

# Neural-Symbolic Reasoning over Knowledge Graph for Multi-stage Explainable Recommendation

Anonymous Author(s)

## Abstract

Recent work on recommender systems has considered external knowledge graphs as valuable sources of information, not only to produce better recommendations but also to provide explanations of why the recommended items were chosen. Pure rule-based symbolic methods provide a transparent reasoning process over knowledge graph but lack generalization ability to unseen examples, while deep learning models enhance powerful feature representation ability but are hard to interpret. Moreover, direct reasoning over large-scale knowledge graph can be costly due to the huge search space of pathfinding. We approach the problem through a novel coarse-to-fine neural symbolic reasoning method called NSER. It first generates a coarse-grained explanation to capture abstract user behavioral pattern, followed by a fine-grained explanation accompanying with explicit reasoning paths and recommendations inferred from knowledge graph. We extensively experiment on four real-world datasets and observe substantial gains of recommendation performance compared with state-of-the-art methods as well as more diversified explanations in different granularity.

## 1 Introduction

Explainable recommendation has attracted increasing attention from both academic and industry communities (Zhang and Chen 2018), which highlights the importance of system transparency and targets to supply users with more satisfactory and informed recommendation results. In this regard, knowledge graphs (KG) have recently come to prominence for building explainable recommender system, as the graph structure empowers the ability to trace reasoning paths behind recommendations. Early works (Catherine et al. 2017; Catherine and Cohen 2016) propose to model user behavior by a set of symbolic rules from knowledge graph. Despite their transparency, these methods heavily depend on the handcrafted rules, which makes them difficult to generalize to unseen behavior correlations. With the advances of deep learning, several neural-based methods have been proposed to incorporate knowledge graphs into recommender systems, in either *pre-defined* (Ma et al. 2019; Wang et al. 2019) or *post-hoc* (Ai et al. 2018) manner. However, they fail to explicitly model the path routing process,

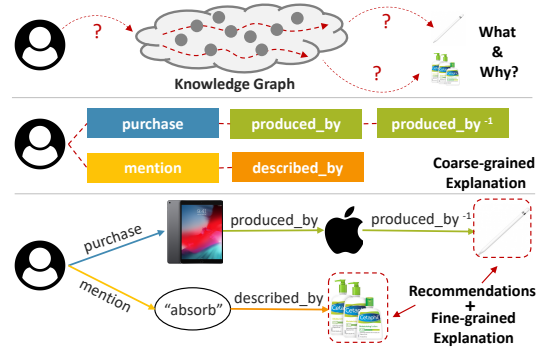


Figure 1: A coarse-to-fine process of neural-symbolic reasoning over knowledge graph for explainable recommendation. The coarse-grained explanation provides abstract and representative user patterns, which is then invoked to produce fine-grained reasoning paths for recommendations.

which impedes the transparency of the recommendations. Recent work (Xian et al. 2019) study *in-progress* explainable recommendation by adopting reinforcement learning for path inference in knowledge graph, but the recommendation performance is sacrificed by the explanation generation process and the paths are not sufficiently diversified.

Other previous work has exploited symbolic rules using neural networks to encode prior knowledge to advance the intent representation and performance with good interpretability. (Karpathy, Johnson, and Fei-Fei 2015) empirically expose interactions within neural network structures. (Santa Cruz et al. 2018) develop neural network modules to compose boolean algebraic operations on visual classifiers, with an ability to synthesize them for complex visual concept expressions. However, despite success in visual reasoning tasks, no existing work seeks to exploit these neural symbolic methods in explainable recommendations.

To leverage the best of both worlds, we propose a coarse-to-fine neural symbolic reasoning method called NSER to infuse explicit symbolic reasoning into the neural modeling process for explainable recommendation, which induces an intermediate reasoning template before concrete path reasoning over knowledge graph for recommendation and explanation. As illustrated in Figure 1, NSER first learns to generate a coarse-grained explanation that depicts abstract but representative user behavioral pattern. Such explana-

tion is characterized by composition of neural relation modules, which facilitates personalization of both explanation and recommendation by refactoring these modules. Then, NSER produces a fine-grained explanation containing concrete reasoning paths based on the user pattern, and the end items in the paths are naturally acquired for recommendation. The use of such two-stage framework not only yields high-quality recommendation results, but also provides a form of hierarchical explanations for better interpretability.

The contributions of this paper are threefold. First, we emphasize the importance of explainable recommendation with knowledge graph, especially in-progress explanations that fully match the actual decision making for recommender systems, as opposed to predefined or post-hoc explanations. Second, we marry the merits of deep learning with the interpretability of symbolic graph reasoning and propose a two-stage coarse-to-fine reasoning model for the problem. Third, we experiment on four real-word benchmarks showing that our model yields both good recommendation results and versatile hierarchical explanations. To the best of our knowledge, this is the first paper that studies the generation of coarse-to-fine explanations for recommendations via multi-stage neural symbolic reasoning process.

## 2 Problem Formulation

A **recommendation knowledge graph**  $\mathcal{G}_R$  is defined as a set of triples,  $\mathcal{G}_R = \{(e_h, r, e_t) \mid e_h, e_t \in \mathcal{E}, r \in \mathcal{R}\}$  with entity set  $\mathcal{E}$  and relation set  $\mathcal{R}$ . Each triple  $(e_h, r, e_t)$  represents a fact indicating head entity  $e_h$  interacts with tail entity  $e_t$  through relation  $r$ . The entity set  $\mathcal{E}$  contains  $N_{\mathcal{E}}$  disjoint subsets of entities, i.e.,  $\mathcal{E} = \mathcal{U} \cup \mathcal{I} \cup \mathcal{E}_3 \cup \dots \cup \mathcal{E}_{N_{\mathcal{E}}}$ , where  $\mathcal{U}$  and  $\mathcal{I}$  respectively stand for “users” and “items” in the context of recommendation. We are particularly interested in the relation of user–item interaction, denoted by  $r_{ui} \in \mathcal{R}$ , which in practice represents “purchase” in e-commerce recommendation. Formally, the problem of explainable recommendation over knowledge graph is defined as follows.

**Definition 1 (Problem Definition)** *Given an incomplete knowledge graph  $\mathcal{G}_R$ , the goal is to recover all missing facts  $\{(u, r_{ui}, i) \mid u \in \mathcal{U}, i \in \mathcal{I}\}$  such that each pair  $(u, i)$  is derived from a reasoning path from  $u$  to  $i$ , which serves as an explanation towards the recommendation  $i$  for user  $u$ .*

The challenges of the problem are threefold. First, the explainable path  $L_{ui}^\pi$  is expected to be generated in an **in-progress** manner and is supposed to explicitly explain how the model makes decisions step by step over the knowledge graph towards a potentially good item for recommendation. Second, the search space to generate explainable paths can be very large in large-scale knowledge graphs. It is necessary to narrow down the scope through a coarse-to-fine process that first generate abstract patterns to sketch user behavior prior to reasoning about concrete paths for recommendation. Third, the method should support personalization in both recommendation and explanation.

## 3 Methodology

We propose a method called Neural-Symbolic Explainable Recommendation (NSER) that takes advantage of both in-

terpretable symbolic reasoning as well as of powerful neural representation learning. As shown in Figure 2, our model makes in-progress explainable recommendations using a two-stage coarse-to-fine reasoning process over KG. It first learns to generate coarse-grained patterns by composing various reasoning components of relation symbols, and then reasons about fine-grained paths consisting of entity symbols for recommendation. Both pattern and path are treated as explanations, where the former represents abstract user behaviors while the latter describes concrete decisions made by the system. In the following, we first introduce neural representation learning for symbols in KG, followed by the two-stage process for explainable recommendation.

### Neural-Symbolic Representation Learning

In the knowledge graph, symbols such as entities and relations can directly be used for reasoning via predefined rules, but such rule-based method lack any generalization ability to unseen rules. Therefore, we neuralize these symbols via deep representation learning and make these neural symbols serve as building blocks in the subsequent reasoning stage.

To be specific, each entity  $e \in \mathcal{E}$  is regarded as a  $d$ -dimensional vector, denoted by  $\mathbf{e} \in \mathbb{R}^d$ , known as an **entity embedding**. In the task of path reasoning for recommendation, we assume that the next entity  $e'$  leading to a good target item is determined by relation  $r$  given the start user  $u$  and the preceding entity  $e$ . Hence, we treat each relation  $r \in \mathcal{R}$  as a reasoning component or a function, denoted by  $\phi_r : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}^d$ , which maps vectorized user and the preceding entity  $(\mathbf{u}, \mathbf{e})$  to the successor entity embedding  $\mathbf{e}'$ . In this work, we approximate  $\phi_r$  by a neural network parameterized by  $\Theta_r$ , written as  $\mathbf{e}' = \phi_r(\mathbf{u}, \mathbf{e}; \Theta_r)$ , and call it the **neural relation module** of relation  $r$ . We treat the linear composition of such modules as an instantiation of the corresponding metapath. A tree-structured composition of modules constructs coarse-grained abstract pattern, which will later be invoked to generate fine-grained reasoning paths for explainable recommendation.

Let  $\Theta_{\mathcal{G}_R} = \{\mathbf{e} \mid \forall e \in \mathcal{E}\} \cup \{\Theta_r \mid \forall r \in \mathcal{R}\}$ , the set of all learnable parameters for the neural symbolic representation. The goal of neural-symbolic representation learning is to optimize over  $\Theta_{\mathcal{G}_R}$  such that neural symbols can be composed in a certain way leading to good reasoning ability for decision making. In particular, let  $\mathcal{I}_u^+$  be the set of positive items of user  $u$ , and  $\mathcal{L}_u^\pi = \{L_{ui}^\pi \mid i \in \mathcal{I}_u^+\}$ , the set of user–item paths from  $u$  to any positive items via metapath  $\pi$ . Now, given a user  $u$  and metapath  $\pi$ , we aim to minimize the negative log likelihood of all positive paths being generated, with the following objective function  $\ell_{\text{path}}(u, \pi)$ :

$$\ell_{\text{path}}(u, \pi) = \mathbb{E}_{L_{ui}^\pi \sim \mathcal{L}_u^\pi} \left[ - \sum_{j=1}^{|\pi|} \log P(e_j \mid u, e_{j-1}, r_j) \right] \quad (1)$$

With the help of the neuralized entity embeddings and neural relation modules, the probability term in Eq. 1 can be approximated as  $P(e_j \mid u, e_{j-1}, r_j) \approx \frac{\exp(s(\mathbf{e}_j))}{\sum_{e \in \mathcal{E}_j} \exp(s(\mathbf{e}))}$ , where

$s(\cdot) = \langle \cdot, \phi_{r_j}(\mathbf{u}, \hat{\mathbf{e}}_{j-1}; \Theta_{r_j}) \rangle$  and  $\mathcal{E}_j \subseteq \mathcal{E}$  denotes the subset of all entities that share the same type as  $e_j$ .

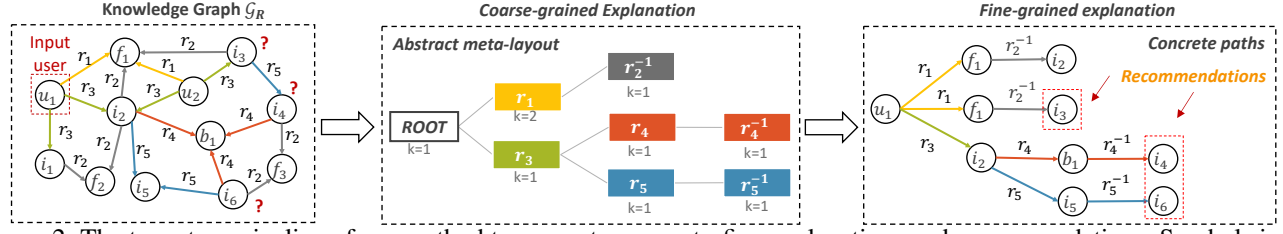


Figure 2: The two-stage pipeline of our method to generate coarse-to-fine explanations and recommendations. Symbols in the knowledge graph are represented as entity embeddings and neural relation modules. A coarse-grained explanation (abstract meta-layout) is first generated to describe representative user patterns for recommendation. A following fine-grained explanation is produced based on the pattern to provide explicit reasoning paths whose end nodes serve as recommendations.

#### Algorithm 1 Abstract Meta-Layout Generation

```

1: Input: metapaths  $\{\pi_1, \dots, \pi_M\}$  with  $\{y_1, \dots, y_M\}$ 
2: Output: abstract meta-layout  $T$ 
3: Initialize  $T$  by merging  $\{\pi_1, \dots, \pi_M\}$ 
4:  $\forall x \in \text{Leaves}(T), \pi_j: \text{if } x \in \pi_j \text{ then } k_x \leftarrow y_j$ 
5:  $\text{RECURSIVEUPDATE}(T.\text{root})$ 
6: return  $T$ 
7: procedure  $\text{RECURSIVEUPDATE}(\text{node } x)$ 
8:    $\forall c \in \text{Children}(x), \text{RECURSIVEUPDATE}(c)$ 
9:   if  $x$  is root then  $k_x \leftarrow 1$  return
10:   $C \leftarrow \{k_c \mid c \in \text{Children}(x), k_c > 0\}$ 
11:  if  $C = \emptyset$  then  $k_x \leftarrow 0$ 
12:  else  $k_x \leftarrow \min(C), \forall c \in \text{Children}(x) : k_c \leftarrow \lfloor k_c / k_x \rfloor$ 

```

Optimizing over the loss function in Eq. 1 empowers the neural relation modules to fit positive paths, but it fails to distinguish the relative importance among them. Thus, we impose an additional ranking loss on the final output entity. Suppose there is an off-the-shelf but less accurate teacher model  $h$  that estimates the similarity between a user  $u$  and an item  $i$ , i.e.,  $h : (u, i) \mapsto [0, 1]$ . We define the negative item set  $\mathcal{I}_{ui}^-$  with respect to an item  $i$  and  $\mathcal{I}_u^+$  to be  $\mathcal{I}_{ui}^- = \{i^- \mid h(u, i) > h(u, i^-), i^- \in \mathcal{I}_u^+\}$ . The ranking loss  $\ell_{rk}(u, \pi)$  with respect to user  $u$  and metapath  $\pi$  is defined as:

$$\ell_{rk}(u, \pi) = \mathbb{E}_{L_{u,i} \sim \mathcal{L}_{u,i}^+} \left[ \mathbb{E}_{i^- \sim \mathcal{I}_{ui}^-} [\sigma(\langle \hat{e}_{|\pi|}, i^- \rangle - \langle \hat{e}_{|\pi|}, i \rangle)] \right], \quad (2)$$

where  $\sigma(\cdot)$  is sigmoid function, and  $\hat{e}_{|\pi|}$  is the output entity embedding from the last neural relation module  $\phi_{r_{|\pi|}}$ .

By aggregating Eq. 1 and 2 across all users and metapaths in the knowledge graph  $\mathcal{G}_R$ , the overall objective becomes:

$$\ell_{\text{all}}(\Theta_{\mathcal{G}_R}) = \sum_{u \in \mathcal{U}, \pi \in \mathcal{G}_R} \ell_{\text{path}}(u, \pi) + \lambda \ell_{rk}(u, \pi), \quad (3)$$

where  $\lambda$  is the weighting factor for the ranking loss.

#### Neural-Symbolic Explainable Recommendation

Next, we introduce the coarse-to-grain process of making explainable recommendations via neural-symbolic reasoning over the knowledge graph. We start by giving the formal definition of the coarse-grained explanation called **abstract meta-layout** that provides abstract user behavior patterns. Formally, an abstract meta-layout  $T$  is defined as a rooted tree where each node  $x$  is associated with a relation  $r_x \in \mathcal{R}$

and a non-negative integer  $k_x$  indicating sampling size of entities within node  $x$ . As illustrated in Figure 2 (middle panel), the abstract meta-layout can be regarded as a specification of a neural network architecture. We can assemble a tree-structured neural network by instantiating each node  $x \in T$  with the corresponding neural relation module  $\phi_{r_x}$ . Each module takes as input the  $d$ -dimensional vector generated by its parent module and emits a new vector to be passed to the child module.

**Generating Coarse-grained Explanation** We propose a heuristic approach to generate abstract meta-layout as follows. Considering that the relation sequence along any root-to-leaf path in the layout forms a metapath, we can construct the layout by simply determining the number of paths to be sampled for each metapath and then combining the set of metapaths that will be used to sample paths. Formally, let  $y_j$  be the number of paths sampled for metapath  $\pi_j$  and  $M$  be the total number of metapaths in  $\mathcal{G}_R$ . For any user  $u$ , we solve the following optimization problem to obtain the optimal  $\{y_1, \dots, y_M\}$ :  $\max_{\{y_j\}} \sum_j y_j \cdot v(u, \pi_j)$  subject to  $\sum_j y_j \leq K, 0 \leq y_j \leq k_j, 1 \leq j \leq M$ , where  $K$  is the maximum number of paths to be sampled in total,  $k_j$  is the maximum number of paths sampled for  $\pi_j$ . Here, we also introduce a heuristic function  $v(u, \pi_j) = \mathbb{E}_{L \in \mathcal{L}_{u,i}^+} [\log P(L \mid u, \pi_j)]$ , which reflects the estimated importance of metapath  $\pi_j$  leading to good items for recommendation to user  $u$ . After obtaining  $\{y_1, \dots, y_M\}$ , we can generate an abstract meta-layout  $T$  for user  $u$  using Algorithm 1. The basic idea is to first construct the layout tree by merging all metapaths in  $\{\pi_j \mid y_j > 0\}$ , which determines the tree structure and the relation in each node.

**Generating Fine-grained Explanation** Given user  $u$  and the abstract meta-layout  $T$ , the final step is to generate concrete paths in the knowledge graph for recommendation that also serve as detailed explanations. The path generation process is illustrated in Fig. 2 (right panel). Specifically, as previously discussed, we first assemble a tree-structured neural network based on  $T$ . By feeding in the user embedding  $\mathbf{u}$ , each neural relation module  $\phi_{r_x}$  at tree node  $x$  can be “executed” one-by-one in breadth-first order to output an intermediate  $d$ -dimensional vector, denoted by  $\hat{e}_{r_x}$ . Then, let  $\tau_x$  be the set of paths generated at node  $x$  and  $\tau_{\text{root}} = \{\{u\}\}$ . For any other node  $x$ , we generate the paths as  $\tau_x = \{L \cup \{r_x, e\} \mid L \in \tau_{\text{parent}(x)}, \forall e \in \mathcal{E} \text{ and } (\text{lastnode}(L), r_x, e) \in \mathcal{G}_R, \text{rank}(\langle \hat{e}_{r_x}, e \rangle) \leq k_x\}$ .

Dataset	CDs & Vinyl				Clothing				Cell Phones				Beauty			
Measures (%)	NDCG	Recall	HR	Prec.	NDCG	Recall	HR	Prec.	NDCG	Recall	HR	Prec.	NDCG	Recall	HR	Prec.
DeepCoNN	4.218	6.001	13.857	1.681	1.310	2.332	3.286	0.229	3.636	6.353	9.913	0.999	3.359	5.429	9.807	1.200
CKE	4.620	6.483	14.541	1.779	1.502	2.509	4.275	0.388	3.995	7.005	10.809	1.070	3.717	5.938	11.043	1.371
HeteroEmbed	5.563	<u>7.949</u>	<u>17.556</u>	<u>2.192</u>	<u>3.091</u>	<u>5.466</u>	<u>7.972</u>	<u>0.763</u>	<u>5.370</u>	<u>9.498</u>	<u>13.455</u>	<u>1.325</u>	<u>6.399</u>	<u>10.411</u>	<u>17.498</u>	<u>1.986</u>
PGPR	<u>5.590</u>	7.569	16.886	2.157	2.858	4.834	7.020	0.728	5.042	8.416	11.904	1.274	5.449	8.324	14.401	1.707
NSER (Ours)	<b>6.868</b>	<b>9.376</b>	<b>19.692</b>	<b>2.562</b>	<b>3.689</b>	<b>6.340</b>	<b>9.275</b>	<b>0.975</b>	<b>6.313</b>	<b>11.086</b>	<b>15.531</b>	<b>1.692</b>	<b>7.061</b>	<b>10.948</b>	<b>18.099</b>	<b>2.270</b>

Table 1: Main experimental results evaluating recommendation ranking quality of our method compared to other approaches on four Amazon datasets. The results are computed based on a top-10 recommendation in the test set and are given as percentages (%). The best results are highlighted in bold font and the best baseline results are underlined.

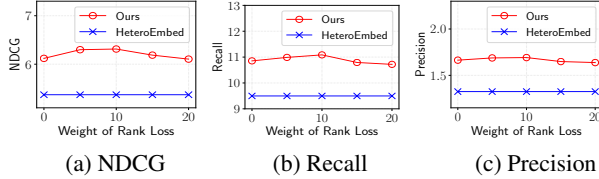


Figure 3: Ranking loss results on Cell Phones dataset.

The final reasoning paths are collected from all leaf nodes.

## 4 Experiments

We experiment on four domains of Amazon e-commerce datasets (He and McAuley 2016): *CDs and Vinyl*, *Clothing*, *Cell Phones* and *Beauty*. Each dataset is considered as an individual benchmark that constitutes a knowledge graph with respectively  $388.5M$ ,  $36.5M$ ,  $37.2M$  and  $37.3M$  triples. We adopt the same training and test splits as (Xian et al. 2019). We consider as baselines the latest *in-progress* method PGPR (Xian et al. 2019) along with selective baselines included in their study. We also include HeteroEmbed (Ai et al. 2018), a strong *post-hoc* method for explainable recommendation over KG, which is used as teacher model in Eq. 2. Four metrics are adopted to evaluate all models: Normalized Discounted Cumulative Gain (**NDCG**), **Recall**, Hit Rate (**HR**), and Precision (**Prec.**).

**Recommendation Performance** We first show the recommendation performance of our proposed NSER method compared to the baselines. The results on four datasets are shown in Table 1. We observe that our method outperforms the best baseline HeteroEmbed by a large margin across all settings. For example, on the Clothing dataset, our model achieves 6.340% in Recall, which is higher than 5.466% of HeteroEmbed and 4.834% of PGPR. Similar trends can also be observed for other metrics. It is interesting to see our model also outperforms the teacher model HeteroEmbed substantially. One reason is that the similarity estimation by the teacher network largely reduces the search space for the correct target items, so our model can further discover patterns for each user to make better recommendations.

**Influence of Ranking Loss** In this experiment, we show how the weighting factor  $\lambda$  in Eq. 3 affects the performance of neural-symbolic representation learning. We experiment on the Cell Phones dataset and choose different values of  $\lambda \in \{0, 5, 10, 15, 20\}$ .  $\lambda = 0$  means no ranking loss is imposed for training. All other settings remain the same as in the previous experiment. The results are plotted in Fig. 3, including our model (red curves) and the best baseline Het-

Dataset	Cell Phones				Beauty			
Method	NDCG	Recall	HR	Prec.	NDCG	Recall	HR	Prec.
uniform	4.545	7.229	10.192	1.087	6.293	9.256	15.564	1.918
prior	6.255	10.842	15.097	1.659	6.880	10.393	17.258	2.224
heuristic	6.313	11.086	15.531	1.692	7.061	10.948	18.099	2.270

Table 2: Influences of three different abstract meta-layouts.

eroEmbed (blue curves). We observe two interesting trends. First, our model consistently outperforms HeteroEmbed under all settings of  $\lambda$  in terms of all metrics. Even without the ranking loss, our model can still guarantee a high quality of recommendation. Second, a larger weight of the ranking loss may not always entail a better performance. Instead, there is a trade-off between the ranking (Eq. 2) and path regularization (Eq. 1). The reason is that if the ranking loss plays a dominant role, the model will pay less attention to the path fitting function, and consequently, it may not be able to find the correct paths to reach a promising item.

**Effectiveness of Abstract Meta-Layout** Now, we evaluate the effectiveness of our abstract meta-layout for recommendation. We specifically consider three layouts that are all generated by Algorithm 1 but based on different  $\{y_1, \dots, y_M\}$ . The first one (*uniform*) uses  $y_j = \sum_i^K \mathbb{1}\{X_i = j\}$  where  $X_i \in [M]$  is a random variable from a uniform distribution. The second one (*prior*) uses  $y_j$  based on a prior  $k_j$ , i.e.,  $y_j = \lceil K \cdot k_j / \sum_i k_i \rceil$ . The third one (*heuristic*) is our own method, with  $y_j$  obtained based on a heuristic function. The results on Cell Phones and Beauty datasets are reported in Table 2. We observe that our generated heuristic program exhibits better recommendation performance than the other two. This is because the heuristic value  $v(u, \pi_j)$  that measures the quality of metapath  $\pi_j$  is directly estimated from neural symbols.

## 5 Conclusions and Future Work

In this paper, we study the problem of explainable recommendation by explicitly reasoning over knowledge graph. We propose a neural-symbolic approach (NSER) that absorbs both explainability of symbolic reasoning and generalization ability of deep learning. The model generates two-stage coarse-to-fine explanations, where coarse-grained explanation describes representative user pattern and is used to generate fine-grained explanation for final explanation and recommendation. We extensively evaluate our model on several real-world benchmarks and show that the approach delivers strong recommendation results.

## References

- [Ai et al. 2018] Ai, Q.; Azizi, V.; Chen, X.; and Zhang, Y. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*.
- [Catherine and Cohen 2016] Catherine, R., and Cohen, W. 2016. Personalized recommendations using knowledge graphs: A probabilistic logic programming approach. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 325–332. ACM.
- [Catherine et al. 2017] Catherine, R.; Mazaitis, K.; Eskenazi, M.; and Cohen, W. 2017. Explainable entity-based recommendations with knowledge graphs. *arXiv preprint arXiv:1707.05254*.
- [He and McAuley 2016] He, R., and McAuley, J. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on World Wide Web*, 507–517.
- [Karpathy, Johnson, and Fei-Fei 2015] Karpathy, A.; Johnson, J.; and Fei-Fei, L. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.
- [Ma et al. 2019] Ma, W.; Zhang, M.; Cao, Y.; Jin, W.; Wang, C.; Liu, Y.; Ma, S.; and Ren, X. 2019. Jointly learning explainable rules for recommendation with knowledge graph. In *The World Wide Web Conference*, 1210–1221. ACM.
- [Santa Cruz et al. 2018] Santa Cruz, R.; Fernando, B.; Cherian, A.; and Gould, S. 2018. Neural algebra of classifiers. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 729–737. IEEE.
- [Wang et al. 2019] Wang, X.; Wang, D.; Xu, C.; He, X.; Cao, Y.; and Chua, T.-S. 2019. Explainable reasoning over knowledge graphs for recommendation. *AAAI*.
- [Xian et al. 2019] Xian, Y.; Fu, Z.; Muthukrishnan, S.; de Melo, G.; and Zhang, Y. 2019. Deep reinforcement learning over knowledge graph inference for e-commerce recommendation. In *Proceedings of SIGIR 2019*.
- [Zhang and Chen 2018] Zhang, Y., and Chen, X. 2018. Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192*.