Faithfully Explainable Recommendation via Neural Logic Reasoning

Yaxin Zhu,*\ Yikun Xian,*\ Zuohui Fu,*\ Gerard de Melo, Yongfeng Zhang
Rutgers University, New Brunswick
{yaxin.a.zhu,yikun.xian,zuohui.fu}@rutgers.edu
{gerard.demelo,yongfeng.zhang}@rutgers.edu

Abstract

Knowledge graphs (KG) have become increasingly important to endow modern recommender systems with the ability to generate traceable reasoning paths to explain the recommendation process. However, prior research rarely considers the faithfulness of the derived explanations to justify the decision-making process. To the best of our knowledge, this is the first work that models and evaluates faithfully explainable recommendation under the framework of KG reasoning. Specifically, we propose neural logic reasoning for explainable recommendation (LOGER) by drawing on interpretable logical rules to guide the path-reasoning process for explanation generation. We experiment on three large-scale datasets in the e-commerce domain, demonstrating the effectiveness of our method in delivering high-quality recommendations as well as ascertaining the faithfulness of the derived explanation.

1 Introduction

Compared with traditional recommender systems (RS), explainable recommendation is not only capable of providing high-quality recommendation results but also offers personalized and intuitive explanations (Zhang and Chen, 2020). Incorporating a knowledge graph (KG) into recommender systems has become increasingly popular, since KG reasoning is able to generate explainable paths connecting users to relevant target item entities. At the same time, there is increasing demand for systems to ascertain the faithfulness of the generated explanation, i.e., assess whether it faithfully reflects the reasoning process of the model and is consistent with the historic user behavior.

However, previous work has largely neglected faithfulness in KG-enhanced explainable recommendation (Xian et al., 2020a; Fu et al., 2020a). A number of studies (Lakkaraju et al., 2019; ter Hoeve et al., 2018; Wu and Mooney, 2018) argue that

faithful explanations should also be personalized and gain the capability to reflect the personalized user historic behavior. However, to the best of our knowledge, none of the existing explainable recommendation models based on KGs have considered faithfulness in the explainable reasoning process and its evaluation on the generated explainable paths. For instance, PGPR (Xian et al., 2019; Zhao et al., 2020) infers explainable paths over the KG without considering personalized user behavior, and its prediction on next potential entities is merely based on the overall knowledge-driven rewards. CAFE (Xian et al., 2020b) builds user module profiles to guide the path inference procedure. However, as illustrated in Subramanian et al. (2020), such neural module networks only implicitly abstract the reasoning process and lack of considering the faithfulness of explanations.

In this paper, we propose a new KG-enhanced recommendation model called LOGER to produce faithfully explainable recommendation via neural logic reasoning. To fully account for heterogeneous information and rules about users and items from the KG, we leverage an interpretable neural logic model for logical reasoning, enhanced by a general graph encoder that learns KG representations to capture semantic aspects of entities and relations. These two components are iteratively trained via the EM algorithm by marrying the merits of interpretability of logical rules and the expressiveness of KG embeddings. Subsequently, the learned rule weights are leveraged to guide the path reasoning to generate faithful explanations. The derived logical rules are expected to be consistent with historic user behavior and the resulting paths genuinely reflect the decision making process in KG reasoning. We experiment on three large-scale datasets for e-commerce recommendation that cover rich user behavior patterns. The results demonstrate the superior recommendation performance achieved by our model compared to the state-of-the-art base-

---
*Equal contribution

lines, with the guarantee of the faithfulness on the generated path-based explanations. The contributions of this paper are threefold.

- We highlight the significance of considering faithfulness in explainable recommendation.
- We propose a novel approach that incorporates interpretable logical rules into KG path reasoning for recommendation and explanation generation.
- We experiment on three large-scale datasets showing promising recommendation performance as well as faithful path-based explanation.

2 Problem Formulation

A knowledge graph (KG) for recommendation is defined as $\mathcal{G} = \{(e_h, r, e_t) \mid e_h, e_t \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ denotes the entity set consisting of sets of users $\mathcal{U}$, items $\mathcal{I}$, and other entities, while $\mathcal{R}$ denotes the relation set. Each triplet (e_h, r, e_t) represents a fact indicating head entity e_h interacts with tail entity e_t via relation r. In recommendation tasks, we are particularly interested in user–item interactions $\{(u, r_{ui}, v) \mid u \in \mathcal{U}, r_{ui} \in \mathcal{R}, v \in \mathcal{I}\}$ with the special relation r_{ui} meaning *purchase* in e-commerce or *like* in movie recommendation.

The problem of KG reasoning for explainable recommendation is formulated as follows. Given an incomplete KG $\mathcal{G}$ with missing user–item interactions, for every user $u \in \mathcal{U}$, the goal is to select a set of items as recommendations $\{v|(u, r_{ui}, v) \notin \mathcal{G}, v \in \mathcal{I}\}$ along with a set of paths as explanations connecting each pair of the user and a predicted item. The key challenge is to not only guarantee the recommendation quality with the rich information in KG, but also generate faithful explanations that reflect the actual decision-making process of the recommendation model and are consistent with historic user behavior.

3 Proposed Method

We introduce the novel neural LOGic Explainable Recommender (LOGER) for producing faithfully explainable recommendations with a KG. As illustrated in Fig. 1, it consists of three components: (i) a KG encoder for learning embeddings of KG entities and relations to capture their semantics, (ii) a neural logic model for conducting interpretable logical reasoning to make recommendations, and (iii) a rule-guided path reasoner for generating faithfully explainable paths. Both KG encoder and neural logic model are trained iteratively via the EM algorithm (Neal and Hinton, 1998) so that they mutually

benefit to make recommendations via logical reasoning. Additionally, personalized rule importance scores are derived for every user and leveraged to guide the path reasoning for faithful explanation generation.

3.1 KG Encoder

Let X_{hrt} be a binary random variable indicating whether a triplet (e_h, r, e_t) is true or not, $X_{\mathcal{G}} = \{X_{hrt} \mid (e_h, r, e_t) \in \mathcal{G}\}$ be a random variable regarding all observed triplets in the KG $\mathcal{G}$, and $X_H = \{X_{hrt} \mid (e_h, r, e_t) \in H\}$ be a random variable of hidden user–item interactions in $H = \{(u, r_{ui}, v) \mid u \in \mathcal{U}, v \in \mathcal{I}, (u, r_{ui}, v) \notin \mathcal{G}\}$. The KG encoder is generally defined as a triplet-wise function $f_\theta : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \mapsto [0, 1]$ parametrized by θ that maps each triplet to a real-valued score. For any triplet $(e_h, r, e_t) \in \mathcal{G} \cup H$, we can interpret its truth probabilistically via the KG encoder f_θ as $q(X_{hrt}|\theta) = \mathsf{Bernoulli}(X_{hrt}|f_\theta(e_h, r, e_t))$. The KG encoder f_θ can be instantiated with any existing KG embedding (Ji et al., 2020) or graph neural network (Wu et al., 2020) model.

3.2 Neural Logic Model

We focus on composition rules for user–item interactions, i.e., r_{ui} is a composition of relations $r_1, \ldots, r_j$ if $(u, r_1, e_1) \wedge \cdots \wedge (e_{j-1}, r_j, v) \Rightarrow (u, r_{ui}, v)$, $\forall u \in \mathcal{U}, v \in \mathcal{V}, e_1, \ldots, e_{j-1} \in \mathcal{E}$. Given a set of logical rules L mined from the KG, the goal of this component is, for every user $u \in \mathcal{U}$, to emit a set of personalized rule importance scores $y_u = \{y_{u,l}\}_{l \in L}$ to capture the historic user behavior. To achieve this, we build upon Markov Logic Networks (Qu and Tang, 2019), an interpretable probabilistic logic reasoning method that models the joint distribution of all triplets via a set of logical rules L, i.e., $p(X_{\mathcal{G}}, X_H|w) = \frac{1}{Z} \exp\left(\sum_{l \in L} w_l n_l\right)$, where $w = \{w_l\}_{l \in L}$ with w_l being the global weight of rule $l \in L$, and n_l denotes the number of true groundings of rule l over observed and hidden triplets. Accordingly, we define the personalized rule importance score to be $y_{u,l} = \frac{w_l n_l(u)}{\sum_{l' \in L} n_{l'}(u)}$, where $n_l(u)$ is the number of groundings of rule l over the observed triplets in $\{(u, r_{ui}, v) \in \mathcal{G}\}$. However, it is intractable to directly maximize the log likelihood of observed triplets to learn the global weights w, i.e., $\max_w \log p(X_{\mathcal{G}}|w)$. Instead, we employ the EM algorithm to iteratively optimize the objective to acquire optimal global weights.
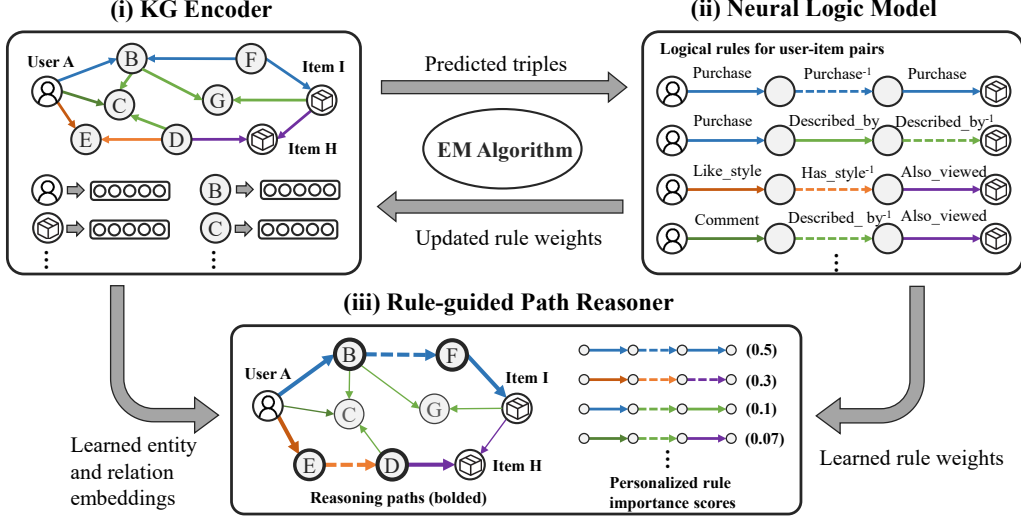
Figure 1: Illustration of the proposed method for explainable recommendation including (i) a KG encoder, (ii) a neural logic model, and (iii) a rule-guided path reasoner.

E-Step We introduce a mean-field variational distribution $q(X_H|\theta) \approx \prod_{(e_h,r,e_t) \in H} q(X_{hrt}|\theta)$ over hidden user–item interactions in H. The goal of the E-step is to estimate $q(X_H|\theta)$ by minimizing the KL divergence between $q(X_H|\theta)$ and the posterior distribution $p(X_H|X_{\mathcal{G}}, w)$ with fixed w. For each triplet $(e_h, r, e_t) \in H$, we denote by L_{hrt} the set of rules associated with the triplet and by G_{hrt} the corresponding groundings of all logical rules in L_{hrt}. Following Qu and Tang (2019), the optimal $q(X_H|\theta)$ can be achieved under the fixed-point condition, i.e., $q(X_{hrt}|\theta) \approx p(X_{hrt}|X_{G_{hrt}}, w)$, for all $(e_h, r, e_t) \in H$. Here, $q(X_{hrt}|\theta)$ is approximated by the KG encoder f_θ, and $p(X_{hrt}|X_{G_{hrt}}, w)$ can be estimated with the global weights w of the rules in L_{hrt} from the last iteration:

$$p(X_{hrt} = 1|X_{G_{hrt}}, w) = \sigma\left(\frac{\sum_{l \in L_{hrt}} w_l}{|L_{hrt}|}\right), \quad (1)$$

where $\sigma(\cdot)$ is the sigmoid function. In other words, if a hidden triplet (e_h, r, e_t) is asserted to be true by the rules (e.g., $p(X_{hrt} = 1 \mid X_{G_{hrt}}, w) > 0.5$), the probability $q(X_{hrt} = 1 \mid \theta)$ given by the KG encoder is also expected to be high. Therefore, to learn the parameter θ, we aim to maximize the log-likelihood function over all observed triplets in $\mathcal{G}$ and the plausibly true hidden triplets in $H^+ = \{(e_h, r, e_t) \mid p(X_{hrt} = 1|X_{G_{hrt}}, w) \geq \tau\}$, which leads to the objective

$$\ell(\theta) = \sum_{(e_h,r,e_t) \in \mathcal{G} \cup H^+} \log q(X_{hrt} = 1 \mid \theta), \quad (2)$$

where τ is a hyperparameter.

M-Step The goal of the M-step is to learn the global rule weights w by maximizing the log-likelihood function $E_{q(X_H)}[\log p(X_{\mathcal{G}}, X_H; w)]$ given a fixed θ from the E-step. Since the log-likelihood term models the joint distribution over all triplets, which is hard to compute for a large KG, we approximate it with the pseudolikelihood (Besag, 1975): $\ell_{PL}(w) = \sum_{(e_h,r,e_t) \in \mathcal{G} \cup H} E_{q(X_H|\theta)}[\log p(X_{hrt}|X_{G_{hrt}}, w)]$. Then, we can invoke gradient ascent to acquire the optimal w, with the gradient defined as:

$$\nabla_{w_l} \ell_{PL}(w_l) = \sum_{(e_h,r,e_t) \in \mathcal{G}} \frac{1 - p_{hrt}}{|L_{hrt}|} + \sum_{(e_h,r,e_t) \in H} \frac{q(X_{hrt} = 1|\theta) - p_{hrt}}{|L_{hrt}|}, \quad (3)$$

where $p_{hrt} = p(X_{hrt} = 1|X_{G_{hrt}}, w)$. Once the optimal global weights are acquired, we can make a recommendation by calculating the ranking score of a user $u \in \mathcal{U}$ and an item $v \in \mathcal{I}$ as $q(X_{urv}|\theta) + \alpha p(X_{urv} = 1|X_{G_{urv}}, w)$, where $r = r_{ui}$ and $\alpha \in \mathbb{R}$ is a hyperparameter.

3.3 Rule-Guided Path Reasoner

We draw on the KG encoder f_θ and the personalized rule importance scores y_u from the last two steps to generate explainable paths for every user u. Specifically, we train an LSTM-based path reasoning network ϕ that takes the start user embedding as input and predicts a sequence of entities and relations to form a path. For every user u, we restrict the reasoner to generate the paths that follow the rules with the largest scores in y_u. The details of ϕ

3085

and path reasoning are described in the Appendix.

4 Experiment

Dataset We experiment on three domain-specific e-commerce datasets from Amazon, namely *Cellphones*, *Grocery*, and *Automotive*. There are two requirements that lead to the selection of these categories in our experiments. First, the constructed KG should contain rich user behavior patterns, e.g., user mentioned features or preferred styles, etc. This is the major difference from most of the existing work (Zhao et al., 2019), which only extends knowledge on the item side. Second, the KGs are assumed to be large-scale. We select several large subsets from Fu et al. (2020b), where the constructed KG can be regarded as an updated version of those of Ai et al. (2019) based on the Amazon review dataset (Ni et al., 2019). The remaining three datasets are the ones that satisfy both of the aforementioned requirements. Statistical details of datasets are provided in the Appendix.

Baselines & Metrics We consider several state-of-art baselines in the following experiments. **CKE** (Zhang et al., 2016) uses semantic representations derived from TransR (Lin et al., 2015) to enhance the matrix factorization process. **RippleNet** (Wang et al., 2018) is a hybrid method combining regularization and path formats, and augmenting user representations with a memory-network-like approach. **PGPR** (Xian et al., 2019) designed a policy-guided graph search algorithm for recommendation over KGs. **HeteroEmbed** (Ai et al., 2018) aims to learn the embeddings of a heterogeneous graph including users, items, and relations for recommendation. **KGAT** (Wang et al., 2019) explicitly models higher-order KG connectivity and learns node representations by propagating the embedding of neighbors with corresponding importance discriminated by an attention mechanism. We adopted the same metrics as Ai et al. (2018) to evaluate the recommendation performance of all models: **Precision**, **Recall**, Normalized Discounted Cumulative Gain (**NDCG**), and Hit Rate (**HR**).

4.1 Recommendation Results

We first evaluate the recommendation quality of our model. The results of all methods across all three datasets are reported in Table 1. In general, our method significantly outperforms all state-of-the-art baselines on all metrics. Taking *Cellphones*

as an example, our method achieves an improvement of 6.01% in NDCG against the best baseline (underlined), and an improvement of 5.82% in Hits@10. Similar trends can be observed on other benchmarks as well. Note that both our model and HeteroEmbed adopt TransE for KG representation learning, yet our model achieves better performance, mainly attributed to the iterative learning of graph encoder and neural logic model.

4.2 Faithfulness of Explanation

We aim to measure whether the generated explainable paths are consistent with the historic user behavior via a faithfulness metric and a user study.

Measuring Faithfulness Inspired by previous work (Maaten and Hinton, 2008; Serrano and Smith, 2019; Subramanian et al., 2020), we define the faithfulness to be the Jensen–Shannon (JS) divergence of rule-related distributions from training and test sets. Specifically, we randomly sample 50 users from the training set. For each user u, we further sample around 1,000 paths between the user and the connected item nodes, and calculate the rule distribution over these paths, denoted by $F(u)$. We compare the proposed LOGER with two baselines, PGPR, and KGAT, each of which is used to generate 20 explainable paths for every selected user in the test phase. Similarly, we can calculate the rule distribution over these 20 paths, denoted by $Q_f(u)$. The JS scores are defined as follows.

$$\mathrm{JS}_f = \mathbb{E}_{u \sim \mathcal{U}}[D_{\mathrm{JS}}(Q_f(u) \parallel F(u))]$$
$$\mathrm{JS}_w = \mathbb{E}_{u \sim \mathcal{U}}[D_{\mathrm{JS}}(Q_w(u) \parallel F(u))]$$

Here, $Q_w(u)$ is the rule weight distribution derived from the personalized rule importance scores of our method or the path weights of baselines. Smaller values of two JS scores correspond to better faithfulness of the explainable paths. This faithfulness evaluation is motivated in terms of the consistency of the explainable paths with respect to the user historic behavior.

User Study Additionally, we conduct a user study to evaluate the faithfulness of the explainable paths. We display 50 sampled KG paths starting from one user towards purchased items in the training set to represent examples of user historical behaviors. For comparison, we also present 10 explainable paths generated by three methods for the same user in the test dataset. We ask 20 human subjects to rank these methods based on whether

	Cellphones				Grocery				Automotive			
	Precision	Recall	NDCG	HR	Precision	Recall	NDCG	HR	Precision	Recall	NDCG	HR
CKE	0.0360	0.1760	0.1847	0.3067	0.0612	0.2528	0.3070	0.4511	0.0458	0.1871	0.2257	0.3621
RippleNet	0.0419	0.2141	0.2177	0.3715	0.0591	0.2682	0.2858	0.4800	-	-	-	-
PGPR	0.0462	0.2148	0.2366	0.3801	0.0649	0.2710	0.3174	0.4926	0.0589	0.2315	0.2804	0.4409
KGAT	0.0476	0.2274	0.2365	0.3835	0.0702	0.2916	0.3381	0.5020	0.0601	0.2500	0.2859	0.4514
HeteroEmbed	<u>0.0527</u>	<u>0.2543</u>	<u>0.2626</u>	<u>0.4226</u>	<u>0.0785</u>	<u>0.3316</u>	<u>0.3701</u>	<u>0.5572</u>	<u>0.0695</u>	<u>0.2923</u>	<u>0.3314</u>	<u>0.5082</u>
LOGER	**0.0622**	**0.2977**	**0.3227**	**0.4808**	**0.0906**	**0.3754**	**0.4370**	**0.6121**	**0.0743**	**0.3091**	**0.3653**	**0.5346**

Table 1: Recommendation quality of all methods on three datasets. The results are computed based on the top-10 recommendation on the test set. The best results are highlighted in bold and the second best results are underlined.

	Cellphones			Grocery		
	JS_f	JS_w	Avg. Rank	JS_f	JS_w	Avg. Rank
PGPR	0.56	0.49	2.52	0.42	0.38	2.27
KGAT	0.53	0.45	2.14	0.39	0.41	2.08
LOGER	**0.47**	**0.32**	**1.52**	**0.34**	**0.28**	**1.75**

Table 2: Results of measuring the faithfulness of the generated paths obtained by three methods. Bold numbers indicate the best results.

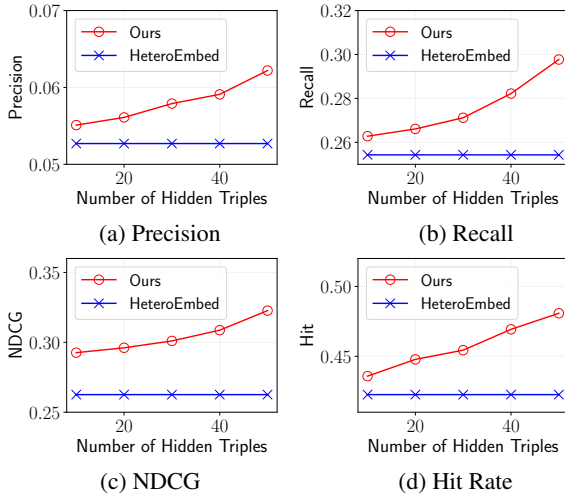(a) Precision

(b) Recall

(c) NDCG

(d) Hit Rate

Figure 2: Recommendation quality under varying sizes of estimated hidden triples.

the generated paths are consistent with those from the training set. Then, we calculate the average ranking scores (**Avg. Rank**) by averaging the rank given by each human tester on each method.

Results The results on the *Cellphones* and *Grocery* datasets are reported in Table 2. We observe that our method LOGER achieves the lowest JS scores and average ranking score, which reveal the effectiveness of our model in producing more faithful explanations in both quantitative measurements and in the user study.

4.3 Ablation Study

We further study how hidden triplets used in training KG encoder (Eq. 2) influence the recommendation performance. We experiment on

the *Cellphones* data under different sizes of hidden triplet sets H^+. We choose the sizes of $\{10, 20, 30, 40, 50\}$ and keep all other settings unchanged. The results are plotted in Fig. 2, including our model (red circles) and the best baseline HeteroEmbed (blue crosses). We find that our model consistently outperforms the baseline in all the metrics under different numbers of hidden triplets. Better recommendation performance can be achieved with more hidden triplets included in training the KG encoder, because more candidate items will enhance the capability of our model to discern the logical rules of good quality and hence benefit the recommendation prediction.

5 Conclusion

In this paper, we propose LOGER for faithfully explainable recommendation, which generates explainable paths based on personalized rule importance scores via neural logic reasoning that adequately captures historic user behavior. We experiment on three large-scale datasets for e-commerce recommendation showing superior recommendation quality of LOGER as well as the faithfulness of the generated explanations both quantitatively and qualitatively. We hope to encourage future work that values explainability and in particular the faithfulness of explanations. Our code is available at https://github.com/orcax/LOGER.

References

Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*.

Qingyao Ai, Yongfeng Zhang, Keping Bi, and W Bruce Croft. 2019. Explainable product search with a dynamic relation embedding model. *TOIS*.

Julian Besag. 1975. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 24(3):179–195.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Zuohui Fu, Yikun Xian, Ruoyuan Gao, Jieyu Zhao, Qiaoying Huang, Yingqiang Ge, Shuyuan Xu, Shijie Geng, Chirag Shah, Yongfeng Zhang, et al. 2020a. Fairness-aware explainable recommendation over knowledge graphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 69–78.

Zuohui Fu, Yikun Xian, Yaxin Zhu, Yongfeng Zhang, and Gerard de Melo. 2020b. Cookie: A dataset for conversational recommendation over knowledge graphs in e-commerce. *arXiv preprint arXiv:2008.09237*.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. 2019. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 131–138.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187. AAAI Press.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854.

Radford M Neal and Geoffrey E Hinton. 1998. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197.

Meng Qu and Jian Tang. 2019. Probabilistic logic neural networks for reasoning. In *Advances in Neural Information Processing Systems*, pages 7712–7722.

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? *arXiv preprint arXiv:1906.03731*.

Sanjay Subramanian, Ben Bogin, Nitish Gupta, Tomer Wolfson, Sameer Singh, Jonathan Berant, and Matt Gardner. 2020. Obtaining faithful interpretations from compositional neural networks. *arXiv preprint arXiv:2005.00724*.

Maartje ter Hoeve, Anne Schuth, Daan Odijk, and Maarten de Rijke. 2018. Faithfully explaining rankings in a news recommender system. *arXiv preprint arXiv:1805.05447*.

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*, pages 417–426. ACM.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *KDD '19*.

Jialin Wu and Raymond J Mooney. 2018. Faithful multimodal explanation for visual question answering. *arXiv preprint arXiv:1809.02805*.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Yikun Xian, Zuohui Fu, Qiaoying Huang, Shan Muthukrishnan, and Yongfeng Zhang. 2020a. Neural-symbolic reasoning over knowledge graph for multi-stage explainable recommendation. *AAAI DLGMA Workshop*.

Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *SIGIR*.

Yikun Xian, Zuohui Fu, Handong Zhao, Yingqiang Ge, Xu Chen, Qiaoying Huang, Shijie Geng, Zhou Qin, Gerard de Melo, S Muthukrishnan, et al. 2020b. Cafe: Coarse-to-fine neural symbolic reasoning for

explainable recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1645–1654.

Wei Zhang, Quan Yuan, Jiawei Han, and Jianyong Wang. 2016. Collaborative multi-level embedding learning from reviews for rating prediction. In *IJ-CAI*.

Yongfeng Zhang and Xu Chen. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends in Information Retrieval*.

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*.

Wayne Xin Zhao, Gaole He, Hong-Jian Dou, J. Huang, Siqi Ouyang, and Ji-Rong Wen. 2019. Kb4rec: A data set for linking knowledge bases with recommender systems. *Data Intelligence*, 1:121–136.

A Detail of Rule-Guided Path Reasoning

Our LSTM-based path reasoner ϕ is based on the graph walker in Moon et al. (2019). It takes as input the embedding of the current entity e_{t-1} and outputs the embeddings of the next relation r_t and the next entity e_t, i.e., $r_t, e_t = \phi(\mathbf{e}_{t-1})$. In particular, the next relation embedding $\mathbf{r}_t$ is defined as:

$$\alpha_t = \sigma(W_\alpha \mathbf{e}_{t-1} + b_\alpha),$$
$$\mathbf{r}_t = \sum_{r \in \mathcal{R}} \alpha_{t,r} r,$$

where W_α, b_α are parameters and α_t are the attention weights over all relations in the KG. The next entity embedding $\mathbf{e}_t$ is defined as:

$$\mathbf{z}_t = \mathbf{e}_{t-1} + \mathbf{r}_t$$
$$\mathbf{i}_t = \sigma(W_i[\mathbf{e}_{t-1}; \mathbf{c}_{t-1}] + b_i)$$
$$\mathbf{c}_t = (1 - \mathbf{i}_t) \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(W_c[\mathbf{z}_t; \mathbf{e}_{t-1}] + b_c)$$
$$\mathbf{o}_t = \sigma(W_o[\mathbf{z}_t, \mathbf{e}_{t-1}, \mathbf{c}_t] + b_o)$$
$$\mathbf{e}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t)$$

Here, $[;]$ denotes concatenation, $\odot$ is elementwise multiplication, i_t, o_t are vectors passing through corresponding gates, and z_t is the context vector.

During training, for every user and its observed user–item triplets, we sample a set of training paths following the rules, with numbers proportional to the rule weights. The goal is to make the path reasoner ϕ generate paths that are close to the training samples, which can be optimized by the hinge loss.

The inference pipeline using the trained path-reasoning network is described in Alg. 1. Starting with a user u encoded as $\mathbf{e}_0 = \mathbf{u}$, the estimated entity embedding $\mathbf{e}_t$ and relation embedding $\mathbf{r}_t$ at the t-th hop is obtained by the model ϕ. At each hop, for all potential neighbors, we calculate a ranking score based on the dot-product of the neighbor and estimated $(\mathbf{e}_t, \mathbf{r}_t)$. After ranking these neighbors based on such scores, we can filter a set of candidate neighbors and invoke a Beam Search to identify a set of paths as well as corresponding items for u.

B Implementation Details

In order to guarantee path connectivity, we add reverse relations into the knowledge graph, i.e., if $(e_h, r, e_t) \in \mathcal{G}$, then $(e_t, r^{-1}, e_h) \in \mathcal{G}$. We restrict the length of candidate rules to be 3. We adopt TransE (Bordes et al., 2013) as the KG encoder

---

Algorithm 1 Rule-guided path reasoning

1: **Input:** KG $\mathcal{G}$, user u, item v, rule set L
2: **Output:** a set of paths P
3: **procedure** MAIN()
4:    $P \leftarrow \{\{u\}\}$.
5:    **for** $t \leftarrow 1$ to T **do** $\triangleright$ T is path length.
6:       $P_{\text{curr}} \leftarrow \{\}$.
7:       **for** path $p \in P$ **do**
8:          $e_{t-1} \leftarrow$ last node of p.
9:          $V_{\text{curr}} \leftarrow \{\}$.
10:         **for** $(e_{t-1}, r', e') \in \mathcal{G}$ **do**
11:            $\hat{\mathbf{e}}_t, \hat{\mathbf{r}}_t = \phi(\mathbf{e}_{t-1})$
12:            $s = \langle \hat{\mathbf{e}}_t, \mathbf{e}' \rangle + \langle \hat{\mathbf{r}}_t, \mathbf{r}' \rangle$.
13:            $V_{\text{curr}} \leftarrow V_{\text{curr}} \cup \{(r', e', s)\}$.
14:         $P_{\text{curr}} \leftarrow P_{\text{curr}} \cup \{p \cup \{r', e'\} | \text{rank}(s) \leq \beta, (r', e', s) \in V_{\text{curr}}\}$.
15:         $P \leftarrow P_{\text{curr}}$.
16:      $P \leftarrow \{p | p \in P, \text{rule}(p) \in L, \texttt{lastnode}(p) = v\}$.
17:    **return** P.

---

f_θ, with the dimensionality of entity and relation embeddings set as 100.

To learn the global rule weights, we first generate the hidden triplet set according to the result of the KG encoder. For each user, the top 50 estimated items with the highest scores predicted by KG encoder are taken as the hidden triplet set H^+. The threshold τ is set to 0.5 and the weighting factor α is set to 0.3 by default. In the path reasoning algorithm, we set the neighboring size β to 10. Other training details can be found in Table 3.

Parameter	Cellphones	Grocery	Automotive
# of epochs	4	2	3
KGE batch size	512	512	512
KGE optimizer	Adam	Adam	Adam
KGE learning rate	1e-4	1e-4	1e-4
NLM learning rate	1e-5	1e-5	1e-5
# of sample node	100	100	100

Table 3: Training detail for three datasets. KGE = KG encoder. NLM = neural logic model.

C Dataset Statistics

The statistics of our datasets are shown in Table 4.

Dataset	Cellphones	Grocery	Automotive
#Users	61,254	57,822	95,445
#Items	47,604	40,694	78,557
#Interactions	607,673	709,280	1,122,776
#Entities	169,331	173,369	270,543
#Relations	45	45	73
#Triples	3,117,051	3,742,954	4,580,318

Table 4: Overall statistics of three datasets.