understand why a recommendation is made. Even advanced news recommendation methods [1, 51, 52, 54], such as those that use complex encoders to derive user and news representations, often treat these *Encoder* as "Black-Box" networks (e.g., as shown in Fig. 2 pipeline (a)). This fact can have a negative impact on user satisfaction and engagement, as users may not trust the recommendations or understand why they are being recommended. Therefore, providing convincing explanations is crucial for intelligent news recommendation systems [28].

Fortunately, **Knowledge Reasoning** has emerged as an effective solution to the two major problems in news recommendation, as demonstrated by several recent studies [6, 24, 50, 64]. By exploring multi-hop connection paths between users and items in a knowledge graph (KG), it can provide explicit explanations for recommendations. Moreover, this approach assists to eliminate behavioral bias resulting from factors such as "user-misclick" or "clickbait" [48], and can utilize candidate news to understand the user's current intention [43]. Additionally, it is also able to infer users' fine-grained interests for different types of news content [45, 51, 66], thus helping to create personalized news representations for each user. As depicted in Fig. 2 pipeline (b), the reasoning process removes the negative influence of irrelevant behaviors, such as click noise (e.g., *clicked news 2*), and reveals the reasons behind recommendations. For instance, it can uncover the user's fine-grained interest in topics such as Sports or news content such as American Sports League (e.g., NBA or MLB), rather than irrelevant content like Adidas Inc. This approach can eliminate biases related to news content and offer more precise recommendations based on denoising user behaviors.

Existing methods for knowledge reasoning often treat the reasoning process as a *Markov Decision Process* (MDP) and optimize it using reinforcement-based strategies. However, we argue that these methods are not suitable for adapting to the rapidly changing news recommendation scenario due to two limitations:

Slow Convergence Speed (P3). Most methods are hindered by the issue of sparse reward signals, resulting in slow convergence speed, particularly in large-scale reasoning networks.

Complex Pre-processing Steps (P4). Many methods rely on pre-processing meta-paths [64] or all feasible paths [24] to generate ground-truth labels for supervised training. They are highly susceptible to ground-truth paths and extremely time-consuming.

Therefore, combining *Reasoning* with news recommendations is a rewarding and challenging research field. In this paper, we propose a novel news recommendation model: Reinforced and Contrastive Heterogeneous Network Reasoning Model for Explainable News Recommendation (RCENR). We introduce the **News Heterogeneous Network Reasoning Recommendation** (NHN-R^2) framework, consisting of three components: reinforced subgraph generation, news heterogeneous network reasoning, and co-enhanced news recommendation. First, we construct a News Heterogeneous Network (NHN) that integrates user interaction information and various types of news content. Next, we design a reinforced subgraph generation module to create specific user/news subgraphs for each user-candidate news pair and apply them to both the recommendation and reasoning tasks. For the recommendation task, we extract collaborative signals from the absorbing nodes in the subgraph (for P1) to enhance the recommendation. For the reasoning task, we first extract overlapping nodes and multi-hop connection paths between users/news subgraphs and then calculate their credit scores as the model's explainability metric (for P2), which reveals user interest in specific news content. More importantly, the generated subgraphs expand the dimension of the reasoning scope from "1-D" to "2-D". Finally, our model outputs credible and diverse multi-hop paths as the explanations for each recommended news.

In this framework, efficient training of the complex model is a critical challenge. Recently, Contrastive Learning, which automatically generates positive/negative samples through data augmentation, has demonstrated remarkable success in both computer vision [5, 14] and natural language processing [10, 58] fields. Therefore, we utilize the concept of contrastive learning and propose a novel **Multi-task Reinforced and Contrastive Optimizing (MR&CO)** framework that fuses Reinforcement Learning (RL) and Contrastive Learning (CL) to train the NHN-R^2 framework. Specifically, we treat the subgraph generation process as a Markov Decision Process (MDP) and design a novel Contrastive Reward, along with three other rewards, for self-supervised training. This approach can accelerate convergence by manually constructed reward signals (for P3) and eliminate pre-processing procedures (for P4), promoting connections between subgraphs while efficiently removing redundant nodes within subgraphs. We implement an end-to-end multi-task optimization to facilitate collaboration between different tasks.

To summarize, our major contributions are as follows:

- We propose a news heterogeneous network reasoning recommendation framework (NHN-R^2), which generates subgraphs for enhancing recommendation performance and providing a credible and diverse explanation of recommendation results.
- We propose a joint multi-task optimization framework (MR&CO), which combines the concepts of self-supervised contrastive learning and unsupervised reinforcement-based strategy for efficient markov decision process training and collaboration among multi-task.
- Our model performs experiments using the real-world MIND dataset and demonstrates superior performance compared to existing news recommendation and reasoning methods.

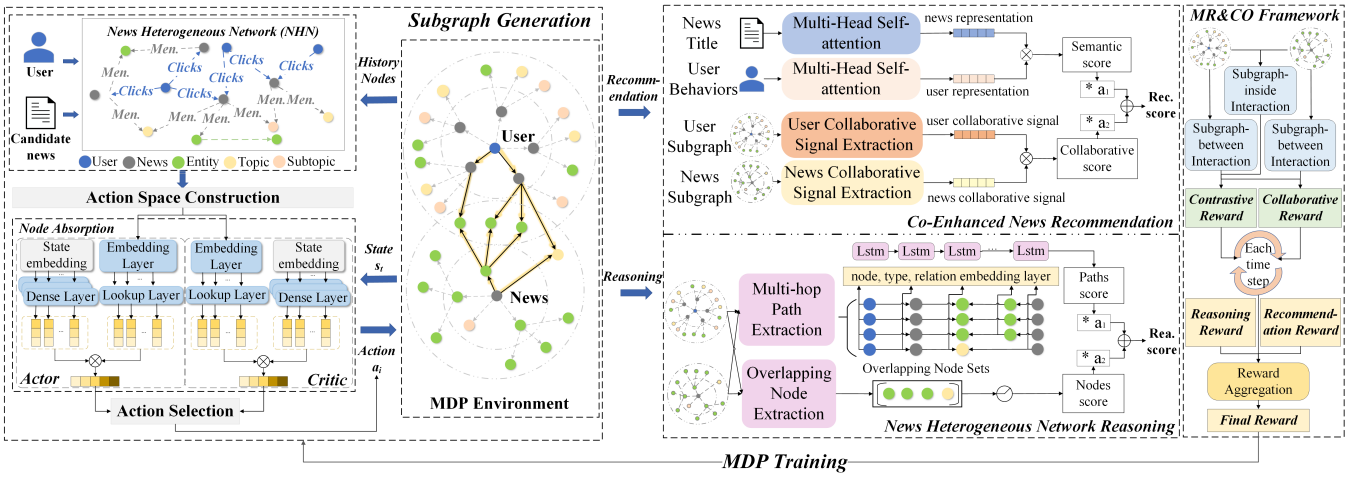


Figure 3: Overall framework of the RCENR model.

2 METHODOLOGY

This section first provides important definition of the News Heterogeneous Network and its associated model tasks, followed by a detailed explanation of the framework and components of RCENR, including reinforced subgraph generation, news heterogeneous network reasoning, and co-enhanced news recommendation.

2.1 Preliminaries

We build a News Heterogeneous Network and clarify the task of News Heterogeneous Network Reasoning Recommendation.

News Heterogeneous Network (NHN). The News Heterogeneous Network (NHN) is defined as a graph $\mathcal{N} = (\mathcal{V}, \mathcal{R}_n)$, where $|\mathcal{V}| \geq 1$ and $|\mathcal{R}_n| \geq 1$ represent the sets of nodes/entities and links/relationships, respectively. The network consists of users ($\mathcal{U}$), candidate news (N_c), user-clicked news (N_u), entities ($\mathcal{E}_{kg}$), topics (C), and subtopics (C_s). Each user is directly connected to the news they have clicked, and each news node is directly connected to its semantic nodes (entities, topics, and subtopics). The network also incorporates external knowledge in the form of a knowledge graph $\mathcal{G} = \{(h, r, t) \mid h, t \in \mathcal{E}_{kg}, r \in \mathcal{R}_n\}$ that is aggregated to the NHN through news entity alignment. Each news entity is associated with a set of triplets $\{(e, r, e_t) \mid e_t \in \mathcal{N}_h, r \in \mathcal{R}_n\}$ in $\mathcal{G}$, where $\mathcal{N}_h$ denotes the set of neighbor entities. The $\mathcal{R}_n$ contains "clicks-relation", "mentions-relation", and original external knowledge relation.

News Heterogeneous Network Reasoning Recommendation (NHN-R^2) task. Based on the News Heterogeneous Network (NHN), we recommend a list of k news, $\{n_1, n_2, \ldots, n_k\}$, to a given user ($u$). Furthermore, for each recommended news (n_i), our model provides an explanation by outputting a set of M-Hop paths ($\Delta_{u \Leftrightarrow n_i}$) connecting u and n_i in the NHN. An M-Hop path is a continuous trajectory in the NHN, denoted as $p^M = \{e_0, r_1, e_1, r_2, e_2, \cdots, r_M, e_M\}$, that connects a source node (u) and a target node (n_i).

2.2 RCENR Framework

Our method comprises three main components, as shown in Fig. 3. The core of the model is the generation of a unique personalized subgraph for each user-candidate news pair $\langle u, n \rangle$ in the News Heterogeneous Network, which is treated as a Markov Decision Process (MDP) and shown in the **Left Part** of Fig. 3. The subgraph is centered around the user/news node and expands by searching for connected nodes to absorb at each step within a fixed number. Once the subgraphs are generated, they are utilized to enhance downstream reasoning and recommendation tasks, as depicted in the **Middle Part** of Fig. 3. In particular, we conduct subgraph reasoning from a "2-D" perspective, which improves the diversity of reasoning paths while reducing computational complexity. Moreover, the subgraph enables us to explore collaboration signals and update the original user/news representation, thereby enhancing the recommendation by uncovering potential relationships between user behaviors and news content. To further improve subgraph generation and multi-task collaboration, we develop an end-to-end optimization framework, as shown in the **Right Part** of Fig. 3.

2.3 Reinfored Subgraph Generation

In this section, we represent the process of generating the subgraph as a Markov Decision Process [40] and use the A3C algorithm [29] to generate the subgraph by absorbing valuable connected nodes.

2.3.1 Markov Decision Process. The Markov Decision Process can be represented as $M = \{\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}\}$, where $\mathcal{S}$ and $\mathcal{A}$ are the sets of states and actions, $\mathcal{R}$ is the set of rewards obtained by the agent after interacting with the environment, and $\mathcal{P}$ represents the transition probability. We represent the user/news subgraphs as $\mathcal{S}_{G_o} = (\mathcal{E}_o, \mathcal{R}_o, \mathcal{T}_o)$, where o represents either the user or the candidate news. Here, $\mathcal{E}_o \in \mathcal{E}$, $\mathcal{R}_o \in \mathcal{R}_n$, and $\mathcal{T}_o \in \mathcal{T}$ denote the ontology, relationship, and type of the absorbing nodes, respectively.

Specifically, the *User-node* and *News-node* are initially anchored as the center of the subgraph, and the next-hop of history absorbing nodes is defined as the action space. At each time step (t), the agent selects a fixed number of nodes ($\mathcal{S}^t_{G_o}$) from the action space based on the calculated value of each action.

- **States**: We use $s_t \in \mathcal{S}$ to represent the state of each time step. To differentiate between node attributes, we integrate the *type of node* in the state (s_t). Specifically, we define the state as a five-tuple: $(o_a, \mathcal{E}_t, \mathcal{R}_t, \mathcal{T}_t, E_t)$, where o_a is the original central node, $\mathcal{E}_t$ represents the historical actions, $\mathcal{R}_t$ represents the relationships of historical actions, $\mathcal{T}_t$ represents the type of historical actions, and E_t represents the actions of current time step. The initial state

is defined as $s_0 = (o_a, \{N_o\}, \emptyset, \{\mathcal{T}_{N_o}\}, \{N_o\})$. Here, $\emptyset$ denotes the empty set, and $\{N_o\}$ and $\{\mathcal{T}_{N_o}\}$ represent the first-order neighbors of the central node and corresponding node type, respectively.

- **Actions**: Based on the state (s_t), we define the action space as the neighboring nodes of the current nodes (E_t) in the NHN: $\mathcal{A}_{s_t} = \{(r_i, e_i, t_i) \mid (e_h, r_i, e_i) \in \mathcal{N}, e_h \in E_t, e_i \notin \mathcal{E}_t\}$. The *Agent* is used to determine the quality of each action and select the absorbing nodes.
- **Rewards**: To evaluate the effectiveness of actions in interacting with the MDP environment and to assess the overall reliability of the subgraph, we design two types of rewards: Immediate Rewards ($\mathcal{R}_I$) and Terminal Rewards ($\mathcal{R}_T$).
- **Transition possibility**: For a given state (s_t) and action ($a_i \in \mathcal{A}_{s_t}$), the transition to the next state $s_{t+1} = (o_a, \mathcal{E}_{t+1}, \mathcal{R}_{t+1}, \mathcal{T}_{t+1}, E_{t+1})$ is calculated using the probability derived from the *Agent*. The *Agent* is a policy network that determines this probability using Eq. (1).

$$\mathcal{P}\left(s_{t+1} = (o_a, \mathcal{E}_{t+1}, \mathcal{R}_{t+1}, \mathcal{T}_{t+1}, E_{t+1}) \mid s_t, \mathcal{A}_{s_t}\right) = 1. \tag{1}$$

2.3.2 Subgraph Absorbing Strategy. We use the A3C algorithm with an *Actor* and a *Critic* for subgraph generation, where the *Actor* creates the absorbing policy, while the *Critic* assesses action quality. Specifically, the subgraph generation process consists of embeddings generation for states/actions and node absorption.

Embedding Generation. We create the state embedding ($\mathbf{s}_t$) by combining the embeddings of four attributes: $\mathbf{O}$, $\mathbf{E}_t$, $\mathbf{R}_t$, and $\mathbf{T}_t$, as follows: $\mathbf{s}_t = [\overline{\mathbf{O}} \oplus \overline{\mathbf{E}_t} \oplus \overline{\mathbf{R}_t} \oplus \overline{\mathbf{T}_t}]$. Similarly, for each action $a_i = (r_i, e_i, t_i)$, we generate the action embedding ($\mathbf{a}_i$) by combining its three attribute embeddings: $\mathbf{R}_i$, $\mathbf{T}_i$, and $\mathbf{E}_i$, as follows: $\mathbf{a}_i = [\mathbf{R}_i \oplus \mathbf{T}_i \oplus \mathbf{E}_i]$. Here, the $\mathbf{O} \in \mathbb{R}^d$, $\mathbf{E}_t \in \mathbb{R}^{n \times d}$, $\mathbf{R}_t \in \mathbb{R}^{n \times d}$, and $\mathbf{T}_t \in \mathbb{R}^{n \times d}$ represent the embedding of the central node, ontology, relationship, and type, respectively. The symbol $\overline{\mathbf{V}}$ denotes average product, and the symbol $\oplus$ denotes element-wise sum product. Specifically, the node embedding ($\mathbf{E}_i$) stores the main information of each action, the relationship embedding ($\mathbf{R}_i$) contains contextual information, and the type embedding ($\mathbf{T}_i$) distinguishes heterogeneous nodes. Moreover, the state and action share the same attribute embedding.

Node Absorption. Given a specific state (s_t) and action (a_i), the probability of a node being absorbed, $\pi_\beta(s_t, a_i) \in \mathbb{R}^1$, is computed by the Actor network. The Actor learns a node absorption policy (π_β) that calculates the probability distribution of action (a_i) based on the state (s_t). Furthermore, we use a Critic network [22] to estimate the Q-value of each action, $Q_\theta(s_t, a_i) \in \mathbb{R}^1$, which is used to evaluate the reward received after interaction with the MDP environment. The transfer policy (π_β) and the Q-value (Q_θ) are obtained by fully connected layers in the Actor and Critic networks and are normalized using the *softmax* and *sigmoid* functions, respectively. The complete process is shown in the Eqs. (2) and (3), where $\mathbf{W}_{\beta 1}$, $\mathbf{W}_{\beta 2}$, $\mathbf{W}_{\theta 1}$, $\mathbf{W}_{\theta 2}$ are the trainable parameters.

$$\pi_\beta(s_t, a_i) = \text{Softmax}\left(\mathbf{W}_{\beta_1} \times \text{ReLU}\left(\mathbf{W}_{\beta_2} \times (\mathbf{s}_t \oplus \mathbf{a}_i)\right)\right). \tag{2}$$

$$Q_\theta(s_t, a_i) = \text{Sigmoid}\left(\mathbf{W}_{\theta_1} \times \text{ReLU}\left(\mathbf{W}_{\theta_2} \times (\mathbf{s}_t \oplus \mathbf{a}_i)\right)\right). \tag{3}$$

Then, we select the nodes with *Top-d* transfer probabilities as the absorbing nodes, and further incorporate them into the subgraph ($\mathcal{S}_{G_o}$) at each time step, $\mathcal{S}^t_{G_o} = \left\{e_i \mid \pi_\beta(s_t, a_i) \in Top_{d_t}\right\}$, where $d_t \in D$ is a hyperparameter indicating the maximum number of nodes absorbed at each time step (t). In this paper, we set $D = [5, 3, 2]$ and the total number of hops (t) is set to 3. If a node fails to find a suitable next-hop, a "self-loop" relation is added to the node. Besides, to prevent the initial subgraph from influencing the subsequent ones, we use an alternating generation approach. For a user-news pair $\langle u, n \rangle$, we refer to the subgraph being explored as the **Varied-subgraph** and the one that remains constant as the **Fixed-subgraph**. These roles are exchanged at each time step.

2.4 News Heterogeneous Network Reasoning

Once the subgraphs are generated, we apply them to the reasoning task. Given a user-news pair $\langle u, n \rangle$ and their subgraphs, we conduct news heterogeneous network reasoning as following.

Reasoning Method. According to the structure of the subgraph, we design two methods of reasoning: *multi-hop paths* and *overlapping nodes*. To explore the paths between user (u) and news (n), we sample all possible paths using a Random Walks Algorithm [27]. These paths are denoted by $p_i \in \Delta_{u \Leftrightarrow n}$, which is a sequence of nodes and relationships that exist in the multi-hop connections between u and n. Additionally, we regard the overlapping nodes in both user and news subgraphs as $\mathrm{N}_{over} = \{e_i \mid e_i \in S_{G_u}, e_i \in S_{G_n}\}$.

Reasoning Score. We employ the reasoning paths and overlapping nodes to calculate the reasoning scores with Eq. (4), where α_1 and α_2 are hyperparameters, $\Delta_{u \Leftrightarrow n}$ denotes the set of connection paths, and N_{over} denotes the set of overlapping nodes. We encourage the reasoning scores of positive user-news pairs to be greater than those of negative pairs, indicating a stronger potential intention between the user and the interactive news.

$$\text{score}_\psi(u, n) = \alpha_1 \times \sum_{p_i \in \Delta_{u \Leftrightarrow n}} \text{score}_{path}(p_i) + \alpha_2 \times \sum_{e_i \in \mathrm{N}_{over}} \text{score}_{node}(e_i). \tag{4}$$

For the *path score*, we utilize the long-term and short-term memory (LSTM) [16] network to capture contextual information. The output vector of the last layer in the LSTM is used to calculate the *path score* (Eq. (5)). For the *node score*, we map the nodes embedding and corresponding types embedding to the same latent space using a dense layer, and then calculate the *node score* using another dense layer to measure their significance (Eq. (6)).

$$\text{score}_{path}(p_i) = \sigma\left(\mathbf{W}^p_\psi \times \text{Tanh}\left(\text{LSTM}\left(\mathbf{p}_i\right)\right)\right), \tag{5}$$

$$\text{score}_{node}(e_i) = \sigma\left(\mathbf{W}^n_\psi \times \text{Tanh}\left(\mathbf{W}^t_\psi \times (\mathbf{e}_i + \mathbf{t}_i)\right)\right), \tag{6}$$

where $\mathbf{p}_i = [(\mathbf{e}_1 + \mathbf{r}_1) \oplus (\mathbf{e}_2 + \mathbf{r}_2) \oplus \ldots \oplus (\mathbf{e}_M + \mathbf{r}_M)]$ is the representation of path p_i, $\mathbf{W}^p_\psi$, $\mathbf{W}^t_\psi$ and $\mathbf{W}^n_\psi$ are the trainable parameters, $\mathbf{e}_i$ and $\mathbf{t}_i$ is the embedding of e_i and t_i. $\sigma(\cdot)$ is the *sigmoid* function.

2.5 Co-Enhanced News Recommendation

In the recommendation task, we consider both collaborative and semantic information to be equally important. However, due to the sparsity of interactions, identifying collaborations can be a critical challenging. To address this issue, we use the generated subgraph to uncover collaborative signals and enhance the recommendations.

News/User Representation To facilitate comparisons between models, we utilize the State-Of-The-Art *NRMS* [54] framework to obtain semantic representations of both users and news. The news title word embedding matrix is denoted as $\mathbf{E}_\mathrm{T} = [\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_L]$, while the user click behavior embedding matrix is denoted as $\mathbf{U}_{click} = [\mathbf{R}_{click_1}, \mathbf{R}_{click_2}, \ldots, \mathbf{R}_{click_H}]$. We process these matrices using multi-head attention and an additional attention layer to obtain the user representation ($\mathbf{U}$) and the news representation ($\mathbf{R}$).

Collaborative Signal. Nodes within the subgraph reveal both user interests and the degree to which candidate news content aligns with those interests, thus serving as collaborative signals. Therefore, we merge node embedding ($\mathbf{e}_i$) with their corresponding type embedding ($\mathbf{t}_i$) to obtain representations of user and news collaborative signals using Eq. (7), denoted by $\mathbf{C}_u$ and $\mathbf{C}_n$, respectively.

$$\mathbf{C}_o = \sigma\left(\frac{1}{|\mathcal{S}_{G_o}|} \times \sum_{e_i \in \mathcal{S}_{G_o}} \text{Tanh}\left(\mathbf{e}_i + \mathbf{t}_i\right)\right), \tag{7}$$

where the o can denote the user (u) or news (n) and the $|\mathcal{S}_{G_o}|$ is the total number of nodes in subgraph ($\mathcal{S}_{G_o}$).

Co-Enhanced Recommendation. We set hyperparameters to balance the weights of the semantic and collaborative scores in news recommendations by incorporating news/user representation and collaborative signals for co-enhanced news recommendation, as shown in Eq. (8), where α_1 and α_2 are hyperparameters.

$$\text{score}_\phi\left(u, n\right) = \alpha_1 \times \text{score}_{sem}\left(\mathbf{U}, \mathbf{R}\right) + \alpha_2 \times \text{score}_{col}\left(\mathbf{C}_u, \mathbf{C}_n\right). \tag{8}$$

Specifically, *semantic score* is calculated by taking the dot product with Eq. (9), and *collaborative score* is obtained by measuring cosine similarity with Eq. (10). $\sigma\left(\cdot\right)$ is the *sigmoid* function.

$$\text{score}_{sem}\left(\mathbf{U}, \mathbf{R}\right) = \sigma(\mathbf{U}^{\mathrm{T}}\mathbf{R}). \tag{9}$$

$$\text{score}_{col}\left(\mathbf{C}_u, \mathbf{C}_n\right) = \sigma(\text{Cosine}\left(\mathbf{C}_u, \mathbf{C}_n\right)). \tag{10}$$

3 MR&CO FRAMEWORK

For efficient collaborative training, we employ a joint multi-task optimization (MR&CO) framework for end-to-end model training and faster convergence. It contains two main components: *reinforced & contrastive optimization* and *joint multi-task optimization*.

3.1 Reinforced & Contrastive Optimization

We design dual reward signals to evaluate the effectiveness of an agent's interaction with its MDP environment over different time steps. Meanwhile, we aggregate these reward signals to derive the final reward for self-supervised MDP training. To optimize the Actor and Critic networks of the subgraph generation, we utilize the temporal difference [39] and the policy gradient [41].

3.1.1 Dual Reward Signals Design. We construct two types of reward signals for real-time adjustment and final judgment of the *Agent*: the Immediate Rewards $\mathcal{R}_I$ and Terminal Rewards $\mathcal{R}_T$.

1) Immediate Reward (IR). Immediate reward measures the effectiveness of each action selection at each time step.

Collaborative Reward (IR-ColR). The **IR-ColR** measures the effectiveness of actions in the subgraph. Following the pipeline of subgraph generation, IR-ColR promotes a "CLOSER" relationship between the *varied-subgraph* and the *fixed-subgraph* during subgraph generation. We compute the similarity between the *varied-subgraph* and the center node of the *fixed-subgraph* using Eq. (11).

$$\mathcal{R}_I^{Col}\left(t\right) = \mathbb{E}_{a \in \mathcal{S}_{G_o}^t}\left[\text{Cosine}\left(\mathbf{o}_a, \frac{\sum_{e_i \in \mathcal{E}_t}\left(\mathbf{e}_i \oplus \mathbf{r}_i\right)}{|\mathcal{E}_t|} + \left(\mathbf{e}_a \oplus \mathbf{r}_a\right)\right)\right], \tag{11}$$

where $\mathbf{o}_a$ are the embedding of central node in *fixed-subgraph*, $\mathbf{e}_i \in \mathbb{R}^d$, $\mathbf{r}_i \in \mathbb{R}^d$, and $\mathbf{e}_a \in \mathbb{R}^d$, $\mathbf{r}_a \in \mathbb{R}^d$ are the ontology embedding and relationship embedding of both the history node and the current node, and $\mathcal{E}_t$ is the set of history nodes in *varied-subgraph*.

Contrastive Reward (IR-ConR). The **IR-ConR** reflects how well an action performs within a subgraph, while the IR-ColR only considers the impact between subgraphs and ignores node interactions within the subgraph. To improve the precision and efficiency of the subgraph generation process, we utilize the thought of self-supervised contrastive learning to quickly detect redundant nodes and speed up model convergence without supervised labels.

Specifically, we define two types of IR-ConR for subgraph generation: intra-hop and inter-hop. The intra-hop reward ($\mathcal{R}_{Intra}^{Con}$) is computed for nodes in the same hop and treated as a positive pair to reduce noise. In contrast, the inter-hop reward ($\mathcal{R}_{Inter}^{Con}$) is calculated for nodes in different hops and considered a negative pair to expand the search space for subgraphs. The specific example for sampling positive and negative pairs during subgraph generation is shown in Fig. 4. The calculation of $\mathcal{R}_{Intra}^{Con}$ (Part 1) and $\mathcal{R}_{Inter}^{Con}$ (Part 2) is given in Eq.(12).

$$\mathcal{R}_I^{Con}(t) = \lambda\, \underbrace{\mathbb{E}_{a \in \mathcal{S}_{Go}^t}\left[\sum_{i^+ \in \mathcal{S}_{Go}^t, i^+ \neq a}^{|E_t|} \text{Cosine}\left(\left[\mathbf{e}_a \oplus \mathbf{r}_a\right], \left[\mathbf{e}_{i^+} \oplus \mathbf{r}_{i^+}\right]\right)\right]}_{\text{Part 1: intra-hop contrastive rewards: } \mathcal{R}_{Intra}^{Con}}$$

$$+ (1-\lambda)\, \underbrace{\mathbb{E}_{a \in \mathcal{S}_{Go}^t}\left[\sum_{i^- \in \mathcal{S}_{Go}^{t-1}, i^- \neq a}^{|E_{t-1}|} \left(1 - \text{Cosine}\left(\left[\mathbf{e}_a \oplus \mathbf{r}_a\right], \left[\mathbf{e}_{i^-} \oplus \mathbf{r}_{i^-}\right]\right)\right)\right]}_{\text{Part 2: inter-hop contrastive rewards: } \mathcal{R}_{Inter}^{Con}}, \tag{12}$$

where $\oplus$ denotes element-wise sum product, $|E_t|/|E_{t-1}|$ is the number of absorbed nodes at time step $(t/t-1)$, while $\mathcal{S}_{G_o}^t/\mathcal{S}_{G_o}^{t-1}$ are the current sets of nodes absorbed by the subgraph at time step $t/t-1$. The IR-ConR aims to encourage subgraphs to extend towards each other ($\mathcal{R}_{Inter}^{Con}$), while removing redundant nodes ($\mathcal{R}_{Intra}^{Con}$). We balance the inter-hop and intra-hop rewards by adjusting the hyperparameter λ, which is able to control whether the subgraph should extend outward or absorb inward. In this paper, we set $\lambda = 0.5$.

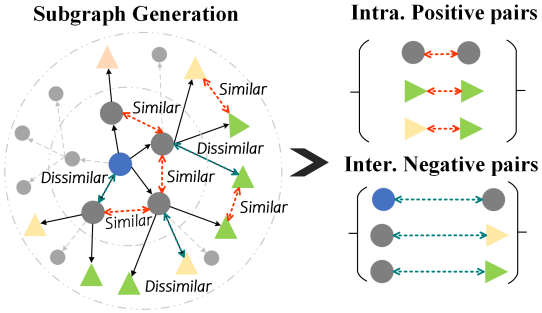


Figure 4: Example of sample pairs in IR-ConR.

2) Terminal Reward (TR): Terminal reward is used to evaluate the suitability of the subgraph for the downstream tasks.

Recommend Reward (TR-RecR). We update the users/news embedding by their subgraph: $\widehat{\mathbf{o}}_a = \mathcal{Y}\left(\mathbf{o}_a, \mathcal{S}_{G_a}\right)$, where $\widehat{\mathbf{o}}_a$ and $\mathbf{o}_a$ denote the representation of the updated user/news and original user/news. Therefore, we define **TR-RecR** by the opposite of the cross-entropy in Eq. (13), measuring whether collaborative signals

from generated subgraphs enhances recommendations.

$$\mathcal{R}_T^{Rec} = \sum_{i \in \mathcal{S}} \log \left(\frac{\exp\left(\mathrm{Y}_{Rec}\left(\widehat{\mathbf{o}}_u, \widehat{\mathbf{o}}_{n_i}\right)^+\right)}{\sum_{j=1}^{K} \exp\left(\mathrm{Y}_{Rec}\left(\widehat{\mathbf{o}}_u, \widehat{\mathbf{o}}_{n_j}\right)^-\right)} \right). \tag{13}$$

Reasoning Reward (TR-ReaR). We consider the quality of the reasoning paths and the number of overlapping nodes in subgraphs as metrics of model explainability. Therefore, we define the **TR-ReaR** as the opposite of the cross-entropy in Eq. (14), with the aim of discovering more credible paths and increasing the number of overlapping nodes between user and interactive news.

$$\mathcal{R}_T^{Rea} = \sum_{i \in \mathcal{S}} \log \left(\frac{\exp\left(\sum_{m=1}^{N} \mathrm{Y}_{Rea}\left(p_m\right) + \tanh\left(\frac{|\mathcal{E}_u \cap \mathcal{E}_i|}{|\mathcal{E}_u| \cdot |\mathcal{E}_i|}\right)\right)^+}{\sum_{j=1}^{K} \exp\left(\sum_{m=1}^{N} \mathrm{Y}_{Rea}\left(p_m\right) + \tanh\left(\frac{|\mathcal{E}_u \cap \mathcal{E}_j|}{|\mathcal{E}_u| \cdot |\mathcal{E}_j|}\right)\right)^-} \right), \tag{14}$$

where $\mathrm{Y}_{Rec}(\cdot)$ and $\mathrm{Y}_{Rea}(\cdot)$ denote the recommendation semantic score function (Eq. (9)) and reasoning path score function (Eq. (5)), $\mathcal{E}$ denotes the set of absorbing nodes, i and j denote positive and negative news samples, and K is the number of negative samples.

3.1.2 Reward Aggregation. Different rewards have distinct impacts on the model's performance. We construct the final reward signal for model training by combining different rewards with the Eq. (15).

$$\mathcal{R}(t) = \left(\sum_{t=0}^{D_t} \gamma \sum_{e_i \in \mathcal{S}_{Go}^t} \left(\mathcal{R}_I^{Col}(t) + \mathcal{R}_I^{Con}(t) \right) \right) + \mathbb{I}_f(t) \left(\beta_1 \mathcal{R}_T^{Rea} + \beta_2 \mathcal{R}_T^{Rec} \right), \tag{15}$$

where γ is the delay factor, D_t represents the number of t-hop absorbed by subgraphs, and $\mathbb{I}_f(t) \in \{0, 1\}$ is a binary indicator that is 1 when subgraph generation is completed and 0 otherwise. α_1 and α_2 are hyperparameters that determine which type of reward is prioritized for supervising user/news subgraph generation. In this paper, we set $\beta_1 = 0.5$, $\beta_2 = 0.5$, and $\gamma = 0.1$.

3.2 Joint Multi-task Optimization

Our model has three main components that are both independent and interdependent. The aim of the end-to-end joint optimization is to create a strong foundation for the upstream task (subgraph generation) that can support the downstream tasks (recommendation and reasoning), while also allowing these downstream tasks to guide the upstream task's generation.

Upstream Task. We use the TD error expectation [39] to train the Critic network, as shown in Eq. (16), while we utilize policy gradient [41] to train the Actor network, as illustrated in Eq. (17).

$$\mathcal{L}_{Critic}(\theta) = \mathbb{E}_{a \sim \pi_\beta}\left[\left(Q_\theta(s_t, a) - q_t \right)^2 \right], \tag{16}$$

$$\mathcal{L}_{Actor}(\beta) = \mathbb{E}_{a \sim \pi_\beta}\left[\left(Q_\theta(s_t, a) - q_t \right) * \pi_\beta(s_t, a) \right], \tag{17}$$

which θ and β are the parameters of Critic and Actor networks. q_t is the target value, which is calculated with the Bellman Eq. (18).

$$q_t = \mathcal{R}(t) + \mathbb{E}_{a \sim \pi_\beta}\left[\gamma \cdot Q_\theta(s_{t+1}, a) \right]. \tag{18}$$

Downstream Tasks. We sample a list of K negative samples: $n_1^-, n_2^-, \ldots, n_K^-$ for each positive candidate news sample: n^+, and construct *cross-entropy* recommendation loss function with Eq. (19).

$$\mathcal{L}_{rec}(\phi) = -\sum_{i \in O^+} \log \left(\frac{\exp\left(\mathrm{score}_\phi(u, n_i)\right)}{\sum_{j=1}^{K} \exp\left(\mathrm{score}_\phi(u, n_j)\right)} \right). \tag{19}$$

To establish reliable reasoning paths between users and interactive news while distinguishing negative news, we employ the *InfoNCE* [31] framework to create a reasoning loss function. This involves randomly sampling negative examples and maximizing the reasoning scores of positive examples, as shown in Eq. (20).

$$\mathcal{L}_{rea}(\psi) = -\sum_{j \in O^-} \log\left(\mathrm{score}_\psi(u, n_j)\right) + \sum_{i \in O^+} log\left(\mathrm{score}_\psi(u, n_i)\right), \tag{20}$$

where ϕ and ψ are the parameters of recommendation and reasoning networks, O^+/ O^- denotes the set of positive/negative news.

Multi-task Optimization Pipelines. We propose a multi-task optimization pipeline that differs from traditional methods like *ADAC* [64], which accumulates the loss of each module. Instead, we focus on inter-task collaboration. In each iteration, we freeze the parameters of subgraph generation (β and θ) and optimize the parameters of recommendation (ϕ) and reasoning (ψ) separately. Then, we freeze ϕ and ψ and optimize β and θ through performance of downstream tasks. The algorithm is shown in Algorithm 1.

Algorithm 1 MR&CO algorithm

Input: Candidate news set (N_c), User set ($\mathcal{U}$), User history behaviors set (N_u), Candidate news titles set ($\mathcal{W}$), News Heterogeneous Network (**NHN**);
Output: Reasoning path set ($\delta_{u \Leftrightarrow n}$), Recommendation ($\mathrm{score}_\phi(u, n)$), Parameters of subgraph generation network (β and θ), Parameters of recommendation network (ϕ), Parameters of reasoning network (ψ);

```
Randomly initialize all parameters;
Set MaxIterations;
for j in 1 : MaxIterations do
    Select batch size B;
    for each pair (u, n) ∈ B do
        Generate user subgraph S_{G_u} and news subgraph S_{G_n};
        Calculate immediate rewards R_I^{Col}, R_I^{Con};
        Calculate reasoning score_ψ and recommendation score_φ;
        Calculate terminal rewards R_T^{Rec}, R_T^{Rea};
        Freeze β and θ, optimize parameters φ and ψ with Eqs. (19) and (20);
        Aggregate terminal and immediate rewards to generate R(t);
        Freeze φ and ψ, optimize parameters β and θ with Eqs. (16) and (17);
    end for
end for
Generate recommendation list and explanation results;
return β, θ, ψ, φ;
```

4 EXPERIMENTS

The experiments are designed to answer the following questions:

- **RQ 1**: How effective is our model in recommendation and reasoning tasks compared to existing outstanding methods?
- **RQ 2**: Dose the MR&CO improve the model's performance?
- **RQ 3**: How effective are the generated subgraphs for both reasoning and recommendation tasks?
- **RQ 4**: Does the model provide reasonable recommendations and explanations for specific recommended news instances?

4.1 Experimental Setting

4.1.1 Dataset Description. The experiment uses Microsoft's open-source news dataset: MIND [57], which contains one million records of news content and user click information from one month of user logs on the MSN News website. We pre-process two versions of the dataset for training and testing our model: **MIND-sample** and **MIND-small**. To ensure the fairness of data, MIND-sample consists of a random selection of 20,000 impressions from MIND-large.

We align news entities with the Wikipedia knowledge graph[2] and capture first-order news neighbor entities to avoid irrelevant entities. Entity embeddings are obtained using TransE [2]. The dataset parameters are shown in Table 1.

Table 1: Detailed statistics of the MIND dataset.

Interaction Data	#Users	#News	# Impression	#Click
MIND-sample	15,427	35,855	20,000	771,350
MIND-small	47,391	54,997	50,000	2,369,550
NHN Data	#Topic	#Subtopic	#News Entity	#External Entity
MIND-sample	17	249	21,342	55,402
MIND-small	17	264	27,759	89,384

4.1.2 Implementation Tricks. During model training, we use three different Adam optimizers [20], one for each of the subgraph generation, recommendation, and reasoning networks. The learning rates for these optimizers are set to (1×10^{-4}), (5×10^{-5}), and (2×10^{-5}), respectively, and we find that (1×10^{-4}) works best in practice.

We use a batch size of 60 and train on 70% of the data, reserving 30% for testing. We evaluate the models using the AUC, MRR, NDCG@5, and NDCG@10 metric. We also introduce a new metric to measure model explainability: the average number of paths, denoted as *Avg.#. Connect Paths* (**AvP**). This metric assesses the ability of the recommended list to explore connection paths.

$$Avg.\#.connect\ paths(AvP) = \frac{\sum_{<u,n>\in C^t} |\Delta_{u \Leftrightarrow n}|}{|C^t|}, \tag{21}$$

where C^t denotes the user-news pair in the test set and $\Delta_{u \Leftrightarrow n}$ denotes the set of multi-hop connection paths between user and recommeded news. The greater the value of this metric, the stronger the evidence the model has to make news recommendations, resulting in a more explainable recommendation model.

4.1.3 Baselines. To prove the superiority of our model in news recommendation and reasoning, we categorize the baseline models into three groups: **Encoder-based News Recommendation** (EB), **Graph-based News Recommendation** (GB), and **Knowledge-based Recommendation** (KB). We select top-performing models from each category. Specifically, knowledge-based models include *RippleNet* [44], *LightGCN* [15], *CKE* [62], *KGCN* [46], *KGAT* [49], *PGPR* [59], *ADAC* [64], and *AnchorKG* [24]. These models incorporate knowledge graph (KG) for recommendation, where *PGPR* and *ADAC* using KG for reasoning recommendations. And *AnchorKG* is the closest to our approach but is focused on recall, which we have adapted for personalized recommendation.

4.2 Overall Performance (RQ1)

Our model[3] has a more outstanding performance compared to the existing models on both the **MIND-sample** and **MIND-small** datasets. Regarding AUC, our model improves by **3.66%** and **5.62%**, respectively. The overall experimental results are shown in Table 2, and the following conclusions are obtained.

- Encoder-based methods, such as *KRED*, *NRMS*, and *NAML*, utilize news content and user behaviors to model interests. However, relying solely on click behaviors can lead to inaccurate user interest representation, while relying solely on content can result in biased news understanding. Our method addresses these issues by incorporating collaborative signals from subgraphs to enhance personalized user/news representations and improve model performance.
- Graph-based methods, such as *GERL*, *GNUD*, and *GNewsRec*, alleviate data sparsity through higher-order information propagation but are highly sensitive to the quality and quantity of interaction data. Our approach uses news content as auxiliary information to overcome the lack of interaction data and generates subgraphs to filter out click noise. Notably, our method outperforms these models with only "10-clicks", compared to their "50-clicks" requirement.
- Knowledge-based methods, such as *KGCN*, *KGAT* and *RippleNet*, and reasoning-based methods, such as *PGPR* and *ADAC*, both incorporate KG to improve recommendation performance and interpretability. However, experimental results prove that these methods are unsuitable for news recommendation scenarios due to massive number of nodes in the NHN and limited interaction data. Our method addresses this by designing NHN with various news semantic information, filtering redundant nodes and selecting valuable ones with NHN-R^2 framework. The *AnchorKG* utilizes subgraphs for "item-to-item" recommendations. However, its integration of MR&CO framework with the concept of CL results in enhanced "item-to-user" recommendations and speeds up convergence.

4.3 MR&CO Validation Experiments (RQ2)

We design MR&CO framework for for end-to-end self-supervised NHN-R^2 framework training. In this section, we aim to investigate the novel framework's improvement of model recommendation accuracy, explainability, time consumption, and complexity.

Accuracy and Explainability. We conduct ablation experiments on MR&CO and the results are shown in Table 3. Compared to RCENR, differences in rewards have minimal impact on recommendation performance. However, immediate rewards (IR-ColR and IR-ConR) are critical for reasoning tasks, with **IR-ConR** being the most crucial, as evidenced by the higher ROC (**-3.607**). Terminal rewards are somewhat less important, with explainability metrics decreasing by **3.155** and **3.380**, but still less than **3.550** and **3.607** of terminal rewards. Our model outperforms reasoning baselines, such as *ADAC* and *PGPR*, and has superior path exploration capabilities. The AvP of our model is better then them, regardless of the type of reward signal removed. These results demonstrate that "2-D" subgraph reasoning expands the search scope and identifies more viable paths, outperforming traditional "point-to-point" reasoning methods. Notably, our model shows significant improvement in both recommendation and reasoning performance compared to the most similar *AnchorKG*, which is attributed to the addition of contrastive learning strategies and MR&CO framework.

Time-consuming and Complexity. To demonstrate the algorithmic efficiency, we compare its total training and testing time with the reasoning recommendation models. We test them on the MIND-sample dataset using an RTX 2080 Ti device over 60 epochs. As shown in Table 4, our model eliminates the steps of *Pre-process* and *Beam-search*, resulting in the lowest model time consumption without sacrificing performance. Furthermore, we compare the computational complexity of the training for three reasoning baseline models, and our model's algorithmic design is found to be superior.

[2]Wikidata is a free and open knowledge base. We download the whole graph from its storage page: https://dumps.wikimedia.org/wikidatawiki/entities/

[3]Our code is available at: https://github.com/JiangHaoPG11/RCENR_code.

Table 2: Overall comparison with baselines of different types.

Type	Model	MIND-sample				MIND-small			
		AUC	MRR	NDCG@5	NDCG@10	AUC	MRR	NDCG@5	NDCG@10
EB	NPA [52]	0.5853	0.2821	0.2987	0.3545	0.6065	0.2859	0.3009	0.3587
	LSTUR [1]	0.6148	0.3007	0.3201	0.3792	0.6313	0.3061	0.3273	0.3850
	DKN [45]	0.6205	0.3103	0.3295	0.3857	0.6392	0.3118	0.3324	0.3900
	GRU [30]	0.6282	0.3125	0.3323	0.3901	0.6589	0.3200	0.3548	0.4113
	KRED [25]	0.6269	0.3048	0.3250	0.3814	0.6407	0.3144	0.3371	0.3928
	KIM [34]	0.6300	0.3203	0.3435	0.3976	0.6426	0.3079	0.3302	0.3877
	NRMS [54]	0.6333	0.3136	0.3324	0.3913	0.6658	0.3287	0.3593	0.4149
	NAML [51]	0.6424	0.3202	0.3467	0.4021	0.6657	0.3276	0.3544	0.4110
GB	GCN [21]	0.5920	0.2956	0.3107	0.3660	0.6147	0.3055	0.3235	0.3777
	GERL [11]	0.6194	0.305	0.3249	0.3799	0.6472	0.3243	0.3472	0.4029
	GNUD [18]	0.6305	0.3136	0.3322	0.3885	0.6568	0.3302	0.3537	0.4100
	User-as-Graph [55]	0.6359	0.3216	0.3443	0.4008	0.6582	0.3350	0.3621	0.4167
	GNewsRec [17]	0.6344	0.3215	0.3464	0.3912	0.6566	0.3248	0.3500	0.4070
KB	LightGCN [15]	0.5182	0.2484	0.2536	0.3114	0.5472	0.2581	0.2616	0.3314
	CKE [62]	0.5377	0.2565	0.2656	0.3231	0.5637	0.2685	0.2861	0.3432
	KGCN [46]	0.5457	0.2574	0.2674	0.3242	0.5715	0.2885	0.2961	0.3522
	KGAT [49]	0.5533	0.2624	0.2730	0.3290	0.5885	0.2991	0.3001	0.3590
	RippleNet [44]	0.6100	0.3004	0.3192	0.3749	0.6301	0.3104	0.3289	0.3949
	PGPR [59]	0.6111	0.3051	0.3245	0.3797	0.6410	0.3151	0.3354	0.3997
	ADAC [64]	0.6214	0.3101	0.3284	0.3804	0.6451	0.3182	0.3396	0.4001
	AnchorKG [24]	0.6393	0.3214	0.3464	0.4008	0.6606	0.3349	0.3623	0.4176
Our model	**RCENR***	**0.6659**	**0.3358**	**0.3635**	**0.4210**	**0.7032**	**0.3641**	**0.3970**	**0.4525**
Growth rate	**IMP**	**3.66%**	**4.87%**	**4.84%**	**4.70%**	**5.62%**	**10.77%**	**10.49%**	**9.06%**

Table 3: Ablation experiments on four type of rewards.

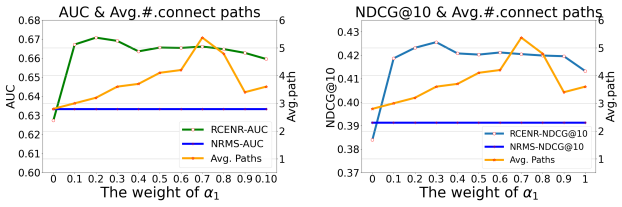
Type	Model	AUC	MRR	NDCG@10	**AvP**	**ROC**
Ours.	**RCENR***	0.6708	0.3390	0.4232	**5.367**	/
Base.	PGPR	0.6110	0.3051	0.3797	**0.863**	**-4.504**
	ADAC	0.6210	0.3101	0.3804	**1.003**	**-4.364**
	AnchorKG	0.6393	0.3214	0.4008	**3.907**	**-1.460**
Abla.	w/o IR-ColR	0.6610	0.3360	0.4186	**1.817**	**-3.550**
	w/o IR-ConR	0.6626	0.3363	0.4199	**1.760**	**-3.607**
	w/o TR-RecR	0.6661	0.3361	0.4193	**2.212**	**-3.155**
	w/o TR-ReaR	0.6598	0.3338	0.4150	**1.937**	**-3.380**

Table 4: Comparison of model time consumption, where L is the connection path length, $|D|$ is the number of candidate nodes in node absorption (10 in this paper), and H is the number of user behaviors. "P-process" refers to Pre-process, "A-train" to Agent training, and "B-search" to Beam search.

Model	P-process	A-train	B-search	**Time**	**Complexity**
PGPR	✗	✓	✓	18h23m	$O(\|D\|^{L\times\|D\|})$
ADAC	✓	✓	✓	15h30m	$O(\|D\|^{L\times\|D\|})$
AnchorKG	✗	✓	✗	12h26m	$O(\|D\|^{L/2\times\|D\|}\times H)$
RCENR*	✗	✓	✗	**7h31m***	$O(\|D\|^{L/2\times\|D\|}\times 2)$ *

4.4 Parameter Sensitivity Experiments (RQ3)

In this section, we evaluate the impact of hyperparameters on both the recommendation and reasoning tasks for validity of subgraphs. These hyperparameters balance the *collaborative signals* and *semantic signals* in the recommendation, and the *connection paths* and *overlapping nodes* in the reasoning. We perform sensitivity experiments on the hyperparameters α_1 and $\alpha_2 = 1 - \alpha_1$ in Eqs. (8) and (4) for recommendation and reasoning, respectively, by varying α_1 from 0 to 1 with a step size of 0.1. Fig. 5 shows the variation curves of model metrics with hyperparameters. Our generated subgraphs assist to significantly improve recommendation accuracy (AUC and NDCG@10) and model explainability metrics (AvP). Specifically, the model performs best for explainability metrics when $\alpha_1 = 0.7$, and peaks in AUC and NDCG when $\alpha_1 = 0.2$. The model consistently outperforms *NRMS* in all cases except when $\alpha_1 = 0$.



(a) AUC-Avg.paths (b) NDCG-Avg.paths

Figure 5: Comparison of parameters sensitivity experiment.

4.5 Case Study (RQ4)

In this section, we test the explainability of our model by presenting two user-recommended news pairs as examples. To illustrate the effectiveness of our model, we carefully selected one straightforward and one complex case for subjective analysis. The results are depicted in Fig. 6, which includes the recommended news and the corresponding explanation paths for each user.

Case 1. For the <*User1950-News34773*> pair, our model filters out the user's clicked news, retaining only those related to the Sports topic, and generates multiple paths connected with the candidate news, *News34773*. In this case, the model produces four explanation paths, which can be broadly categorized into two types. The first is the *news semantic path*, in which the user-clicked news and candidate news both belong to the Topic: Sports, such as: $User1950 \xrightarrow{Clicks}$

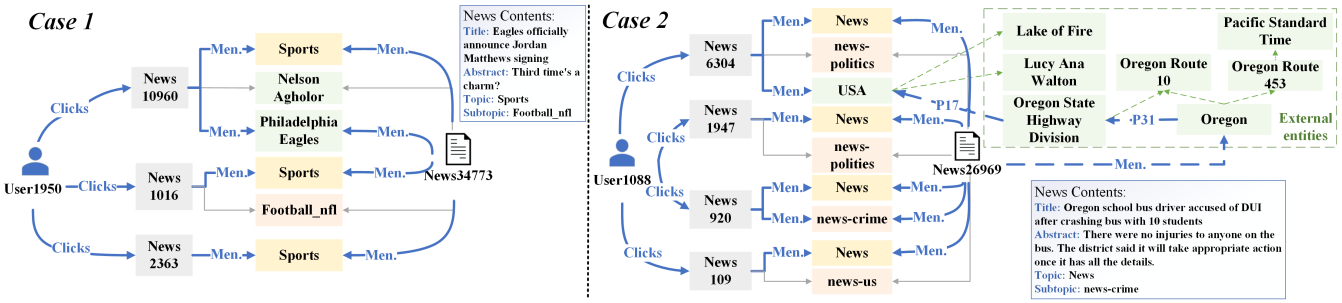


Figure 6: Two examples of recommendation results are shown. In the figure, all nodes are included in the subgraphs, and the blue-highlighted straight lines indicate the presence of multi-hop connection paths which can serve as the explanations.

$News1016 \xrightarrow{Men.} Sports \xleftarrow{Men.} News34773$. The second type is the *news external knowledge path*, in which the user-clicked news and candidate news share the same Entity: Philadelphia Eagles: $User1950 \xrightarrow{Clicks} News10960 \xrightarrow{Men.} Philadelphia\ Eagles \xleftarrow{Men.} News34773$.

Recommended Explanation: Among the multi-hop connection paths, the model recommends *News34773* for *User1950* because the user tends to click on news of Topic: Sports. Moreover, the model provides that the user is interested in the US football team Philadelphia Eagles, rather than the player Nelson Agholor.

Case 2. For complex situations in <*User1088-News26969*>, our model can capture both coarse-grained connection paths (*news semantic path* and *news external knowledge path*) and fine-grained connection paths (*news multi-hop external knowledge path*) that reflect user potential interests. The model filters out irrelevant news and prioritizes news with topics and subtopics of interest, generating coarse-grained explanations, such as: $User1088 \xrightarrow{Clicks} News920 \xrightarrow{Men.} news\text{-}crime \xleftarrow{Men.} News26969$ and $User1088 \xrightarrow{Clicks} News1947 \xrightarrow{Men.} News \xleftarrow{Men.} News26969$. Meanwhile, the model identifies multi-hop connection paths through relationships among external entities, such as: $User1088 \xrightarrow{Clicks} News6304 \xrightarrow{Men.} USA \xleftarrow{P17} Oregon\ State\ Highway\ Division \xleftarrow{P31} Oregon \xleftarrow{Men.} News26969$.

Recommended Explanation: From the diverse connection paths, the model recommends *News26969* for *Users1088* as the user is interested in news about Topic: News and Subtopic: news-crime. Notably, the model finds that the user is more focused on the Oregon State Highway Division in the Oregon State of USA, which provides a higher degree of credibility for the explanation.

In conclusion, our method is able to capture both *explicit user interests* presented in various news content and *potential user interests* embedded in external knowledge facts as explanations, resulting in the generation of diverse and credible explanations.

5 RELATED WORK

Our work is related to the news recommendation and knowledge reasoning recommendation.

5.1 News Recommendation

Existing news recommendation methods can be divided into three main groups. Feature-based methods [3, 7, 12, 13, 36] make recommendations based on interaction information and manually crafted news features. Deep learning-based methods [1, 4, 25, 26, 34, 45, 51–54, 66] utilize complex and sophisticated neural network encoders to represent news and users for recommendations. Graph-based methods [11, 17, 18, 35, 37] take into account the higher-order relationships between users and news to address the sparsity of interaction data. Despite their huge success in improving recommendation accuracy, they have limited explainability and do not provide insights into the reasons for the recommendations.

5.2 Knowledge Reasoning Recommendation

Early knowledge-aware methods [2, 19, 23, 38, 44, 46, 47, 49, 61, 63] use knowledge graph (KG) information to improve data sparsity and model interpretability. Knowledge reasoning is a growing field that seeks to understand multi-hop connections between KG nodes and served as explanations. Existing methods usually incorporate reinforcement learning [9, 24, 28, 33, 42, 50, 59, 60], or adversarial learning [64] into reasoning. The RL-based methods usually train the *Agent* to search on the KG by taking a series of actions from the *source-node* to the *target-node*. For example, the *PGPR* [59] starts with a given user and navigates to potential items, utilizing the historical paths as the explanations. The *AnchorKG* [24] is the few models exploring news reasoning recommendations. Each news article generates a subgraph from KG for "item-to-item" recommendations. However, the complexity and time-consuming of those methods are the reasons that prevent their application to news.

6 CONCLUSIONS

In this paper, we propose an explainable news recommendation method (RCENR) to solve the problem of data sparsity and poor interpretability. Our approach combines a novel news heterogeneous network reasoning recommendation (NHN-R^2) framework and a joint multi-task optimization (MR&CO) framework. The NHN-R^2 uses subgraphs to explore collaborative signals and multi-hop connection paths, while MR&CO utilizes a reinforcement-based strategy and four kinds of manually constructed reward signals for self-supervised model training. Among them, we pioneer the combination of CL with RL, creating IR-ConR to filter redundant nodes and accelerate model convergence. Our sufficient experiments demonstrate that RCENR significantly improves recommendation accuracy and provides diversity and persuasive explanations through novel subgraph reasoning and optimization framework.

ACKNOWLEDGMENTS

This work is supported by the National Key R&D Program of China under grants number 2020YFF0305300 and SQ2020YFF0426386.

REFERENCES

[1] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long-and short-term user representations. In *ACL*. 336–345.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *NeurIPS* 26 (2013).

[3] Michel Capelle, Flavius Frasincar, Marnix Moerland, and Frederik Hogenboom. 2012. Semantics-based news recommendation. In *WIMS*. 1–9.

[4] Davide Castelvecchi. 2016. Can we open the black box of AI? *Nature News* 538, 7623 (2016), 20.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. 1597–1607.

[6] Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications* 141 (2020), 112948.

[7] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *WWW*. 271–280.

[8] Dain C Donelson, John M McInnis, Richard D Mergenthaler, and Yong Yu. 2012. The timeliness of bad earnings news and litigation risk. *The Accounting Review* 87, 6 (2012), 1967–1991.

[9] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. 2019. Explainable recommendation through attentive multi-view learning. In *AAAI*, Vol. 33. 3622–3629.

[10] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP*. 6894–6910.

[11] Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph enhanced representation learning for news recommendation. In *WWW*. 2863–2869.

[12] Anatole Gershman, Travis Wolfe, Eugene Fink, and Jaime G Carbonell. 2011. News personalization using support vector machines. (2011).

[13] Frank Goossen, Wouter IJntema, Flavius Frasincar, Frederik Hogenboom, and Uzay Kaymak. 2011. News personalization using the CF-IDF semantic recommender. In *WIMS*. 1–12.

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*. 9729–9738.

[15] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[17] Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. 2020. Graph neural news recommendation with long-term and short-term interest modeling. *Inf Process Manag* 57, 2 (2020), 102142.

[18] Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020. Graph neural news recommendation with unsupervised preference disentanglement. In *ACL*. 4255–4264.

[19] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *ACL*. 687–696.

[20] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[21] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[22] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[23] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, Vol. 29.

[24] Danyang Liu, Jianxun Lian, Zheng Liu, Xiting Wang, Guangzhong Sun, and Xing Xie. 2021. Reinforced anchor knowledge graph generation for news recommendation reasoning. In *KDD*. 1055–1065.

[25] Danyang Liu, Jianxun Lian, Shiyin Wang, Ying Qiao, Jiun-Hung Chen, Guangzhong Sun, and Xing Xie. 2020. KRED: Knowledge-aware document representation for news recommendations. In *RecSys*. 200–209.

[26] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *IUI*. 31–40.

[27] László Lovász. 1993. Random walks on graphs. *Combinatorics, Paul erdos is eighty* 2, 1-46 (1993), 4.

[28] Ziyu Lyu, Yue Wu, Junjie Lai, Min Yang, Chengming Li, and Wei Zhou. 2022. Knowledge Enhanced Graph Neural Networks for Explainable Recommendation. *IEEE TKDE* (2022).

[29] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *ICML*. 1928–1937.

[30] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *KDD*. 1933–1942.

[31] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[32] Özlem Özgöbek, Jon Atle Gulla, and Riza Cenk Erdur. 2014. A Survey on Challenges and Methods in News Recommendation.. In *WEBIST*. 278–285.

[33] Sung-Jun Park, Dong-Kyu Chae, Hong-Kyun Bae, Sumin Park, and Sang-Wook Kim. 2022. Reinforcement Learning over Sentiment-Augmented Knowledge Graphs towards Accurate and Explainable Recommendation. In *WSDM*. 784–793.

[34] Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021. Personalized news recommendation with knowledge-aware interactive matching. In *SIGIR*. 61–70.

[35] Tao Qi, Fangzhao Wu, Chuhan Wu, Peiru Yang, Yang Yu, Xing Xie, and Yongfeng Huang. 2021. HieRec: Hierarchical User Interest Modeling for Personalized News Recommendation. In *ACL*. 5446–5456.

[36] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW*. 175–186.

[37] TYSS Santosh, Avirup Saha, and Niloy Ganguly. 2020. MVL: Multi-View Learning for News Recommendation. In *SIGIR*. 1873–1876.

[38] 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. *VLDB* 4, 11 (2011), 992–1003.

[39] Richard S Sutton. 1988. Learning to predict by the methods of temporal differences. *Machine learning* 3, 1 (1988), 9–44.

[40] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[41] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *NeurIPS* 12 (1999).

[42] Chang-You Tai, Liang-Ying Huang, Chien-Kun Huang, and Lun-Wei Ku. 2021. User-centric path reasoning towards explainable recommendation. In *SIGIR*. 879–889.

[43] Chenyang Wang, Zhefan Wang, Yankai Liu, Yang Ge, Weizhi Ma, Min Zhang, Yiqun Liu, Junlan Feng, Chao Deng, and Shaoping Ma. 2022. Target Interest Distillation for Multi-Interest Recommendation. In *CIKM*. 2007–2016.

[44] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *CIKM*. 417–426.

[45] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *WWW*. 1835–1844.

[46] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *KDD*. 968–977.

[47] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE TKDE* 29, 12 (2017), 2724–2743.

[48] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *SIGIR*. 1288–1297.

[49] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *KDD*. 950–958.

[50] Xiting Wang, Kunpeng Liu, Dongjie Wang, Le Wu, Yanjie Fu, and Xing Xie. 2022. Multi-level recommendation reasoning over knowledge graphs with reinforcement learning. In *WWW*. 2098–2108.

[51] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with attentive multi-view learning. In *IJCAI*. 3863–3869.

[52] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Npa: Neural news recommendation with personalized attention. In *KDD*. 2576–2584.

[53] Chuhan Wu, Fangzhao Wu, Mingxiao An, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with topic-aware news representation. In *ACL*. 1154–1159.

[54] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with multi-head self-attention. In *EMNLP*. 6389–6394.

[55] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2021. User-as-graph: User modeling with heterogeneous graph pooling for news recommendation. IJCAI.

[56] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Hi-Transformer: Hierarchical Interactive Transformer for Efficient and Effective Long Document Modeling. In *ACL*. 848–853.

[57] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020. Mind: A large-scale dataset for news recommendation. In *ACL*. 3597–3606.

[58] Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *arXiv preprint*

arXiv:2012.15466 (2020).

[59] Yikun Xian, Zuohui Fu, Shan Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *SIGIR*. 285–294.

[60] Yikun Xian, Zuohui Fu, Handong Zhao, Yingqiang Ge, Xu Chen, Qiaoying Huang, Shijie Geng, Zhou Qin, Gerard De Melo, Shan Muthukrishnan, et al. 2020. CAFE: Coarse-to-fine neural symbolic reasoning for explainable recommendation. In *CIKM*. 1645–1654.

[61] Peng Yang, Chengming Ai, Yu Yao, and Bing Li. 2022. EKPN: enhanced knowledge-aware path network for recommendation. *Applied Intelligence* 52, 8 (2022), 9308–9319.

[62] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *KDD*. 353–362.

[63] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Meta-graph based recommendation fusion over heterogeneous information networks. In *KDD*. 635–644.

[64] Kangzhi Zhao, Xiting Wang, Yuren Zhang, Li Zhao, Zheng Liu, Chunxiao Xing, and Xing Xie. 2020. Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs. In *SIGIR*. 239–248.

[65] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *WWW*. 167–176.

[66] Qiannan Zhu, Xiaofei Zhou, Zeliang Song, Jianlong Tan, and Li Guo. 2019. Dan: Deep attention neural network for news recommendation. In *AAAI*, Vol. 33. 5973–5980.

[67] Franziska Zimmer, Katrin Scheibe, Mechtild Stock, and Wolfgang G Stock. 2019. Fake news in social media: Bad algorithms or biased users? *JISTaP* 7, 2 (2019), 40–53.