A Reusable Model-agnostic Framework for Faithfully Explainable Recommendation and System Scrutability

ZHICHAO XU, University of Utah, United States
HANSI ZENG, University of Massachusetts Amherst, United States
JUNTAO TAN, ZUOHUI FU, and YONGFENG ZHANG, Rutgers University, United States
QINGYAO AI, Tsinghua University, Zhongguancun Laboratory, China

State-of-the-art industrial-level recommender system applications mostly adopt complicated model structures such as deep neural networks. While this helps with the model performance, the lack of system explainability caused by these nearly blackbox models also raises concerns and potentially weakens the users' trust in the system. Existing work on explainable recommendation mostly focuses on designing interpretable model structures to generate model-intrinsic explanations. However, most of them have complex structures, and it is difficult to directly apply these designs onto existing recommendation applications due to the effectiveness and efficiency concerns. However, while there have been some studies on explaining recommendation models without knowing their internal structures (i.e., model-agnostic explanations), these methods have been criticized for not reflecting the actual reasoning process of the recommendation model or, in other words, *faithfulness*. How to develop model-agnostic explanation methods and evaluate them in terms of faithfulness is mostly unknown. In this work, we propose a reusable evaluation pipeline for model-agnostic explainable recommendation. Our pipeline evaluates the quality of model-agnostic explanation from the perspectives of faithfulness and scrutability. We further propose a model-agnostic explanation framework for recommendation and verify it with the proposed evaluation pipeline. Extensive experiments on public datasets demonstrate that our model-agnostic framework is able to generate explanations that are faithful to the recommendation model. We additionally provide quantitative and qualitative study to show that our explanation framework could enhance the scrutability of blackbox recommendation model. With proper modification, our evaluation pipeline and model-agnostic explanation framework could be easily migrated to existing applications. Through this work, we hope to encourage the community to focus more on faithfulness evaluation of explainable recommender systems.

CCS Concepts: • **Information systems** → **Recommender systems;**

Additional Key Words and Phrases: Explainable recommendation, faithfulness, scrutability

Zhichao Xu is supported in part by NSF IIS-2007398 and in part by NSF IIS-2205418 and NSF DMS-2134223. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

Authors' addresses: Z. Xu, University of Utah, 50 Central Campus Drive, Salt Lake City, UT, 84112; email: zhichao.xu@utah.edu; H. Zeng, University of Massachusetts Amherst, 140 Governors Drive., Amherst, MA 01003; email: hzeng@cs.umass.edu; J. Tan, Z. Fu, and Y. Zhang, Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ 08854; emails: {juntao.tan, zuohui.fu, yongfeng.zhang}@rutgers.edu; Q. Ai, Department of Computer Science and Technology, Tsinghua University, Zhongguancun Laboratory, West Tsinghua Rd, Haidian District, Beijing, China, 100190; email: aiqy@tsinghua.edu.cn.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
© 2023 Copyright held by the owner/author(s).
1046-8188/2023/08-ART29 $15.00
https://doi.org/10.1145/3605357

ACM Reference format:
Zhichao Xu, Hansi Zeng, Juntao Tan, Zuohui Fu, Yongfeng Zhang, and Qingyao Ai. 2023. A Reusable Model-agnostic Framework for Faithfully Explainable Recommendation and System Scrutability. *ACM Trans. Inf. Syst.* 42, 1, Article 29 (August 2023), 29 pages.
https://doi.org/10.1145/3605357

1 INTRODUCTION

With the emerging development of the Internet over the past few decades, researchers have devoted extensive efforts to the study of **recommender systems (RS)** [1, 11, 44, 69]. As complex **deep neural networks (DNN)** are being introduced and deployed in real-world applications, the blackbox nature of such models has raised several concerns in the RS research community. On the one hand, users could be reluctant to trust an uninterpretable RS model that cannot provide explanations to its recommendation decisions. On the other hand, system designers cannot control the behavior and outputs of an uninterpretable RS model directly, which makes it non-trivial to fix the system even when the exact failure cases have been detected. To overcome these problems and challenges, the idea of designing explainable[1] and scrutable [7, 8] recommendation models has received considerable attention in recent years.

Existing works on explainable recommendation can be broadly classified into two categories: *model-intrinsic* explanations and *model-agnostic* explanations [18, 66, 94]. Model-intrinsic explanations are generated by interpretable recommendation models, while model-agnostic explanations are generated from a separate explanation model by treating the recommendation model as blackbox. Both explanation paradigms have been extensively studied by the RS community, and they have their own strengths and weaknesses. Model-intrinsic explanations can reflect the actual decision mechanism of the RS model, but deploying such applications could be non-trivial in practice. For example, due to historical reasons, deploying model-intrinsic explainable RS means an overhaul to the existing system, which is time- and effort consuming. Also, from the perspective of model design, enhancing model explainability often means the sacrifice of model performance/latency [63], and taking the risk of adversarial attack [17]. Thus, despite recent advances on model-intrinsic explainable models [79, 81, 84], most industrial RS still use uninterpretable (blackbox) model designs [48, 56]. Model-agnostic explanation models, however, can bypass the concerns above in practice by running a separate explanation model that does not disturb the production model. However, whether these explanations are faithful, i.e., reflecting the actual reasoning process of the recommendation model, is often questionable. Existing studies on model-agnostic explainable recommendation methods either ignore the discussion of explanation faithfulness [19, 23, 28, 42, 58, 72, 77, 78] or build explanations that are dependent on the specific choice of blackbox recommendation models [8, 45, 67] and thus cannot be directly applied or compared to other genres of models. To the best of our knowledge, there has not been a reusable framework to build model-agnostic explanations for different RS models and evaluate the quality of these explanations.

Seeing this gap, in this work, we focus on designing a reusable framework to build model-agnostic explanations for different RS models and evaluate the quality of these explanations. We first propose an evaluation pipeline to evaluate model-agnostic algorithms from two perspectives: (1) *faithfulness (to what extent can the explanation model reflect the true reasoning process of the recommendation model)* and (2) *scrutability (whether the explanation model allows the users to correct the system such that the whole explanation system can be further improved).* The model-agnostic

[1] we use the terms interpretable and explainable interchangeably.

explanation framework to be evaluated is composed of two separate parts: a recommendation model to make recommendation decisions, which we refer to as the *blackbox* model,[2] and a separate explanation model to explain the recommendation model's decisions, which we refer to as the whitebox model. By implementing the recommendation model with a model-intrinsic explainable model, the faithfulness and the scrutability of the explanation model can be quantified based on whether the explanation can match the actual explanation provided by the recommendation model.

Following the evaluation pipeline, we propose a model-agnostic explanation framework to operationalize the desiderata of faithful explanation and system scrutability. Inspired by the technique of knowledge distillation, we propose to build a model-agnostic explanation model by taking the outputs of a blackbox recommendation model to train a whitebox explainable model and generate explanations accordingly. In this work, we adopt two types of RS models, i.e., the aspect-based models and knowledge-graph-based models, for both the blackbox recommendation model and the whitebox explanation model to show how we can apply the evaluation pipeline and explanation framework to different types of models. In addition to the aforementioned model categories, we provide concrete guidelines to adapt the proposed explanation framework to other genres of models. As for scrutability, we provide quantitative and qualitative studies to show that our explanation framework can be potentially used to manipulate the blackbox recommendation's behaviors to enhance the scrutability of blackbox recommendation model. To the best of our knowledge, our work is among the first efforts to propose a model-independent and reusable evaluation and learning framework for model-agnostic explainable recommendation. Our contributions can be summarized as follows:

- We propose a reliable evaluation pipeline for model-agnostic explainable recommendation with a focus on explanation faithfulness and system scrutability. Our pipeline can be used on different genres of recommendation models with proper modification.
- We propose a model-agnostic explanation framework for recommendation with a knowledge-distillation-style training strategy to improve explanation faithfulness and system scrutability.
- We provide qualitative and quantitative studies to show that our explanation framework could enhance the faithfulness of model-agnostic explanations and the scrutability of blackbox recommendation model.

The rest of this article is organized as follows: We first review the related work (Section 2), and then we give a high-level overview of the proposed model-agnostic explanation framework and the proposed evaluation pipeline (Section 3) and introduce the idea of knowledge-distillation-style training strategy in Section 4. We use two types of explainable recommendation models to detail the designs and implementations of the model-agnostic explanation framework, the evaluation pipeline, and the training strategy in Section 5. Further, we discuss the limitations of this work and provide guidelines to adapt our work to other genres of explainable recommendation models in Section 6. We cover the experimental details and report and analyze the results (Section 7). In addition, we also present the qualitative and quantitative scrutability study (Section 7.3). Finally, we summarize the article and point out possible future research directions (Section 8). Our code implementation is made public.[3]

[2]Because model-agnostic explanation assumes no knowledge on the structure of the recommendation model, here we simply refer to the recommendation model as the blackbox model. Note that this does not mean that the recommendation model has to be implemented with models that are uninterpretable and inaccessible.

[3]https://github.com/zhichaoxu-shufe/Faithful-and-Scrutable-Recommendation-Framework.

2 RELATED WORK

There are four lines of works that are related to this article: (1) model-intrinsic explanation vs. model-agnostic explanation in explainable machine learning and recommendation, (2) faithfulness and scrutability in model-agnostic explainable recommendation, (3) existing model-agnostic explainable RS and evaluation, and (4) knowledge distillation.

2.1 Model-intrinsic Explanation vs. Model-agnostic Explanation

With the development of more complicated **machine learning (ML)** models, such as the "black-box" deep neural networks, there has been an urge for explainable and controllable/scrutable ML models. By whether the explanations are generated by the decision model itself, the existing explanation paradigms can be broadly classified into two categories: *model-intrinsic* (or pre hoc) explanations and *model-agnostic* (or post hoc) explanations [50, 94].

Model-intrinsic approaches aim to design inherently interpretable models, so that the generated explanations can reflect the actual inference mechanism of the decision model [12, 30, 75, 79, 81, 84, 85, 95]. Most of recently proposed explainable recommendation models fall into this category. For example, Zhang et al. [95] propose to combine extracted Aspect-Opinion-Sentiment triplets with vanilla matrix factorization for recommendation explanation. He et al. [30] propose to model user-item-aspect interactions via a tripartite graph. A different line of research propose to leverage the explainable nature of **knowledge graph (KG)** for recommendation. Wang et al. [79] incorporates knowledge graph into recommendation to improve click-through rate. **Knowledge-aware Path Reasoning Network (KPRN)** [81] treats the knowledge graph path as a sequence and train a LSTM neural network for explicit reasoning over KG. Xian et al. [84] use a **reinforcement learning (RL)** algorithm featuring soft reward strategy, user-conditional action pruning, and a multi-hop scoring strategy to generate knowledge graph reasoning paths. Inspired by Path Language Modeling [47], Geng et al. [22] propose PLM-Rec framework to learn knowledge graph representations and generate reasoning paths over the knowledge graph for recommendation in an autoregressive manner. For the experiments in this work, we adopt two types of RS models, i.e., the aspect-based models and knowledge-graph-based models for both the blackbox recommendation model and the whitebox explanation model. We will further provide guidelines on modifying the proposed evaluation pipeline and explanation framework on other genres of explainable recommendation models.

Although various model-intrinsic explainable recommmendation models have been proposed, most industrial RS applications still use uninterpretable (blackbox) model designs [48, 56]. As discussed in previous works [58, 63] explainability and model performance/efficiency could sometimes be conflicting goals in model design that we have to tradeoff. Due to the fact that the deep neural recommendation models being deployed take heterogeneous information sources as input, and are growing more complicated in terms of model structures, it is hard to ask the model to provide human-interpretable explanations without hurting the system performance as well as efficiency. However, the model-agnostic approach aims to design a separate explanation model to explain the decision model's output [2, 3, 23, 38, 58, 72, 77, 86, 89, 96]. With model-agnostic explanation, the decision model's performance and efficiency can be preserved. But it is also criticized for not being able to fully reflect the actual reasoning process of the decision model [5, 9, 36, 66], thus not exhibiting *faithfulness*. Seeing this gap, in this work, we make attempts to improve the faithfulness of the model-agnostic explanation paradigm.

2.2 Faithfulness and Scrutability in Model-agnostic Explainable Recommendation

In explainable ML, *faithfulness measures how accurately the generated explanations reflect the actual reasoning process of the decision model* [25, 29, 31, 35, 36]. A similar concept is *transparency, which*

is whether the explanation explains how the system works, proposed by Tintarev and Masthoff [76] and further refined in References [7, 8]. It is also referred to as *trustworthiness* [13] and related to *completeness* [25].[4]

While the problem of how to improve the faithfulness of model-agnostic explanations has been widely recognized in the explainable ML literature [5, 9, 36], only a few works tackle this problem in explainable recommendation. This discrepancy exists, because RS data and methods are very different from traditional supervised ML settings [54]. The model-agnostic explainable RS model takes user–item pairs as input, and thus the definition and evaluation of faithfulness needs to take both the user and item-side information into account. For example, ter Hoeve et al. [74] take a perturbation approach to generate faithful and model-agnostic explanations for a blackbox news recommendation system. However, their experiments are conducted under the classical learning-to-rank setting, where only a few explicit categorical features are utilized for the recommendation decision, and no representations learning is involved during the model's training phase. Such an intervention approach is not applicable for state-of-the-art DNN models such as References [48, 56]. Zhu et al. [97] propose a faithfully explainable recommendation model with knowledge graph, but their model generates model-intrinsic explanations, and the faithfulness is defined as the similarity between the generated knowledge graph reasoning paths and the paths in the trainset.

Related but different from faithfulness, in the explainable ML and recommendation literature [75, 76], *scrutable/scrutability*[5] is used to describe *whether the system allows its users to correct the system's reasoning or modify the users' preferences such that the system can be improved in the long run.* Scrutability is also among the desiderata for industrial-level RS applications [8, 24], as a scrutable system could enable the system designers to control the its behavior abd debug its failure cases such that the system could be further improved. It is well recognized that scrutability is related to faithfulness, because one cannot learn how to control a system based on explanations that are unfaithful in describing system behaviors. Yet, having faithful explanations does not necessarily improve system scrutability, e.g., we can explain an attention network by checking the attention scores, but we may not know how the changing of these attention scores would affect the final predictions [36, 83].

2.3 Existing Model-agnostic Explainable RS and Evaluation

Tintarev and Masthoff [76] propose in total seven design goals for explainable recommendation, including *effectiveness*, *efficiency*, *persuasiveness*, *satisfication*, *scrutability*, *transparency*, and *trust*. Balog and Radlinski [7] and Nunes and Jannach [55] point out that most of the existing works only focus on one single goal, and only a few consider multiple goals [16, 21]. Balog and Radlinski [7] conducted user study to measure the correlation between the aforementioned goals and the results, not surprisingly, show that these goals are interrelated. However, we notice that some goals are more important than others in certain application scenarios/domains, e.g., *rersuasiveness* in generating post hoc explanations in recipe/music recommendation [46, 87]. More importantly, while various models have been proposed for explainable recommendation, there has not been a universal and model-independent metric to systematically evaluate the quality of generated explanations in terms of faithfulness and scrutability. Peake and Wang [58] propose to mine association rules to explain the recommendations given by matrix factorization, but their evaluation is limited to the explanations' fidelity, i.e., whether the association rules recommend similar items to the matrix factorization model. Some works [3, 23, 96] use user/case study to prove the generated

[4]The terminologies are not yet standardized in literature and we choose the most well-accepted *Faithfulness* throughout the article.

[5]also referred to as "decipherable" or "understandable" [7, 8, 27].

explanations are superior than those from the baseline models in terms of persuasiveness. For example, Ai and Narayanan [3] conduct crowdsourcing to compare model-intrinsic and model-agnostic explanations from *informativeness*, *usefulness*, and *satisfaction*, where usefulness has the same definition as persuasiveness. Similarly, Ghazimatin [23] shows that the proposed explanation framework outperforms other explanation types in terms of usefulness. A recent line of works [23, 38, 46, 72] takes a counterfactual perspective. Ghazimatin et al. [23] propose to use a subset of users' interaction history as counterfactual explanation and use user studies to prove the generated explanations are of higher quality. Tan et al. [72] propose to evaluate the explanations using *necessity* (whether the condition is necessary) and *sufficiency* (whether the condition is sufficient). However, their methods are bound to the specific recommendation algorithm, and the authors fail to properly illustrate how these explanation frameworks can be adapted to other explanation models. Besides, although the counterfactual models can deliver human-interpretable explanations [23, 65, 72], the evaluation of faithfulness has been overlooked in these works. We also include two counterfactual baselines and show that explanations being counterfactually true do not necessarily guarantee they can reflect the actual reasoning process of the recommendation model. In this work, we propose an evaluation pipeline to evaluate the model-agnostic explanations from the perspective of faithfulness and scrutability. Our work is different from existing works from three perspectives: (1) We focus on the evaluation of generated explanations from the perspectives of faithfulness and scrutability, which has been largely overlooked by previous works in explainable recommendation; (2) our evaluation pipeline is not limited to a certain explanation style, e.g., aspects [72] or association rules [58]; and (3) with proper modification, our model-agnostic explanation framework can be used for other genres of recommendation model (detailed in Section 6.3).

2.4 Knowledge Distillation

The idea of knowledge distillation is first brought up by Hinton et al. [32]. By adding soft-targets generated from teacher networks as training objectives, the student network can achieve knowledge transfer [57]. Originally, knowledge distillation is used for model compression and acceleration [32, 68]. It has also been explored for other purposes, such as efficient building blocks for deep models (see Reference [26] for a comprehensive survey). Hofstätter et al. [33] use cross-architecture distillation to enable model compression and performance improvement for dense retrieval. Within the IR community, knowledge distillation has also been studied. Tang and Wang [73] study ranking distillation for model compression. Liu et al. [49] design a knowledge distillation framework for counterfactual recommendation via uniform data. Zhang et al. [96] explore distilling the knowledge from knowledge graph paths into latent factor embeddings in matrix factorization for better recommendation performance. Perhaps the closest works to this work are References [20, 59], where the authors explore self distillation for better performance. Within this work, we aim to provide a model-agnostic explanation framework for explainable recommendation for better faithfulness and scrutability. To achieve this goal, we use the knowledge distillation technique where the student model, i.e., the explanation model is able to learn the reasoning process from the teacher model, i.e., the recommendation.

3 OVERVIEW

We start this section by introducing the problem formulation and presenting a high-level overview to our model-agnostic explanation framework for explainable recommendation (Section 3.1); then we illustrate the design logic for the proposed evaluation pipeline with the focus of improving explanation faithfulness and system scrutability (Section 3.2). In Section 3.3, we discuss the assumptions we use in this work.

Table 1. A Summary of Notations We Use in This Article

$u, \mathcal{U}, i, \mathcal{I}, a, \mathcal{A}, \mathcal{D}$	user, user set, item, item set, aspect, aspect set, interaction set
$\mathcal{I}_u$	items that user u has interacted with previously
$\mathcal{A}_{ui}$	aspects of item i mentioned by user u
$\mathcal{A}_u$	aspect set mentioned by user u
$\mathcal{A}_i$	aspect set of item i
$\overrightarrow{\boldsymbol{u}}, \overrightarrow{\boldsymbol{i}}, \overrightarrow{\boldsymbol{a}}$	user/aspect/item latent factor
$X_{\lvert\mathcal{U}\rvert\times\lvert\mathcal{A}\rvert}$	user–aspect attention matrix
$Y_{\lvert\mathcal{I}\rvert\times\lvert\mathcal{A}\rvert}$	item–aspect quality matrix
s_{ui}	ranking score for user–item pair (u, i)
$\mathcal{A}_{ui}@k$	most important k aspects determined by blackbox recommendation model
$\mathcal{A}'_{ui}@k$	most important k aspects determined by whitebox explanation model
$s(path_{ui}) \in \mathcal{P}_{ui}$	knowledge-graph reasoning path & paths set connecting (u, i)

3.1 Problem Formulation and Overview of Model-agnostic Explanation Framework

We study the task of top-N recommendation task with implicit feedback. Formally, let $\mathcal{U}, \mathcal{I}$ be the set of users and the set of items. We observe the user–item interactions set $\mathcal{D} = \bigcup_{u \in U}\{(u, i)|i \in \mathcal{I}_u\}$ where $\mathcal{I}_u$ is the set of items user u previously interacted with. The task is to provide each user with a ranklist of recommended items and to provide a piece of explanation E to each item. Depending on the specific model choices, the explanations can be in different forms (detailed in Section 5.1). A summary of notations can be found in Table 1.

Our explanation framework consists of two parts: a blackbox recommendation model to generate recommendation decisions and a whitebox explanation model to generate the explanations for the recommendation decisions. We provide an overview of our model-agnostic explanation framework in Figure 1. The blackbox recommendation model is on the left side and the whitebox explanation model on the right side. The input of the whitebox explanation model is the user–item pair, and it will generate explanations for why this item might be relevant to the user.

3.2 Evaluation Pipeline

The proposed evaluation pipeline evaluates a model-agnostic explanation framework from two perspectives, explanation faithfulness and system scrutability. Here we illustrate the high-level logic for our evaluation pipeline and leave the detailed design and implementation of metrics to Section 5.4.

3.2.1 Faithfulness and Its Evaluation. Faithfulness measures to what extent the generated explanations can reflect the actual reasoning process of the recommendation model. From this sense, it is hard to quantify the level of faithfulness if the two parts of the explanation framework utilizes completely different inputs and model structures, e.g., matrix factorization for blackbox recommendation model and association rules for whitebox explanation model [58]. Some works in **natural language processing (NLP)** [35, 36, 83] and **computer vision (CV)** [42] propose to evaluate the faithfulness by measuring the overlap of important inputs with regard to model predictions determined by the decision model and explanation model, i.e., feature attribution, such as the overlap of input tokens or pixels highlighted by the two models. We take a similar view and propose to consider the overlap of important input features to the output of interest.

In our evaluation pipeline, the explanation faithfulness is measured by the overlap between explanation generated by blackbox recommendation model and explanation generated by whitebox explanation model (*Explanations* → *Faithful* → *Reasons* in Figure 1).

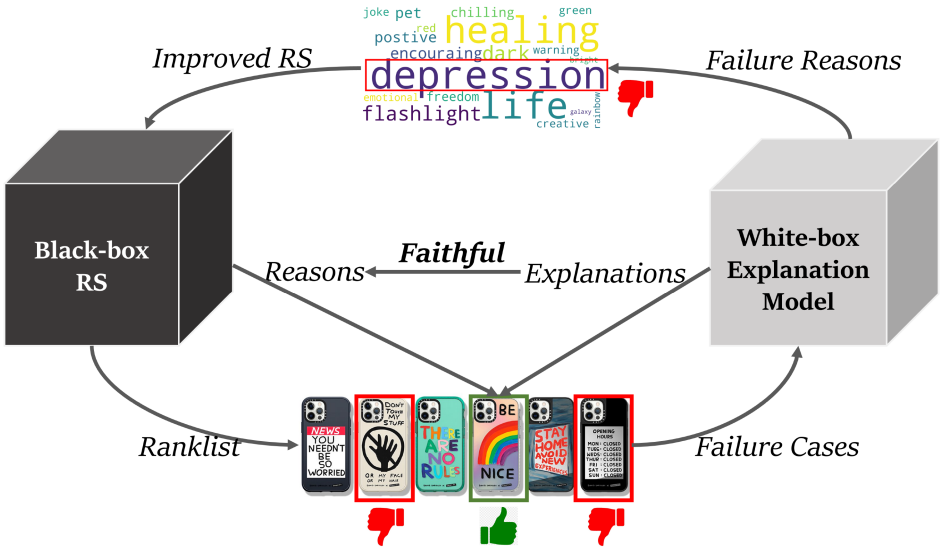


Fig. 1. We use a simple product recommender to explain the proposed model-agnostic explanation framework and evaluation pipeline. The faithfulness is evaluated by the overlap of important features determined by the blackbox recommendation model and the whitebox explanation model. We show a workflow of scrutability analysis in the outer loop.

3.2.2 Scrutability and Its Evaluation. *Scrutability* is another ideal property for industrial-level RS [8, 24, 38, 75]. A scrutable RS model could help system designers analyze failure cases, e.g., a video received *Dislike* on YouTube, and improve the system in the long run. In Reference [75], *scrutability* is defined as *whether the system allows its users to tell it is wrong*. Balog et al. [8] propose a scrutable explainable RS algorithm with set-based preference, but their discussion is limited to model-intrinsic explanations. In model-agnostic explainable RS, the concept of scrutability and its evaluation has not been well studied.

Here in a model-agnostic setting, we adapt the definition of scrutability to *whether the explanation model can take the negative feedback from users, understand the reasons for the failure cases, and improve the system in terms of other explanable recommendation design goals*. The whitebox explanation model is responsible for finding the exact reasons for the failure cases and uses those to help improve the blackbox recommendation model. In our evaluation pipeline, the workflow of system scrutability measurement is shown in the outer loop of Figure 1 (*Ranklist* → *Failure Cases* → *Failure Reasons* → *Improved RS*). Here we should note that, unlike faithfulness, the evaluation of scrutability does not require the blackbox recommendation model and whitebox explanation model to share the same set of input features. A content-based explanation model can still help debug a matrix factorization recommendation model if with specific design (more discussion in Section 6.3).

3.3 Assumptions

Throughout this article, we follow three key assumptions:

- The blackbox recommendation model is already trained, and we only focus on providing a good whitebox explanation model to improve explanation faithfulness and system scrutability.

- We use input features as explanations, which is referred to as the Attribution method [4] and has been common practice in previous explainable AI/ML literature [2, 23, 58, 61, 62, 65, 72, 74].
- Following the previous point, we assume the blackbox recommendation model and the whitebox explanation model share the same set of input features.[6] The whitebox explanation model has access to the blackbox recommendation model's predictions but has no knowledge on the detailed model structures, gradients, and embeddings of the blackbox recommendation model; thus our whitebox explanation model is "model agnostic" and "data-type agnostic" [39]. This setup has also been widely adopted by previous explainable AI/ML and recommendation literature [2, 3, 8, 58, 61, 62, 65, 72, 74, 82].

4 MODEL-AGNOSTIC RECOMMENDATION EXPLANATION WITH KNOWLEDGE DISTILLATION

Since we follow the model-agnostic setting i.e., the whitebox explanation model will have no knowledge w.r.t. the blackbox recommendation other than its predictions, it can be challenging to design a trainable whitebox model, because neither the blackbox recommendation model's structures nor the trained parameters are available. Ideally, the whitebox explanation model should be able to learn the blackbox recommendation model's actual reasoning process. Previous works [32, 33, 68] has shown that with knowledge distillation style training, the student network is able to efficiently and effectively *distill* the knowledge from teacher network. This matches our design goal of explanation faithfulness and system scrutability.

In this work, we propose to train the whitebox model in our model-agnostic explanation framework with a knowledge-distillation-style training strategy. Specifically, the whitebox explanation model is treated as a student network and is trained with the supervision signals generated by the teacher network, i.e., the blackbox recommendation model in our problem setting. By using this knowledge-distillation strategy, the whitebox explanation should be able to learn from blackbox recommendation model's decision pattern and learn its actual reasoning processes, even if their model structures and parameters are different. We should note that we are not the first to use this knowledge-distillation-style training strategy in the explainable AI/ML literature. Bastani et al. [9] term this as *Model Extraction* and use it to train model-agnostic explainer. However, their experiment is conducted under classical ML settings, e.g., classification tasks with random forest as whitebox explanation model. And how to adapt this strategy for model-agnostic explanation for explainable recommendation has not been studied by the IR community.

We present an example figure for our training strategy in Figure 2. Considering recommendation as a personalized ranking task, for each user u, we denote the supervision signals $S_u = [s_{u1}, s_{u2}, \ldots, s_{u|\mathcal{I}|}]$, where $|\mathcal{I}|$ denotes the size of whole item set, and s_{ui} is blackbox recommendation model's predicted ranking score for (u, i); similarly, we let the whitebox explanation model generate the corresponding S'_u in the forward pass. We construct the loss function as standard cross entropy loss used in previous knowledge distillation literature [32]. In addition, we normalize the supervision signals S_u to P_u with a softmax operation with τ as a hyperparameter to smooth the gradients:

$$p_{ui} = \frac{\exp\left(s_{ui}/\tau\right)}{\sum_{j=1}^{\mathcal{I}} \exp(s_{uj}/\tau)}. \tag{1}$$

This P_u reflects the user's preference over $\mathcal{I}$ as a softer probability distribution learned by the blackbox recommendation model, and we use it to train the whitebox explanation model. Similarly,

[6] Also classified as *dataset sharing* in Reference [50].

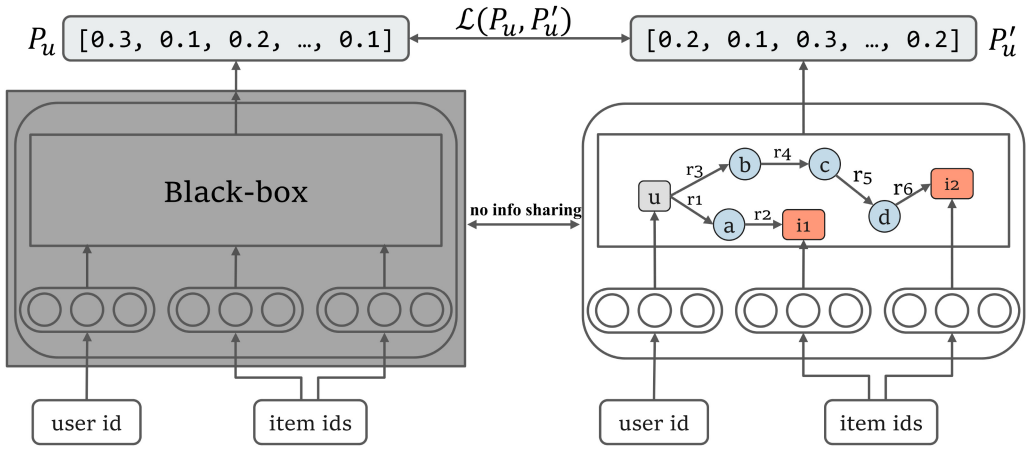


Fig. 2. A closer look of our model-agnostic explanation framework for recommendation, both the blackbox recommendation model and the whitebox explanation model take user id and item ids as input and generate prediction scores P_u and P'_u, respectively. The parameters of the whitebox explanation model is optimized by the cross-entropy loss function $\mathcal{L}(P_u, P'_u)$. Within the whitebox explanation model, the path $u \rightarrow r1 \rightarrow a \rightarrow r2 \rightarrow i1$ means user u interacts item i via aspect a; the path $u \rightarrow r3 \rightarrow b \rightarrow r4 \rightarrow c \rightarrow r5 \rightarrow d \rightarrow r6 \rightarrow i2$ means user u interacts item $i2$ via a multi-hop knowledge-graph reasoning path.

we denote the output probability distribution from the whitebox model's forward as P'_u. And the loss is computed by

$$\mathcal{L} = \sum_{u \in \mathcal{U}} \texttt{CrossEntropy}(P_u, P'_u) = -\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} p_{ui} \log p'_{ui}. \tag{2}$$

We leave the specific details of constructing S_u and transformation from S'_u to P'_u to Section 7.1.3 and Section 5.3, respectively.

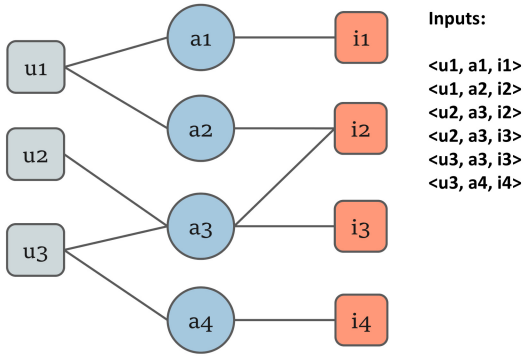
5 DETAILED DESIGN AND IMPLEMENTATION

In this section, we first summarize the model-specific notations (Section 5.1). Then we illustrate how we implement the above model-agnostic explanation framework using two representative genres of models, aspect-based models and knowledge-graph-based models. We further introduce the model-specific design and implementation for our evaluation pipeline metrics (Section 5.4).

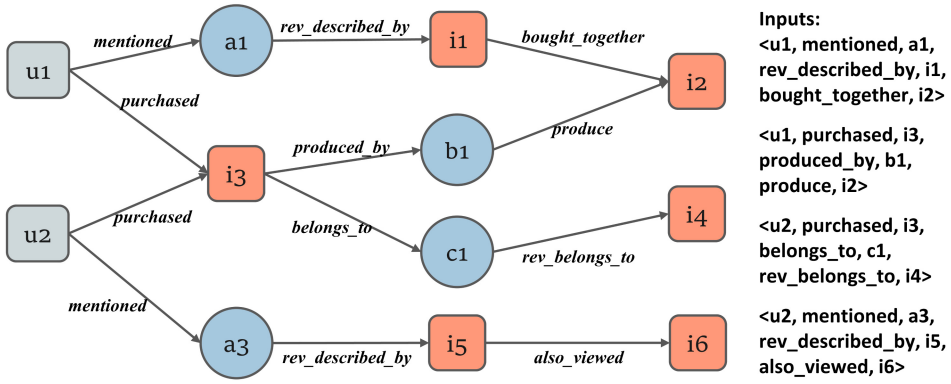
5.1 Model-specific Notations

In this work, we include both explicit feedback-based datasets, i.e., user-generated reviews, and implicit feedback-based datasets, i.e., clicks. For implicit feedback-based datasets, we can observe the click/interaction set $\mathcal{D}$. For explicit feedback-based datasets, each user–item interaction pair $(u, i) \in \mathcal{D}$ is associated with a piece of textual reviews from which we can extract explicit textual features that describes the aspects of the item, e.g., price, novelty, and durability. We denote these explicit textual features as **Aspects a**(s), and the aspects extracted from this user–item interaction as $\mathcal{A}_{ui}$ This user-aspect-item relations form a triparitite graph (example in Figure 3(a)) and has been studied by previous works [30, 80]. In the meantime, we can also extract a product knowledge graph $\mathcal{G}$ from the dataset with entity set $\mathcal{E}$ and relation set $\mathcal{R}$ (example in Figure 3(b)). A summary of notations can be found at Table 1.

In our experiments, we implement the model-agnostic explanation framework with two genres of models. The aspect-based model predicts the ranking score s_{ui} via aspect-level interactions. The



(a) A sample tripartite graph



(b) A sample user-item product knowledge graph

Fig. 3. In Figure 3(a), a sample user-aspect-item tripartite graph is presented where u stands for *User*, a stands for *Aspect*, and i stands for *Item*; the graph nodes are connected via undirected edges. In Figure 3(b), a sample product knowledge graph is presented, additionally, b stands for *Brand*, c stands for *Category*; the user node and item node are connected via multi-hop connections through different categories of nodes and edges. Note that the product knowledge graph is a directed graph.

explanation task is defined as follows: For user u and a recommended item i, find several explicit textual features/aspects from $\mathcal{A}$ and use them to explain why this item is recommended to u. The knowledge-graph-based model predicts the ranking score s_{ui} from learning the representations from product-knowledge-graph $\mathcal{G}$ and interaction set $\mathcal{D}$. The explanation task is defined as follows: For user u and a recommended item i, to generate a knowledge graph reasoning path $path_{ui}$ connecting u and i to explain the recommendation.

We present a running example in the right-hand side of Figure 2, where $u \rightarrow r1 \rightarrow a \rightarrow r2 \rightarrow i1$ means user u interacts item $i1$ because of aspect a and $u \rightarrow r3 \rightarrow b \rightarrow r4 \rightarrow c \rightarrow r5 \rightarrow d \rightarrow r6 \rightarrow i2$ means user u interacts item $i2$ via a multi-hop knowledge-graph reasoning path. For the aspect-based models, we adopt the explanation template used by previous works [30, 95]: `You might be interested in` [ASPECT A], `on which this product performs well`. We will cover more model details in Sections 5.2 and 5.3.

5.2 Blackbox Recommendation Model

We use two representative lines of models: aspect-based models and knowledge-graph-based models. For aspect-based models, we select the **Explicit Factor Model (EFM)** [95] and **Attribute-aware Collaborative Filtering Model (A2CF)** [15]; we select **Policy-Guided Path Reasoning (PGPR)** [84] and **Coarse-to-Fine Neural Symbolic Reasoning (CAFE)** [85] as the representative knowledge-graph-based models. We give a brief introduction here and refer the model details to the original papers.

5.2.1 Explicit Factor Model. In EFM, the user, item, and aspect latent representations are learned by performing matrix factorization over the user–item rating matrix $R_{|\mathcal{U}|\times|\mathcal{I}|}$, the user–aspect attention matrix $X_{|\mathcal{U}|\times|\mathcal{A}|}$, and the item–aspect quality matrix $Y_{|\mathcal{I}|\times|\mathcal{A}|}$. In the inference stage, the predicted ranking score s_{ui} is computed by combining the user–item rating score and user–item similarity score, which is based on the top-k most important product aspects in $\mathcal{A}_u \cap \mathcal{A}_i$. Here the top-$k$ important aspects are determined by acquiring the top-k largest values of the element-wise product between the user–aspect vector from $X_{|\mathcal{U}|\times\mathcal{A}}$ and item–aspect vector from $Y_{|\mathcal{I}|\times|\mathcal{A}|}$.

5.2.2 Attribute-aware Collaborative Filtering Model. The training of A2CF model consists of two stages. First, it leverages a residual feed-forward network to predict the missing values in user–aspect attention matrix $X_{|\mathcal{U}|\times|\mathcal{A}|}$ and item–aspect quality matrix $Y_{|\mathcal{I}|\times|\mathcal{A}|}$. Then it combines the aspect latent factors in stage 1 to build user and item representations and use a deep feed-forward neural network to compute the ranking score. The algorithm originally considers both user–item preference relations and item–item similarity relations for substitute recommendations. To make it compatible with our problem setting, we only use the user–item preference relations.

5.2.3 Policy-guided Path Reasoning. PGPR regards the path reasoning in knowledge graph as a searching algorithm and propose a RL algorithm motivated by Markov Decision Process. The graph reasoner starts from user node u, walks through graph $\mathcal{G}$, and eventually reach item node i via path p_{ui}. In the training stage, the ground-truth paths are not annotated so the authors propose to train the model by interactions $\mathcal{D}$. In the inference stage, the graph reasoner search all the items reachable from the user node and construct the ranklist using the ranking scores of the paths connecting the user node and item nodes.

5.2.4 Coarse-to-Fine Neural Symbolic Reasoning. Similarly to PGPR, CAFE also treats the path reasoning in knowledge graph as a search process, and the same authors improve over PGPR by using a behavior cloning [37] technique to improve the search process. In the training stage, a random walk module is used to sample the paths between training user–item pairs, and then these paths are used to train a behavior cloning module with negative log-likelihood loss. In the inference stage, for each user, CAFE first constructs a layout tree starting from the user node and then ranks the candidate items by sorting the ranking scores from the log-likelihood of the possible paths to the candidate items.

5.3 Whitebox Explanation Model

Correspondingly, we also utilize aspect-based model and KG-based model as our whitebox explanation model. For aspect-based models, we take the common practice in the literature [30] to treat the **User-Aspect-Item (U-A-I)** relations as a tripartite graph and use node embeddings to represent the users, items, and aspects. The recommendation process is divided into two stages: user–aspect match and item–aspect match. Correspondingly, we propose two simple models referred to as **Hard-Aspect-Match (HAM)** model and **Soft-Aspect-Match (SAM)** model. A sample U-A-I tripartite graph gets shown in Figure 3(a).

5.3.1 Hard-Aspect-Match Model. The HAM model utilizes hard user–aspect match and item–aspect match. The hard user–aspect matching score s_{ua} is computed by

$$s_{ua} = \frac{f(\overrightarrow{\boldsymbol{u}}, \overrightarrow{\boldsymbol{a}})}{\sum_{a_j \in \mathcal{A}_u} f(\overrightarrow{\boldsymbol{u}}, \overrightarrow{\boldsymbol{a}_j})}, \tag{3}$$

where $f(\cdot, \cdot)$ is a function to measure the similarity of two vectors, e.g., dot product or cosine similarity. Similarly, we have

$$s_{ia} = \frac{f(\overrightarrow{\boldsymbol{i}}, \overrightarrow{\boldsymbol{a}})}{\sum_{a_j \in \mathcal{A}_u} f(\overrightarrow{\boldsymbol{i}}, \overrightarrow{\boldsymbol{a}_j})}. \tag{4}$$

The matching score of user u interacting with item i via aspect a is computed by

$$s_{uai} = s_{ua} \cdot s_{ia}. \tag{5}$$

The matching score s_{ui} is computed by summing up all aspects in $\mathcal{A}_u \cap \mathcal{A}_i$,

$$s_{ui} = \begin{cases} \sum_{a_j \in \mathcal{A}_u \cap \mathcal{A}_i} (s_{ua_j} \cdot s_{ia_j}), & \text{if } \mathcal{A}_u \cap \mathcal{A}_i \neq \emptyset \\ 0, & otherwise \end{cases}, \tag{6}$$

and we simply set $p_{ui} = s_{ui}$. Here $\sum_i^{|\mathcal{I}|} p_{ui} = 1$, and thus we can train the hard-aspect-match model with cross entropy loss from Equation (2). In HAM, the term "hard" means that we force the user–item interaction to happen through the aspects they have in common.

5.3.2 Soft-Aspect-Match Model. In the SAM model, we relax the constraint on the user–item interaction by allowing the user and item to attend to every aspect a in $\mathcal{A}$ even if a is not in $\mathcal{A}_u$ or $\mathcal{A}_i$. We have

$$s_{ua} = \frac{f(\overrightarrow{\boldsymbol{u}}, \overrightarrow{\boldsymbol{a}})}{\sum_{a_j \in \mathcal{A}} f(\overrightarrow{\boldsymbol{u}}, \overrightarrow{\boldsymbol{a}_j})}, \tag{7}$$

$$s_{ia} = \frac{f(\overrightarrow{\boldsymbol{i}}, \overrightarrow{\boldsymbol{a}})}{\sum_{a_j \in \mathcal{A}} f(\overrightarrow{\boldsymbol{i}}, \overrightarrow{\boldsymbol{a}_j})}, \tag{8}$$

$$s_{ui} = \sum_{a_j \in \mathcal{A}} (s_{ua_j} \cdot s_{ia_j}). \tag{9}$$

Similarly to Equation (6), $p_{ui} = s_{ui}$ and $\sum_{i=1}^{\mathcal{I}} p_{ui} = 1$. As we do not restrict the user–item interaction to happen in shared aspects set, we refer to this model as the Soft-Aspect-Match model.

5.3.3 Knowledge-graph-based Model.
We show a sample knowledge graph in Figure 3(b). We use an existing path-based model KPRN [81]. KPRN treats the knowledge graph paths as sequences and nodes and relations as tokens in a sequence. It trains a LSTM model to score the potential paths $s(path_{ui})$ connecting (u, i) pairs in interactions set $\mathcal{D}$. In the inference stage, it first uses a graph search algorithm to find all potential paths connecting (u, i) and uses the LSTM model to generate a ranking score for each path *path*. The final ranking score is derived by averaging $s(path_{ui})$ in $\mathcal{P}_{ui}$. The model can be formulated as

$$s(path_{ui}) = FFN(LSTM(path_{ui})), \tag{10}$$

$$s_{ui} = \frac{1}{|\mathcal{P}_{ui}|} \sum_{path_{ui} \in \mathcal{P}_{ui}} s(path_{ui}), \tag{11}$$

where *FFN* is a feed-forward neural network, and $|\cdot|$ denotes the size of the set. We convert the ranking scores S_u over $\mathcal{I}$ into probability distribution P_u with a softmax operator similar to Equation (1).

5.4 Detailed Evaluation Pipeline Design and Implementation

We have briefly covered the high-level logic for our evaluation pipeline, and now we dive into more details on how we design and implement the evaluation metrics of explanation faithfulness and system scrutability.

5.4.1 Faithfulness Metric Design. As mentioned in Section 3.2, we opt to evaluate the explanation faithfulness by measuring the overlap of important input features determined by the blackbox recommendation model and whitebox explanation model. Under this setup, the design of faithfulness metric is quite flexible and more model dependent. Here we only provide two sample faithfulness metric designs for aspect-based models and KG-based models and will discuss guidelines for other genres of models later in Section 6.3.

Previous aspect-based explainable recommendation models [10, 30, 34, 95] define their task as to provide item recommendations as well as the aspects that the user may be interested. Bauman et al. [10] evaluate their model by computing precision between the predicted aspects and the ground-truth aspects, i.e., the aspects that appear in users' reviews. Therefore, we believe the overlap of predicted aspects between our blackbox recommendation model and whitebox explanation model should be a good reflection on the faithfulness of whitebox explanation model. Formally, for aspect-based model, we define the faithfulness as the **Average Aspect Overlap (AAO)** per item as follows:

$$AAO = \frac{\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}_u@n} |\mathcal{A}_{ui}@k \cap \mathcal{A}'_{ui}@k|}{|\mathcal{U}| \times n}, \tag{12}$$

where $\mathcal{A}_{ui}@k$ and $\mathcal{A}'_{ui}@k$ are the top-k most important aspects determined by the blackbox recommendation model and whitebox explanation model, respectively; $\mathcal{I}_u@n$ are the blackbox recommendation model's top-n recommended items for u; and $|\cdot|$ is the size of the set.

For knowledge-graph-based model, we propose to evaluate the faithfulness by computing the **normalized Generalized Levenshtein Distance (nGLD)** [50, 52, 92] between blackbox recommendation model generated path $path_{ui}$ and whitebox explanation model generated path $path'_{ui}$. Generalized Levenshtein Distance is originally proposed to measure the similarity of two strings by computing the minimum number of steps to transfer one string to the other through *Insert*, *Delete*, or *Replace*. We propose to evaluate the similarity of two knowledge graph reasoning paths by casting them as two sequences of strings and compute the similarity between the two strings. Specifically, we encode $path_{ui}$ and $path_{ui}$ into two strings str_{ui} and str'_{ui} and compute the nGLD as

$$nGLD = \frac{max\{len(str_{ui}), len(str'_{ui})\} - GLD(str_{ui}, str'_{ui})}{max\{len(str_{ui}), len(str'_{ui})\}}, \tag{13}$$

where *GLD* is the Generalized Levenshtein Distance.[7] From Equation (13) we can see $nGLD \in [0, 1]$ where 1 means two paths are identical and 0 means two paths are totally different.

5.4.2 Scrutability Metric Design. Recall from Section 3.2 that the idea of scrutability in model-agnostic explainable recommendation is to use the whitebox explanation model to help debug/improve the blackbox recommendation model. We start from the logic that after detecting

[7]We skip the detailed computation process here as this is not the focus of this article; interested readers can refer to Reference [52] for details.

one failure case (u, i) (e.g., *Dislike* on Youtube), the explanation model could generate an explanation on why i is being recommended. If the explanation model is correct about this explanation, then removing it from the dataset can prevent this failure item from being recommended, i.e., the rank of the failure item will drop out from the top of ranklist. This reason located by the explanation model could be a historical record of user u when matrix factorization is used as blackbox recommendation model, or an aspect/KG-entity in our experiments. If the failure item i is ranked lower after removing the training instances located by the whitebox explanation model, then it suggests that the agnostic explanation model could be used to interpret the blackbox recommendation model and improve the system scrutability. A more interpretable example is presented in the outer loop of Figure 1.

We propose to measure the scrutability of a model-agnostic explanation algorithm by **Average Position Change (APC)**:

$$APC = \frac{\sum_{f_{ui} \in F} R(f)' - R(f)}{|F|}, \tag{14}$$

where $f_{ui} \in F$ is a failure case, $R(f)$ and $R(f)'$ are the item's rank on the ranklist before and after removing the training instances, and $|\cdot|$ measures the size of set. A higher APC for failure items suggests the effectiveness of the explanation model, i.e., it is able to help find the reason for the failure cases, and in return improve the blackbox recommendation model. We leave the detailed algorithm to Section 7.3.

5.5 Detailed Training of Whitebox Explanation Model

In our model-agnostic explanation framework, we adopt a knowledge-distillation-style training strategy. Specifically, we train the whitebox explanation model using blackbox recommendation model's output as supervision signals. To better understand our whitebox explanation models, we also use ground truth in trainset as supervision signals to train them and use the result on testset as a benchmark. We refer to this training strategy as vanilla training. For simplicity, we refer to SAM model trained with EFM output as SAM-EFM, SAM model trained with A2CF output as SAM-A2CF, and SAM model trained with ground truth as SAM-Vanilla; the same naming pattern also applies for HAM; we refer the KPRN model trained with PGPR output as KPRN-PGPR, KPRN model trained with CAFE output as KPRN-CAFE, and, similarly, KPRN-Vanilla.

5.5.1 Knowledge-distillation-style Training. For user u and item i, let s_{ui} be the blackbox recommendation model's prediction for (u, i), and s'_{ui} be the whitebox prediction on (u, i), thus S_u denotes the blackbox recommendation model's predictions for user u over the item set. We apply a softmax transformation (Equation (1)) to map S_u into probability distribution P_u, and construct whitebox model's output such that it can be treated as probability distribution over $\mathcal{I}$ (detailed in Section 5.3). Then we optimize the whitebox models with the cross entropy loss (Equation (2)) between the two probability distributions. We detail the construction of supervision signals S_u in Section 7.1.3.

5.5.2 Vanilla Training. For each user u in the trainset, we use the negative sampling approach to sample a small set of negatives $\Omega(\mathcal{I} \setminus \mathcal{I}_u)$. Then we correspondingly construct the supervision signals S_u and optimize the following loss function,

$$\mathcal{L} = -\left(\sum_u \sum_{i \in \mathcal{I}_u} \frac{exp(s'_{ui}/\tau)}{\sum_{j \in \Omega(\mathcal{I} \setminus \mathcal{I}_u) \cup \mathcal{I}_u} exp(s'_{uj}/\tau)} - \lambda ||\Phi||^2 \right), \tag{15}$$

where Φ is the set of all model parameters to be learned in the training process, τ is a temperature hyperparameter for gradient smoothing and model convergence, and λ is regularization coefficient to prevent overfitting.

6 DISCUSSIONS

Within this section, we discuss the time complexity as well as limitations of the proposed model-agnostic explanation framework and provide practical guidances on adapting it to other genres of recommendations.

6.1 Time Complexity Analysis

We should note that within the proposed model-agnostic recommendation and explanation framework, the recommendation stage and explanation stage can be executed *asynchronously*, i.e., the framework first generates a small ranklist π_k of k items from the whole collection of size $|\mathcal{I}|$ with the blackbox recommendation model and then generates explanations for these k items with the whitebox explanation model.

Denote the time complexity of blackbox recommendation model's inference for one item as P, and time complexity of generating explanation for one single item as Q, which varies in practice depending on the specific choice of the whitebox explanation model, the overall time complexity for the recommendation stage is $O(P|\mathcal{I}|)$, and the explanation stage is $O(Qk)$; the overall time complexity for our framework is $O(P|\mathcal{I}| + Qk)$. Since in real-world systems the item collection usually consists of millions of items, among which only a small subset of items is shown in the final ranklist, $k \ll |\mathcal{I}|$, it is therefore safe to assume $O(P|\mathcal{I}| + Qk) \approx O(P|\mathcal{I}|)$. Moreover, we do not need to restrict the model choice of the recommendation model to model-intrinsic explainable models and have the freedom to choose uninterpretable but efficient models, such as embedding-based models; this way we can make $P < Q$, and $O(P|\mathcal{I}| + Qk) \approx O(P|\mathcal{I}|) < O(Q|\mathcal{I}|)$.

6.2 Limitations

At this stage, one potential concern from the readers may be how our explanation framework can be truly model agnostic, i.e., the whitebox explanation model should assume no knowledge w.r.t. the blackbox recommendation model, such as input features; Our experiment is currently limited, because for the sake of evaluation, we still need to implement the blackbox recommendation model with model-intrinsic explainable recommendation model and allow the blackbox recommendation model and whitebox explanation model to share the same set of input features. Only with this setting, the faithfulness of the whitebox explanation model could be quantitatively evaluated by whether the explanation generated by the whitebox explanation model can match the explanation generated by the blackbox recommendation.

6.3 Guidelines on Adapting our Framework to Other Recommendation Models

According to the genres of blackbox recommendation model, we can further divide our framework into two separate subcategories:

6.3.1 The Recommendation Model Does Not Use Explicit Content-based Features. Such kinds of recommendation models can include matrix factorization [60] and other complicated interaction-based models [71, 88]: This setting can be intricate, as there has not been a well-established method to quantitatively evaluate the faithfulness of explanations w.r.t. interaction-based models. Unlike faithfulness, we can still measure the system scrutability following the same idea as Section 5.4. For example, similarly to Reference [8], we can implement the explanation model as a model-intrinsic explainable model and use the explicit features as explanation, e.g., the tags of user's previously

interacted items as in Reference [8]; then we measure the Average Position Change after removing the corresponding user–item pair of the tags found by the explanation model from user's historical interactions.

6.3.2 The Recommendation Model Makes Use of Explicit Features. These models make use of content-based features, such as user-generated product reviews [14, 90, 93], Heterogeneous Information Graph or Product Knowledge Graph [22, 84, 85], or hand-crafted learning-to-rank style features [70, 91]. This setting is more straightforward, because as system designer, we already know what features the blackbox recommendation model uses and its detailed model structure. The design goal is changed to designing a similar but simpler whitebox explanation model with same set of features and to train it with the knowledge distillation training strategy in Section 4 such that certain requirements can be met. For example, if the reasoning of recommendation model is too complicated for users to comprehend, then the system designer can design the whitebox explanation model to provide simplified and more straightforward reasonings while maintaining a certain degree of faithfulness with the proposed knowledge-distillation-style training strategy; if the design goal is to provide real-time explanation for an offline recommendation model, e.g., to provide explanation to a certain recommended item that the user clicks through from a promotional email, then the system designer can put the focus of whitebox explanation model on latency, e.g., predefined templates with aspects predicted by an aspect-based explanation model.

7 EXPERIMENTS

In this section, we study the effectiveness of our framework. Specifically, we discuss three research questions:

RQ1: Can our model-agnostic framework and training strategy improve the faithfulness of generated explanations?
RQ2: Can our whitebox explanation model be used to improve the scrutability of blackbox recommendation model?
RQ3: Since our whitebox explanation model can give recommendations by itself, how is the recommendation performance? To what extent can it mimic the blackbox recommendation model's recommendation decisions?

7.1 Experimental Setup

7.1.1 Datasets and Dataset Partitions. We use three categories of data from the famous Amazon Product Review Dataset[8] [53] and one implicit-feedback click dataset Last FM.[9] Amazon datasets focus on product recommendation while Last FM dataset focuses on music recommendation.

For the Amazon datasets, we use the well-established aspect extraction tool Sentires[10] [95] to extract the aspects from the product reviews. We filter out the users, items, and aspects with fewer than five, five, and three entries, respectively. Thus, each user will have at least five interactions. Due to the sparsity of our dataset, we use each user's last two interactions as a testset, the second-to-last interaction as a validation set, and the rest as the trainset. For the Last-FM dataset, we use the preprocessed knowledge graph and train-test split from Reference [6]. Since the Last-FM dataset is constructed from implicit-feedback clicks, we do not conduct aspect-based models' experiments on it. Detailed statistics of the dataset are shown in Table 2.

[8] Amazon Product Review: http://jmcauley.ucsd.edu/data/amazon/links.html.
[9] https://grouplens.org/datasets/hetrec-2011/.
[10] Sentires: https://github.com/evison/Sentires.

Table 2. The Basic Statistics of the Datasets

Dataset	Electronics	Office Product	Tools & Home	Last-FM
#users	3,151	1,983	651	5,941
#items	3,253	957	515	10,303
#aspects	200	452	326	—
#total nodes	7,313	3,712	1,888	33,970
#total relations	14	14	14	14
interaction density	0.31%	1.15%	1.63%	1.73%

7.1.2 Ablations and Baselines. **Ablations:** We use the whitebox explanation model trained with ground-truth (vanilla training) as an ablation. The intuition is that if our knowledge-distillation-style training strategy works, then the result generated by the explanation model should have better faithfulness compared to the result from the explanation model with vanilla training strategy.

We additionally make use of two other model-agnostic counterfactual baselines:

PRINCE [23]: PRINCE propose to find a subset from user's historical interactions and use them as counterfactual explanations. More specifically, the explanation of PRINCE is defined as a user's necessary set of historical actions such as Click, View, and Add-to-cart and if removed will lead to a different recommended item.

CountER [72]: CountER is another comparable baseline that proposes a counterfactual joint optimization problem by solving which it generates explanations based on the explicit aspects of the items.

We adapt both baselines to make them work for the aspect- and KG-based models. For ease of comparison, we also include a weak *Random* baseline, specifically, in the inference stage, we randomly arrange the candidates items (or candidate aspects) in the candidates set.

7.1.3 Construction of Supervision Signals. For aspect-based models, we construct the supervision signals S_u over the whole item set $\mathcal{I}$. For KG-based models, it is computationally expensive to go through $\mathcal{I}$ due to the unparalleled nature of LSTM. We adopt the negative sampling strategy used in the original KPRN paper [81]. Specifically, for each u, we construct ~~S_u~~ P_u by assigning top-20 items from the blackbox recommendation model's ranklist with label 1 and assign the sampled negatives from the rest of ranklist with label 0, i.e., $p_{ui} \in \{0, 1\}$. This can be regarded as $s_{ui} = +\infty$ for positive items and $s_{ui} = -\infty$ for sampled negatives. In practice, we use a ratio of 1:4, i.e., 20 positive items and 80 negative items for each u. More specifically, during training we organize each training batch into mini batches, and each mini batch consists of 1 positive item and 4 negative items. Then we compute the loss with Equation (2), but instead the denominator is calculated over the mini batch rather than the whole item set $\mathcal{I}$. For vanilla training, we use a same 1:4 ratio, but instead the positive items are from the training set rather than the ranklist from the blackbox recommendation model as teacher. And the same mini batch training setup applies to vanilla training for the aspect-based models as well.

7.1.4 Training and Evaluation. We train our model on the trainset and use the validation set to tune hyperparameters. We conduct a full ranking on all datasets, namely, for each user u, we rank all items in $\mathcal{I} \setminus \mathcal{I}_u$. We use the evaluation metrics introduced in Section 5.4. We use the top-20 items on the ranklist and top-5 aspects per (u, i) pair for AAO, top-1 KG path per (u, i) pair for nGLD. We repeat each experiment three times and report the mean values.

7.1.5 Implementation Details. For implementation, we build EFM, A2CF, PGPR, CAFE, HAM, SAM, and KPRN using PyTorch. We train all the models using Adam [43] and SGD [64]. We set

the latent factor size to 128 for all models. We search the learning rate between 1e-1 and 1e-6, L2-regularization between 1e-1 and 1e-4, and the number of negative samples in Equation (15) between 2 and 10. We search the temperature hyperparameter τ for knowledge-distillation-style training and vanilla training separately.

7.2 Comparison of Faithfulness

We report the AAO and nGLD results in Table 3. We find (1) when using EFM as blackbox recommendation model, all whitebox explanation models have better AAO compared to Random. (2) Compared to PRINCE and CountER, the vanilla training approach generally are not able to generate more faithful explanations. For example, in the Electronics dataset, when using EFM as blackbox recommendation model, PRINCE has an AAO of 0.072 and CountER has an AAO of 0.087 compared to HAM-Vanilla's 0.074. The same pattern applies for A2CF. (3) Our knowledge-distillation-style training strategy could enable the whitebox explanation model to generate more faithful explanations compared to the other model-agnostic baselines and the whitebox explanation model with vanilla training strategy. For aspect-based models, HAM-EFM, SAM-EFM, HAM-A2CF, and SAM-A2CF all outperform their vanilla versions and the best counterfactual baselines. For example, compared to the vanilla versions, SAM-A2CF has 0.089 improvement in Electronics, 0.062 in Office, and 0.058 in Tool & Home, and HAM-A2CF has 0.222, 0.120, and 0.119, respectively. Compared with CountER, SAM-A2CF's improvement is 0.041, 0.069, and 0.032 on three datasets, respectively. And the improvement is 0.257, 0.099, and 0.080 for HAM-A2CF. When use knowledge graph-based models as the blackbox model, with the proposed training strategy, the improvement in terms of nGLD is also significant. For example, KPRN-PGPR outperforms best baseline CountER by 0.026, 0.024, and 0.027 in three Amazon datasets, respectively, and outperforms best baseline PRINCE by 0.023 in Last-FM. And the improvement is 0.042, 0.034, and 0.036 over KPRN-Vanilla in three Amazon datasets and 0.028 in Last-FM dataset; the same pattern also applies for KPRN-CAFE and KPRN-Vanilla.

To summarize, the proposed model-agnostic explanation framework can provide more faithful explanations compared to the existing model-agnostic algorithms and our model ablations. The improvement is consistent on both genres of models, suggesting our framework's universality under the model-agnostic and data-type-agnostic settings.

We go further to discuss the evaluation of faithfulness in model-agnostic explanation paradigm. Lakkaraju et al. [45] discuss the definition of faithfulness in the problem of decision model and propose to evaluate the level of faithfulness by the overlap of decision rules set used by the two models. Works in CV and NLP [35, 36, 42, 51, 62, 83] also propose to measure faithfulness by the overlap of important inputs determined by the two models, e.g., important input text tokens and pixels. By definition, the evaluation of faithfulness can be flexible and model dependent, as long as it measures the overlap of inputs or important intermediate features shared by both the blackbox recommendation model and the whitebox explanation model. Therefore, the proposed evaluation pipeline could be easily adapted and extended to other genres of explainable recommendation models.

7.3 Scrutability Analysis

Scrutability is an ideal property for industrial-level RS. Suppose one user purchased the item being recommended by the blackbox recommendation model and was not satisfied with it afterwards. In this case, the blackbox recommendation model generates one failure case. We want to analyze the potential reason behind this failed case with the help of the agnostic explanation model and potentially find methods to fix the problem for the blackbox recommendation model.

In our scrutability analysis, for each failure case in blackbox recommendation model, we use the whitebox explanation model to determine the top-1 most important aspects/KG-path used in

Table 3. The Evaluation of Faithfulness

Datasets		Electronics	Office Product	Tool & Home	Last FM
Aspect-based Models (measured in AAO)					
EFM	Random	0.024	0.011	0.016	—
	PRINCE	0.072	**0.105**	0.084	—
	CountER	**0.087**	0.097	**0.099**	—
	SAM-Vanilla	0.057	0.034	0.029	—
	SAM-EFM	**0.119**	**0.066**	**0.053**	—
	HAM-Vanilla	0.074	0.089	0.058	—
	HAM-EFM	**0.175**†	**0.152**†	**0.141**†	—
A2CF	Random	0.026	0.011	0.015	—
	PRINCE	0.085	0.097	0.102	—
	CountER	**0.112**	**0.145**	**0.136**	—
	SAM-Vanilla	0.064	0.152	0.110	—
	SAM-A2CF	**0.153**	**0.214**	**0.168**	—
	HAM-Vanilla	0.147	0.124	0.097	—
	HAM-A2CF	**0.369**†	**0.244**†	**0.216**†	—
KG-based Models (measured in nGLD)					
PGPR	Random	0.416	0.445	0.431	0.602
	PRINCE	0.439	0.455	0.459	**0.622**
	CountER	**0.443**	**0.469**	**0.462**	0.619
	KPRN-Vanilla	0.427	0.459	0.453	0.617
	KPRN-PGPR	**0.469**†	**0.493**†	**0.489**†	**0.645**†
CAFE	Random	0.405	0.437	0.425	0.597
	PRINCE	0.445	0.459	0.455	**0.612**
	CountER	**0.462**	**0.471**	**0.460**	0.605
	KPRN-Vanilla	0.450	0.469	0.466	0.622
	KPRN-CAFE	**0.482**†	**0.499**†	**0.487**†	**0.657**†

We use the AAO metric and nGLD metric for aspect-based model and KG-based model, respectively. We highlight the whitebox explanation model with best performance within each sub-category. We conduct paired t-test at 0.05 level and mark the whitebox explanation model that are significantly better than other models with the same blackbox recommendation model with †. The left-most column is the blackbox recommendation model while the second column from the left is the whitebox explanation model.

recommending this i to u. After that, we manually *set the aspect/KG-path embedding vector to 0s for* (u, i)[11] and re-run the blackbox recommendation model to generate a new ranklist for u. We are interested to know that whether the items in failure cases will be ranked lower after masking the aspect. We show the algorithm we use for quantitative scrutability analysis in Algorithm 1 and report the APC result in Table 4.

We find that (1) for aspect-based models, the proposed training strategy enables the whitebox explanation model to have significantly higher APC compared to the vanilla strategy. For example, in the Electronics dataset, HAM-A2CF has 1411.41 APC compared to HAM-Vanilla's 457.60, and the number is 45.59 compared to 2.72 for SAM-A2CF and SAM-Vanilla. (2) For aspect-based models, the HAM model has higher APC compared with SAM. For example, in the Electronics dataset, HAM-A2CF has APC of 1411.41 compared to SAM-A2CF's 45.59. The reason is the nature

[11]We refer to this step as "mask the aspect/path for (u, i)."

Table 4. The Quantitative Scrutability Analysis

Datasets		Electronics	Office Product	Tool & Home	Last FM
Aspect-based Models					
EFM	PRINCE	244.82	19.77	34.40	—
	CountER	325.85	23.52	54.19	—
	SAM-Vanilla	2.14	1.42	1.35	—
	SAM-EFM	35.62	12.94	25.27	—
	HAM-Vanilla	386.12	6.65	43.25	—
	HAM-EFM	**1068.43**†	**43.22**†	**124.06**†	—
A2CF	PRINCE	295.80	22.85	41.27	—
	CountER	396.89	26.41	65.28	—
	SAM-Vanilla	2.72	1.47	1.85	—
	SAM-A2CF	45.59	29.53	41.34	—
	HAM-Vanilla	457.60	8.67	63.47	—
	HAM-A2CF	**1411.41**†	**59.52**†	**197.23**†	—
Knowledge graph-based Models					
PGPR	PRINCE	284.71	19.58	37.79	114.28
	CountER	342.90	23.18	54.70	75.92
	KPRN-Vanilla	455.39	60.97	82.37	89.37
	KPRN-PGPR	**1352.49**†	**62.55**†	**205.39**†	**175.26**†
CAFE	PRINCE	287.55	19.91	38.62	132.74
	CountER	352.45	26.82	62.03	77.95
	KPRN-Vanilla	397.19	63.72	87.44	115.84
	KPRN-PGPR	**1178.35**†	**69.24**†	**192.73**†	**186.40**†

We show the result for APC defined in Equation (14). Being positive means the item is ranked lower on the ranklist. For each blackbox recommendation model, we highlight the whitebox explanation model with highest APC. We use † to indicate the model's improvement over other models is significant in terms of paired t-test at the 0.05 level. The leftmost column is the blackbox recommendation model while the second column from the left is the whitebox explanation model.

ALGORITHM 1: Workflow of Scrutability Analysis

1: Train the blackbox recommendation model on the trainset.
2: Use blackbox recommendation model's output as supervision signals to train the whitebox explanation model.
3: For user u, let blackbox recommendation model generate a ranklist.
4: Manually set the non-ground-truth items at **top-3** of the ranklist as failed items.
5: Use the whitebox explanation model to find the **top-1** most important aspects/path for each failure case (u, i).
6: Re-run blackbox recommendation model to generate a ranklist for u again, specifically, for each failure case (u, i), **mask** the corresponding aspect/path during blackbox's inference.
7: Compute the **average position change** of failure items.

of "hard-match" make impact of the removal of aspect significant. (3) Compared to PRINCE and CountER, HAM-A2CF and HAM-EFM have significantly higher APC, which suggests the effectiveness of the proposed training strategy. However, SAM-A2CF and SAM-EFM fail to outperform PRINCE and CountER. The reason is PRINCE and CountER are still "hard-match" model by design, and the removal of important aspects still have significant impact for the item's

Table 5. The Result for Qualitative Analysis of Correct Cases

Aspect-based Models						
Datasets	User	Target Failure Item	Item Type	Masked Aspect	Original Rank	New Rank
Electronics	A12GKGLR2L4MZY	B002R5AM7C	Video Camera	Sound	1	1332
	AVSD11L7NZG3I	B003YH9EZ8	Earbud	Size	3	34
	A2BYV7S1QP2YIG	B004MSQZUU	Headphone	Price	3	67
Office Product	A3BTL4FV60DKAT	B005HFJFK4	Inkjet Printer	Network	2	21
	A1HFT68GJ42LTM	B002K9IHJK	Removable Label Pad	Tape	2	15
	AEL6CQNQXONBX	B00CPXDK2U	View Binder	Price	2	7
Tool & Home	A1MC6BFHWY6WC3	B005NXPSTM	LED Penlight	Batteries	2	8
	A3KPJ1MOGTZVGC	B00B8BPAHQ	Floor Lamp	Bulb	3	11
	A1KJO5VP4K3CHU	B00GY71PT8	Lantern Flashlight	Weight	3	8
Knowledge graph-based Models						
Datasets	User	Target Failure Item	Masked KG Path		Original Rank	New Rank
Last FM	user_2426	product_789	user_0 $\xrightarrow{listened}$ product_3648 $\xrightarrow{rev\_listened}$ user_1945 $\xrightarrow{listened}$ product_789		1	525
	user_837	product_985	user_837 $\xrightarrow{listened}$ product_7490 $\xrightarrow{belongs\_to}$ category_0 $\xrightarrow{rev\_belongs\_to}$ product_837		2	1091
	user_405	product_1300	user_405 $\xrightarrow{listened}$ product_3536 $\xrightarrow{sang\_by}$ artist_907 $\xrightarrow{rev\_sang\_by}$ product_1300		2	927

Within each dataset, we show three cases where the failed items are ranked significantly lower after masking the *top-1* important aspect/knowledge graph path for the failed user–item pair. The results of aspect-based models are generated with A2CF as blackbox recommendation model and HAM as whitebox explanation model, and the results of knowledge graph-based models are generated with PGPR as blackbox recommendation model and KPRN as whitebox explanation model.

Table 6. The Result for Qualitative Analysis of Cases Where the Proposed Framework Fails

Aspect-based Models						
Datasets	User	Target Failure Item	Item Type	Masked Aspect	Original Rank	New Rank
Electronics	A1EARN5PUVIF1S	B002R5AM7C	Video Camera	Feature	1	2
	A2BVOBG7YDSVOZ	B003YH9EZ8	Earbud	Manual	2	2
Office Product	ABDR6IJ93HFIO	B005HFJFK4	Inkjet Printer	Price	1	1
	A1V4VVBQBFXRHC	B002K9IHJK	Removable Label Pad	Plastic	2	3
Tool & Home	A2MSBIA18RXYQC	B005NXPSTM	LED Penlight	Output	1	2
	A55PCTJ6NINET	B005Z29U6S	LED Penlight	Output	1	2
Knowledge graph-based Models						
Datasets	User	Target Failure Item	Masked KG Path		Original Rank	New Rank
Last FM	user_612	product_16	user_612 $\xrightarrow{listened}$ product_379 $\xrightarrow{rev\_listened}$ user_492 $\xrightarrow{listened}$ product_16		1	1
	user_4465	product_95	user_4465 $\xrightarrow{listened}$ product_126 $\xrightarrow{rev\_listened}$ user_2294 $\xrightarrow{listened}$ product_95		1	1

Within each dataset, we show two cases where the failed items are ranked *not significantly* lower after masking the *top-1* important aspect/knowledge graph path for the failed user–item pair. The results of aspect-based models are generated with A2CF as blackbox recommendation model and HAM as whitebox explanation model; and the results of knowledge graph-based models are generated with PGPR as blackbox recommendation model and KPRN as whitebox explanation model.

position on the ranklist. (4) The result for KG-based models is consistant with the aspect-based models. We observe that both KPRN-PGPR and KPRN-CAFE has the highest APC compared to the baselines and its ablation. Interestingly, we find that KPRN-Vanilla outperforms baselines in three Amazon datasets, but is outperformed PRINCE in Last FM datasets. This might be because of the different path patterns and distributions between the Amazon and Last FM dataset.

We also present qualitative studies to help readers understand how the proposed framework achieves system scrutability (Table 5) and its limitation (Table 6). From Table 5, we can see that when the whitebox explanation model is able to extract meaningful aspects for (u, i) pair, the blackbox recommendation model could further effectively rank the failed items to the lower part of the ranklist after masking those aspects. For example, item B002R5AM7C is ranked lower after masking the aspect Sound. Similar pattern holds for the knowledge graph-based models; for example, for $(user\_2426, product\_789)$, the rank of $product\_789$ drops from 1 to 525 after setting the embedding vector of $product\_3648$ and $user\_1945$ to 0s. However, this also suggests that our whitebox explanation model effectively distills the knowledge learned by the blackbox recommendation model and achieves better faithfulness.

We can also observe some limitations of our framework from Table 6. (1) When the whitebox explanation model is not able to find the important features, i.e., aspects or knowledge graph paths, removing these features makes hardly any effect on the ranks of the target items. For example, item

B002K9IHJK is not ranked lower when the masked aspect is `Plastic`. This suggests our framework can be improved by designing better knowledge distillation training strategy to train the whitebox explanation model. (2) Whether our framework can achieve scrutability is also limited by the features set used by the recommendation and the explanation models and is related to the patterns of the dataset itself. For example, item B005NXPSTM is not ranked lower when the masked aspect is `Output`, and `Output` is an ambiguous aspect. For knowledge graph-based models, by manually examining the cases where items are not ranked lower, we find that most of the paths belongs to the metapath $user\_a \xrightarrow{listeded} product\_b \xrightarrow{rev\_listened} user\_c \xrightarrow{listened} product\_d$, which is the most common metapath on the dense Last FM dataset. Between one (u, i) pair, there might be hundred of such paths; therefore, removing one of them does not make the item being ranked lower. Thus suggests that when applying the proposed framework, we need to take dataset-specific factors into consideration.

The scrutability of our framework is potentially helpful from two perspectives: (1) Using the agnostic model could help us understand how blackbox recommendation model works. In our implementation, the whitebox explanation model helps us better profile user preferences as well as item features and conduct more personalized recommendations. (2) With the information extracted by the whitebox explanation model, we can effectively control the results of an originally not-controllable blackbox recommendation model. This is particularly important for debugging production models online and could potentially lead to new algorithms and techniques for the study of negative feedback and conversational recommendation with non-explainable recommendation models.

7.4 RS Performance and Decision Pattern

Since our whitebox explanation model is also able to generate recommendation decisions by itself, we additionally include a subsection to discuss its performance. Also, *fidelity* is defined as the extent to which the explanation model could mimic the decision model's decision pattern [35, 58, 70], i.e., the similarity of ranklists or the overlap of recommended items given by the two models. We show the comparison of recommendation performance (evaluated by *hit rate@20* and *nDCG@20*) in Table 7 and fidelity in Table 8.

7.4.1 Recommendation Performance. From Table 7, we find that (1) EFM, A2CF, PGPR, and CAFE can deliver good recommendation performance, and thus it is suitable to use them as blackbox recommendation models. (2) Compared to the vanilla versions, the whitebox explanation models trained with the proposed knowledge-distillation-style training strategy can also deliver competitive performance, even if no ground-truth labels are used in training. For example, in the Electronics dataset, KPRN-PGPR outperforms both vanilla PGPR and KPRN in terms of hit rate and nDCG, and KPRN-CAFE outperforms vanilla CAFE and KPRN as well. This finding is consistent with recent knowledge-distillation papers [20, 59]. It may suggest we could utilize the same training strategy to design better model-intrinsic explainable recommendation models.

7.4.2 Fidelity of Explanation Model. We evaluate the fidelity from both the item level and ranklist level. Similarly to previous works [58], we measure the item-level fidelity by the overlap of recommended items given by blackbox recommendation model and whitebox explanation model via **Average Item Overlap (AIO)**,

$$AIO = \frac{\sum_{u \in \mathcal{U}} |\mathcal{I}_u@n \bigcap \mathcal{I}'_u@n|}{|\mathcal{U}|}, \tag{16}$$

where $\mathcal{I}_u@n$ and $\mathcal{I}_u@n'$ are the blackbox recommendation model's and whitebox explanation model's top-n recommended items for user u, respectively, and $|\cdot|$ denotes the size of set. In

Table 7. The Recommndation Performance (Hit Rate@20, nDCG@20) Measured in %

Datasets	Electronics		Office Product		Tool & Home		Last FM	
Models / Metrics	HR	nDCG	HR	nDCG	HR	nDCG	HR	nDCG
Random	0.74	0.09	0.89	0.13	7.24	1.92	—	—
EFM	5.54	1.25	7.26	1.68	16.28	4.71	—	—
A2CF	**7.03**	1.52	7.61	1.55	**27.95**	**7.08**		
PGPR	6.11	1.58	7.92	2.05	15.51	4.95	**43.19**	**5.15**
CAFE	6.57	**1.64**	**8.11**	**2.12**	16.62	5.13	41.74	4.82
SAM-Vanilla	5.21	1.06	5.59	1.10	**17.66**	**5.28**	—	—
HAM-Vanilla	5.15	1.15	3.96	0.77	15.20	4.49	—	—
KPRN-Vanilla	**6.72**	**1.59**	**8.69**	**2.32**	13.67	3.96	39.92	4.45
SAM-EFM	4.45	1.20	5.59	1.24	15.36	3.94	—	—
HAM-EFM	4.88	1.09	7.01	1.59	13.97	3.77	—	—
SAM-A2CF	6.65	1.36	6.45	1.18	**26.96**	**7.08**	—	—
HAM-A2CF	4.95	1.25	7.06	1.47	17.05	4.54	—	—
KPRN-PGPR	7.19	1.71	8.57	2.36	14.29	4.18	40.72	4.52
KPRN-CAFE	**7.32**	**1.75**	**8.72**	**2.42**	15.52	4.33	41.44	4.62

We highlight the model with best Performance within each sub-category. Note we do not include CountER and PRINCE, because they cannot generate recommendation decisions.

Table 8. The Evaluation of Fidelity

Datasets		Electronics		Office Product		Tool & Home		Last FM	
Models / Metrics		ρ	AIO	ρ	AIO	ρ	AIO	ρ	AIO
EFM	Random	0.05	0.14	0.03	0.33	0.02	0.44	—	—
	SAM-Vanilla	0.59	8.75	0.59	1.92	0.33	6.65	—	—
	SAM-EFM	**0.85**	**11.52**	**0.64**	**5.57**	**0.44**	**8.83**	—	—
	HAM-Vanilla	0.46	2.64	0.31	1.57	0.35	3.66	—	—
	HAM-EFM	**0.63**	**2.75**	**0.34**	**3.16**	**0.43**	**5.18**	—	—
A2CF	Random	0.01	0.16	0.02	0.27	0.05	0.39	—	—
	SAM-Vanilla	0.28	4.85	0.61	1.92	0.26	6.65	—	—
	SAM-A2CF	**0.95**	**14.52**	**0.94**	**14.77**	**0.27**	**9.53**	—	—
	HAM-Vanilla	0.22	2.15	0.21	1.46	0.26	3.35	—	—
	HAM-A2CF	**0.29**	**2.36**	**0.39**	**4.60**	**0.34**	**6.78**	—	—
PGPR	Random	0.02	0.17	0.03	0.35	0.04	0.45	0.03	0.18
	KPRN-Vanilla	0.31	3.56	0.30	2.95	0.34	4.72	0.53	7.42
	KPRN-PGPR	**0.45**	**4.72**	**0.38**	**3.70**	**0.45**	**6.69**	**0.60**	**9.04**
CAFE	Random	0.03	0.15	0.03	0.34	0.04	0.45	0.04	0.20
	KPRN-Vanilla	0.33	3.72	0.35	3.12	0.33	4.62	**0.55**	**7.76**
	KPRN-CAFE	**0.47**	**4.85**	**0.39**	**3.85**	**0.42**	**6.59**	**0.64**	**9.39**

We use Spearman Rank Correlation Coefficient (Spearman's ρ) and AIO. We highlight the whitebox explanation model that is significantly better its vanilla version by paired t-test at 0.05 level. The left-most column is the blackbox recommendation model the whitebox explanation model compares the ranklist to. The second column from the left is the whitebox explanation model.

practice, we calculate the overlap between the top-20 items on the ranklists for each user. The Item Overlap is calculated for each user and is averaged to derive AIO. We also evaluate the ranklist-level fidelity by the similarity between the two whole ranklists via *Spearman Rank Coefficient* ρ [40, 41].

For a sample of size n, the n raw scores X_i, Y_i are converted to ranks $R(X_i)$ and $R(Y_i)$, and Spearman's Rank Correlation ρ is computed by

$$\rho = \frac{cov(R(X), R(Y))}{\sigma_{R(X)}\sigma_{R(Y)}} \in [-1, 1], \tag{17}$$

where $cov(R(X), R(Y))$ is the covariance of the rank variables, $\sigma_{R(X)}$ and $\sigma_{R(Y)}$ are the standard deviations of the rank variables, and we have

$$\begin{aligned} cov(R(X), R(Y)) &= E[(X - E[X])(Y - E[Y])] \\ &= E[XY] - E[X]E[Y]. \end{aligned} \tag{18}$$

Specifically, in our implementation, there are no tied ranks of the items, so we compute ρ as

$$\rho = 1 - \frac{6\sum_{i \in \mathcal{I}} d_i^2}{n(n^2 - 1)}, \tag{19}$$

where d_i is the difference of item i's rank between two ranklists and $n = |\mathcal{I}|$. $\rho = -1$ means the two ranks are perfect negative correlation, while $\rho = 1$ means perfect positive correlation; $\rho = 0$ means no correlation between two ranks.

We report the results for fidelity in Table 8. We find the following: (1) within the scope of aspect-based models, SAM has better fidelity compared to HAM model. This might be because the "hard-match" nature of HAM limits the candidate items so that it cannot learn the blackbox recommendation model's decision pattern well; (2) for both aspect-based models and KG-based models, the whitebox explanation model with the proposed training strategy has signiciantly better Spearman Rank Coefficient as well as AIO. To summarize, the proposed training strategy enables the whitebox explanation model to generate more similar recommendation decisions to the blackbox recommendation model, compared to its vanilla versions.

8 CONCLUSION AND FUTURE WORK

In this work, we propose a reusable evaluation pipeline for model-agnostic explainable recommendation. Our pipeline focuses on faithfulness and scrutability, which have not been the focus of previous model-agnostic explainable recommendation literature. We further propose a model-agnostic explanation framework with a knowledge-distillation-style training strategy. Extensive qualitative and quantitative studies demonstrate that our explanation framework could enhance the faithfulness of model-agnostic explanations and the recommender system scrutability. Our framework could be potentially used in other IR tasks such as negative feedback, conversational recommendation/product search and dense retrieval, which will be the focus of our future work.

ACKNOWLEDGMENTS

We thank the reviewers for their valuable comments and suggestions.

REFERENCES

[1] Charu C. Aggarwal et al. 2016. *Recommender Systems*. Vol. 1. Springer.
[2] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms* 11, 9 (2018), 137.
[3] Qingyao Ai and Lakshmi Narayanan Ramasamy. 2021. Model-agnostic vs. Model-intrinsic interpretability for explainable product search. arXiv:2108.05317. Retrieved from https://arxiv.org/abs/2108.05217.
[4] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2017. Towards better understanding of gradient-based attribution methods for deep neural networks. arXiv:1711.06104. Retrieved from https://arxiv.org/abs/1711.06104.
[5] Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A diagnostic study of explainability techniques for text classification. arXiv:2009.13295. Retrieved from https://arxiv.org/abs/2009.13295.

[6] Giacomo Balloccu, Ludovico Boratto, Gianni Fenu, and Mirko Marras. 2022. Hands on explainable recommender systems with knowledge graphs. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 710–713.
[7] Krisztian Balog and Filip Radlinski. 2020. Measuring recommendation explanation quality: The conflicting goals of explanations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 329–338.
[8] Krisztian Balog, Filip Radlinski, and Shushan Arakelyan. 2019. Transparent, scrutable and explainable user models for personalized recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 265–274.
[9] Osbert Bastani, Carolyn Kim, and Hamsa Bastani. 2017. Interpretability via model extraction. arXiv:1706.09773. Retrieved from https://arxiv.org/abs/1706.09773.
[10] Konstantin Bauman, Bing Liu, and Alexander Tuzhilin. 2017. Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 717–725.
[11] James Bennett, Stan Lanning, et al. 2007. The netflix prize. In *Proceedings of KDD Cup and Workshop*, Vol. 2007. 35.
[12] Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User Model. User-adapt. Interact.* 12, 4 (2002), 331–370.
[13] Oana-Maria Camburu, Eleonora Giunchiglia, Jakob Foerster, Thomas Lukasiewicz, and Phil Blunsom. 2019. Can I trust the explainer? Verifying post-hoc explanatory methods. arXiv:1910.02065. Retrieved from https://arxiv.org/abs/1910.02065.
[14] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural attentional rating regression with review-level explanations. In *Proceedings of the World Wide Web Conference*. 1583–1592.
[15] Tong Chen, Hongzhi Yin, Guanhua Ye, Zi Huang, Yang Wang, and Meng Wang. 2020. Try this instead: Personalized and interpretable substitute recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 891–900.
[16] Henriette Cramer, Vanessa Evers, Satyan Ramlal, Maarten Van Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo, and Bob Wielinga. 2008. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Model. User-adapt. Interact.* 18, 5 (2008), 455–496.
[17] Yashar Deldjoo, Tommaso Di Noia, and Felice Antonio Merra. 2021. A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks. *ACM Comput. Surv.* 54, 2 (2021), 1–38.
[18] Mengnan Du, Ninghao Liu, and Xia Hu. 2019. Techniques for interpretable machine learning. *Commun. ACM* 63, 1 (2019), 68–77.
[19] Glenn Fung, Sathyakama Sandilya, and R Bharat Rao. 2008. Rule extraction from linear support vector machines via mathematical programming. In *Rule Extraction from Support Vector Machines*. Springer, 83–107.
[20] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. In *International Conference on Machine Learning*. PMLR, 1607–1616.
[21] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. 2014. How should I explain? A comparison of different explanation types for recommender systems. *Int. J. Hum.-Comput. Stud.* 72, 4 (2014), 367–382.
[22] Shijie Geng, Zuohui Fu, Juntao Tan, Yingqiang Ge, Gerard De Melo, and Yongfeng Zhang. 2022. Path language modeling over knowledge graphsfor explainable recommendation. In *Proceedings of the ACM Web Conference 2022*. 946–955.
[23] Azin Ghazimatin, Oana Balalau, Rishiraj Saha Roy, and Gerhard Weikum. 2020. PRINCE: Provider-side interpretability with counterfactual explanations in recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 196–204.
[24] Azin Ghazimatin, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. 2021. ELIXIR: Learning from user feedback on explanations to improve recommender models. In *Proceedings of the Web Conference 2021*. 3850–3860.
[25] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *Proceedings of the IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA'18)*. IEEE, 80–89.
[26] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *Int. J. Comput. Vis.* 129, 6 (2021), 1789–1819.
[27] Stephen J Green, Paul Lamere, Jeffrey Alexander, François Maillet, Susanna Kirk, Jessica Holt, Jackie Bourque, and Xiao-Wen Mak. 2009. Generating transparent, steerable recommendations from textual descriptions of items. In *Proceedings of the 3rd ACM Conference on Recommender Systems*. 281–284.
[28] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2018. Local rule-based explanations of black box decision systems. arXiv:1805.10820. Retrieved from https://arxiv.org/abs/1805.10820.
[29] Leo A. Harrington, Michael D. Morley, A. Šcedrov, and Stephen G. Simpson. 1985. *Harvey Friedman's Research on the Foundations of Mathematics*. Elsevier.

[30] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1661–1670.

[31] Bernease Herman. 2017. The promise and peril of human evaluation for model interpretability. arXiv:1711.07414 (2017). Retrieved from https://arxiv.org/abs/1711.07414.

[32] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. arXiv:1503.02531. Retrieved from https://arxiv.org/abs/1503.02531.

[33] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. arXiv:2010.02666. Retrieved from https://arxiv.org/abs/2010.02666.

[34] Yidan Hu, Yong Liu, Chunyan Miao, Gongqi Lin, and Yuan Miao. 2022. Aspect-guided syntax graph learning for explainable recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2022), 1–14. DOI: https://doi.org/10.1109/TKDE.2022.3221847

[35] Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? arXiv:2004.03685. Retrieved from https://arxiv.org/abs/2004.03685.

[36] Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. arXiv:1902.10186. Retrieved from https://arxiv.org/abs/1902.10186.

[37] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *J. Artif. Intell. Res.* 4 (1996), 237–285.

[38] Vassilis Kaffes, Dimitris Sacharidis, and Giorgos Giannopoulos. 2021. Model-agnostic counterfactual explanations of recommendations. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*. 280–285.

[39] Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. 2020. Model-agnostic counterfactual explanations for consequential decisions. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 895–905.

[40] Maurice G. Kendall. 1938. A new measure of rank correlation. *Biometrika* 30, 1/2 (1938), 81–93.

[41] Maurice G. Kendall et al. 1948. The advanced theory of statistics. Vols. 1. *The Advanced Theory of Statistics. Vols. 1* (1948).

[42] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*. PMLR, 2668–2677.

[43] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*, Yoshua Bengio and Yann LeCun (Eds.). http://dblp.uni-trier.de/db/conf/iclr/iclr2015.html#KingmaB14.

[44] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

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

[46] Lei Li, Yongfeng Zhang, and Li Chen. 2020. Generate neural template explanations for recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 755–764.

[47] Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. 2020. Connecting the dots: Event graph schema induction with path language modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. 684–695.

[48] Sen Li, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, Xiao-Ming Wu, and Qianli Ma. 2021. Embedding-based product retrieval in taobao search. arXiv:2106.09297. Retrieved from https://arxiv.org/abs/2103.09297.

[49] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A general knowledge distillation framework for counterfactual recommendation via uniform data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 831–840.

[50] Andreas Madsen, Siva Reddy, and Sarath Chandar. 2021. Post-hoc interpretability for neural NLP: A survey. arXiv:2108.04840. Retrieved from https://arxiv.org/abs/2108.04840.

[51] Ana Marasović, Chandra Bhagavatula, Jae Sung Park, Ronan Le Bras, Noah A Smith, and Yejin Choi. 2020. Natural language rationales with full-stack visual reasoning: From pixels to semantic frames to commonsense graphs. arXiv:2010.07526. Retrieved from https://arxiv.org/abs/2010.07526.

[52] Andres Marzal and Enrique Vidal. 1993. Computation of normalized edit distance and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* 15, 9 (1993), 926–932.

[53] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*. 165–172.

[54] Caio Nóbrega and Leandro Marinho. 2019. Towards explaining recommendations through local surrogate models. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. 1671–1678.

[55] Ingrid Nunes and Dietmar Jannach. 2017. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Model. User-Adapt. Interact.* 27, 3 (2017), 393–444.
[56] Aditya Pal, Chantat Eksombatchai, Yitong Zhou, Bo Zhao, Charles Rosenberg, and Jure Leskovec. 2020. PinnerSage: Multi-modal user embedding framework for recommendations at pinterest. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2311–2320.
[57] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22, 10 (2009), 1345–1359.
[58] Georgina Peake and Jun Wang. 2018. Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2060–2069.
[59] Zhen Qin, Le Yan, Yi Tay, Honglei Zhuang, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2021. Born again neural rankers. arXiv:2109.15285. Retrieved from https://arxiv.org/abs/2109.15285.
[60] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv:1205.2618. Retrieved from https://arxiv.org/abs/1205.2618.
[61] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1135–1144.
[62] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
[63] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender Systems Handbook*. Springer, 1–35.
[64] Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *Ann. Math. Stat.* (1951), 400–407.
[65] Alexis Ross, Ana Marasović, and Matthew E. Peters. 2020. Explaining nlp models via minimal contrastive editing (mice). arXiv:2012.13985. Retrieved from https://arxiv.org/abs/2012.13985.
[66] Cynthia Rudin. 2018. Please stop explaining black box models for high stakes decisions. *Stat* 1050 (2018), 26.
[67] Ivan Sanchez, Tim Rocktaschel, Sebastian Riedel, and Sameer Singh. 2015. Towards extracting faithful and descriptive representations of latent variable models. In *AAAI Spring Symposium on Knowledge Representation and Reasoning (KRR): Integrating Symbolic and Neural Approaches*, 4–1.
[68] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv:1910.01108. Retrieved from https://arxiv.org/abs/1910.01108.
[69] J. Ben Schafer, Joseph Konstan, and John Riedl. 1999. Recommender systems in e-commerce. In *Proceedings of the 1st ACM Conference on Electronic Commerce*. 158–166.
[70] Jaspreet Singh and Avishek Anand. 2020. Model agnostic interpretability of rankers via intent modelling. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 618–628.
[71] Harald Steck and Dawen Liang. 2021. Negative interactions for improved collaborative filtering: Don't go deeper, go higher. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 34–43.
[72] Juntao Tan, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. 2021. Counterfactual explainable recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1784–1793.
[73] Jiaxi Tang and Ke Wang. 2018. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2289–2298.
[74] Maartje ter Hoeve, Anne Schuth, Daan Odijk, and Maarten de Rijke. 2018. Faithfully explaining rankings in a news recommender system. arXiv:1805.05447. Retrieved from https://arxiv.org/abs/1805.05447.
[75] Nava Tintarev and Judith Masthoff. 2007. A survey of explanations in recommender systems. In *Proceedings of the IEEE 23rd International Conference on Data Engineering Workshop*. IEEE, 801–810.
[76] Nava Tintarev and Judith Masthoff. 2015. Explaining recommendations: Design and evaluation. In *Recommender Systems Handbook*. Springer, 353–382.
[77] Khanh Hiep Tran, Azin Ghazimatin, and Rishiraj Saha Roy. 2021. Counterfactual explanations for neural recommenders. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1627–1631.
[78] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. J. Law Technol.* 31 (2017), 841.
[79] 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 *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 417–426.

[80] Nan Wang, Hongning Wang, Yiling Jia, and Yue Yin. 2018. Explainable recommendation via multi-task learning in opinionated text data. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 165–174.

[81] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5329–5336.

[82] Sarah Wiegreffe and Ana Marasović. 2021. Teach me to explain: A review of datasets for explainable nlp. arXiv: 2102.12060. Retrieved from https://arxiv.org/abs/2102.12060.

[83] Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. arXiv:1908.04626. Retrieved from https://arxiv.org/abs/1908.04626.

[84] Yikun Xian, Zuohui Fu, Shan Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 285–294.

[85] 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 *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1645–1654.

[86] Zhichao Xu and Daniel Cohen. 2023. A lightweight constrained generation alternative for query-focused summarization. arXiv:2304.11721. Retrieved from https://arxiv.org/abs/2304.11721.

[87] Zhichao Xu, Yi Han, Tao Yang, Anh Tran, and Qingyao Ai. 2022. Learning to rank rationales for explainable recommendation. arXiv:2206.05368. Retrieved from https://arxiv.org/abs/2206.05368.

[88] Zhichao Xu, Yi Han, Yongfeng Zhang, and Qingyao Ai. 2020. E-commerce recommendation with weighted expected utility. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1695–1704.

[89] Zhichao Xu, Hemank Lamba, Qingyao Ai, Joel Tetreault, and Alex Jaimes. 2023. Counterfactual editing for search result explanation. arXiv:2301.10389. Retrieved from https://arxiv.org/abs/2301.10389.

[90] Zhichao Xu, Hansi Zeng, and Qingyao Ai. 2021. Understanding the effectiveness of reviews in e-commerce top-n recommendation. arXiv:2106.09665. Retrieved from https://arxiv.org/abs/2106.29665.

[91] Tao Yang, Zhichao Xu, and Qingyao Ai. 2022. Effective exposure amortizing for fair top-k recommendation. arXiv:2204.03046. Retrieved from https://arxiv.org/abs/2204.03046.

[92] Li Yujian and Liu Bo. 2007. A normalized levenshtein distance metric. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 6 (2007), 1091–1095.

[93] Hansi Zeng, Zhichao Xu, and Qingyao Ai. 2021. A zero attentive relevance matching networkfor review modeling in recommendation system. arXiv:2101.06387 [cs.IR]. Retrieved from https://arxiv.org/abs/2101.06389.

[94] Yongfeng Zhang and Xu Chen. 2018. Explainable recommendation: A survey and new perspectives. arXiv:1804.11192. Retrieved from https://arxiv.org/abs/1804.11192.

[95] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. 83–92.

[96] Yuan Zhang, Xiaoran Xu, Hanning Zhou, and Yan Zhang. 2020. Distilling structured knowledge into embeddings for explainable and accurate recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 735–743.

[97] Yaxin Zhu, Yikun Xian, Zuohui Fu, Gerard de Melo, and Yongfeng Zhang. 2021. Faithfully explainable recommendation via neural logic reasoning. arXiv:2104.07869. Retrieved from https://arxiv.org/abs/2104.07869.

Received 29 July 2022; revised 26 April 2023; accepted 6 June 2023