CI-OCM: Counterfactual Inference towards Unbiased Outfit Compatibility Modeling

Liqiang Jing
Shandong University
jingliqaing6@gmail.com

Minghui Tian
Shandong University
tianminghui99@gmail.com

Xiaolin Chen
Shandong University
cxlicd@gmail.com

Teng Sun
Shandong University
stbestforever@gmail.com

Weili Guan
Monash University
weili.guan@monash.edu

Xuemeng Song
Shandong University
sxmustc@gmail.com

ABSTRACT

As a key task to support intelligent fashion shop construction, outfit compatibility modeling, which aims to estimate whether the given set of fashion items makes a compatible outfit, has attracted much research attention. Although previous efforts have achieved compelling success, they still suffer from the spurious correlation between the category matching and outfit compatibility, which hurts the generalization of the model and misleads the model to be biased. To tackle this problem, we introduce the causal graph tool to analyze the causal relationship among variables of outfit compatibility modeling. In particular, we find that the spurious correlation is attributed to the direct effect of the category information on outfit compatibility prediction by the causal graph. To remove this bad effect from the category information, we present a novel counterfactual inference framework for outfit compatibility modeling, dubbed as CI-OCM. Thereinto, we capture the direct effect of the category information on model prediction in the training phase and then subtract it from the total effect in the testing phase to achieve debiased prediction. Extensive experiments on two splits of a widely-used dataset (*i.e.,* under the independent identically distribution and out-of-distribution assumptions) clearly demonstrate that our CI-OCM can achieve significant improvement over the existing baselines. In addition, we released our code to facilitate the research community[1].

CCS CONCEPTS

• **Information systems → Retrieval tasks and goals**; *World Wide Web.*

KEYWORDS

Outfit Compatibility Modeling; Counterfactual Inference; Fashion Analysis

[1] https://github.com/LiqiangJing/CI-OCM.

MCFR '22, October 14, 2022, Lisboa, Portugal
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9498-7/22/10…$15.00
https://doi.org/10.1145/3552468.3555363

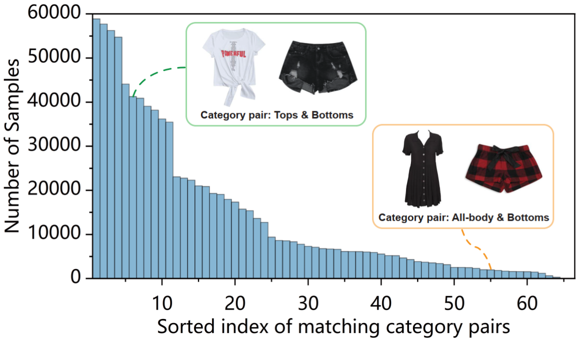


Figure 1: Compatible sample distribution over category matching pairs of Polyvore Outfits dataset, which are sorted according to their corresponding number of samples.

ACM Reference Format:
Liqiang Jing, Minghui Tian, Xiaolin Chen, Teng Sun, Weili Guan, and Xuemeng Song. 2022. CI-OCM: Counterfactual Inference towards Unbiased Outfit Compatibility Modeling. In *Proceedings of the 1st Workshop on Multimedia Computing towards Fashion Recommendation (MCFR '22), October 14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3552468.3555363

1 INTRODUCTION

Recent years have witnessed an increasing trend of purchasing clothes online because of the convenience of e-commerce. One typical clothing shopping scenario is that users would like to buy compatible complementary garments to match those they already have, to make a favorable outfit. For example, the user may want to buy a casual Jeans to match a white T-shirt she/he previously bought. Therefore, compatible garment recommendation has great potential for boosting sales. Towards this end, its essential task of outfit compatibility modeling, which aims to automatically determine whether the fashion items in an outfit are compatible, has received extensive research attention.

Early studies of outfit compatibility modeling focus on mining the pair-based matching relationship between fashion items in one outfit. For example, the Type-aware method [37] utilizes an end-to-end model to simultaneously learn the pair-based similarity and compatibility in one outfit. Obviously, the pair-based methods ignore the contextual relationship among all fashion items in the whole outfit. To learn the contextual relationship, many sequence-based works [11, 17] emerge, which capture the relationship of the whole outfit in a predefined order. The sequence-based methods have achieved compelling success, however, they are sensitive to the position of the fashion items in the outfit and hence may result in a non-robust prediction. In the past few years, the advanced graph-based methods have been proposed, which treat an outfit as a graph where fashion items are nodes.

Although existing efforts have attained promising results, they may suffer from fitting the spurious correlations in category matching [10, 37], rather than learning visual matching relationships. This tends to hurt the generalization of the model and easily leads to suboptimal performance. We argue that the root cause of this problem is that the dataset inevitably has a bias in the collection process. As shown in Figure 1, the distribution of category matching pairs has a long-tail characteristic. If a model is trained to estimate the head category matching a bigger compatibility score, the head category matching is more likely to prevail over the tail category matching during the testing phase. Even if the category matching bias misleads the model severely, it also can benefit the model with some prior knowledge, *e.g.,* tops and bottoms are more likely to match than bottoms and all-body. Therefore, our key challenge is how to dislodge the harmful effect and keep the beneficial effect of the category matching on outfit compatibility.

To tackle this challenge, we present a counterfactual inference framework for outfit compatibility modeling, dubbed CI-OCM. To sort out cause and effect between variables among outfit compatibility modeling, we embrace the causal graph [22], where we analyze the harmful effect and beneficial effect of the category matching on the compatibility prediction. In particular, the bad effect is attributed to its direct effect on the outfit compatibility prediction and the good effect comes from the indirect effect of the category matching on the outfit compatibility prediction by considering the complete outfit representation. Finally, we resort to counterfactual inference [23] to eliminate the bad effect and keep the good effect of the category matching on the outfit compatibility prediction. To verify the generalization of the model, we conduct extensive experiments on two splits of the Polyvore Outfits dataset. In addition to following the original independent and identically distributed (IID) assumption [4], we also carry out experiments on the dataset, which is split based on the out-of-distribution (OOD) assumption.

Our main contribution can be summarized as follows:

- We explore the causal relationship among variables existing in outfit compatibility modeling by the causal graph, pointing out the harmful effect hidden in the category matching.
- We present a novel counterfactual inference framework for outfit compatibility modeling, which eliminates the harmful effect and keeps the beneficial effect of the category matching on outfit compatibility prediction.
- We conducted extensive experiments on two splits of a widely-used public dataset, and the result shows the superiority of CI-OCM compared with existing baselines.

The rest of the paper is organized as follows. Section 2 briefly reviews the related work. In Section 3, we detail the proposed CI-OCM. Experimental results are given in Section 4, followed by the conclusion and future work in Section 5.

2 RELATED WORK

Our work is related to outfit compatibility modeling and causal inference.

2.1 Outfit Compatibility Modeling

Outfit compatibility modeling aims to determine whether the items in an outfit are compatible, which has been catching more and more attention due to its enormous application in e-commerce. Existing methods of outfit compatibility modeling can be roughly divided into three categories. The first is the pair-based outfit compatibility modeling methods [30, 31, 35, 37], which aims to learn the compatibility of the outfit based on pairs of fashion items. Obviously, this category of methods overlooks the context relationship among fashion items in the whole outfit. In light of this, several researchers resorted to sequence-based methods, which capture the context relationship among items based on a pre-defined sequence of fashion items. For example, Han *et al.* [11] views an outfit as a fixed-order sequence and uses a bi-directional LSTM (Bi-LSTM) [9] model to learn the compatibility between items. Despite their compelling success, sequence-based methods are sensitive to the order of fashion items, *i.e.,* their performances highly rely on the order information of fashion items in the outfit. Therefore, the third category of methods, namely the graph-based methods, aiming to learn the outfit compatibility based on graph neural networks [29] is proposed and achieves remarkable performance [5, 6, 14].

Although existing methods have achieved great success, they overlook the spurious correlation between the category of the fashion item and the outfit compatibility, which may hurt the robustness of existing models and cause performance degradation.

2.2 Casual Inference

Causal inference is a new science to study the causal relationship between variables [8, 21, 23]. Its general purpose is to empower models with the ability to pursue causal effects so as to obtain a more robust prediction. Since its advance in mining the causal effects, the causal inference technology has been widely applied to multiple domains, such as recommendation system [39, 40, 43], text classification [7, 24], multimodal sentiment analysis [34], scene graph generation [36], and visual question answer [18, 38], and video moment retrieval [41], demonstrating its effectiveness in eliminating the harmful effects of dataset bias on models. The causal inference methods applied to deep learning techniques can be roughly divided into two groups. The first group [3, 28, 41] introduces do-operation based on backdoor adjustment, which forces the model to incorporate all possible inputs into consideration. The second group of methods introduces counterfactual inference to achieve unbiased prediction based on biased data [16, 18, 24]. Although causal inference has achieved enormous success in many research domains, there is relatively sparse literature on the outfit compatibility modeling task.

3 METHOD

In this section, we first introduce the relevant theoretical foundations in the area of causal inference [20, 22, 23, 26], and then formulate the research problem. Finally, we integrate causal inference to the outfit compatibility modeling task and detail the proposed model.

3.1 Preliminaries

Causal Graph. The causal graph reflects the causal relationships between variables, which can be exhibited as a directed acyclic graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. $\mathcal{V}$ is the set of nodes in the directed acyclic graph, which represents a set of variables. $\mathcal{E}$ denotes the set of edges in the directed acyclic graph, which represents the set of causal relationships among variables. We adopt capital letters to denote random variables and lowercase letters to refer to the observed values of random variables. Figure 2(a) illustrates a causal graph example that consists of three variables (*i.e.,* X, Y and M). From this graph, the variable X directly affects the variable Y (*i.e.,* $X \longrightarrow Y$) and indirectly affects the variable Y by the mediator variable M (*i.e.,* $X \longrightarrow M \longrightarrow Y$).

Counterfactual. To formulate the causal relationship between variables, we resort to the counterfactual notations[23]. When setting X to x and M to m, the value of Y can be denoted as follows,

$$Y_{x,m} = Y(X = x, M = m), \tag{1}$$

where $Y_{x,m}$ denotes the value of Y when it receives $X = x$ and $M = m$. Suppose in the real scenario, the variable X takes the value x and the variable M takes the value m (*i.e.,* $Y_{x,m} = Y_{x,M_x} = Y(X = x, M = M_x) = Y(X = x, M = M(X = x))$). In the factual scenario, X takes the same value for both M and Y. However, in the counterfactual scenario, X can simultaneously set to different values (*e.g.,* x and x^*) for M and Y. For example, in the causal effect of after-school classes (X) on the test score (Y), the amount of homework serves as a mediator (M). In the real scenario, a person can only choose between "attending after-school classes" ($X = x$) and "not attending after-school classes" ($X = x^*$), and the number of homework is denoted as M_x and M_{x^*} respectively. So the test score can only be $Y(X = x, M = M_x)$ or $Y(X = x^*, M = M_{x^*})$. But in the counterfactual scenario, we can infer the situation that "what his/her test score would be if a person has the homework of after-school class and does not attend after-school classes", *i.e.,* $Y(X = x^*, M = M_x)$. Figure 2(c) and (d) provides two examples of the counterfactual notations.

Causal Effect. The causal effect of X on Y is the comparison of the outcome variable Y when its ancestor variable X is under different treatments (*i.e.,* X is set to different values) [25, 27], which is also named as the total effect (TE). Usually, $X = x$ means "under the treated condition" and $X = x^*$ means "under the untreated condition". For example, in the question "whether after-school classes improve the test score", $X = x$ means "taking after-school classes" while $X = x^*$ means "not taking after-school classes". Under the two treatments $X = x$ and $X = x^*$, the TE of $X = x$ on Y can be defined as follows,

$$TE = Y_{x,M_x} - Y_{x^*,M_{x^*}}, \tag{2}$$

where Y_{x,M_x} and $Y_{x^*,M_{x^*}}$ denotes the test score Y when receiving after-school classes or not. The natural direct effect (NDE) of X on the outcome variable Y is defined as the effect of a change of X from x^* to x when the mediator M is unaffected by the change of X, which is formulated as:

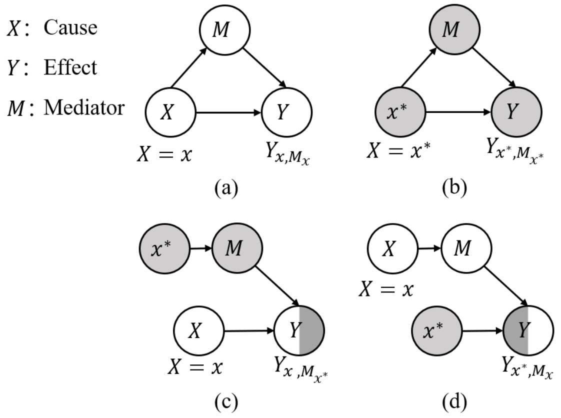


Figure 2: Example of counterfactual notations. White nodes are at $X = x$, gray nodes are at $X = x^*$, and half white and half gray nodes indicate the prediction of Y when X takes different values for the direct and indirect paths. (a) and (b) show examples of factual scenarios. (c) and (d) show examples of counterfactual scenarios.

$$NDE = Y_{x,M_{x^*}} - Y_{x^*,M_{x^*}}. \tag{3}$$

Meanwhile, the TE can be decomposed into NDE and the total indirect effect (TIE) [23] as follows,

$$TE = NDE + TIE. \tag{4}$$

TIE is defined as the effect of a change of X from x^* to x on Y by the indirect path $X \longrightarrow M \longrightarrow Y$ when $X \longrightarrow Y$ is blocked, which is formulated as,

$$TIE = TE - NDE = Y_{x,M_x} - Y_{x,M_{x^*}}. \tag{5}$$

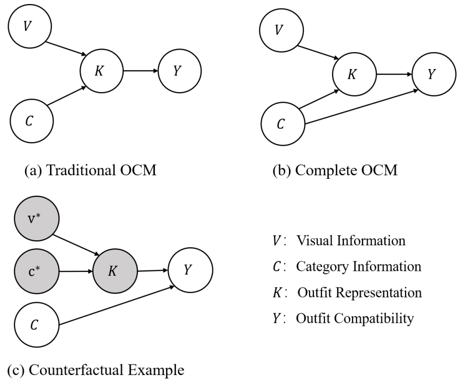
3.2 Problem Formulation

Formally, we first declare some notations. We use boldface uppercase letters (e.g., $\mathbf{X}$) and boldface lowercase letters (e.g., $\mathbf{x}$) to denote matrices and vectors, respectively. We employ non-bold letters (e.g., i and N) to represent scalars. If not clarified, all vectors are in the column forms.

Suppose we have a training set $\mathcal{D} = \{(O^1, y_1), \cdots, (O^T, y_T)\}$, where each sample consists of an outfit O^i and its compatibility label y_i. In partIcular, $y_i = 1$, if fashion items in O^i are compatible, and $y_i = 0$ otherwise. T is the total number of samples in the training set. Each outfit is associated with a set of m fashion items, defined as $O = \{o_1, o_2, \cdots, o_m\}$ (we omit the index of O_i for simplify), where o_j is the j-th fashion item in the outfit O. Each fashion item o_j has its corresponding product image V_j and category information $C_u \in C$, $u \in \{1, 2, \cdots, N_c\}$, where $C = \left\{c_1, c_2, \cdots, c_{N_c}\right\}$ refers to the complete set of categories covering all items and N_c is the total number of categories.

The goal of the outfit compatibility modeling task is to assess the compatibility score of a given outfit O through designing a model $\mathcal{F}$ as follows,

$$\hat{y} = \mathcal{F}\left(\left\{O_j\right\}_{j=1}^{m} \mid \Theta_F\right), \tag{6}$$



(a) Traditional OCM

(b) Complete OCM

(c) Counterfactual Example

Figure 3: Causal graphs in different cases. (a) shows the causal graph of traditional outfit compatibility modeling. (b) shows the causal graph of outfit compatibility modeling considering the direct effect of category. (c) shows the causal graph of outfit compatibility modeling in counterfactual scenarios.

where $\hat{y}$ is the prediction compatibility score for a given outfit, and Θ_F is a set of parameters of the model $\mathcal{F}$ to be learned.

3.3 Counterfactual Inference for Outfit Compatibility Modeling

In this subsection, we first propose the causal graph of outfit compatibility modeling, and then detail the CI-OCM implementation. Finally, we show the training and testing processes of the CI-OCM.

3.3.1 Causal Graph of Outfit Compatibility Modeling. Analyzing the existing work on outfit compatibility modeling, we found that most of them consist of two main components: an outfit representation modeling component and a compatibility prediction component. The former is used to encode visual and category information of the outfit to learn the outfit representation, and the latter aims to calculate the compatibility score based on the outfit representation. In fact, we can abstract the typical outfit compatibility modeling framework with a causal graph, as shown in Figure 3(a), where V, C, K, and Y denote the visual information, category information, outfit representation, and outfit compatibility prediction, respectively.

In particular, paths $V \rightarrow K$ and $C \rightarrow K$ represent the direct effects of the visual information and category information on the outfit representation, respectively, while the path $K \rightarrow Y$ refers to the direct effect of the outfit representation on the compatibility prediction. Therefore, the outfit representation can be expressed as $K_{c,v} = K(C = c, V = v)$, and the compatibility score can be expressed as $Y_k = Y(K = k)$. Combining the above two formulas, we obtain the complete compatibility score prediction expression $Y_k = Y_{K_{c,v}} = Y(K = K(C = c, V = v))$.

Previous works ignore the direct effect of the category information on outfit compatibility prediction, which is the harmful effect of category information due to the spurious correlation between the category matching and outfit compatibility. Therefore, we take into account the direct effect of category information on outfit compatibility prediction and redraw the complete causal graph as shown in Figure 3(b), where the path $C \rightarrow Y$ represents the direct effect of category information on outfit compatibility prediction. In Figure 3(b), the expression of the compatibility score Y can be rewritten as $Y_{c,k} = Y_{c,K_{c,v}} = Y(C = c, K = K(C = c, V = v))$ following the definition in Section 3.1.

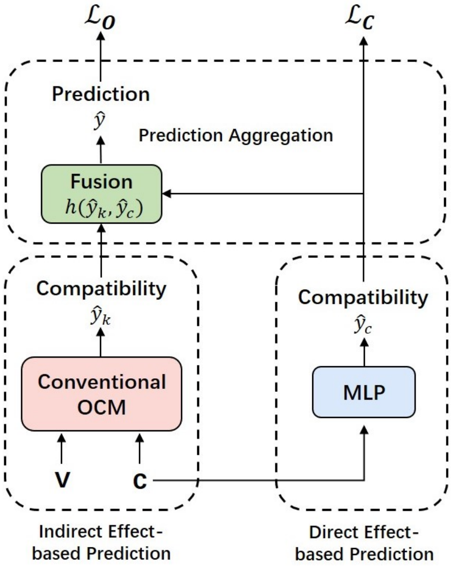


Figure 4: The framework of CI-OCM, which mainly consists of three components: indirect effect-based prediction, direct effect-based prediction, and prediction aggregation.

3.3.2 CI-OCM. In order to mitigate the harmful effect of the category information on the outfit compatibility prediction, we design a novel causal inference framework for outfit compatibility modeling. Following the causal graph in Figure 3(b), the proposed CI-OCM can be realized with three key modules: indirect effect-based prediction, direct effect-based prediction, and prediction aggregation, where the former two modules correspond to paths $(C, V) \rightarrow K \rightarrow Y$ and $C \rightarrow Y$, respectively, while the last module is used for calculating the prediction $Y_{c,k}$.

Indirect Effect-based Prediction. This module aims to measure the outfit compatibility based on the entangled effect of the visual and category information, i.e., the indirect path $(C, V) \rightarrow K \rightarrow Y$. Let $\hat{y}_k = Y_k = Y(K = K(C = c, V = v))$ represent the predicted outfit compatibility based on both visual and category information of composing items of an outfit. For this module, we can utilize existing outfit compatibility modeling work, such as the state-of-the-art OCM-CF [33], to implement it.

Direct Effect-based Prediction. This module targets at evaluating the outfit compatibility simply based on the category information of items, i.e., the direct path $C \rightarrow Y$. Let $\hat{y}_c = Y_c = Y(C = c)$ denote the compatibility prediction score obtained by this module. For simplify, we implement the category compatibility module using a multi-layer perceptron (MLP). To embed the category, we first introduce an embedding matrix to convert the category of each fashion item to a vector as follows,

$$\mathbf{c}_i = \mathbf{W}_c \hat{\mathbf{c}}_i, i = 1, \cdots, m, \tag{7}$$

where $\mathbf{W}_c \in \mathbb{R}^{D_c \times N_c}$ is the to-be-learned embedding matrix. $\hat{\mathbf{c}}_i \in \mathbb{R}^{N_c}$ is the one-hot vector indicating the index of the category of i-th item in the category set C. $\mathbf{c}_i \in \mathbb{R}^{D_c}$ is the embedding of the category of i-th fashion item in the outfit. Next, we concatenate all category vectors of the outfit to input into the MLP as follows,

$$\begin{aligned} \hat{y}_c &= Y_c(C = c) \\ &= \mathbf{W}_3 \left[\psi\left(\mathbf{W}_2 \left[\psi\left(\mathbf{W}_1[\mathbf{c}_1; \cdots; \mathbf{c}_m] + \mathbf{b}_1\right)\right] + \mathbf{b}_2\right)\right] + b_3, \end{aligned} \tag{8}$$

where [;] denote the concatenate operation, $\mathbf{W}_1 \in \mathbb{R}^{D \times m * D_c}$, $b_1 \in \mathbb{R}^D$, $\mathbf{W}_2 \in \mathbb{R}^{D \times D}$, $b_2 \in \mathbb{R}^D$, $\mathbf{W}_3 \in \mathbb{R}^{1 \times D}$, and $b_3 \in \mathbb{R}^1$ are the parameters of the fully connected layer to be learned. $\psi(\cdot)$ is the Relu activation function.

Prediction Aggregation. To achieve the overall compatibility prediction $Y_{c,k}$, we utilize a fusion function $h(\cdot)$ to aggregate the compatibility prediction scores of the two modules (*i.e.,* outfit compatibility module and category compatibility module) into a final prediction score. In particular, for the fusion strategy, we use the RUBi [1] strategy, which has demonstrated its advance in many feature aggregation tasks [2, 18], to obtain the compatibility prediction score $\hat{y}$ as follows,

$$\begin{cases} \hat{y} = Y(C = c, K = K(C = c, V = v)) = h\left(\hat{y}_k, \hat{y}_c\right), \\ \quad h\left(\hat{y}_k, \hat{y}_c\right) = RUBi\left(\hat{y}_k, \hat{y}_c\right) = \hat{y}_k \sigma\left(\hat{y}_c\right), \end{cases} \tag{9}$$

where $\sigma(\cdot)$ is a sigmoid function that scales $\hat{y}_c$ to a matching probability between $[0, 1]$ to adjust the dependence of the prediction score on the traditional outfit compatibility prediction $\hat{y}_k$.

3.3.3 *Training and Testing.*

We detail the training and testing stages of the proposed CI-OCM in this subsection.

Training. In the training stage, similar to the conventional model, we supervise the model by binary cross entropy loss function [10] as follows,

$$\mathcal{L}_O = \sum_{(O,y) \in \mathcal{D}} -y \log\left(\sigma\left(\hat{y}\right)\right) - (1 - y) \log\left(1 - \sigma\left(\hat{y}\right)\right), \tag{10}$$

where $\sigma(\cdot)$ denotes the sigmoid function, $\hat{y}$ denotes the compatibility prediction score, and y is the ground truth which denotes whether the outfit is compatible or not.

To make the category module learn category matching bias in the training set, we also introduce a binary cross entropy loss function to supervise $\hat{y}_c$. Formally, the loss function is defined as,

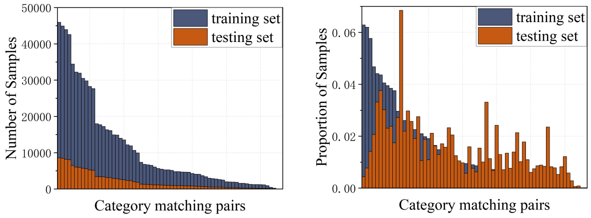
$$\mathcal{L}_C = \sum_{(O,y) \in \mathcal{D}} -y \log\left(\sigma\left(\hat{y}_c\right)\right) - (1 - y) \log\left(1 - \sigma\left(\hat{y}_c\right)\right). \tag{11}$$

where $\hat{y}_c$ is the prediction compatibility score of the category module. Towards the optimization of CI-OCM, we derive the total loss by adding L_C and L_O as follows,

$$\mathcal{L} = \mathcal{L}_O + \alpha \mathcal{L}_C, \tag{12}$$

where α is the trade-off hyper-parameter.

Counterfactual Inference. As mentioned in Section 3.3.1, the key to eliminating the category matching bias is to remove the direct effect of category information on the compatibility prediction (*i.e.,* the path $C \rightarrow Y$). In the following, we will use a counterfactual inference approach to remove the adverse effect of category matching bias on the compatibility prediction in the testing stage.



(a) Polyvore Outfits.

(b) Polyvore Outfits-OOD.

Figure 5: Distribution of category pairs for the Polyvore Outfits and Polyvore Outfits-OOD datasets. For clarity, we show the number of samples rather than the proportion of samples for the original Polyvore Outfits dataset. The horizontal coordinate refers to the sorted category pair index according to the number of corresponding samples in the training set.

As shown in Figure 3(b), C affects Y through two paths, the indirect path $C \rightarrow K \rightarrow Y$ and the direct path $C \rightarrow Y$. Following the counterfactual notations defined in Section 3.1, we can obtain NDE of C on Y as follows,

$$NDE = Y\left(C = c, K = K_{c^*,v^*}\right) - Y\left(C = c^*, K = K_{c^*,v^*}\right), \tag{13}$$

where c^*, v^* refer to the reference values of C, V (*i.e.,* "under the untreated condition"). According to Eqn.(2), the TE of C to Y can be written as,

$$TE = Y\left(C = c, K = K_{c,v}\right) - Y\left(C = c^*, K = K_{c^*,v^*}\right). \tag{14}$$

According to section 3.3.1, the elimination of the category matching bias can be achieved by subtracting the NDE from the TE, expressed formally as follows,

$$TE - NDE = Y\left(C = c, K = K_{c,v}\right) - Y\left(C = c, K = K_{c^*,v^*}\right). \tag{15}$$

Since the outfit compatibility score is calculated according to Eqn.(9), we have $Y(C = c, K = K_{c,v}) = \hat{y}_k \sigma(\hat{y}_c)$ and $Y(C = c, K = K_{c^*,v^*}) = \hat{y}_* \sigma(\hat{y}_c)$, where $\hat{y}_*$ is a hyper-parameter referring to the $\hat{y}_k$ in the case of $K = K_{c^*,v^*}$, which can be tuned according to the validation set. Ultimately, we can derive the unbiased outfit compatibility prediction $\hat{y}$ during the testing phase as follows,

$$\hat{y} = \hat{y}_k \sigma\left(\hat{y}_c\right) - \hat{y}_* \sigma\left(\hat{y}_c\right). \tag{16}$$

4 EXPERIMENT

To evaluate the performance of the proposed CI-OCM, we conducted extensive experiments on two splits of a widely-used dataset (*i.e.,* under the IID and OOD assumptions) to answer the following research questions.

- **RQ1.** Does CI-OCM outperform state-of-the-art baselines?
- **RQ2.** How is the qualitative performance of CI-OCM? do the hyperparameters affect the model performance?

4.1 Settings

4.1.1 *Dataset.*

To verify the effectiveness of the proposed CI-OCM, we adopted the Polyvore Outfits dataset [37]. Notably, the split of the Polyvore Outfits dataset follows the IID assumption, which is similar to most works of deep learning [11, 32]. Thus the deleterious effect of the long-tailed distribution on the model's generalization cannot be observed. For this reason, we constructed an OOD dataset where distributions of matching category pairs in the testing and

Algorithm 1 The split of Polyvore Outfits under OOD assumption.

```
Input: Complete data set $\mathcal{D}$, the maximum capacity of the testing set $N_t$.
Output: OOD testing set $\mathcal{D}_{test}$ and remaining set $\mathcal{D}_{remain}$.
Initialize parameters: $\mathcal{D}_{test} = \emptyset$ ,$\mathcal{D}_{remain} = \emptyset$.
Compute the distribution of proportion $P$ of each compatible category pair in all compatible category pairs.
Set $P' = \frac{1/P}{SUM(1/P)}$.
Get the numbers of all category pairs $\phi_{ood}$ in the testing set by set $\phi_{ood} = P' * N_t$.
repeat
    Randomly select an outfit $O$ from the dataset $\mathcal{D}$.
    if $O$ corresponds to the category matching distribution $\phi_o < \phi_{ood}$ then
        Set $\mathcal{D}_{test} = \mathcal{D}_{test} \cup \{O\}$, $\phi_{ood} = \phi_{ood} - \phi_o$
    else
        $\mathcal{D} = \mathcal{D} - \{O\}$
        $\mathcal{D}_{remain} = \mathcal{D}_{remain} \cup \{O\}$
    end if
until $D$ is $\emptyset$ or $|\mathcal{D}_{test}| = N_t$.
```

training sets are significantly different, to evaluate the model's generalization ability.

Polyvore Outfits. The Polyvore dataset is collected from the Polyvore fashion website[2], where each outfit is comprised of rich multimodal information, *e.g.,* product images, text descriptions, attribute information, and category information. The Polyvore Outfits dataset contains $68,306$ outfits with $365,054$ fashion items, and the average number of items in one outfit is 5.3. The numbers of outfits in the training set, validation set, and testing set are $53,306$, $5,000$, and $10,000$, respectively. To observe the distributions of the category matching pairs in this dataset, we visualized the distribution of category matching pairs on the training and testing sets of the Polyvore Outfits dataset in Figure 5(a). It can be seen that the category matching distributions are apparent long-tail distributions in training and testing sets, and the distributions on the training and testing sets are very similar. The similar distributions in training and testing sets mean that the bias in the training and testing sets is almost the same. Hence the generalization ability of our de-biased CI-OCM cannot be verified.

Polyvore Outfits-OOD. Considering that the existing dataset is not suitable for testing the harmful effects of category matching bias on the model for the aforementioned reason, we propose Algorithm 1 to construct Polyvore Outfits-OOD datasets with significantly different distributions for the training and testing sets. Algorithm 1 uses the idea of the Knapsack problem. We first calculated the maximum number of category matching pairs in the testing set, then used the greedy algorithm to randomly extract outfits into the testing set according to the category matching capacity. We extracted the testing set $\mathcal{D}_{test}$ containing $2,851$ outfits from the complete Polyvore Outfits dataset with Algorithm 1, and then we randomly selected $3,000$ outfits from the remaining outfits set $\mathcal{D}_{remain}$ as the validation set. As a result, we obtained the Polyvore Outfits-OOD dataset, where the number of outfits in training set, validation set, and testing set are $62,455$, $3,000$, and $2,851$, respectively. In addition, we also visualized the distribution of category pairs in the training and testing sets in Figure 5(b). It can be seen that the category matching pairs are significantly different in training set and testing set.

4.1.2 Implementation Details. We implemented the proposed CI-OCM with Pytorch[3] [19]. We use OCM-CF [33] for $Y_k(\cdot)$, and the model parameters following the setting in the original paper. We optimized all models using the Adam [12] optimizer, and set the learning rate and the batch size as $5e^{-5}$ and 64, respectively. We used the exponential decay for learning rate adjustment and *gama* is set to 0.985. We applied the weighting parameters α within $\{1e^{-5}, 1e^{-4}, 1e^{-3}, 1e^{-2}\}$ and obtained the optimal performance at $1e^{-3}$. We utilized grid search strategy for optimal $\hat{y}_*$ in Eqn.(16) in the range $[0, 10]$ with the step size of 1 and achieved the optimal performance at 1.

4.1.3 Evaluation Metric. Most existing work [6, 11, 35, 37] on outfit compatibility modeling evaluates the model via two tasks: fill-in-the-blank and outfit compatibility prediction. However, since all candidate items in the fill-in-the-blank task belong to the same category, it cannot effectively evaluate our model in dealing with the category matching distribution bias. Therefore, in this paper, we only employed the outfit compatibility prediction task, which evaluates the compatibility of an outfit containing a number of fashion items, to verify the proposed CI-OCM. Similar to existing studies [11, 37], we utilize AUC (area under the ROC curve) [42] as the evaluation metric for the outfit compatibility prediction task.

4.1.4 Baselines. To validate the effectiveness of our CI-OCM, we compared it with the following state-of-the-art methods, including pair-based, sequence-based, and graph-based models.

- **Bi-LSTM [11]** arranges all items in an outfit into a predefined order based on categories. It treats outfit compatibility modeling as a sequential prediction problem and uses a Bi-LSTM for prediction. For fairness, we removed textual information from the released model.
- **Type-aware [37]** evaluates the compatibility of fashion item pairs based on a category space rather than a separate generic space. We used the author's code and retrained the model with only image information.
- **SCE-NET [35]** learns different similarity conditions and uses a weighting module to combine different embeddings to represent fashion item pairs. Similar to Type-aware, we removed the regularization of textual information from the released model.
- **NGNN [6]** maps the features of fashion items into the category space to construct the item graph, where the node embeddings are updated using Gated Recurrent Unit (GRU) [15], and an attention mechanism is used to summarize the clothing compatibility scores.
- **Context-aware [5]** constructs a graph containing all fashion items in the dataset. Each node will receive information from the outfit evolving the node and other outfits to learn an

[2] https://www.polyvore.com/.

[3] https://pytorch.org/.

Table 1: Performance comparison among different methods with respect to AUC metric. The best results are in boldface, while the second best are underlined.

Method	Polyvore Outfits	Polyvore Outfits-OOD
Bi-LSTM	0.68	0.65
SCE-NET	0.83	0.82
Type-aware	<u>0.87</u>	0.78
NGNN	0.75	0.68
Context-aware	0.81	0.77
HFGN	0.84	0.70
OCM-CF	**0.92**	<u>0.84</u>
CI-OCM	**0.92**	**0.86**

embedding containing information of the contextual items. We calculated the compatibility score based on node embedding of the outfit in the testing phase.

- **HFGN [14]**, different from NGNN, designs an R-view attention graph and an R-view score graph for evaluating outfit compatibility scores on a category-oriented outfit graph with GCNs [13].
- **OCM-CF [33]** combines NGNN outfit representation learning and hidden complementary factor learning to facilitate outfit compatibility evaluation from multiple hidden factors.

4.2 On Model Comparison (RQ1)

Table 1 shows the performance comparison among different methods for outfit compatibility prediction on Polyvore Outfits and Polyvore Outfits-OOD. For clarity, we divided the baselines into three groups: sequence-based method (*i.e.,* Bi-LSTM), pair-based methods (*i.e.,* Type-aware and SCE-NET), and graph-based methods (*i.e.,* NGNN, Context-aware, HFGN and OCM-CF). From Table 1, we have the following observations. 1) Bi-LSTM performs the worst on the outfit compatibility prediction task compared to other baselines. This demonstrates that it is unreasonable to present the outfit as an ordered list of fashion items and may cause error accumulation because this method predicts the compatibility between the next item and the previous one. 2) Surprisingly, most graph-based approaches (*i.e.,* , NGNN, Context-aware, and HFGN) do not surpass the pair-based approaches (*i.e.,* SCE-NET and Type-aware). This may be due to following reasons. NGNN and HFGN initialize fashion item nodes in a category space, which may mislead the model to rely too much on category information for prediction, resulting in incorrect outfit compatibility predictions. Regarding Context-aware, it only embeds each item into a single space, ignoring the category information of items, which leads to the poor performance. 3) The proposed CI-OCM outperforms the optimal baseline model on the Polyvore Outfits-OOD dataset, and achieves comparative performance with state-of-the-art baselines on the Polyvore Outfits dataset. This verifies that our model can mitigate the deleterious effects of category matching bias and has better generalization ability. 4) For all baselines, as expected, they all suffer from performance decrease when the testing set is set to the OOD case. This indicates that existing methods do have poor generalization ability due to the harmful effect of the category bias on the outfit compatibility prediction.

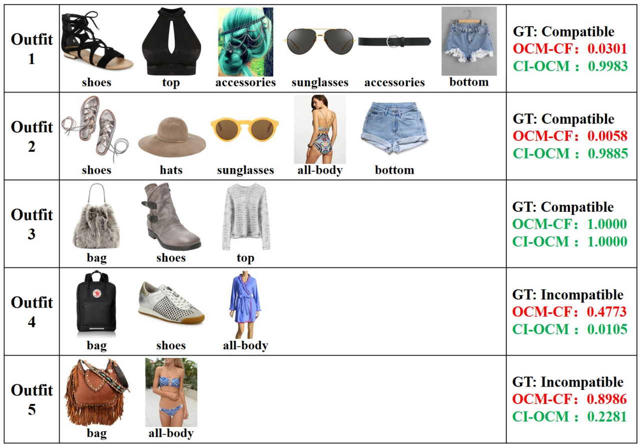


Figure 6: Prediction results of CI-OCM and OCM-CF over 5 testing outfits. The text below each image indicates the item's coarse-grained category. "GT" means ground truth. The results in green and red colors indicate the correct and incorrect predictions, respectively.

4.3 On Case Study (RQ2)

To gain more deep insights into the influence of counterfactual inference, we intuitively compared CI-OCM and the best baseline (*i.e.,* OCM-CF) with several testing outfits in Figure 6. Towards comprehensive comparison, we selected three kinds of outfits: a) compatible outfit with tail category matching pairs (*i.e., Outfit 1* and *Outfit 2*), b) compatible outfit with head category matching pairs (*i.e., Outfit 3*), and c) incompatible outfit with head category matching pairs (*i.e., Outfit 4* and *Outfit 5*).

As we can see, CI-OCM performs better than OCM-CF in cases of both compatible outfits with tail category matching pairs and incompatible outfits with head category matching pairs. Specifically, for *Outfit 1* and *Outfit 2* that contain tail category matching pairs (*e.g.,* (bottoms, hats), (sunglasses, hats), and (hats, all-body)), CI-OCM correctly predicts their compatibility, while OCM-CF may be misled by the tail category pairs in these two outfits and hence gives the incorrect prediction. Similar observation can be found for *Outfit 4* and *Outfit 5*, which are incompatible outfits with head category matching pairs. These observations validate that the category matching bias does exist and the effectiveness of integrating counterfactual inference in eliminating the category matching bias towards unbiased outfit compatibility modeling. In addition, we also observed that both CI-OCM and OCM-CF can acquire the accurate compatibility score concerning *Outfit 3*, which is a compatible outfit with head category matching pairs (*e.g.,* (bag, shoes), (bag, tops), and (shoes, tops)). This suggests that equipped with counterfactual inference, our model can still keep the beneficial effect of the category matching on outfit compatibility modeling.

5 CONCLUSION AND FUTURE WORK

In this work, we analyze the causal relationship among variables in outfit compatibility modeling with the causal graph, and present a novel counterfactual inference framework for unbiased outfit compatibility modeling, named CI-OCM. Extensive experiments on two splits (*i.e.,* under the independent identically distribution and out-of-distribution assumptions) of a widely-used benchmark clearly demonstrate the superiority of our CI-OCM. Importantly, we empirically found that the category matching bias does exist, and our CI-OCM can eliminate such category matching bias. In the future, we plan to explore the unbiased personalized outfit compatibility modeling and multimodal outfit compatibility modeling.

REFERENCES

[1] Rémi Cadène, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, and Devi Parikh. 2019. RUBi: Reducing Unimodal Biases for Visual Question Answering. In *Annual Conference on Neural Information Processing Systems.* 839–850.

[2] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. 2020. Counterfactual Samples Synthesizing for Robust Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition.* IEEE/CVF, 10797–10806.

[3] Konstantina Christakopoulou, Madeleine Traverse, Trevor Potter, Emma Marriott, Daniel Li, Chris Haulk, Ed H. Chi, and Minmin Chen. 2020. Deconfounding User Satisfaction Estimation from Response Rate Bias. In *Conference on Recommender Systems.* ACM, 450–455.

[4] Aaron Clauset. 2011. A brief primer on probability distributions. In *Santa Fe Institute.*

[5] Guillem Cucurull, Perouz Taslakian, and David Vázquez. 2019. Context-Aware Visual Compatibility Prediction. In *Conference on Computer Vision and Pattern Recognition.* IEEE/CVF, 12617–12626.

[6] Zeyu Cui, Zekun Li, Shu Wu, Xiaoyu Zhang, and Liang Wang. 2019. Dressing as a Whole: Outfit Compatibility Learning Based on Node-wise Graph Neural Networks. In *The World Wide Web Conference.* ACM, 307–317.

[7] Fuli Feng, Jizhi Zhang, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Empowering Language Understanding with Counterfactual Reasoning. In *Findings of the Association for Computational Linguistics.* ACL, 2226–2236.

[8] Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. 2016. *Causal inference in statistics: A primer.* John Wiley & Sons.

[9] Alex Graves. 2012. *Supervised Sequence Labelling with Recurrent Neural Networks.* Studies in Computational Intelligence, Vol. 385. Springer.

[10] Weili Guan, Haokun Wen, Xuemeng Song, Chung-Hsing Yeh, Xiaojun Chang, and Liqiang Nie. 2021. Multimodal Compatibility Modeling via Exploring the Consistent and Complementary Correlations. In *Proceedings of the International Conference on Multimedia.* ACM, 2299–2307.

[11] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S. Davis. 2017. Learning Fashion Compatibility with Bidirectional LSTMs. In *Proceedings of the International Conference on Multimedia.* ACM, 1078–1086.

[12] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations.*

[13] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations.*

[14] Xingchen Li, Xiang Wang, Xiangnan He, Long Chen, Jun Xiao, and Tat-Seng Chua. 2020. Hierarchical Fashion Graph Network for Personalized Outfit Recommendation. In *Proceedings of the International Conference on Research and Development in Information Retrieval.* ACM, 159–168.

[15] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. 2016. Gated Graph Sequence Neural Networks. In *International Conference on Learning Representations.*

[16] Rishabh Mehrotra, Prasanta Bhattacharya, and Mounia Lalmas. 2020. Inferring the Causal Impact of New Track Releases on Music Recommendation Platforms through Counterfactual Predictions. In *Conference on Recommender Systems.* ACM, 687–691.

[17] Takuma Nakamura and Ryosuke Goto. 2018. Outfit Generation and Style Extraction via Bidirectional LSTM and Autoencoder. *CoRR* abs/1807.03133 (2018).

[18] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. Counterfactual VQA: A Cause-Effect Look at Language Bias. In *Conference on Computer Vision and Pattern Recognition.* IEEE/CVF, 12700–12710.

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Annual Conference on Neural Information Processing Systems.* 8024–8035.

[20] Judea Pearl. 2001. Direct and Indirect Effects. In *Proceedings of the Conference in Uncertainty in Artificial Intelligence.* Morgan Kaufmann, 411–420.

[21] Judea Pearl. 2009. *Causality.* Cambridge university press.

[22] Judea Pearl et al. 2000. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress* 19, 2 (2000).

[23] Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect.* Basic books.

[24] Chen Qian, Fuli Feng, Lijie Wen, Chunping Ma, and Pengjun Xie. 2021. Counterfactual Inference for Text Classification Debiasing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing.* ACL, 5434–5445.

[25] James Robins. 1986. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical modelling* 7, 9-12 (1986), 1393–1512.

[26] James M Robins. 2003. Semantics of causal DAG models and the identification of direct and indirect effects. *Oxford Statistical Science Series* (2003), 70–82.

[27] Donald B Rubin. 1978. Bayesian inference for causal effects: The role of randomization. *The Annals of Statistics* (1978), 34–58.

[28] Masahiro Sato, Sho Takemori, Janmajay Singh, and Tomoko Ohkuma. 2020. Unbiased Learning for the Causal Effect of Recommendation. In *Conference on Recommender Systems.* ACM, 378–387.

[29] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *Trans. Neural Networks* 20, 1 (2009), 61–80.

[30] Xuemeng Song, Fuli Feng, Jinhuan Liu, Zekun Li, Liqiang Nie, and Jun Ma. 2017. NeuroStylist: Neural Compatibility Modeling for Clothing Matching. In *Proceedings of the International Conference on Multimedia.* ACM, 753–761.

[31] Xuemeng Song, Xianjing Han, Yunkai Li, Jingyuan Chen, Xin-Shun Xu, and Liqiang Nie. 2019. GP-BPR: Personalized Compatibility Modeling for Clothing Matching. In *Proceedings of the International Conference on Multimedia.* ACM, 320–328.

[32] Xuemeng Song, Liqiang Jing, Dengtian Lin, Zhongzhou Zhao, Haiqing Chen, and Liqiang Nie. 2022. V2P: Vision-to-Prompt based Multi-Modal Product Summary Generation. In *The International Conference on Research and Development in Information Retrieval.* ACM, 992–1001.

[33] Tianyu Su, Xuemeng Song, Na Zheng, Weili Guan, Yan Li, and Liqiang Nie. 2021. Complementary Factorization towards Outfit Compatibility Modeling. In *Proceedings of the International Conference on Multimedia.* ACM, 4073–4081.

[34] Teng Sun, Wenjie Wang, Liqiang Jing, Yiran Cui, Xuemeng Song, and Liqiang Nie. 2022. Counterfactual Reasoning for Out-of-distribution Multimodal Sentiment Analysis. *CoRR* abs/2207.11652 (2022).

[35] Reuben Tan, Mariya I. Vasileva, Kate Saenko, and Bryan A. Plummer. 2019. Learning Similarity Conditions Without Explicit Supervision. In *International Conference on Computer Vision.* IEEE/CVF, 10372–10381.

[36] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. 2020. Unbiased Scene Graph Generation From Biased Training. In *Conference on Computer Vision and Pattern Recognition.* IEEE/CVF, 3713–3722.

[37] Mariya I. Vasileva, Bryan A. Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David A. Forsyth. 2018. Learning Type-Aware Embeddings for Fashion Compatibility. In *European Conference on Computer Vision (Lecture Notes in Computer Science, Vol. 11220).* Springer, 405–421.

[38] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. 2020. Visual Commonsense R-CNN. In *Conference on Computer Vision and Pattern Recognition.* IEEE/CVF, 10757–10767.

[39] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. In *The Conference on Knowledge Discovery and Data Mining.* ACM, 1717–1725.

[40] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be Cheating: Counterfactual Recommendation for Mitigating Clickbait Issue. In *The International Conference on Research and Development in Information Retrieval.* ACM, 1288–1297.

[41] Xun Yang, Fuli Feng, Wei Ji, Meng Wang, and Tat-Seng Chua. 2021. Deconfounded Video Moment Retrieval with Causal Intervention. In *The International Conference on Research and Development in Information Retrieval.* ACM, 1–10.

[42] Hanwang Zhang, Zheng-Jun Zha, Yang Yang, Shuicheng Yan, Yue Gao, and Tat-Seng Chua. 2013. Attribute-augmented semantic hierarchy: towards bridging semantic gap and intention gap in image retrieval. In *Proceedings of the International Conference on Multimedia.* ACM, 33–42.

[43] 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 *The International Conference on Research and Development in Information Retrieval.* ACM, 11–20.