Recommendation with Causality enhanced Natural Language Explanations

Jingsen Zhang[†]
zhangjingsen@ruc.edu.cn
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China

Xu Chen[†*]
xu.chen@ruc.edu.cn
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China

Jiakai Tang[†]
tangjiakai5704@gmail.com
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China

Weiqi Shao[†]
shaoweiqi@ruc.edu.cn
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China

Quanyu Dai
daiquanyu@huawei.com
Huawei Noah's Ark Lab

Zhenhua Dong
dongzhenhua@huawei.com
Huawei Noah's Ark Lab

Rui Zhang
rayteam@yeah.net
www.ruizhang.info

ABSTRACT

Explainable recommendation has recently attracted increasing attention from both academic and industry communities. Among different explainable strategies, generating natural language explanations is an important method, which can deliver more informative, flexible and readable explanations to facilitate better user decisions. Despite the effectiveness, existing models are mostly optimized based on the observed datasets, which can be skewed due to the selection or exposure bias. To alleviate this problem, in this paper, we formulate the task of explainable recommendation with a causal graph, and design a causality enhanced framework to generate unbiased explanations. More specifically, we firstly define an ideal unbiased learning objective, and then derive a tractable loss for the observational data based on the inverse propensity score (IPS), where the key is a sample re-weighting strategy for equalizing the loss and ideal objective in expectation. Considering that the IPS estimated from the sparse and noisy recommendation datasets can be inaccurate, we introduce a fault tolerant mechanism by minimizing the maximum loss induced by the sample weights near the IPS. For more comprehensive modeling, we further analyze and infer the potential latent confounders induced by the complex and diverse user personalities. We conduct extensive experiments by comparing with the state-of-the-art methods based on three real-world datasets to demonstrate the effectiveness of our method.

CCS CONCEPTS

• **Information systems → Recommender systems**.

∗ Corresponding author.
† Beijing Key Laboratory of Big Data Management and Analysis Methods.

WWW '23, May 1–5, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583260

KEYWORDS

Explainable Recommendation, Natural Language Explanations

ACM Reference Format:
Jingsen Zhang[†], Xu Chen[†*], Jiakai Tang[†], Weiqi Shao[†], Quanyu Dai, Zhenhua Dong, and Rui Zhang. 2023. Recommendation with Causality enhanced Natural Language Explanations. In *Proceedings of the ACM Web Conference 2023 (WWW '23), May 1–5, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3543507.3583260

1 INTRODUCTION

Explainable recommendation has been recognized as an important problem in both of the academic and industry communities. It basically aims to solve the problem of "why an item is recommended to a user?", which can help to enhance the recommendation persuasiveness, user satisfaction and system transparency [36]. To achieve explainable recommendation, people have designed a large amount of models [7, 15, 16, 18, 24, 35, 38], among which the methods for generating natural language explanations are becoming more and more popular due to their capabilities on delivering richer and more accessible information [36]. In general, these methods regard user reviews as the explanations [16, 19], and the task of explainable recommendation is converted to the review generation problem. More specifically, early models like NRT [19] and Att2Seq [9] generate explanations completely based on the user/item IDs (or additionally the rating information). Due to the lack of informative guidance, these explanations usually contain a large amount of general words [16], which are less effective for assisting user decisions. To solve this problem, recent models, such as NETE [16] and PETER [18] firstly extract item features (*e.g.*, product quality, clothing style) from the user reviews, and then regard them as inputs to generate more specific and informative explanations.

While the above models have achieved many successes, they are optimized based on the observational review datasets (*e.g.*, Amazon and Yelp), which may have been biased by the user intrinsic preference or previous recommender systems (RecSys). For example, in Figure 1(a), the user is a digital fan. Thus we observe that most of her interactions and reviews are about items like computers, phones and digital cameras. However, for the other products (*e.g.*, clothes), only a few interactions and reviews can be observed. Based on such

a dataset, the estimated explanations for the clothes can be not well. In the user reviews, we may observe that the feature "item quality" is mentioned with a much higher frequency due to the user reviewing habits. However, providing explanations should not be related with the reviewing habits. The system should have the capability of choosing any feature for explanation. When the system selects appearance or price as the explanation feature, the model may not work well due to the insufficient training corpus. To further verify such item- and feature-level biases, we conduct a preliminary study based on three real-world datasets including the TripAdvisor[1], Amazon[2] and Yelp[3]. We focus on the relation between the observation[4] frequency and the number of items (or features) with this frequency. The results are presented in Figure 1(b). We can see a small number of items (or features) are observed with much higher frequency than the other ones. If one directly learns models based on such datasets, the parameters would be biased towards the frequently observed items/features, and the explanations generated for the underrepresented items/features can be unsatisfied.

To alleviate the above problem, we propose to formulate the explainable recommendation task with a causal graph to understand the bias formation mechanism and design an unbiased explainable recommender framework based on the inverse propensity score (IPS). While this seems to be an interesting idea, it is not easy due to the following challenges: to begin with, while recent years have witnessed many promising unbiased recommender models [27], they are mostly designed for the item-level bias. However, in our problem, the biases come from both of the item- and feature-level. How to jointly correct them in a unified framework is still not clear. Then, the recommendation datasets can by quite sparse and noisy, the estimated IPS may deviate from the real one, how to handle the estimation error to guarantee the final performance needs our careful designs. In addition, user personalities in real-world scenarios can be quite diverse. Thus, there may exist latent confounders, which may invalid the basic causal assumptions. How to model and infer them may also challenge our idea.

To overcome the above challenges, we firstly define an unbiased learning objective considering both of the item- and feature-level biases, and then derive a tractable loss for the observational data based on IPS, where the basic idea is to impose smaller weights to the items/features with higher observational frequencies, and assign larger weights to the long-tail items/features. For handling the prediction error of IPS, we firstly estimate it from the noisy data, and then assume that the real IPS should be not far from the estimated one. At last, we minimize the maximum loss induced by the sample weights near the estimated IPS. For handling the potential latent confounders, we leverage neural networks to model and infer them, and the obtained results are incorporated into the IPS and user preference estimation processes. Based on all the above designs, we finally propose an unbiased explainable recommender framework, where we call it as **USER** for short.

In a summary, the main contributions of this paper can be concluded as follows: (1) we propose to build an unbiased explainable recommender framework based on causal inference, which, to the best of our knowledge, is the first time in the recommendation domain. (2) To achieve the above idea, we design a framework to jointly correct the item- and feature-level biases, where we propose a fault tolerant IPS estimation strategy, and also model and infer the potential latent confounders. (3) We conduct extensive experiments to demonstrate the effectiveness of our model based on three real-world datasets. To benefit the research community, we have released our framework at https://gitee.com/mindspore/models/tree/master/research/recommend/user.

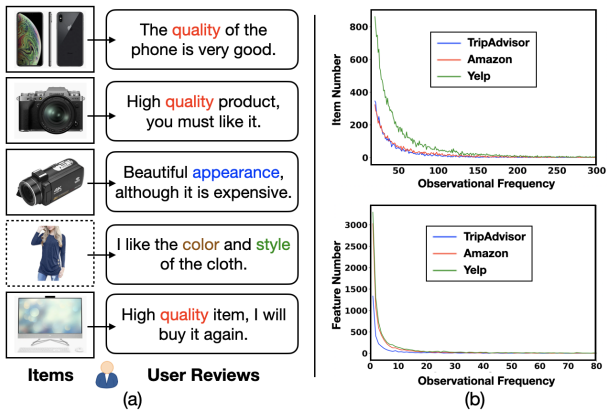


Figure 1: (a) Examples of the biased item interactions and feature mentions. (b) Statistics on the relation between the observational frequency and the number of items (or features) based the datasets of TripAdvisor, Amazon and Yelp.

2 PRELIMINARIES

2.1 RecSys with Natural Language Explanations

Natural language explanations hold the promise of explaining recommendations according to the user preference in a flexible and informative manner. In practice, it is hard to obtain the ground truth of the explanations. Thus, people leverage user reviews, which contain rich user preferences, to approximate the real explanations [8, 16, 18]. Formally, suppose we have a user set $\mathcal{U}$ and an item set $\mathcal{I}$, the observed ratings and reviews[5] from the users to the items are collected in $\mathcal{R} = \{(r_{ui}, \boldsymbol{s}_{ui}) | u \in \mathcal{U}, i \in \mathcal{I}\}$, where r_{ui} is the rating falling into the range of $[1, 5]$. $\boldsymbol{s}_{ui} = \{s_{ui}^1, s_{ui}^2, ..., s_{ui}^{l_{ui}}\}$ is the review posted by user u on item i, s_{ui}^k is the kth word in the review, the word vocabulary is defined as $\mathcal{V}$, and l_{ui} is the review length. For each review $\boldsymbol{s}_{ui}$, there is a feature set $\boldsymbol{f}_{ui}$ associated with it, indicating the review contents. For example, in Figure 1(a), for the review of "The quality of the phone is very good", the feature is "quality". We define by $\mathcal{F}$ the set of all features. Given $\mathcal{U}$, $\mathcal{I}$, $\mathcal{F}$ and $\mathcal{R}$, the task of natural language explainable recommendation aims to learn a model, such that for a give user-item pair and a feature set, the model can accurately predict the review and rating.

To accomplish the above task, people have designed a lot of models [8, 9, 16, 18, 19]. Generally speaking, there are usually two parts in these models: (1) review prediction and (2) rating prediction. For the first part, suppose we have a review $\boldsymbol{s}_{ui} = \{s_{ui}^1, s_{ui}^2, ..., s_{ui}^{l_{ui}}\}$,

[1]https://www.kaggle.com/datasets/andrewmvd/trip-advisor-hotel-reviews
[2]https://jmcauley.ucsd.edu/data/amazon/
[3]https://www.yelp.com/dataset
[4]Here, an observation can be an item interaction or a feature mention.
[5]Following the common practice in this domain, we assume that each rating is accompanied with a user review.

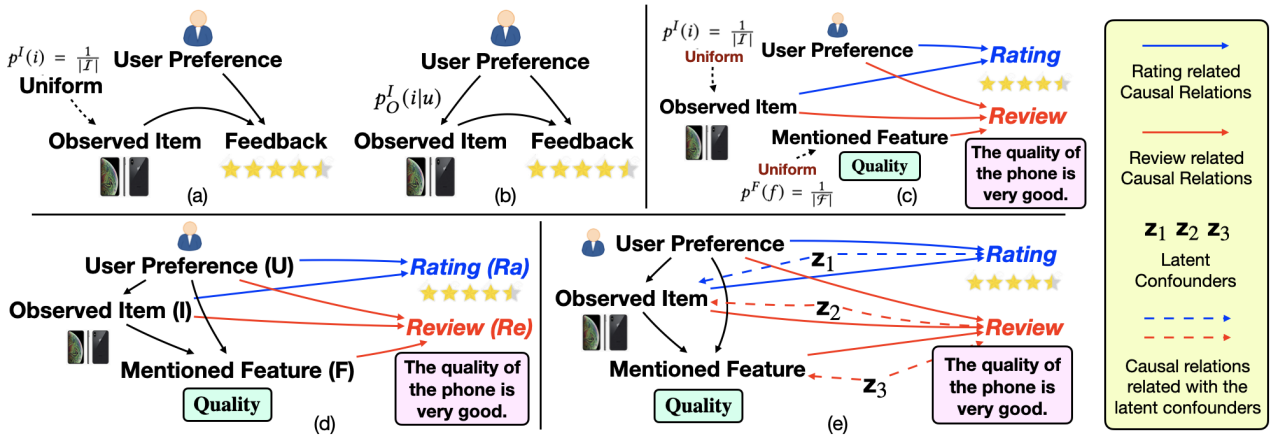


Figure 2: (a) The causal graph for generating ideal datasets in general recommendation. (b) The causal graph for generating observational datasets in general recommendation. (c) The causal graph for generating ideal datasets in explainable recommendation. (d) The causal graph for generating observational datasets in explainable recommendation. (e) The causal graph for explainable recommendation with latent confounders.

then the model is optimized to maximize the likelihood of observing this review. The loss function is $L^S_{u,i,f} = \frac{1}{l_{ui}} \sum_{t=1}^{l_{ui}} -\log g^S(s = s^t_{ui} | u, i, f, \boldsymbol{s}^{1:(t-1)}_{ui})$, where g^S can be implemented with any sequential architecture like GRU [16, 19] and LSTM [8, 9]. The output of g^S is a $\mathcal{V}$-sized softmax layer, and "$s = s^t_{ui}$" means that the s^t_{ui}th element should be maximized. $\boldsymbol{s}^{1:(t-1)}_{ui} = \{s^1_{ui}, s^2_{ui}, ..., s^{t-1}_{ui}\}$ is the set of words before time step t. For generating more informative explanations, people usually extract a feature f from $\boldsymbol{s}_{ui}$, and input it at each step [16, 18]. For the second part, the model is optimized by minimizing the distance between the predicted and real ratings, and the loss is $L^R_{u,i} = (r_{ui} - g^R(u, i))^2$, where g^R is the model for predicting the user-item ratings. Since the review and rating can be associated (*e.g.*, on the sentiment polarity), g^S and g^R are usually designed by sharing the architectures and parameters [16, 18], which unifies $L^S_{u,i,f}$ and $L^R_{u,i}$ into a multi-task learning framework.

2.2 Causal Understanding of Debiased RecSys

Debiased recommendation is becoming more and more popular [4, 12, 26, 27, 32]. In this section, we provide a causal perspective to understand this problem. For general recommendation, where we only consider the user, item and rating, the ideal loss function should evaluate all the user-item pairs [8, 27], that is: $L_{\text{ideal}} = \frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} L^R_{u,i} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathrm{E}_{i \sim p^I(i)}[L^R_{u,i}]$, where $p^I(i) = \frac{1}{|\mathcal{I}|}$ $(\forall i \in \mathcal{I})$ is the uniform distribution on the item set. This ideal objective is equal to optimizing the model with the dataset generated according to the causal graph in Figure 2(a). However, in practice, the observed datasets do not follow this causal graph, since the observation[6] of an item should be related with the user due to the diverse personalities or the personalization requirements of the previous recsys (see Figure 2(b)). In order to obtain debiased recommender models based on the observed datasets, the following loss function can be adopted $L_{\text{debias}} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{I}_u|} \sum_{i \in \mathcal{I}_u} \frac{p^I(i)}{p^I_O(i|u)} L^R_{u,i}$, where $p^I_O(i|u)$ is the probability of observing the interaction of item i given user u (a.k.a, propensity score). $\mathcal{I}_u$ is the set of items interacted by user u. For the unbiasedness of this objective, we have the following theory, where we present the proof in Appendix A.

Theorem 1 (Unbiasedness of L_{debias}). *Suppose each item in $\mathcal{I}_u$ is a random variable, and independently sampled from the observational probability $p^I_O(i|u)$, then $E[L_{debias}] = L_{ideal}$.*

Remark. Above, we provide a causal understanding of the debiased recommendation by analyzing the causal graphs leveraged for generating the ideal and observed datasets. These causal graphs enable us to more intuitively understand the bias formation mechanisms, and inspire us to improve debiased recommender models from the causal perspective (*e.g.*, capturing the potential latent confounders).

3 THE USER FRAMEWORK

In this section, we firstly define the ideal unbiased learning objective for explainable recommendation, and correspondingly show the causal graph for ideal dataset generation. Then, we analyze the causal graph for generating the observed datasets, and propose a debiased loss function for the observed datasets based on IPS. In the next, we design a fault tolerant mechanism for the IPS estimated from the noisy and sparse recommendation datasets. At last, we model and infer the potential latent confounders, and incorporate them into the IPS and review/rating prediction processes. In the following, we introduce our framework more in detail.

3.1 Ideal Learning Objective

In our explainable recommendation task, the ideal objective should evaluate the reviews and ratings for all the users, items and features, thus we have the following ideal loss:

$$L^e_{\text{ideal}} = \frac{1}{|\mathcal{U}||\mathcal{I}||\mathcal{F}|} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \sum_{f \in \mathcal{F}} L^S_{u,i,f} + \frac{\alpha}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} L^R_{u,i}$$

$$= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathrm{E}_{i \sim p^I(i)}[\mathrm{E}_{f \sim p^F(f)}[L^S_{u,i,f}]] + \frac{\alpha}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathrm{E}_{i \sim p^I(i)}[L^R_{u,i}],$$

[6] An item may be observed because the user actively selects this item or the item is recommended according to the user preference.

where α is a hyper-parameter balancing the importances between the review and rating predictions. $p^F(f) = \frac{1}{|\mathcal{F}|}$ $(\forall f \in \mathcal{F})$ is the uniform distribution on the feature set. This objective needs to be optimized based on the datasets generated according to the causal graph in Figure 2(c), where the observation probabilities of the items and features are not influenced by any other factors, but follow uniform distributions. However, such datasets can only be obtained by conducting a large amount of online experiments, which are too expensive [27]. Thus, we derive a tractable unbiased loss for the observed datasets, which are much more accessible.

3.2 Unbiased Loss for the Observed Datasets

To derive the unbiased loss function, we assume that the observed datasets are generated according to the causal graph in Figure 2(d). The rationalities of this causal graph are presented as follows: **U** $\rightarrow$ **I**: the items are observed due to the user active selections or system recommendations, which are naturally influenced by the user personalized preferences. **U** $\rightarrow$ **F** and **I** $\rightarrow$ **F**: since the reviews are written by the users for the items, the mentioned features in the review are influenced by the user preferences and observed items. **U, I** $\rightarrow$ **Ra**: the ratings are given by the users on the items, thus they are determined by the user preferences and observed items. **U, I, F** $\rightarrow$ **Re**: the reviews are posted by the users for the items around the features, which form the edges from **U, I, F** to **Re**.

Based on the above causal graph, we design the following unbiased loss function:

$$L^e_{\text{debias}} = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\left\{\frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}\frac{p^I(i)}{p^I_O(i|u)}\left[\frac{1}{|\mathcal{F}_{ui}|}\sum_{f\in\mathcal{F}_{ui}}\frac{p^F(f)}{p^F_O(f|u,i)}L^S_{u,i,f}\right]\right\} + \frac{\alpha}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}\frac{p^I(i)}{p^I_O(i|u)}L^R_{u,i},$$

where $\mathcal{F}_{ui}$ is the set of features[7] mentioned in the review of user u for item i. $p^F_O(f|u,i)$ is the probability of observing feature f given the user-item pair (u,i). The unbiasedness of L^e_{debias} is shown in the following theory, and the proof is presented in Appendix B.

Theorem 2 (Unbiasedness of L^e_{debias})**.** *Suppose each item in $\mathcal{I}_u$ and each feature in $\mathcal{F}_{ui}$ are independently sampled from $p^I_O(i|u)$ and $p^F_O(f|u,i)$, respectively, then $E[L^e_{debias}] = L^e_{ideal}$.*

3.3 IPS with Fault Tolerant Mechanism

In most of the previous works [4, 12, 26, 27, 32], IPS is estimated from the recommendation datasets, which can be highly sparse[8] and noisy, making it hard to obtain the real IPS. To alleviate this problem, we introduce a fault tolerant mechanism to handle the estimation error. More specifically, we firstly predict an initial IPS (which can be inaccurate), and then assume that the real IPS fall into an ϵ-ball centered at the initial IPS. At last, we minimize the maximum loss induced by the IPS in this ϵ-ball. Such optimization is equal to minimize the upper bound of the loss function with the real IPS, which is demonstrated to be effective in our experiments.

[7] $|\mathcal{F}_{ui}| = 1$ in our problem.

[8] In our problem, we not only have to handle the user-item matrix, but also need to process the user-item-feature tensor, which is much sparser.

Formally, we solve the following optimization problem:

$$\begin{aligned}\min_{\Theta_g}\max_{\Theta_p}\ & \frac{1}{|\mathcal{U}||\mathcal{I}||\mathcal{F}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_u}\frac{1}{p^I_O(i|u)|\mathcal{I}_u|}\frac{1}{p^F_O(f|u,i)}L^S_{u,i,f} \\ &+ \frac{\alpha}{|\mathcal{U}||\mathcal{I}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_u}\frac{1}{p^I_O(i|u)|\mathcal{I}_u|}L^R_{u,i} \\ \text{s.t. }& |p^I_O(i|u) - \hat{p}^I_O(i|u)| \le \epsilon,\ |p^F_O(f|u,i) - \hat{p}^F_O(f|u,i)| \le \epsilon, \\ & \forall u \in \mathcal{U},\ i \in \mathcal{I}_u,\ f \in \mathcal{F}_{ui},\end{aligned} \tag{1}$$

where $\hat{p}^I_O(i|u)$ and $\hat{p}^F_O(f|u,i)$ are the initially estimated IPS. Straightforwardly, $p^I_O(i|u)$ (or $p^F_O(f|u,i)$) can be realized with free parameters for each user-item pair (or user-item-feature triplet). However, this requires nearly $|\mathcal{U}||\mathcal{I}|$ (or $|\mathcal{U}||\mathcal{I}||\mathcal{F}|$) parameters, which may easily over-fit the training data, and be hard to generalize. In practice, we learn models to estimate $p^I_O(i|u)$ (or $p^F_O(f|u,i)$), where the model parameters are collected in Θ_p, and we introduce their specifications later. Θ_g is the set of model parameters for predicting the reviews and ratings. ϵ defines the tolerance level. Smaller ϵ requires more accurate initial IPS to quickly discover the real IPS within a small range. Larger ϵ indicates lower confidence on the initial IPS, and our model needs to find the real IPS in a wider space.

Actually, the above problem has close relation with the ideal learning objective L^e_{ideal}. To reveal such relation, we rewrite the constraints in (1) as regularizers, and define the following objective:

$$\begin{aligned}L_{\text{debias}} = \max_{\Theta_p}\ & \frac{1}{|\mathcal{U}||\mathcal{I}||\mathcal{F}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_u}\frac{1}{p^I_O|\mathcal{I}_u|}\frac{1}{p^F_O}L^S_{u,i,f} \\ &+ \frac{\alpha}{|\mathcal{U}||\mathcal{I}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_u}\frac{1}{p^I_O|\mathcal{I}_u|}L^R_{u,i} \\ &- \lambda_S\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}|\hat{p}^F_O - p^F_O| - \lambda_R\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}|\hat{p}^I_O - p^I_O|,\end{aligned}$$

where $p^I_O = p^I_O(i|u)$ and $p^F_O = p^F_O(f|u,i)$, λ_R and λ_S are regularization parameters, and we slightly change (1) by constraining all the user-item pairs. Then we have the following theory:

Theorem 3 (Theoretical justification)**.** *Suppose: (1) p^{I*}_O and p^{F*}_O are the real propensity scores, p^I_O and p^F_O are the propensity scores used in our objective and $\hat{p}^I_O$ and $\hat{p}^F_O$ are the initially estimated propensity scores. All the propensity scores fall into the range of $[\kappa_1, \kappa_2]$, where $0 < \kappa_1 < \kappa_2 < 1$. (2) $|\frac{L^R_{u,i}}{p^I_O}| \le \Delta$ and $|\frac{L^S_{u,i,f}}{p^I_O p^F_O}| \le \Delta_1$, if we set $\lambda_S = \frac{\Delta_1\kappa_2}{|\mathcal{U}||\mathcal{I}||\mathcal{F}|}$, $\lambda_R = \frac{\Delta|\mathcal{F}|+\Delta_1\kappa_2}{|\mathcal{U}||\mathcal{I}||\mathcal{F}|}$. Then the following inequality holds with probability as least $1-\eta$.*

$$L^e_{ideal} \le L_{debias} + Const, \tag{2}$$

where Const is a constant related with η.

This theory relates our objective with the ideal loss, which theoretically reveals the rationality of our framework. The proof of this theory is presented in the Appendix C.

3.4 Modeling the Latent Confounders

In our problem, there can be three types of latent confounders (see Figure 2(e)), that is, (i) the confounder between **I** and **Ra** (*e.g.*, the promotion of the items), (ii) the confounder between **I** and

Re (*e.g.*, the temporal factors. In winter, the user may interacted with the cotton, and the warmth is more likely to be commented), and (iii) the confounder between **F** and **Re** (*e.g.*, the behaviors of following the other reviews). We assume that the above latent confounders can be fully represented by three embeddings z_1, z_2 and z_3, respectively. Similar to [2, 22], we infer the latent confounders by neural networks based on the user, item and feature information. In specific, we let $z_1 = g_1(u, i)$, $z_2 = g_2(u, i)$ and $z_3 = g_3(u, i, f)$. Intuitively, if the latent confounders are well represented, then they should lead to larger likelihoods of the item/feature observations. Thus we have the following cross-entropy loss: $L_1 = -\sum_{i\in I} y_i \log(\hat{y}^I(i|u, z_1))$, $L_2 = -\sum_{i\in I} y_i \log(\hat{y}^I(i|u, z_2))$ and $L_3 = -\sum_{f\in F} y_f \log(\hat{y}^F(f|u, i, z_3))$, where we delay the specifications of $g_1(\cdot)$, $g_2(\cdot)$ and $g_3(\cdot)$ in the following section. y_i and y_f indicate whether item i and feature f are observed. $\hat{y}^I$ and $\hat{y}^F$ predict the probabilities of the item and feature observations. In addition to L_1, L_2 and L_3, we also constrain z_1, z_2 and z_3 by incorporating them into objective (1). Suppose Θ_c is the set of parameters of g_1, g_2 and g_3, then we finally have the following objective:

$$\min_{\Theta_g,\Theta_c}\max_{\Theta_p}\{L_{\text{final}}:\frac{1}{N|\mathcal{F}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_u}\frac{1}{p^I_{O2}(i|u,z_2)|\mathcal{I}_u|}\frac{1}{p^F_O(f|u,i,z_3)}L^S_{u,i,f}$$
$$+\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_u}(\frac{\alpha}{p^I_{O1}(i|u,z_1)N|\mathcal{I}_u|}L^R_{u,i}-\lambda_R|p^F_O(f|u,i,z_3)-\hat{p}^F_O(f|u,i)|)$$
$$-\lambda_S\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_u}(|p^I_{O1}(i|u,z_1)-\hat{p}^I_O(i|u)|+|p^I_{O2}(i|u,z_2)-\hat{p}^I_O(i|u)|)$$
$$+L_1+L_2+L_3\},$$

where $N = |\mathcal{U}||\mathcal{I}|$, λ_R and λ_S are Lagrange multipliers. They play similar roles as the tolerance level ϵ. Larger λ_R (or λ_S) imposes stricter constraint on the distance between the real and initial IPSs, while smaller λ_R (or λ_S) means that this distance is allowed to be larger. We use different item-level propensities to capture the confounders related to the review and rating behaviors separately. g^S and g^R in L^S and L^R are updated to $g^S(s = s^t_{ui}|u, i, f, z_2, z_3, \boldsymbol{s}^{1:(t-1)}_{ui})$ and $g^R(u, i, z_1)$, respectively.

Model specifications. For $p^I_{Ok}(i|u, z)$ ($k = 1$ or 2) and $p^F_O(f|u, i, z)$, we have $p^I_{Ok}(i|u, z) = \text{softmax}(\boldsymbol{E}_{\mathcal{I}}(\boldsymbol{e}_u \odot z))$ and $p^F_O(f|u, i, z) = \text{softmax}(\boldsymbol{E}_{\mathcal{F}}(\boldsymbol{e}_u \odot \boldsymbol{e}_i \odot z))$, where z is the representation of the latent confounders, $\boldsymbol{E}_{\mathcal{I}} \in R^{\mathcal{I}\times d}$, $\boldsymbol{E}_{\mathcal{U}} \in R^{\mathcal{U}\times d}$ and $\boldsymbol{E}_{\mathcal{F}} \in R^{\mathcal{F}\times d}$ are the embedding matrices for the items, users and features. $\boldsymbol{e}_u \in R^d$ and $\boldsymbol{e}_i \in R^d$ are the uth and ith columns of $\boldsymbol{E}_{\mathcal{U}}$ and $\boldsymbol{E}_{\mathcal{I}}$, respectively, representing the embeddings of user u and item i. $\odot$ is the element-wise product. For g_1, g_2 and g_3, we implement them with two layer fully connected neural networks, where we use ReLU as the activation functions, and the inputs are the concatenation of the user-item or user-item-feature embeddings. For $\hat{y}^I$ and $\hat{y}^F$, we leverage similar architectures as used for g_1, g_2 and g_3, but the output layers are softmax to predict the item/feature observation probabilities. g^S and g^R are determined according to the specific models our framework is applied to.

4 RELATED WORK

Our framework stands on the intersection between explainable recommendation and debiased recommendation. In this section, we discuss the previous work in these fields. **Relation with explainable recommendation.** In the past few years, people have proposed a lot of explainable recommendation(EXR) models [36]. They are based on different techniques such as rule mining [1, 24], attention mechanism [6, 7] and various auxiliary information such as knowledge graph [31, 35] and user reviews [18]. Among different explanation strategies, producing natural language explanation is an important method, which can deliver more flexible and user-accessible information [18]. For example, [8, 9] leverage LSTM to generate user reviews. [16, 18] propose to generate controllable explanations based on the product features. [10, 11] build explainable fairness aware recommender models. While the above studies have greatly promoted the field of EXR, they mainly focus on the model perspective. However, in this paper, we consider EXR on the data biases, which significantly differs from the previous work. **Relation with debiased recommendation.** Recent years have witnessed many promising studies on debiased recommendation [5]. For example, [27] proposes an IPS based method to re-weight the training samples for data bias correction. [26] further extends this work to the implicit feedback. To improve the robustness, [32] designs a doubly robust model to achieve unbiased recommendation. To more comprehensively consider the diverse nature of the user preferences and item properties, [30, 33, 37] propose to model and infer the latent confounders in the recommendation domain. By incorporating a small amount of uniform data, [3] design a causal embedding strategy to overcome the data bias problem. While the above studies have achieved many promising results, they focus on improving the recommendation performance. However, we aim to enhance the explanation quality, which, to the best of our knowledge, is the first time in the recommendation domain.

Table 1: Statistics of the datasets.

	TA-HK	AZ-MT	YELP
# Users	9,765	7,506	27,147
# Items	6,280	7,360	20,265
# Features	2,825	6,473	8,548
# Interactions	169,389	235,459	676,433
Sparsity	99.72%	99.57%	99.99%
Domain	Hotel	E-commerce	Restaurant

5 EXPERIMENTS

5.1 Experiment Setup

Datasets. Our experiments are based on three public available datasets: **TripAdvisor-HongKong (TA-HK)** is a dataset crawled from a well-known travel website called TripAdvisor[9]. It contains user ratings and reviews on the hotels in Hong Kong. **Amazon Movies&TV (AZ-MT)** is an e-commerce dataset, where we can access the user preferences on the videos in terms of the ratings and reviews. **Yelp Challenge 2019 (YELP)** is a dataset reflecting user preferences on the restaurants. In our experiments, we directly use the datasets[10] released in [17], where the item features have been provided for each user review. The dataset statistics are presented in Table 1. We can see our datasets can cover different domains, which can help to demonstrate the generality of our framework.

[9]https://www.tripadvisor.com
[10]https://github.com/lileipisces/EXTRA

Table 2: Overall comparison between our framework and baselines. For BLEU and ROUGE, the results are percentage values with "%" omitted. For each dataset and evaluation metric, we use bold fonts to label the best performance. "-" means the evaluation metric is not available for the model. The performance improvements of our framework are significant under paired t-test with $p < 0.05$.

Metrics	BLEU (%)		ROUGE-1 (%)			ROUGE-2 (%)			MAE
	BLEU-1	BLEU-4	F1	Recall	Precision	F1	Recall	Precision	
	TA-HK dataset								
MF	-	-	-	-	-	-	-	-	1.494
SVD++	-	-	-	-	-	-	-	-	0.679
Att2Seq	13.788	0.622	13.981	13.134	16.766	1.514	1.479	1.751	-
NRT	10.853	0.306	14.830	12.751	19.368	0.799	0.658	1.137	0.699
NETE	14.960	0.641	15.690	14.904	18.082	1.815	1.844	1.989	0.664
NETE-USER	**15.923**	**0.769**	**26.060**	**18.204**	**58.303**	**3.456**	**2.831**	**5.733**	**0.647**
PETER	18.718	2.148	22.344	20.647	27.547	4.616	4.412	5.729	0.710
PETER-USER	**19.462**	**2.522**	**26.897**	**23.560**	**37.931**	**6.512**	**5.888**	**9.465**	**0.692**
	AZ-MT dataset								
MF	-	-	-	-	-	-	-	-	1.465
SVD++	-	-	-	-	-	-	-	-	0.866
Att2Seq	10.595	0.465	13.258	11.025	19.557	1.306	1.117	1.873	-
NRT	11.305	0.388	14.735	11.748	21.818	1.309	1.058	1.931	0.815
NETE	13.070	0.433	15.592	13.128	21.162	1.458	1.318	1.914	0.723
NETE-USER	**15.701**	**0.761**	**17.518**	**15.235**	**24.025**	**2.001**	**1.834**	**2.536**	**0.714**
PETER	16.602	1.898	21.664	18.140	32.142	4.320	3.775	6.225	0.732
PETER-USER	**17.169**	**2.038**	**23.234**	**19.247**	**35.575**	**4.990**	**4.340**	**7.404**	**0.705**
	YELP dataset								
MF	-	-	-	-	-	-	-	-	2.516
SVD++	-	-	-	-	-	-	-	-	0.976
Att2Seq	10.634	0.328	11.871	10.631	15.573	0.715	0.653	0.968	-
NRT	9.515	0.343	11.430	9.983	15.562	0.764	0.697	1.037	0.852
NETE	7.966	0.210	11.238	9.320	16.569	0.525	0.438	0.815	0.867
NETE-USER	**12.918**	**0.541**	**23.948**	**16.410**	**56.050**	**2.744**	**2.130**	**5.079**	**0.859**
PETER	17.426	1.928	22.078	20.325	28.578	4.530	4.225	6.104	0.877
PETER-USER	**17.664**	**1.941**	**22.747**	**20.718**	**30.703**	**4.558**	**4.260**	**6.516**	**0.838**

Baselines. We compare our framework with the following representative baselines: **Att2Seq** [9] is an LSTM model for generating user reviews directly based on the user/item ID and rating information. **NRT** [19] is also a review generation model based user/item ID, but its backbone is the gated recurrent unit (GRU). **NETE** [16] is a controllable review generation model, where the text sequence is decoded by taking the user, item and feature information as input. **PETER** [18] also leverages features as input to generate more informative explanations, but it uses transformer as the main architecture, which can usually achieve the state-of-the-art performance. We apply our framework on **NETE** and **PETER**, which are both feature enhanced explainable recommender models. The obtained methods are called **NETE-USER** and **PETER-USER**, respectively. For the task of rating prediction, we also compare our framework with the following simple but effective models: **MF** [14] is the well-known matrix factorization model, where the users and items are represented by latent vectors, and the user-item preferences are estimated by inner-product based on these vectors. **SVD++** [13] is a variant of MF, where the preference is estimated by taking the user history information into consideration.

Implementation details. In general, the experiments in the domain of debiased recommendation should follow the paradigm of "biased training and unbiased evaluation". To this end, we use 50% of each user interactions as the biased training set. For building the validation/testing sets, we follow the previous work [3, 20, 25, 26, 34, 39] to sample from the other interactions based on the inverse item/feature observation frequencies, where the more frequently observed items/features are sampled with lower probabilities. We set the splitting ratio between the validation and testing sets as 1:1. In the experiments, we set the maximum length of the generated explanations as 15, and the vocabulary $\mathcal{V}$ is constructed by 20000 most frequently mentioned words. We tune the hyper-parameters of our model by grid search. In specific, we tune the learning rate and hidden size in the ranges of [0.1, 0.01, 0.001] and [32, 64, 128, 256], respectively. The batch size for all models is set as 128 and the weight of $L^R_{u,i}$ is tuned in [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]. We implement the baselines with the codes released in [16, 18] at https://github.com/lileipisces/NLG4RS. The parameters are set as the optimal values reported in the original paper or tuned in the same ranges as our model's. To evaluate the explanation quality, two commonly used metrics including BLEU [23] and ROUGE [21] are leveraged for model comparisons. To evaluate the recommendation performance, we use MAE [19] as the evaluation metric.

5.2 Overall Performance

The overall comparison results are presented in Table 2. We can see: for the explanation task, the winner between Att2Seq and NRT varies on different datasets and evaluation metrics, and the performance gap is not large, which suggests that these models may have similar explanation capabilities. By introducing item features as input, NETE and PETER can achieve better performances than Att2Seq and NRT in most cases. This is as expected, since the features can provide much valuable information to describe the review content and better guide the sentence decoding process. It

Table 3: Performance comparison between our framework and its variants, where "ROUGE " represents "ROUGE-1-F1 (%)". We use bold fonts to label the best performance.

Method	TA-HK		AZ-MT		YELP	
	ROUGE	MAE	ROUGE	MAE	ROUGE	MAE
PETER	22.344	0.710	21.664	0.732	22.078	0.877
PETER-USER-T	19.848	0.714	19.421	0.917	12.354	0.958
PETER-USER-L	22.897	0.696	22.016	0.711	21.560	0.855
PETER-USER-I	26.636	0.718	22.336	0.744	21.606	0.964
PETER-USER-F	22.237	0.706	21.292	0.710	21.940	0.847
PETER-USER	**26.897**	**0.692**	**23.234**	**0.705**	**22.747**	**0.838**

is encouraging to see that, by imposing our framework on NETE and PETER, the performance can be significantly enhanced, which are consistent on all the datasets and metrics. Considering that the evaluation datasets are unbiased, this result demonstrates that our framework can provide better explanations if we treat different items and features equally in the testing phase. In the original models, the parameters are directly learned based on the biased datasets, thus they cannot perform well when the testing sample distribution is changed to be unbiased. However, our framework can effectively correct the item- and feature-level biases, which achieves much better performance than the original models. For the rating prediction task, we find that by incorporating the history information, SVD++ can achieve much better performance than MF on all the datasets. In most cases, NRT, NETE and PETER perform better than MF and SVD++. This is not surprising, because in the first three models, the rating and review prediction problems are formulated as a multi-task learning framework, which makes them can be mutually enhanced by each other. By applying our framework on NETE and PETER, the performances are improved, which demonstrates the debiasing effect of our framework.

5.3 Ablation Studies

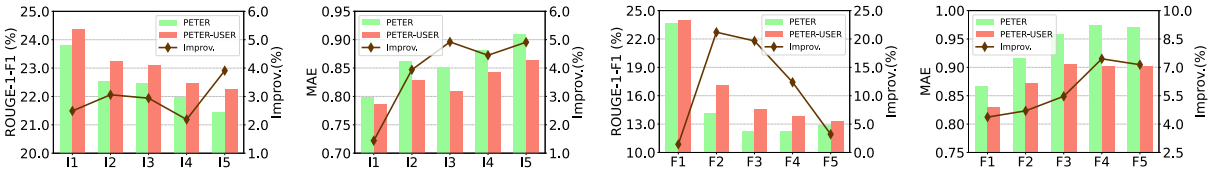
In this section, we conduct ablation studies to study the contributions of different components of our framework. More specifically, suppose the original model is X, then we compare our framework with its four variants: in *X-USER-T*, we remove the IPS fault tolerant mechanism, and only remain the initially estimated IPS. In *X-USER-L*, we do not model the latent confounders. In *X-USER-I*, we do not correct the item-level bias (*i.e.*, removing p^I_{O1} and p^I_{O2} in L_{final}). In *X-USER-F*, we do not correct the feature-level bias (*i.e.*, removing p^F_O in L_{final}). We use PETER as the original model, and the conclusions on NETE are similar and omitted. We set the parameters as their optimal values tuned in the above section. The comparison results are presented in Table 3. We can see: in most cases, dropping the IPS fault tolerant mechanism lowers the performance more severely than ignoring the latent confounders. We speculate that the recommendation datasets can be too sparse and noisy to obtain accurate IPS, thus introducing fault tolerant mechanism for the estimated IPS can well bound the real objective function, and help to generate better explanations. While modeling latent confounders can indeed bring better performance, but the improvements are not large. We argue that the confounder structures can be very complicated, representing them with unified embeddings can be too coarse, and fail to provide enough priors for effective parameter learning. Correcting the item- and feature-level biases are both important, which is evidenced by the lowered performances of X-USER-I and X-USER-F against X-USER. For all datasets, X-USER-F can achieve better performance than X-USER-I on MAE, because the feature-level bias is irrelevant with the rating prediction task. However, for the explanation task, the performance of X-USER-F is comparable or worse than that of X-USER-I, which suggests that removing the feature-level bias is important to enhance the explanation quality. The best performance is achieved when we combine all the components, which demonstrates that they are all necessary.

5.4 Influence of the Item/Feature Popularity

To better understand the debiasing effect of our framework, we report the performance on the items/features with different popularity levels. In specific, we firstly sort the items and features in the training set according to their frequencies, where more popular items are ranked higher. Then we cluster the items/features ranked between 0%-20%, 20%-40%, 40%-60%, 60%-80% and 80%-100% into five groups. According to the item groups, we separate the original testing set into five subsets I_1, I_2, I_3, I_4 and I_5, for example, in I_1, the items are all ranked between 0%-20% in the training set. Similarly, based on the feature groups, the testing set can also be divided into five subsets, and we denote them as F_1, F_2, F_3, F_4 and F_5, respectively. In the experiments, we remain the training set the same as the previous experiments, while the final performances are reported on the above ten sub-testing sets. We use YELP as the experiment dataset, and the model parameters are set as their optimal values tuned in Table 2. From the results presented in Figure 3, we can see: from the item perspective, the performance improvement brought by our framework is larger on the items with lower observational frequencies. The conclusion is consistent on both of the explanation generation and rating prediction tasks. The reason can be that the items with lower frequencies are higher weighted in our framework. They can be optimized more sufficiently, and thus the performance on them can be improved larger. From the feature perspective, the performance improvement is small when the features are too unpopular. We find that the feature frequencies in the last groups (*e.g.*, F_5) are mostly "1". In such a scenario, while we have higher weighted these samples, the information provided for each feature is too limited and not diverse enough, which impacts the model generalization capability and performance improvements.

5.5 Studies on the Generated Explanations

5.5.1 Qualitative studies. For more intuitively understanding our framework, we present many examples from the TA-HK dataset. In specific, we use PETER as the original model, and compare the explanations generated from PETER and PETER-USER. We set the model parameters as their optimal values tuned in Table 2. For references, we also present (i) the ground truth of the explanations, and (ii) the observation frequencies of the items and features, which are computed as the ratio between the current and maximum item interaction (or feature mention) times. From the results shown in Table 4, we can see, in the first case, when the item and feature have high observation frequencies, both PETER and PETER-USER can capture the key word "amenities". For the second and third cases, where the items and features are underrepresented in the dataset (*i.e.*, with lower observation frequencies), PETER-USER



(a) Results with different item popularity.

(b) Results with different feature popularity.

Figure 3: Performance on the items/features with different popularities. "Improv." means the performance improvement ratio.

Table 4: Qualitative studies. In each case, the first line indicates the item and feature observation frequencies (omitting "%"). The second line presents the true explanations. The third and forth lines show the results of PETER and PETER-USER, respectively. The features in the real and generated explanations are labeled by bold fonts.

Model	Explanation
$Freq_i$=4.50; $Freq_f$=28.63	
Ground Truth	Clean and comfortable with good **amenities**.
PETER	The **amenities** are good.
PETER-USER	**Amenities** were good.
$Freq_i$=1.35; $Freq_f$=1.31	
Ground Truth	Set in a nice **surrounding**.
PETER	The **hotel** is located in a great location.
PETER-USER	**Surroundings** are great and the **staff** are friendly and helpful.
$Freq_i$=0.75; $Freq_f$=0.58	
Ground Truth	Had a nice **steak** in the Italian restaurant.
PETER	The **hotel** is located in the heart of the city.
PETER-USER	The hotel is a good **steak** house and the **food** is good.

can generate more accurate and informative explanations around the given features, while PETER only produces general words for the explanations. More specifically, in the second case, PETER-USER can accurately generate the explanation around the feature "surrounding", while PETER regards "hotel" as the main review contents. In the third case, PETER-USER can reasonably provide explanations on the feature "steak", but PETER fails to capture such information. The reasons behind the above phenomena can be that in our framework, the items/features with lower observation frequencies are higher weighted to enhance their importances in the training phase. However, in the original model, these items/features are mostly neglected, thus the performance is not satisfied.

5.5.2 Quantitative studies. To study whether the generated explanations can indeed help users, we further design questionnaires to ask the feelings of the users on the explanations. In specific, we focus on two aspects of the explanations [28], that is, (i) effectiveness: whether the explanations can communicate useful information on the items, and (ii) persuasiveness: whether the explanations can persuade users to make decisions. Based on these aspects, we design the following two questions [29]: **Q1**: *Does the explanation help you to learn more about the recommended item?* **Q2**: *Does the explanation help you to make fast decisions?* For each of these questions, the annotator is required to give a rating ranging from 1 to 5 to indicate her agreement (*i.e.*, 1-strongly disagree, 2-disagree, 3-neural, 4-agree and 5-strongly agree) on the question. In the experiments,

Table 5: Quantitative studies. For each question, larger result indicates better performance. The better results between our framework and the original model are labeled by bold fonts.

Dataset	TA-HK		AZ-MT		YELP		Average	
Question	Q1	Q2	Q1	Q2	Q1	Q2	Q1	Q2
NETE	2.69	3.15	2.30	2.60	2.57	2.67	2.52	2.81
NETE-USER	**3.51**	**3.87**	**2.90**	**3.01**	**3.19**	**3.29**	**3.20**	**3.39**
PETER	3.01	3.16	2.43	2.49	3.12	3.25	2.85	2.96
PETER-USER	**3.28**	**3.63**	**2.68**	**2.89**	**3.18**	**3.46**	**3.05**	**3.33**

we randomly select 50 samples from the testing set of each dataset, and 12 annotators with different backgrounds are employed from a university. We leverage NETE and PETER as the original models, and compare the explanations generated from them with the ones produced by imposing our framework. The results are reported as the average rating across different annotators and samples, which are presented in Table 5. We can see: for both questions, our framework can lead to better performance than the original model on all the datasets, and the results are consistent for both NETE and PETER. On average, our framework can improve the performance of the original model by about 17.0% and 16.6% on Q1 and Q2, respectively. This observation demonstrates the effectiveness of our framework on improving the explanation effectiveness and persuasiveness, which demonstrates its potential in real-world settings. While we have noticed that there can be general limitations for the questionnaire studies (*e.g.*, discrepancies between different annotator understandings on the explanations), under the same limitations, our framework can always achieve better performances, which suggests its superiority.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose the task of debiased explainable recommendation for the first time. For solving this task, we build a principled framework to jointly correct the item- and feature-level biases, and design fault tolerant IPS mechanism and latent confounder modeling strategy to improve this framework. Extensive experiments demonstrate that our framework can bring improved explanation and recommendation performances for the state-of-the-art models. This paper opens a novel direction on explainable recommendation, and we believe there still left much room for improvement. For example, one can leverage IPS normalization or doubly robust methods to lower the variance of the loss function. In addition, this paper mainly focuses on natural language explanations, it is interesting to extend the debiasing idea to the other types of explanations. At last, the user preferences in real-world scenarios can be dynamic, how to design debiased models considering the temporal influence is also important.

ACKNOWLEDGMENTS

This work is supported in part by National Natural Science Foundation of China (No. 62102420 and No. 61832017), Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China, and Public Computing Cloud, Renmin University of China. The work is sponsored by Huawei Innovation Research Programs. We gratefully acknowledge the support from Mindspore[11], CANN (Compute Architecture for Neural Networks) and Ascend Ai Processor used for this research.

REFERENCES

[1] 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.

[2] Ioana Bica, Ahmed Alaa, and Mihaela Van Der Schaar. 2020. Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders. In *International Conference on Machine Learning*. PMLR, 884–895.

[3] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *Proceedings of the 12th ACM conference on recommender systems*. 104–112.

[4] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. AutoDebias: Learning to debias for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 21–30.

[5] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions. *arXiv preprint arXiv:2010.03240* (2020).

[6] Jingwu Chen, Fuzhen Zhuang, Xin Hong, Xiang Ao, Xing Xie, and Qing He. 2018. Attention-driven factor model for explainable personalized recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 909–912.

[7] Xu Chen, Yongfeng Zhang, and Zheng Qin. 2019. Dynamic explainable recommendation based on neural attentive models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 53–60.

[8] Felipe Costa, Sixun Ouyang, Peter Dolog, and Aonghus Lawlor. 2018. Automatic generation of natural language explanations. In *Proceedings of the 23rd international conference on intelligent user interfaces companion*. 1–2.

[9] Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. 623–632.

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

[11] Yingqiang Ge, Juntao Tan, Yan Zhu, Yinglong Xia, Jiebo Luo, Shuchang Liu, Zuohui Fu, Shijie Geng, Zelong Li, and Yongfeng Zhang. 2022. Explainable fairness in recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 681–691.

[12] Siyuan Guo, Lixin Zou, Yiding Liu, Wenwen Ye, Suqi Cheng, Shuaiqiang Wang, Hechang Chen, Dawei Yin, and Yi Chang. 2021. Enhanced doubly robust learning for debiasing post-click conversion rate estimation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 275–284.

[13] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 426–434.

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

[15] Chenliang Li, Cong Quan, Li Peng, Yunwei Qi, Yuming Deng, and Libing Wu. 2019. A capsule network for recommendation and explaining what you like and dislike. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 275–284.

[16] 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.

[17] Lei Li, Yongfeng Zhang, and Li Chen. 2021. Extra: Explanation ranking datasets for explainable recommendation. In *Proceedings of the 44th International ACM SIGIR conference on Research and Development in Information Retrieval*. 2463–2469.

[18] Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized transformer for explainable recommendation. *arXiv preprint arXiv:2105.11601* (2021).

[19] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 345–354.

[20] Dawen Liang, Laurent Charlin, and David M Blei. 2016. Causal inference for recommendation. In *Causation: Foundation to Application, Workshop at UAI. AUAI*.

[21] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[22] Jing Ma, Ruocheng Guo, Chen Chen, Aidong Zhang, and Jundong Li. 2021. Deconfounding with networked observational data in a dynamic environment. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 166–174.

[23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[24] 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.

[25] Yuta Saito. 2020. Asymmetric tri-training for debiasing missing-not-at-random explicit feedback. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 309–318.

[26] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 501–509.

[27] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*. PMLR, 1670–1679.

[28] Nava Tintarev and Judith Masthoff. 2007. A survey of explanations in recommender systems. In *2007 IEEE 23rd international conference on data engineering workshop*. IEEE, 801–810.

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

[30] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded recommendation for alleviating bias amplification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1717–1725.

[31] 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.

[32] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly robust joint learning for recommendation on data missing not at random. In *International Conference on Machine Learning*. PMLR, 6638–6647.

[33] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. 2018. The deconfounded recommender: A causal inference approach to recommendation. *arXiv preprint arXiv:1808.06581* (2018).

[34] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1791–1800.

[35] 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.

[36] Yongfeng Zhang, Xu Chen, et al. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101.

[37] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 11–20.

[38] 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.

[39] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling user interest and conformity for recommendation with causal embedding. In *Proceedings of the Web Conference 2021*. 2980–2991.

[11] https://www.mindspore.cn

A PROOF OF THEORY 1

Proof. By taking expectation on L_{debias}, we have:

$$\begin{aligned}
\mathrm{E}[L_{\text{debias}}] &= \mathrm{E}[\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}\frac{p^I(i)}{p_O^I(i|u)}L_{u,i}^R] \\
&= \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}\mathrm{E}_{i\sim p_O^I(i|u)}[\frac{p^I(i)}{p_O^I(i|u)}L_{u,i}^R] \\
&= \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}\{\sum_{i\in\mathcal{I}}p_O^I(i|u)\times\frac{p^I(i)}{p_O^I(i|u)}L_{u,i}^R\} \\
&= \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}\{\sum_{i\in\mathcal{I}}p^I(i)L_{u,i}^R\} \\
&= \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}\mathrm{E}_{i\sim p^I(i)}[L_{u,i}^R] \\
&= \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\mathrm{E}_{i\sim p^I(i)}[L_{u,i}^R] \\
&= L_{\text{ideal}},
\end{aligned} \tag{3}$$

where the second equation holds because i is the only random variable and independently sampled from $p_O^I(i|u)$. The second last equation holds because $\mathrm{E}_{i\sim p^I(i)}[L_{u,i}^R]$ is irrelevant with i. □

B PROOF OF THEORY 2

Proof. By taking expectation on L_{debias}^e, we have:

$$\begin{aligned}
&\mathrm{E}[L_{\text{debias}}^e] \\
=&\mathrm{E}[\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\left\{\frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}\frac{p^I(i)}{p_O^I(i|u)}\left[\frac{1}{|\mathcal{F}_{ui}|}\sum_{f\in\mathcal{F}_{ui}}\frac{p^F(f)}{p_O^F(f|u,i)}L_{u,i,f}^S\right]\right\}+ \\
&\mathrm{E}[\frac{\alpha}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}\frac{p^I(i)}{p_O^I(i|u)}L_{u,i}^R] \\
=&\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\left\{\frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}\mathrm{E}_{i\sim p_O^I}\left[\frac{p^I(i)}{p_O^I(i|u)}\mathrm{E}_{f\sim p^F}[L_{u,i,f}^S]\right]\right\}+ \\
&\frac{\alpha}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}\mathrm{E}_{i\sim p^I}[L_{u,i}^R] \\
=&\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\mathrm{E}_{i\sim p^I}\left[\mathrm{E}_{f\sim p^F}[L_{u,i,f}^S]\right]+\frac{\alpha}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\mathrm{E}_{i\sim p^I}[L_{u,i}^R] \\
=&L_{\text{ideal}}^e,
\end{aligned}$$

where the second equation holds because $|\mathcal{F}_{ui}|=1$. □

C PROOF OF THEORY 3

For easy analysis, we introduce some notations. Let

$$L_{\text{ideal}}^S = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\mathrm{E}_{i\sim p^I(i)}[\mathrm{E}_{f\sim p^F(f)}[L_{u,i,f}^S]],$$

$$L_{\text{ideal}}^R = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\mathrm{E}_{i\sim p^I(i)}[L_{u,i}^R],$$

then $L_{\text{ideal}}^e = L_{\text{ideal}}^S + \alpha L_{\text{ideal}}^R$.

Let

$$L_{\text{debias}}^S = \frac{1}{|\mathcal{U}||\mathcal{I}||\mathcal{F}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_u}\frac{1}{|\mathcal{I}_u|p_O^I p_O^F}L_{u,i,f}^S,$$

$$L_{\text{debias}}^R = \frac{1}{|\mathcal{U}||\mathcal{I}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_u}\frac{1}{|\mathcal{I}_u|p_O^I}L_{u,i}^R,$$

then $L_{\text{debias}} = L_{\text{debias}}^S + \alpha L_{\text{debias}}^R - \lambda_S\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}|\hat{p}_O^F - p_O^F| - \lambda_R\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}|\hat{p}_O^I - p_O^I|$.

To prove this theory, we firstly bound L_{ideal}^R and L_{ideal}^S with L_{debias}^R and L_{debias}^S, respectively. Then we combine these results to obtain the inequality (2).

C.1 Bounding L_{ideal}^R with L_{debias}^R

To begin with, we have:

$$L_{\text{ideal}}^R - E_{p_O^{I*}}[L_{\text{debias}}^R] = \frac{1}{|\mathcal{U}||\mathcal{I}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}(1-\frac{p_O^{I*}}{p_O^I})L_{u,i}^R \tag{4}$$

Let $s_u = \frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}\frac{p^I(i)}{p_O^I(i|u)}L_{u,i}^R$, then $L_{\text{debias}}^R = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}s_u$. According to the Hoeffding's inequality, if $s_u\in[a_u,b_u]$ and $|a_u-b_u|\le B_1$, then the expectation of L_{debias}^R is bounded by the following value with probability at least $1-\eta_1$:

$$E_{p_O^{I*}}[L_{\text{debias}}^R] \le L_{\text{debias}}^R + B_1\sqrt{\frac{1}{2|\mathcal{U}|}log(\frac{2|\mathcal{H}|}{\eta_1})}. \tag{5}$$

Let $C = B_1\sqrt{\frac{1}{2|\mathcal{U}|}log(\frac{2|\mathcal{H}|}{\eta_1})}$, $\beta_{u,i} = \frac{L_{u,i}^R}{|\mathcal{U}||\mathcal{I}|p_O^I}$, then we have the following inequality holds with probability at least $1-\eta_1$:

$$\begin{aligned}
L_{\text{ideal}}^R &= L_{\text{ideal}}^R - E_{p_O^{I*}}[L_{\text{debias}}^R] + E_{p_O^{I*}}[L_{\text{debias}}^R] \\
&\le \frac{1}{|\mathcal{U}||\mathcal{I}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}(1-\frac{p_O^{I*}}{p_O^I})L_{u,i}^R + L_{\text{debias}}^R + C \\
&= \sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}\frac{L_{u,i}^R}{|\mathcal{U}||\mathcal{I}|p_O^I}(p_O^I - p_O^{I*}) + L_{\text{debias}}^R + C \\
&= \sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}\beta_{u,i}(p_O^I - \hat{p}_O^I) + \beta_{u,i}(\hat{p}_O^I - p_O^{I*}) + L_{\text{debias}}^R + C \\
&\le \sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}\beta_{u,i}(p_O^I - \hat{p}_O^I) + \Delta(\kappa_2-\kappa_1) + L_{\text{debias}}^R + C \\
&\le \sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}\beta_{u,i}(2\kappa_2 - |p_O^I - \hat{p}_O^I|) + L_{\text{debias}}^R + C_1 \\
&\le 2\kappa_2\Delta - \frac{\Delta}{|\mathcal{U}||\mathcal{I}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}|p_O^I - \hat{p}_O^I| + L_{\text{debias}}^R + C_1 \\
&= L_{\text{debias}}^R - \frac{\Delta}{|\mathcal{U}||\mathcal{I}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}|p_O^I - \hat{p}_O^I| + C_2
\end{aligned}$$

where $C_1 = C + \Delta(\kappa_2-\kappa_1)$, $C_2 = C_1 + 2\kappa_2\Delta$.

C.2 Bounding L_{ideal}^S with L_{debias}^S

Similar to the above section, we have the following inequality: $L_{\text{ideal}}^S - E_{p_O^{I*},p_O^{F*}}[L_{\text{debias}}^S] = \frac{1}{|\mathcal{U}||\mathcal{I}||\mathcal{F}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}(1-\frac{p_O^{I*}p_O^{F*}}{p_O^I p_O^F})L_{u,i,f}^S$.
Let $\bar{s}_u = \frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}\frac{1}{|\mathcal{F}_{ui}|}\sum_{f\in\mathcal{F}_{ui}}\frac{p^I(i)p^F(f)}{p_O^I(i|u)p_O^F(f|u,i)}L_{u,i,f}^S$, then $L_{\text{debias}}^S =$

$\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\bar{s}_u$. According the Hoeffding's inequality, suppose $\bar{s}_u \in [c_u, d_u]$, $|c_u - d_u| \le B_2$, then the expectation of L^S_{debias} is bounded by the following value with probability at least $1-\eta_2$:

$$E_{p_O^{I*},p_O^{F*}}[L^S_{\text{debias}}] \le L^S_{\text{debias}} + B_2\sqrt{\frac{1}{2|\mathcal{U}|}log(\frac{2|\mathcal{H}|}{\eta_2})}. \tag{6}$$

Let $C_3 = B_2\sqrt{\frac{1}{2|\mathcal{U}|}log(\frac{2|\mathcal{H}|}{\eta_2})}$, $\beta_{u,i,f} = \frac{L^S_{u,i,f}}{|\mathcal{U}||\mathcal{I}||\mathcal{F}|p_O^I p_O^F}$, then we have the following inequality holds with probability at least $1-\eta_2$:

$$\begin{aligned}
&L^S_{\text{ideal}}\\
=&L^S_{\text{ideal}} - E_{p_O^{I*},p_O^{F*}}[L^S_{\text{debias}}] + E_{p_O^{I*},p_O^{F*}}[L^S_{\text{debias}}]\\
\le&\frac{1}{|\mathcal{U}||\mathcal{I}||\mathcal{F}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}(1-\frac{p_O^{I*}p_O^{F*}}{p_O^I p_O^F})L^S_{u,i,f} + L^S_{\text{debias}} + C_3\\
=&\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}\beta_{u,i,f}(p_O^I p_O^F - p_O^{I*}p_O^{F*}) + L^S_{\text{debias}} + C_3\\
=&\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}\beta_{u,i,f}(p_O^I p_O^F - p_O^I\hat{p}_O^F + p_O^I\hat{p}_O^F - \hat{p}_O^I\hat{p}_O^F)\\
&+ L^S_{\text{debias}} + C_4\\
=&\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}\beta_{u,i,f}p_O^I(p_O^F - \hat{p}_O^F) + \beta_{u,i,f}\hat{p}_O^F(p_O^I - \hat{p}_O^I)\\
&+ L^S_{\text{debias}} + C_4\\
\le&\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}\beta_{u,i,f}p_O^I(2\kappa_2 - |p_O^F - \hat{p}_O^F|) + \beta_{u,i,f}\hat{p}_O^F(2\kappa_2 - |p_O^I - \hat{p}_O^I|)\\
&+ L^S_{\text{debias}} + C_4\\
\le&\frac{4\Delta_1\kappa_2^2}{|\mathcal{F}|} - \frac{\Delta_1\kappa_2}{|\mathcal{U}||\mathcal{I}||\mathcal{F}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}(|p_O^F - \hat{p}_O^F|) + (|p_O^I - \hat{p}_O^I|)\\
&+ L^S_{\text{debias}} + C_4\\
=&L^S_{\text{debias}} - \frac{\Delta_1\kappa_2}{|\mathcal{U}||\mathcal{I}||\mathcal{F}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}(|p_O^F - \hat{p}_O^F|) + (|p_O^I - \hat{p}_O^I|) + C_5
\end{aligned} \tag{7}$$

where $C_4 = \Delta_1(\kappa_2^2 - \kappa_1^2) + C_3$ and $C_5 = \frac{4\Delta_1\kappa_2^2}{|\mathcal{F}|} + C_4$.

C.3 Bounding L_{ideal} with L_{debias}

Suppose we define X_1 as $L^R_{\text{ideal}} \le L^R_{\text{debias}} - \frac{\Delta}{|\mathcal{U}||\mathcal{I}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}|p_O^I - \hat{p}_O^I| + C_2$, X_2 as $L^S_{\text{ideal}} \le L^S_{\text{debias}} - \frac{\Delta_1\kappa_2}{|\mathcal{U}||\mathcal{I}||\mathcal{F}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}(|p_O^F - \hat{p}_O^F|) + (|p_O^I - \hat{p}_O^I|) + C_5$ and Y as $L^e_{\text{ideal}} \le L_{\text{debias}} + Const$. Based on the above results, we know $P(X_1) \ge 1-\eta_1$ and $P(X_2) \ge 1-\eta_2$. Since $X_1 \cap X_2 \to Y$, we have: $P(Y) \ge P(X_1 \cap X_2) = 1 - P(\overline{X_1} \cup \overline{X_2}) \ge 1 - P(\overline{X_1}) - P(\overline{X_2}) \ge 1 - \eta_1 - \eta_2$. Let $\eta = \eta_1 + \eta_2$, we have Y holds with probability at least $1-\eta$.