mathematics

MDPI

Article

CGCLER: Causality-Enhanced Graph Contrastive Learning for Explainable Recommendation

Ying Guan, Zhiwen Yang * and Mengtao Duan

School of Computer and Communication Engineering, Northeastern University, Qinhuangdao 066004, China; guanying@neuq.edu.cn (Y.G.); duanmengtao@mails.neu.edu.cn (M.D.)
* Correspondence: 202312248@stu.neu.edu.cn

Abstract

Explainable recommendation has been conceptualized as a joint ranking task encompassing both items and explanations within contemporary recommender system research. The modeling of user–item–explanation triplets can be effectively facilitated by Graph Neural Networks (GNNs) due to their powerful representation learning capabilities. However, various observable and unobservable confounding factors, such as the user's mood at the time of purchase and product popularity, may cause users to make decisions that diverge from their genuine preferences. These confounding variables significantly affect GNN models, resulting in inconsistent or inaccurate representations of the relationships among users, items, and explanations. To address these challenges, we propose Causality-enhanced Graph Contrastive Learning for Explainable Recommendation (CGCLER). This method enables item–explanation joint ranking by distinguishing causal and confounding features at the graph-node representation level. Guided by the backdoor adjustment principle, CGCLER further introduces a backdoor-inspired graph contrastive learning objective that constructs representation-level perturbation views by combining causal features with randomly sampled confounding features, thereby encouraging representations that are less affected by varying confounding contexts. The effectiveness of CGCLER is evaluated through experiments on three publicly available datasets.

Keywords: explainable recommendation; causal inference; graph neural networks; contrastive learning

MSC: 68T07; 68T05; 68R10

Academic Editor: Jonathan Blackledge

Received: 17 July 2026
Revised: 13 August 2026
Accepted: 15 August 2026
Published: 17 August 2026

1. Introduction

Recommendation systems constitute some of the most effective approaches for addressing the challenge of information overload by filtering content according to users' preferences. These systems are widely implemented across social networking platforms, e-commerce websites, and various other online services, thereby becoming integral to daily life. As research in recommendation systems progresses, there is an increasing focus on explainable recommendation methods, which seek to provide users comprehensible explanations for the recommended items [1,2]. Explainable recommendations can be broadly categorized into two types: model-explainable and model-unexplainable [3]. The former refers to recommendation models that inherently generate explanations alongside the recommended items, while the latter involves the development of separate explanation models designed to provide explanations for specific recommendations.

Explanations in recommender systems exhibit notable diversity in both form and source, encompassing feature-based [4], path-based [5], sentence-based [6], and aspect-based [2] approaches. This heterogeneity presents significant challenges for the comparative evaluation of explainable recommendation models, as conventional metrics often fail to adequately capture interpretability nuances across different paradigms. To address these challenges, Li et al. [7] proposed a scalable framework that constructed an explanation set by aggregating high-frequency sentences extracted from user reviews. Candidate sentences were then ranked according to their relevance to a target item, with the top-K sentences presented as explainable justifications. Although this review-derived explanation construction improves the scalability of explanation evaluation, it may introduce structural co-occurrence bias. Popular items tend to accumulate more reviews and therefore more explanation candidates, while generic explanation sentences may repeatedly co-occur with many items. Consequently, explanation ranking may overemphasize frequent or popularity-driven explanation patterns rather than explanations that are truly informative for a specific user–item pair.

Building upon previously identified evaluation challenges, commonly used metrics for recommendation performance—such as Normalized Discounted Cumulative Gain (NDCG) and Recall—can be effectively adapted to evaluate the quality of explanations. Specifically, higher Recall and NDCG scores for the set of explanations associated with an item indicate greater accuracy and relevance in interpretability. To integrate recommendation and explanation generation, this framework formulates a joint ranking task in which the model simultaneously ranks items for recommendation and orders explanations from a predefined set tailored to specific user–item pairs (see Figure 1).

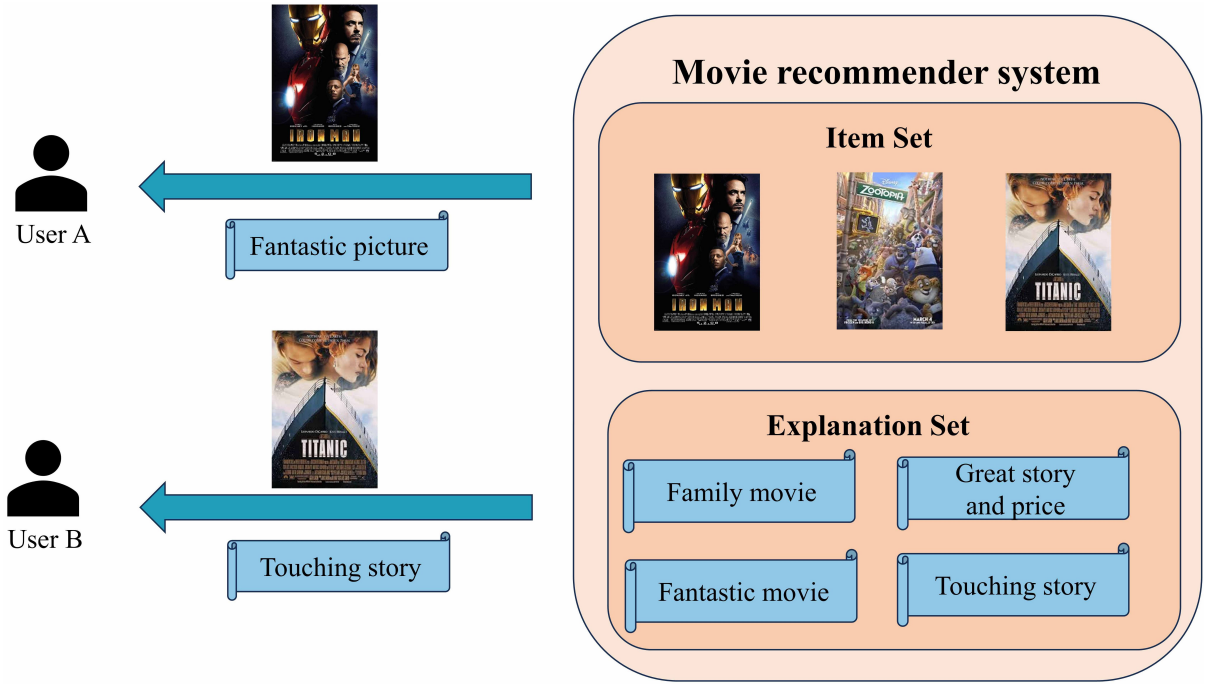

Figure 1. A toy example of the item–explanation joint ranking task for explainable recommendation.

The confounding issue in this setting differs from that in standard collaborative filtering. Standard recommendation debiasing mainly focuses on user–item interactions, whereas item–explanation joint ranking additionally involves user–explanation and item–explanation associations. A popular item can affect not only the observed user–item interaction but also the frequency with which its associated explanations appear in the graph. Similarly, a generic explanation may obtain high ranking scores because it frequently

co-occurs with many users or items, even when it is not the most specific explanation for a target user–item pair. Therefore, directly applying conventional recommendation debiasing methods to the user–item graph is insufficient for this task; the confounding-related signals propagated through explanation nodes should also be considered.

Graph Neural Networks (GNNs) have achieved considerable success in diverse domains, including rumor detection, social network analysis, and knowledge graph completion. Their strength lies in end-to-end encoding of graph topology into discriminative node representations. In the context of explainable recommendation systems, GNNs provide a natural modeling framework wherein users, items, and explanations are represented as graph nodes, while their interactions—specifically user–item–explanation triples—are represented as edges. By employing stacked graph convolutional layers, GNN-based approaches effectively capture high-order collaborative effects among these entities, thereby establishing a robust foundation for the joint ranking task outlined above.

Nevertheless, these GNN-based models often do not explicitly address confounding effects. They are therefore vulnerable to diverse confounding factors, which may induce spurious correlations and degrade generalization capability. For instance, product popularity can serve as a confounding variable, biasing the model toward recommending popular items rather than those that genuinely align with user preferences. To mitigate this issue, several studies have proposed deconfounded recommendation frameworks targeting specific confounders. Zhang et al. [8] identify product popularity as a confounding variable influencing the relationship between items and user clicks and propose to alleviate this bias through causal intervention using backdoor adjustment in causal inference. Similarly, He et al. [9] consider video length as a confounding factor affecting user clicks and employ a Mixture of Experts (MoE) architecture to enable efficient backdoor adjustment, thereby improving recommendation accuracy in accordance with user preferences. More recently, Zhu et al. [10] explicitly modeled both single-domain and cross-domain confounders in dual-target cross-domain recommendation and proposed to leverage backdoor adjustment to eliminate their negative influences.

Despite these advancements, real-world recommendation scenarios involve numerous confounding factors, and approaches that focus on a single variable often struggle to enhance model generalizability. Moreover, certain confounders may be latent and unobservable, including factors such as product quality or a user's mood during system interaction. Recent studies [11–13] have proposed deconfounding strategies based on front-door adjustment to address the challenges posed by unobservable confounders. Nevertheless, these methods necessitate the identification of intermediary variables that remain unaffected by confounders, a requirement that is particularly challenging within the context of item–explanation joint ranking.

To overcome the aforementioned limitations, we propose CGCLER, a unified causality-enhanced framework for item–explanation joint ranking. In this task, confounding effects may be propagated not only through user–item interactions but also through user–explanation and item–explanation associations. Rather than explicitly specifying a single confounder or constructing an observable mediator, CGCLER follows a representation-level feature separation strategy. Specifically, the effects of confounding factors are embedded in node representations, and CGCLER learns to distinguish causal features from confounding features through learnable masks. In this context, causal features refer to stable and discriminative representation components that are useful for accurate item–explanation ranking, whereas confounding features denote components that may introduce spurious correlations or noisy signals and negatively affect model generalization. This terminology is used at the representation level and does not imply that the learned masks provide a strictly identifiable causal decomposition.

To improve model generalizability, we introduce a backdoor-inspired graph contrastive learning strategy. CGCLER randomly combines causal features with confounding features sampled from other nodes and encourages stable representations under different confounding contexts. This design does not require the explicit identification of every real-world confounder. Instead, the learned confounding features are used to construct representation-level perturbations. This encourages the model to maintain a stable relationship between causal features and ranking labels when the confounding context changes.

Our principal contributions are summarized as follows:

(1) For the item–explanation joint ranking task, we propose CGCLER, a causality-enhanced graph contrastive learning framework that alleviates the adverse effects of confounders. By exploiting the learned causal features for prediction, this framework aims to achieve accurate item and explanation rankings that are less affected by spurious correlations.
(2) We develop a backdoor-inspired graph contrastive learning strategy. By randomly integrating causal and confounding features during contrastive training, this strategy encourages the model to learn stable representations under different confounding contexts and improves model generalizability.
(3) We conduct experiments on three publicly available datasets to evaluate CGCLER's performance. The results show that CGCLER generally achieves better performance than representative methods, while ablation studies further validate the effectiveness of its main components.

2. Related Work

2.1. Explainable Recommendation

In recent years, explainable recommendation has attracted significant interest from both academia and industry, emerging as a pivotal research area within the field of recommendation systems [1,2,4,14–16]. Explanations in recommendation systems manifest in diverse forms, including visual highlights [17], item neighbors [14], knowledge graph paths [5], word clouds [16], item features [4], predefined templates [2], automatically generated text [15], and retrieved text [18], among others. Nevertheless, in many studies, explanations are regarded as incidental byproducts of recommendation models, exhibiting heterogeneous formats and lacking standardized quantitative metrics for evaluating explanation quality. Notably, only a limited number of studies have formally conceptualized explanation generation as an integral component of the recommendation process, and even fewer have explored the joint optimization of item ranking and explanation relevance.

Li et al. [7] initiated a paradigm shift by conceptualizing explanation generation as a ranking task analogous to item recommendation, employing ranking-based metrics such as NDCG to evaluate explanation quality. They constructed (user, item, explanation) triplets and formulated item recommendation and explanation generation as a unified joint learning problem. Building upon that framework, Chen et al. [19] adopted the PITF framework [20] to model the user–item–explanation tensor, while Li et al. [6] addressed the sparsity issue inherent in triplet data by decomposing the (user, item, explanation) triplet into pairwise (user–explanation) and (item–explanation) relationships. Additionally, they incorporated pre-trained language models to capture semantic correlations among explanations. Despite these advancements, existing joint learning approaches predominantly focus on modeling associative relationships within triplet data, neglecting confounding factors that may distort genuine user–item–explanation interactions. This critical limitation motivates the development of our causality-enhanced framework.

2.2. GNN-Based Recommendation

GNNs have exhibited remarkable proficiency in representation learning, significantly advancing recommendation systems, especially in the area of collaborative filtering. In this context, user–item interaction data can be effectively modeled as a bipartite graph. To capture collaborative signals between users and items, Wang et al. [21] proposed NGCF, a pioneering GNN-based recommendation framework that substantially improved the modeling of item–user relevance. Subsequently, He et al. [22] observed that NGCF relied exclusively on identity (ID) features and that certain graph convolutional components, including nonlinear activation functions, contributed only marginally to performance. This observation motivated the development of LightGCN, which eliminates these redundant components to enhance both computational efficiency and recommendation accuracy.

Several studies [23–25] have incorporated explanatory information into GNN-based models by utilizing user reviews to extract attributes or tags as explanations, thereby constructing heterogeneous graphs comprising user, item, and explanation nodes. To address the complexity of ternary (user–item–explanation) interactions, Liu et al. [25] introduced A2-GCN, which utilizes an attention-driven aggregation mechanism to fuse heterogeneous information modalities, effectively capturing the intricate interdependencies among entities. Similarly, Chen et al. [23] developed TGCN, which performs graph convolution at both node and type granularities to aggregate representations in the heterogeneous graph. To alleviate negative transfer in multi-task joint learning, Wei et al. [24] decomposed the user–item–explanation heterogeneous graph into three homogeneous subgraphs—user–user, item–item, and explanation–explanation—thereby facilitating task-specific node representations and reducing conflicts between item recommendation and explanation ranking tasks. This approach enhanced performance in explainable recommendation systems.

Although GNN-based methods have achieved considerable advancements in modeling heterogeneous interactions, they often neglect potential confounding factors intrinsic to graph data—such as product popularity or frequency-driven explanation co-occurrences—that can distort the learning of genuine user preferences and subsequently impair model generalizability. In contrast, this study models causal and confounding features within node representations through learnable masks and employs backdoor-inspired contrastive learning to effectively alleviate the adverse effects of these confounders, thereby promoting the development of more accurate and robust explainable recommendation models.

2.3. Causal Inference in Recommendation

Causal inference has received increasing attention in recommendation systems due to its capacity to distinguish genuine causal relationships from spurious correlations, thereby improving both the accuracy and interpretability of recommendations. An expanding corpus of research employs causal inference to address confounding effects in recommendations. For instance, Zhang et al. [8] identified product popularity as a confounder influencing the relationship between items and user clicks and proposed the use of backdoor adjustment to mitigate its biased impact. He et al. [9] targeted video length as a confounder in user–item interactions, implementing backdoor adjustment via a Mixture of Experts (MoE) framework to better align recommendations with genuine user preferences. More recently, Zhu et al. [10] identified both single-domain and cross-domain confounders in dual-target cross-domain recommendation, and proposed a causal deconfounding framework that leveraged backdoor adjustment to preserve the positive effects of such confounders while eliminating their negative influences. To handle unobserved confounders, Xu et al. [26] utilized invariant item features as mediators between items and user behaviors, applying front-door adjustment to alleviate confounding effects. Similarly, Zhu et al. [12] employed user click behavior as a mediator linking item features to

purchase decisions, leveraging front-door adjustment to counteract spurious correlations introduced by confounders. Alternatively, Guo et al. [13] extended front-door adjustment to multi-criteria recommendation scenarios, leveraging users' multi-dimensional ratings as mediators to mitigate confounding bias in overall rating prediction.

Existing causal recommendation methods generally rely on explicitly modeling specific confounders or identifying observable mediators, assumptions difficult to meet in the item–explanation joint ranking task studied here. The heterogeneous, high-dimensional explanation data complicate defining meaningful mediators to isolate causal relationships among users, items, and explanations. To relax these requirements, recent studies have explored representation-level feature separation as an alternative approach to mitigating confounding effects. CAL [27] separates causal and shortcut features through causal attention and combines these features to approximate backdoor adjustment for graph classification. DCCL [28] disentangles user interest and conformity through contrastive learning for conventional item recommendation. Unlike CAL, which focuses on graph-level classification, and DCCL, which models user–item interactions for item recommendation, CGCLER is designed for item–explanation joint ranking over a user–item–explanation graph. It incorporates causal and confounding feature separation into the three edge-type subgraphs adopted from ExpGCN and employs a backdoor-inspired contrastive learning objective to alleviate confounding effects within a unified framework, without requiring explicit mediator identification or individual confounder modeling. This design enables representation-level feature separation to account for confounding-related signals propagated through user–item, user–explanation, and item–explanation associations within the joint ranking framework.

3. Problem Formulation

Formally, we define the key entities of the item–explanation joint ranking task as follows. Let U denote the set of users, I the set of items, and E the set of candidate explanations. For a given user–item pair (u, i), the model outputs an item preference score s_{ui}, which quantifies the preference of user u for item i. For a user–item–explanation triplet (u, i, e), the model further computes an explanation relevance score s_{uie}, which measures how well explanation e aligns with the preference of user u for item i. The top-K item recommendation list for user u is generated by ranking items in I, as defined in Equation (1):

$$\underset{i \in I}{\arg\max}^{K} s_{ui} \tag{1}$$

Similarly, for a specified user–item pair (u, i), the top-K explanation list is obtained by ranking candidate explanations in E, as shown in Equation (2):

$$\underset{e \in E}{\arg\max}^{K} s_{uie} \tag{2}$$

4. The Proposed CGCLER Model

The overall architecture of the CGCLER model is shown in Figure 2, which consists of four main modules: graph construction, node representation, backdoor-inspired contrastive learning, and recommendation and explanation prediction. The following sections detail each component of the model: Section 4.1 describes graph construction, Section 4.2 explains the graph convolution mechanism and the scoring functions for item and explanation ranking, Section 4.3 presents the causal perspective of the item–explanation joint ranking task, Section 4.4 introduces representation-level disentanglement, Section 4.5 presents the

backdoor-inspired contrastive learning, and Section 4.6 discusses the optimization objective for recommendation and explanation prediction.

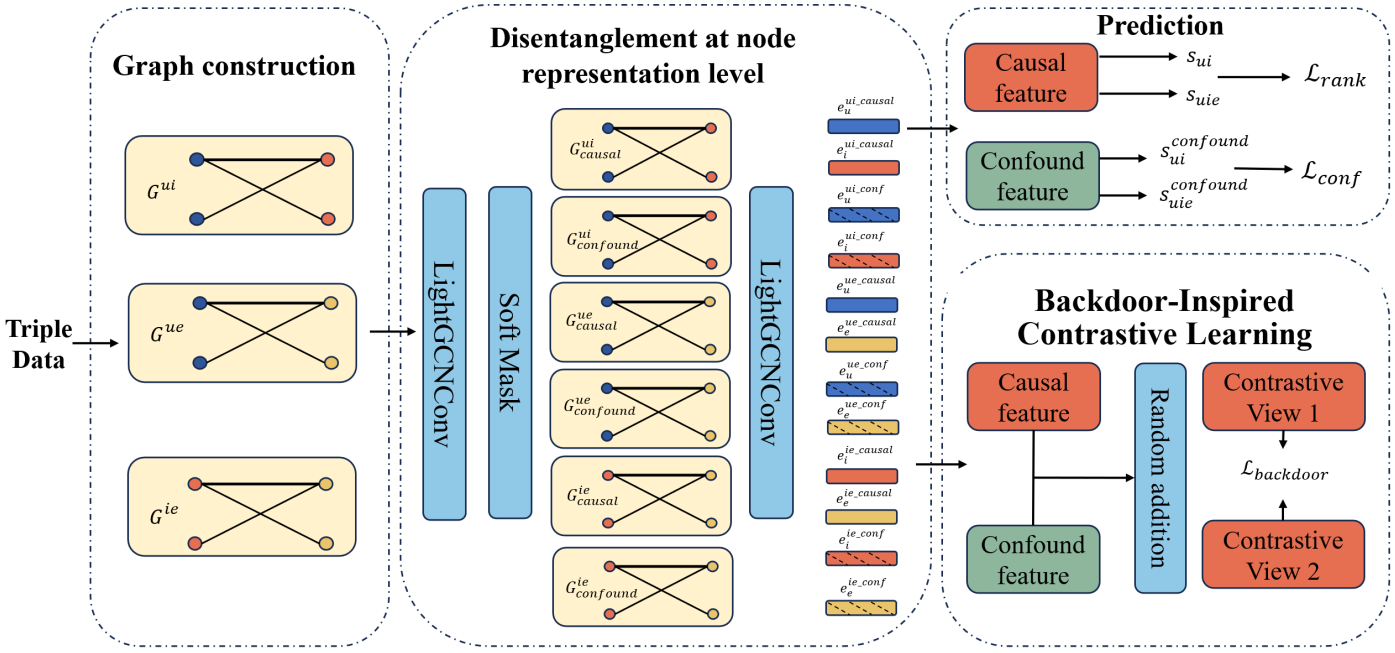

Figure 2. An illustration of the architecture of our proposed CGCLER model.

4.1. Graph Construction

We define a graph $G = \{A, V\}$ with a node set V that includes user nodes U, item nodes I, and explanation nodes E. The adjacency matrix of the graph is represented as $A \in \mathbb{R}^{|V|\times|V|}$. Based on the types of nodes connected by edges, we categorize edges into three types: user–item edges $\langle v_u, v_i \rangle$, user–explanation edges $\langle v_u, v_e \rangle$, and item–explanation edges $\langle v_i, v_e \rangle$. Following the approach in ExpGCN [24], the original heterogeneous graph is divided into three subgraphs, G_{ui}, G_{ue}, and G_{ie}, based on edge types.

The co-occurrence counts between users or items and explanations capture fine-grained information about user preferences and the alignment between items and explanations. Therefore, we use these counts as edge weights for edges connecting user or item nodes to explanation nodes. This allows the model to capture structural co-occurrence patterns among users, items, and explanations. Let W_{ue}, W_{ie}, and W_{ui} denote the edge weight matrices of subgraphs G_{ue}, G_{ie}, and G_{ui}, respectively. For a user–item–explanation triplet (u, i, e), let $y_{uie} = 1$ if the triplet exists and $y_{uie} = 0$ otherwise. In subgraph G_{ue}, the edge weight between user u and explanation e, denoted by $(W_{ue})_{u,e}$, is defined by aggregating over items:

$$(W_{ue})_{u,e} = \sum_{i \in I} y_{uie} \tag{3}$$

Similarly, the edge weight in subgraph G_{ie} is obtained by aggregating over users. In contrast, subgraph G_{ui} represents observed user–item interactions, and its edge weight is specified as follows:

$$(W_{ui})_{u,i} = \begin{cases} 1, & \text{if user } u \text{ interacts with item } i \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

For each bipartite subgraph, its adjacency matrix A can be constructed from the corresponding edge weight matrix W as follows:

$$A = \begin{bmatrix} 0 & W \\ W^T & 0 \end{bmatrix} \tag{5}$$

Subgraphs G_{ue} and G_{ie} are weighted, while G_{ui} is unweighted, reflecting interaction information among user, item, and explanation nodes in the constructed graph.

4.2. Graph Convolution and Scoring

After constructing the three subgraphs, we apply LightGCNConv [22] independently to each of them. The node embedding update formula after each graph convolution layer is given by:

$$e_p^{(k+1)} = \sum_{q \in \mathcal{N}_p} \frac{A_{pq}}{\sqrt{d_p d_q}} e_q^{(k)} \tag{6}$$

Here, $e_p^{(k+1)}$ represents the embedding of node p after $k+1$ layers of convolution, $\mathcal{N}_p$ denotes the set of neighbors of node p, and d_p is the weighted degree of node p. The LightGCNConv operation can be expressed in matrix form as follows:

$$X^{(k+1)} = LightGCNConv(A, X^{(k)}) = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} X^{(k)} \tag{7}$$

where $X^{(k)} \in \mathbb{R}^{N \times d}$ is the node embedding matrix at layer k, N represents the number of nodes, d is the embedding dimension, and D is a diagonal matrix with elements D_{pp} corresponding to the weighted degrees of the nodes. The final node embeddings are computed by averaging the embeddings obtained from all convolution layers (except the 0th layer) as shown below:

$$e_p = \sum_{k=1}^{K} \frac{e_p^{(k)}}{K} \tag{8}$$

To further enhance the connection between item recommendation and explanation ranking, we incorporate the embeddings learned from explanation-related subgraphs into the initial embeddings used in the user–item subgraph. Specifically, the user embedding derived from G_{ue} and the item embedding derived from G_{ie} are added to the corresponding 0th-layer user and item embeddings in G_{ui}, respectively, as illustrated in the following equations:

$$e_u^{ui(0)} = e_u^{(0)} + e_u^{ue} \tag{9}$$

$$e_i^{ui(0)} = e_i^{(0)} + e_i^{ie} \tag{10}$$

In these equations, $e_u^{(0)}$ and $e_i^{(0)}$ are the initial embeddings for users and items, while e_u^{ue} and e_i^{ie} represent the embeddings of user and item nodes derived from convolution on subgraphs G_{ue} and G_{ie}, respectively.

Each node type may appear in two subgraphs simultaneously, for instance, user nodes appear in both G_{ue} and G_{ui}. The embeddings learned from different subgraphs are specialized for different tasks: embeddings from G_{ue} and G_{ie} focus on explanation ranking, while those from G_{ui} are tailored for item recommendation. The predicted relevance score for an explanation e given a user u and an item i is computed as follows:

$$s_{uie} = \langle e_u^{ue}, e_e^{ue} \rangle + \langle e_i^{ie}, e_e^{ie} \rangle \tag{11}$$

where s_{uie} is the relevance score, $\langle \cdot, \cdot \rangle$ denotes the dot product, e_u^{ue} is the user embedding from G_{ue}, and e_e^{ue} is the explanation embedding from G_{ue}. Similar terms apply for item

embeddings e_i^{ie} and explanation embeddings e_e^{ie} in subgraph G_{ie}. The predicted score for an item i given a user u is calculated as:

$$s_{ui} = \langle e_u^{ui}, e_i^{ui} \rangle \tag{12}$$

For both item and explanation ranking tasks, we use the Bayesian Personalized Ranking (BPR) loss function [29] for training. For a given triplet, negative samples for the item recommendation task are drawn from items with which the user has not interacted, and for the explanation ranking task, explanations not mentioned by the user when interacting with the item are used as negative samples. The loss function is defined as follows:

$$\mathcal{L}_{rank} = \sum_{u \in U} \sum_{i \in I_u} \left(\sum_{j \notin I_u} -\ln \sigma(s_{ui} - s_{uj}) - \sum_{m \in E_{ui}} \sum_{n \notin E_{ui}} \ln \sigma(s_{uim} - s_{uin}) \right) \tag{13}$$

where $\sigma(\cdot)$ represents the sigmoid function, I_u denotes the set of items that user u has interacted with, $i \in I_u$ and $j \notin I_u$ denote positive and negative item samples, respectively, E_{ui} is the set of explanations associated with the user–item pair (u, i), and $m \in E_{ui}$ and $n \notin E_{ui}$ denote positive and negative explanation samples, respectively.

4.3. Causal Perspective in the Item–Explanation Joint Ranking Task

To analyze the underlying causal mechanisms within the item–explanation joint ranking task, we conducted a causal examination of the learning process of GNNs. The assumed relationships among eight variables—user preference U, item attributes A, (hidden) confounders H, graph data G, causal feature C, confounding feature F, node representation R, and model prediction Y—are illustrated in the causal graph (see Figure 3). Directed edges in the graph represent assumed causal dependencies among these variables, providing a conceptual view of the joint ranking process.

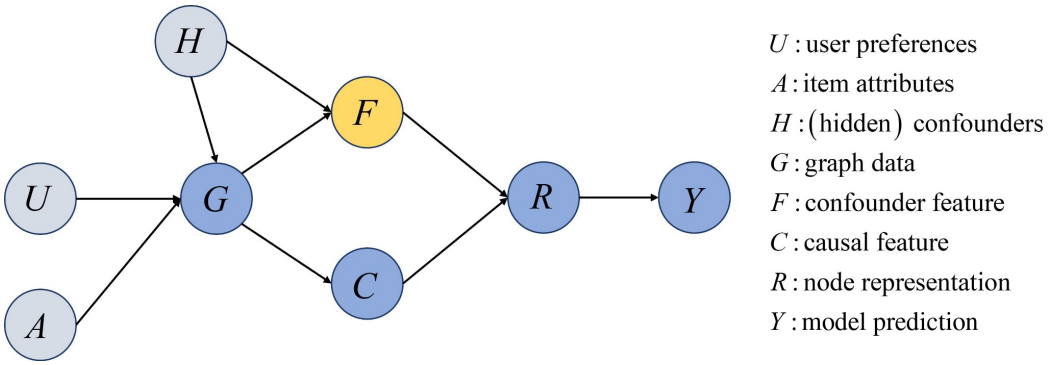

Figure 3. Causal graph for graph learning on the item–explanation joint ranking task.

- $U \rightarrow G \leftarrow A$: User preferences and item attributes jointly determine the observed interaction triplets within the dataset.
- $H \rightarrow G$: H denotes confounding factors that may influence interaction decisions but are not explicitly represented in the constructed user–item–explanation graph, including observable proxies (e.g., product popularity) and unobservable factors (e.g., product quality, user's mood).
- $C \leftarrow G \rightarrow F \leftarrow H$: This relationship shows that the graph data impact the presence of both causal and confounding features. C represents data that accurately reflect user–item matching, while F captures data influenced by biases due to confounders, which may not reliably represent user–item compatibility.

- $F \rightarrow R \leftarrow C$: Given the graph data, GNNs learn node representations informed by both causal and confounding features to characterize user, item, and explanation nodes.
- $R \rightarrow Y$: Using the learned node representations, the GNN model predicts recommended items and explanations.

Under the assumed causal graph, two backdoor paths may exist: $C \leftarrow G \rightarrow F \rightarrow R \rightarrow Y$ and $C \leftarrow G \leftarrow H \rightarrow F \rightarrow R \rightarrow Y$. In an ideal setting where the confounding feature F is observed and forms a valid adjustment set, backdoor adjustment can be used to accurately isolate the causal relationship between C and Y. Using do-calculus, we can express this relationship as follows:

$$P(Y|do(C)) = \sum_{f \in \Gamma} P(Y|C, f)P(f) \tag{14}$$

In this equation, Γ represents all possible values of F, $P(Y|C, f)$ is the conditional probability given causal feature C and confounding feature F, and $P(f)$ denotes the prior probability of F. This approach effectively mitigates the confounding effect caused by hidden variables, allowing us to capture the true causal relationship between C and Y.

However, several challenges arise when implementing this calculation:

1. **Feature Distinction**: Identifying causal and confounding features within the graph data is complex.
2. **Unstructured Data Representation**: Even if causal and confounding features are distinguishable, determining the value set for F is difficult due to the unstructured nature of graph data.
3. **Computational Feasibility**: Even if the value set of F can be determined, evaluating $P(Y|C, f)$ for all possible confounding feature values may incur prohibitive computational costs.

Equation (14) provides the theoretical motivation for reducing the influence of confounding features. In the item–explanation graph learning scenario, however, the exact confounders are usually latent, and F should not be directly treated as a fully identifiable adjustment set. In the following sections, we explore methods to address these challenges, thereby improving the robustness of our model in the item–explanation joint ranking task.

4.4. Disentanglement at the Representation Level

The non-Euclidean structure of graph data presents significant challenges in distinguishing between causal and confounding features at the data level. Nonetheless, during the graph convolution process, confounding factors are ultimately encoded into node representations, providing a basis for disentangling causal and confounding information. Our approach utilizes a soft-mask module to estimate the causal features C and confounding features F at the representation level. The soft-mask module comprises $M_c \in \mathbb{R}^{N \times d}$ and $M_o \in \mathbb{R}^{N \times d}$, where N denotes the number of nodes. This technique provides a tractable representation-level way to address challenges (1) and (2) discussed in the causal perspective subsection. Specifically, we employ a multi-layer perceptron (MLP) to implement the soft-mask module, as shown below:

$$(M_c, M_o) = \text{expand}_d\Big(\text{softmax}_{row}\Big(\text{MLP}\Big(\text{LightGCNConv}(A, X^{(0)})\Big)\Big)\Big) \tag{15}$$

$$X^c = X^{(0)} \odot M_c, \quad X^o = X^{(0)} \odot M_o \tag{16}$$

$$M_c + M_o = 1 \tag{17}$$

Here, A is the adjacency matrix of the corresponding subgraph, $X^{(0)} \in \mathbb{R}^{N\times d}$ represents the initial node embedding matrix, and $H = \text{LightGCNConv}(A, X^{(0)}) \in \mathbb{R}^{N\times d}$ is the output of the first LightGCN convolution. The MLP is implemented as a one-layer linear projection from d to 2, without a hidden layer, and produces two logits for each node. The operator softmax_{row} denotes a row-wise softmax over the two logits of each node. The operator expand_d expands the two node-level coefficients along the feature dimension and returns two masks $M_c, M_o \in \mathbb{R}^{N\times d}$. The first convolution is used to estimate graph-aware mask coefficients, while the resulting masks are applied to $X^{(0)}$ to initialize the causal and confounding branches. M_c acts as a causal feature mask for obtaining X^c, while M_o serves as a confounding feature mask for obtaining the complementary component X^o. The operator $\odot$ denotes element-wise multiplication, and $1 \in \mathbb{R}^{N\times d}$ is an all-one matrix.

The soft-mask module is applied once, following the initial convolution layer, for each of the three subgraphs. Consequently, each subgraph contains two separate node embedding matrices, resulting in six subgraphs in total: $G_{ue}^{causal} = \{A_{ue}, X_{ue}^c\}$, $G_{ue}^{confound} = \{A_{ue}, X_{ue}^o\}$, $G_{ie}^{causal} = \{A_{ie}, X_{ie}^c\}$, $G_{ie}^{confound} = \{A_{ie}, X_{ie}^o\}$, $G_{ui}^{causal} = \{A_{ui}, X_{ui}^c\}$, and $G_{ui}^{confound} = \{A_{ui}, X_{ui}^o\}$. Note that, following the embedding aggregation rule in Equation (8), initial node embeddings in these subgraphs do not directly contribute to final node embeddings. The soft-mask module is not applied before the first convolution layer since, while the initial embeddings may be influenced by confounding features, they still retain valuable information that can be effectively leveraged through information propagation. For instance, even a product chosen due to social influence (herd mentality) still partially reflects the user's preferences.

Confounding features can mislead model learning, with features captured from confounding aspects of the training set often generalizing poorly. To minimize the impact of these features, we propose a heuristic loss function, defined as follows:

$$s_{ui}^{confound} = \left\langle e_u^{ui\_conf}, e_i^{ui\_conf} \right\rangle \tag{18}$$

$$s_{uie}^{confound} = \left\langle e_u^{ue\_conf}, e_e^{ue\_conf} \right\rangle + \left\langle e_i^{ie\_conf}, e_e^{ie\_conf} \right\rangle \tag{19}$$

$$\mathcal{L}_{conf} = \sum_{u\in U}\sum_{i\in I_u}\left(\sum_{j\notin I_u}(s_{ui}^{confound} - s_{uj}^{confound})^2 + \sum_{m\in E_{ui}}\sum_{n\notin E_{ui}}(s_{uim}^{confound} - s_{uin}^{confound})^2\right) \tag{20}$$

Here, $e_u^{ui\_conf}$ and $e_i^{ui\_conf}$ denote embeddings learned from the subgraph $G_{ui}^{confound} = \{A_{ui}, X_{ui}^o\}$. Similar meanings apply to $e_u^{ue\_conf}$ and related terms for users and items. By minimizing this loss, the score gap between positive and negative samples in the confounding branch is reduced. This objective follows the common assumption in causal representation learning that confounding or shortcut features should not alone provide reliable evidence for the target label, thereby discouraging the model from relying on these features for ranking decisions. It should be noted that $\mathcal{L}_{conf}$ alone does not guarantee that the learned confounding branch corresponds to actual real-world confounders. When this constraint is overemphasized, a degenerate mask solution may also reduce the ranking discriminability of the confounding branch. Therefore, the learned decomposition should be interpreted as a representation-level separation between ranking-relevant and confounding-sensitive components.

Following the application of the soft-mask module, the model can now make more accurate predictions based on causal features by maximizing the score gap between positive and negative samples. We redefine the ranking loss from Equation (13) as follows:

$$\mathcal{L}_{rank} = \sum_{u \in U} \sum_{i \in I_u} \left(\sum_{j \notin I_u} -\ln \sigma(s_{ui}^{causal} - s_{uj}^{causal}) - \sum_{m \in E_{ui}} \sum_{n \notin E_{ui}} \ln \sigma(s_{uim}^{causal} - s_{uin}^{causal}) \right) \tag{21}$$

4.5. Backdoor-Inspired Contrastive Learning

As illustrated in Figure 3, the backdoor paths $C \leftarrow G \rightarrow F \rightarrow R \rightarrow Y$ and $C \leftarrow G \leftarrow H \rightarrow F \rightarrow R \rightarrow Y$ introduce confounding influences that lead the model to learn spurious, unstable relationships between causal features C and labels Y during training. Following the motivation of backdoor adjustment in Equation (14), the model should reduce the dependence of predictions on varying confounding features while preserving the effect of causal features. However, as noted in the causal perspective subsection, due to the extensive number of nodes and the wide range of confounding feature values, direct enumeration over confounding states is computationally infeasible.

Following insights from previous studies [27,28,30], we introduce a random addition technique to construct representation-level perturbation views via contrastive learning. Specifically, CGCLER samples confounding features from other nodes and adds them to the causal features of the target node. This operation provides a tractable way to expose causal features to diverse confounding contexts, encouraging the learned causal representations to remain stable when the confounding context changes. During training, uniform random sampling is used as a practical strategy for constructing diverse views; this does not imply that real-world confounders are assumed to strictly follow a uniform distribution. Inspired by this backdoor-adjustment perspective, we define the contrastive learning loss as follows:

$$z_i' = \frac{e_i^{causal} + e_{j'}^{confound}}{\left\| e_i^{causal} + e_{j'}^{confound} \right\|} \tag{22}$$

$$z_i'' = \frac{e_i^{causal} + e_{j''}^{confound}}{\left\| e_i^{causal} + e_{j''}^{confound} \right\|} \tag{23}$$

$$\mathcal{L}_{backdoor} = \sum_{i \in \mathcal{N}} -\log \frac{\exp\left((z_i')^T z_i'' / \tau\right)}{\sum_{j \in \mathcal{N}} \exp\left((z_i')^T z_j'' / \tau\right)} \tag{24}$$

In this formulation, e_i^{causal} denotes the embeddings of node i generated via convolution on subgraph $G_{\cdot}^{causal}$, and $e_{j'}^{confound}$ represents the embedding of a randomly selected node j' from subgraph $G_{\cdot}^{confound}$. For each subgraph, j' and j'' are sampled from nodes of the same type as node i. For example, user nodes are combined only with user-node confounding features, item nodes only with item-node confounding features, and explanation nodes only with explanation-node confounding features. Cross-type random mixing is not used. Here, $\mathcal{N}$ denotes the set of nodes, $N = |\mathcal{N}|$ is the number of nodes, and τ is the temperature parameter. Minimizing $\mathcal{L}_{backdoor}$ aligns perturbation views that share the same causal feature but are combined with different confounding features, thereby encouraging stable node representations under changes in the confounding context. This objective is consistent with the stability motivation underlying Equation (14).

Together with the confounding-branch constraint, this contrastive objective helps CGCLER distinguish stable causal features from unstable confounding features at the representation level, thereby improving the robustness of the learned representations for item–explanation joint ranking.

4.6. Model Optimization

The overall loss function used in CGCLER is given in Equation (25):

$$\mathcal{L} = \mathcal{L}_{rank} + \lambda_1 \mathcal{L}_{conf} + \lambda_2 \mathcal{L}_{backdoor} + \lambda_3 \|\theta\|_2 \tag{25}$$

Here, λ_1 and λ_2 are hyperparameters controlling the intensity of confounding feature separation and backdoor-inspired contrastive learning, respectively, while λ_3 regulates the strength of L2 regularization, with θ representing the model parameters. The complete CGCLER architecture is depicted in Figure 2, with the detailed pseudocode provided in Algorithm 1.

Algorithm 1 Learning algorithm for CGCLER

Require: Training set D, hyperparameters λ_1, λ_2, λ_3, number of subgraph layers for G_{ue}, G_{ie}, and G_{ui} (m and n respectively)
Ensure: Optimal model parameters

Initialize model parameters θ
Construct subgraphs G_{ui}, G_{ue}, G_{ie}; adjacency matrices A_{ui}, A_{ue}, A_{ie}; initial embeddings $X_{ui}^{(0)}, X_{ue}^{(0)}, X_{ie}^{(0)}$
for each epoch in Epochs **do**
  for each mini-batch in D **do**
    $X_{ui}^{(1)} = \text{LightGCNConv}(A_{ui}, X_{ui}^{(0)})$, $X_{ue}^{(1)} = \text{LightGCNConv}(A_{ue}, X_{ue}^{(0)})$, $X_{ie}^{(1)} = \text{LightGCNConv}(A_{ie}, X_{ie}^{(0)})$
    Apply soft-mask module to obtain causal feature mask M_c and confounding feature mask M_o
    Use M_c to obtain causal feature embeddings matrices $X_{ui\_caus}^{(0)}, X_{ue\_caus}^{(0)}, X_{ie\_caus}^{(0)}$ for subgraphs G_{ui},G_{ue},G_{ie}
    Use M_o to obtain confounding feature embeddings $X_{ui\_conf}^{(0)}, X_{ue\_conf}^{(0)}, X_{ie\_conf}^{(0)}$ for the same subgraphs
    for $k \in [1 \dots n]$ **do**
      $X_{ui\_caus}^{(k)} = \text{LightGCNConv}(A_{ui}, X_{ui\_caus}^{(k-1)})$
      $X_{ui\_conf}^{(k)} = \text{LightGCNConv}(A_{ui}, X_{ui\_conf}^{(k-1)})$
    end for
    for $k \in [1 \dots m]$ **do**
      $X_{ue\_caus}^{(k)} = \text{LightGCNConv}(A_{ue}, X_{ue\_caus}^{(k-1)})$
      $X_{ue\_conf}^{(k)} = \text{LightGCNConv}(A_{ue}, X_{ue\_conf}^{(k-1)})$
      $X_{ie\_caus}^{(k)} = \text{LightGCNConv}(A_{ie}, X_{ie\_caus}^{(k-1)})$
      $X_{ie\_conf}^{(k)} = \text{LightGCNConv}(A_{ie}, X_{ie\_conf}^{(k-1)})$
    end for
    Aggregate embeddings for causal features: $e_u^{ui\_caus} = \sum_{k=1}^{n} \frac{e_{u(k)}^{ui\_caus}}{n}$, $e_i^{ui\_caus} = \sum_{k=1}^{n} \frac{e_{i(k)}^{ui\_caus}}{n}$
    Aggregate embeddings for confounding features: $e_u^{ui\_conf} = \sum_{k=1}^{n} \frac{e_{u(k)}^{ui\_conf}}{n}$, $e_i^{ui\_conf} = \sum_{k=1}^{n} \frac{e_{i(k)}^{ui\_conf}}{n}$
    Similarly obtain $e_u^{ue\_caus}, e_e^{ue\_caus}, e_u^{ue\_conf}, e_e^{ue\_conf}, e_i^{ie\_caus}, e_e^{ie\_caus}, e_i^{ie\_conf}, e_e^{ie\_conf}$.
    Compute $\mathcal{L}_{conf}$ as per Equation (20)
    Compute $\mathcal{L}_{rank}$ as per Equation (21)
    Compute $\mathcal{L}_{backdoor}$ as per Equation (24)
    Calculate total loss $\mathcal{L}$ as per Equation (25)
    Update all trainable parameters to minimize $\mathcal{L}$
  end for
end for

5. Experimental Setup

5.1. Dataset

Our experiments utilized three publicly available datasets provided by Li et al. [7]: Amazon Movies & TV, TripAdvisor, and Yelp. Following established research protocols, we filtered users, items, and explanations with fewer than five interactions in each dataset. The statistical details of the processed datasets are presented in Table 1. These datasets cover different topics and vary in size and sparsity, providing a practical evaluation setting across different data characteristics.

Table 1. Dataset statistics.

Dataset	Amazon Movies & TV	TripAdvisor	Yelp
Users	7621	31,792	37,032
Items	6985	23,018	21,833
Explanations	1811	6487	2524
(u, i) pairs	92,824	482,272	356,140
(u, i, e) triplets	122,815	495,426	506,040

In this experimental configuration, positive item samples were items that had observed interactions with the corresponding user, and positive explanation samples were explanations appearing in observed user–item–explanation triplets. Negative sampling was used only during training, with a fixed uniform negative sampling rate of 20 across all datasets. Each dataset was randomly split into training, validation, and test sets in a 70:15:15 ratio. During validation and testing, item ranking was conducted over all candidate items in I, and explanation ranking was conducted over all candidate explanations in E for each evaluated user–item pair.

5.2. Evaluation Metrics

For both the item recommendation task and the explanation ranking task, we employed two widely recognized metrics, Recall@K and NDCG@K, to assess the performance of our model. We report the item-ranking and explanation-ranking metrics separately and additionally use GM as a single aggregate indicator. Since GM becomes small when either task performs poorly, it penalizes imbalanced performance across the two tasks and is suitable for summarizing the joint item–explanation ranking objective. The GM for the two index scores s_1 and s_2 is calculated as follows:

$$GM = \sqrt{s_1 s_2}.$$

In addition to the standard item recommendation and explanation ranking evaluation, we conducted stratified evaluation and representation-level diagnostic analysis. For stratified evaluation, item popularity and explanation frequency were computed only from the training split to avoid test leakage. Items and explanations were sorted by their training-set occurrence counts and divided into low-, medium-, and high-frequency groups using tertile-based splitting. Items or explanations with the same count at a boundary were assigned to the same group to avoid arbitrary separation. Zero-count nodes that did not appear in the training split were assigned to the low-frequency group. The all group corresponded to the original full test set. For each stratified group, positive test targets were restricted to the specified group, while ranking was still conducted over the complete item or explanation candidate set. We report R@10, R@20, N@10, and N@20 for each group and mainly discuss R@20 and N@20 because they are more stable on sparse stratified subsets. For representation-level diagnostics, we extracted causal and confounding branch

embeddings from the trained CGCLER model, computed the causal norm, confounding norm, and confounding share, and analyzed their relationship with popularity and frequency through correlation analysis and PCA visualization.

5.3. Baselines

To evaluate the effectiveness of CGCLER, we compared it against the following six baseline models:

- **PITF-J** [20]: This model extends the label recommendation framework PITF by incorporating parameter sharing across joint tasks. In PITF, the embeddings of users and items, originally utilized for label recommendations, can also be directly applied to item recommendations through dot-product calculations.
- **AMCF** [31]: The Attentive Multitask Collaborative Filtering model introduces a novel item refactoring task aimed at establishing mappings between items and aspects, thereby enhancing the interpretability of recommendations.
- **BPER-J** [6]: This model focuses on joint ranking for Bayesian Personalized Explanation Ranking. It extends PITF to facilitate item–explanation joint ranking.
- **A^2-GCN** [25]: The Attribute-aware Attentive Graph Convolution Network employs graph convolutional neural networks to characterize interactions within user–item–attribute triplets.
- **TGCN** [23]: The Tag Graph Convolution Network utilizes type-aware node sampling and convolution aggregation for node-level and type-level aggregation. It also designs a TransTag regularization term to enhance the performance of tag-aware recommendations by identifying user preferences.
- **ExpGCN** [24]: The Explanation-aware Graph Convolution Network learns task-oriented node representations by aggregating information from various subgraphs, thereby mitigating performance degradation caused by conflicts among different task learning objectives in the item–explanation joint ranking task.

All baselines were evaluated under the same processed data split and full-ranking protocol described above.

5.4. Hyperparameter Setting

For all baseline models, we used the RecBole toolbox (version 1.0.1) [32] as the unified framework for data processing, model training, and evaluation, thereby ensuring consistency in data division, random seeds, and evaluation metrics for fair comparisons and reproducibility. All models employed the Adam optimizer, with weight decay set to one of the following values: $\{1 \times 10^{-6}, 1 \times 10^{-5}, 1 \times 10^{-4}\}$. The learning rate was similarly adjusted across the values $\{1 \times 10^{-4}, 1 \times 10^{-3}, 1 \times 10^{-2}\}$. The maximum number of training epochs was set to 500, and an early stopping strategy was implemented, allowing for a maximum of 10 rounds without improvement before halting training.

For models based on Graph Neural Networks (GNNs), the number of convolutional layers was varied within the range of {1, 2, 3, 4, 5}. All model parameters were initialized using the Xavier uniform distribution. To account for variability, each model was tested at least five times with different random seeds, and the mean and standard deviation of each evaluation metric were reported as the experimental results. For CGCLER, we fixed the embedding dimension to $d = 128$, the number of layers to four for G_{ui} and two for both G_{ue} and G_{ie}, the learning rate to 10^{-3}, the optimizer weight decay to 10^{-6}, the training/evaluation batch sizes to 16,384/32,768, the temperature to $\tau = 0.8$, and the regularization coefficient to $\lambda_3 = 10^{-5}$. The dataset-specific loss weights were selected according to validation performance: $(\lambda_1, \lambda_2) = (1.1, 0.1)$ for Amazon Movies & TV,

$(0.5, 0.066)$ for TripAdvisor, and $(4, 0.002)$ for Yelp. Since M_c and M_o are learned continuous complementary soft masks, no additional mask-size hyperparameter was required.

The learnable-parameter counts of ExpGCN/CGCLER were 2.10M/2.10M, 7.85M/7.85M, and 7.86M/7.86M on Amazon Movies & TV, TripAdvisor, and Yelp, respectively. CGCLER added only 774 soft-mask parameters under $d = 128$, accounting for less than 0.04% of the trainable parameters on all datasets. The main additional computational cost came from propagating both causal and confounding branches and computing the mini-batch contrastive loss.

To evaluate the computational overhead of CGCLER, we compared CGCLER with ExpGCN on the Amazon Movies & TV dataset using the same processed data split, training negative-sampling setting, full-ranking validation protocol, and GPU server equipped with an NVIDIA GeForce RTX 4090. For each model, we conducted five runs with different random seeds. Epoch 0 was excluded as a warm-up epoch. For each run, the training, validation, and total epoch times were first averaged over the post-warm-up epochs before early stopping; Table 2 reports the mean and standard deviation across the five runs.

Table 2. Runtime comparison with ExpGCN on the Amazon Movies & TV dataset.

Model	Runs	Train (s/Epoch)	Valid. (s/Epoch)	Total (s/Epoch)	Relative Total
ExpGCN	5	12.86 ± 0.42	22.35 ± 0.87	35.21 ± 0.62	$1.00\times$
CGCLER	5	64.59 ± 2.87	21.74 ± 0.12	86.33 ± 2.98	$2.45\times$

As shown in Table 2, CGCLER increases the total per-epoch runtime from 35.21 to 86.33 s/epoch, corresponding to about $2.45\times$ that of ExpGCN. This overhead is expected because CGCLER propagates causal and confounding branches over the three edge-type subgraphs and optimizes an additional contrastive objective; since the number of additional trainable parameters is negligible, the overhead mainly arises from the added computation path rather than from model size.

6. Results and Discussion

6.1. Performance Comparison (RQ 1)

Tables 3–5 present the results comparing various models across three datasets. In these tables, Recall@K and NDCG@K are denoted as R@K and N@K, respectively, with all indicator values expressed as percentages (the percentage symbol is omitted for brevity). Each metric's best-performing models are highlighted in bold. All metrics are reported as mean $\pm$ standard deviation over five random seeds. The p-values were computed on GM N@20 using a two-tailed paired t-test, where CGCLER was compared with ExpGCN. ** and *** denote $p < 0.01$ and $p < 0.001$, respectively. The experimental results show that CGCLER generally improves most joint-task and single-task metrics. Regardless of dataset size and sparsity, CGCLER yields effective enhancements, attributable to several key factors:

1. CGCLER separates node representations into causal and confounding features, which helps the model distinguish stable collaborative signals from shortcut or spurious patterns.
2. Item recommendation and explanation ranking decisions are based on the extracted causal features rather than on embeddings influenced by confounding features post-convolution.
3. The backdoor-inspired contrastive learning module exposes causal features to different confounding contexts, encouraging stable causal representations and improving generalization without explicitly enumerating all confounding states in the adjustment formulation.

4. The confounding-feature constraint reduces the ranking discriminability of the confounding branch, thereby decreasing the model's reliance on shortcut signals such as popularity-related effects, frequency-driven explanation co-occurrences, and other confounding-sensitive patterns in the user–item–explanation graph.

Table 3. Comparisonbetween different models on the Amazon Movies & TV dataset.

Model	Item Recommendation				Explanation Ranking				Geometric Mean (GM)				p
Metrics	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**GM N@20**
PITF-J	3.73 ± 0.07	6.12 ± 0.12	2.00 ± 0.04	2.64 ± 0.05	38.25 ± 0.14	45.05 ± 0.13	28.73 ± 0.10	30.60 ± 0.09	11.94 ± 0.12	16.61 ± 0.17	7.58 ± 0.08	8.98 ± 0.13	–
AMCF	3.92 ± 0.09	6.51 ± 0.13	2.10 ± 0.04	2.78 ± 0.06	34.62 ± 0.14	41.02 ± 0.13	26.17 ± 0.10	27.94 ± 0.09	11.64 ± 0.13	16.34 ± 0.18	7.41 ± 0.08	8.82 ± 0.14	–
BPER-J	4.18 ± 0.10	6.83 ± 0.15	2.31 ± 0.05	3.01 ± 0.07	34.32 ± 0.15	42.87 ± 0.15	23.67 ± 0.10	26.02 ± 0.09	11.98 ± 0.14	17.11 ± 0.21	7.39 ± 0.09	8.84 ± 0.15	–
A^2-GCN	4.39 ± 0.11	7.22 ± 0.17	2.39 ± 0.06	3.14 ± 0.07	32.54 ± 0.15	40.17 ± 0.15	23.21 ± 0.11	25.31 ± 0.09	11.95 ± 0.16	17.03 ± 0.22	7.45 ± 0.10	8.91 ± 0.17	–
TGCN	4.88 ± 0.13	7.86 ± 0.20	2.64 ± 0.07	3.43 ± 0.08	34.28 ± 0.17	41.58 ± 0.16	24.47 ± 0.12	26.48 ± 0.10	12.93 ± 0.18	18.07 ± 0.25	8.04 ± 0.11	9.53 ± 0.18	–
ExpGCN	5.25 ± 0.15	8.53 ± 0.22	2.86 ± 0.08	3.72 ± 0.09	40.63 ± 0.20	47.75 ± 0.19	30.21 ± 0.15	32.17 ± 0.13	14.60 ± 0.20	20.19 ± 0.28	9.29 ± 0.13	10.94 ± 0.22	– (ref)
CGCLER	**5.43 ± 0.17**	**8.93 ± 0.27**	**2.98 ± 0.09**	**3.90 ± 0.11**	**41.41 ± 0.25**	**48.60 ± 0.24**	**30.71 ± 0.18**	**32.70 ± 0.16**	**15.00 ± 0.24**	**20.83 ± 0.33**	**9.53 ± 0.15**	**11.25 ± 0.15**	0.0047 **

Note: Bold values indicate the best performance for each metric. ** denotes statistical significance at $p < 0.01$.

Table 4. Comparison between different models on the TripAdvisor dataset.

Model	Item Recommendation				Explanation Ranking				Geometric Mean (GM)				p
Metrics	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**GM N@20**
PITF-J	2.12 ± 0.06	3.52 ± 0.09	1.16 ± 0.03	1.55 ± 0.04	14.44 ± 0.07	19.66 ± 0.09	10.15 ± 0.05	11.54 ± 0.05	5.53 ± 0.07	8.23 ± 0.11	3.42 ± 0.04	4.23 ± 0.12	–
AMCF	2.46 ± 0.07	4.01 ± 0.11	1.41 ± 0.04	1.85 ± 0.05	13.50 ± 0.07	18.19 ± 0.09	9.46 ± 0.05	10.85 ± 0.05	5.76 ± 0.08	8.53 ± 0.12	3.65 ± 0.05	4.48 ± 0.13	–
BPER-J	2.68 ± 0.09	4.32 ± 0.13	1.54 ± 0.05	2.01 ± 0.06	13.66 ± 0.08	18.35 ± 0.10	9.58 ± 0.06	10.97 ± 0.06	6.06 ± 0.09	8.90 ± 0.14	3.84 ± 0.06	4.69 ± 0.15	–
A^2-GCN	2.15 ± 0.08	3.52 ± 0.11	1.22 ± 0.04	1.61 ± 0.05	12.22 ± 0.08	16.87 ± 0.09	8.27 ± 0.05	9.63 ± 0.05	5.12 ± 0.09	7.71 ± 0.13	3.17 ± 0.05	3.94 ± 0.14	–
TGCN	2.71 ± 0.10	4.33 ± 0.14	1.58 ± 0.06	2.05 ± 0.06	13.60 ± 0.09	18.31 ± 0.11	9.47 ± 0.06	10.85 ± 0.06	6.07 ± 0.11	8.90 ± 0.15	3.87 ± 0.07	4.71 ± 0.17	–
ExpGCN	2.94 ± 0.11	4.73 ± 0.16	1.70 ± 0.06	2.22 ± 0.07	15.67 ± 0.11	21.18 ± 0.13	10.79 ± 0.08	12.41 ± 0.07	6.79 ± 0.12	10.01 ± 0.18	4.29 ± 0.08	5.25 ± 0.20	– (ref)
CGCLER	**3.33 ± 0.15**	**5.31 ± 0.21**	**1.96 ± 0.08**	**2.53 ± 0.10**	**15.97 ± 0.14**	**22.03 ± 0.18**	**11.04 ± 0.10**	**12.82 ± 0.10**	**7.29 ± 0.16**	**10.82 ± 0.24**	**4.66 ± 0.10**	**5.70 ± 0.18**	0.0007 ***

Note: Bold values indicate the best performance for each metric. *** denotes statistical significance at $p < 0.001$.

Table 5. Comparison between different models on the Yelp dataset.

Model	Item Recommendation				Explanation Ranking				Geometric Mean (GM)				p
Metrics	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**GM N@20**
PITF-J	3.29 ± 0.11	5.51 ± 0.17	1.76 ± 0.06	2.35 ± 0.07	15.08 ± 0.11	20.97 ± 0.12	9.45 ± 0.07	11.08 ± 0.06	7.04 ± 0.12	10.75 ± 0.17	4.08 ± 0.07	5.10 ± 0.17	–
AMCF	4.55 ± 0.17	7.51 ± 0.25	2.40 ± 0.08	3.19 ± 0.10	13.73 ± 0.11	19.48 ± 0.12	8.47 ± 0.07	10.05 ± 0.07	7.90 ± 0.15	12.09 ± 0.21	4.51 ± 0.09	5.66 ± 0.20	–
BPER-J	4.60 ± 0.18	7.62 ± 0.28	2.43 ± 0.09	3.23 ± 0.11	14.59 ± 0.12	21.12 ± 0.14	8.88 ± 0.08	10.67 ± 0.07	8.19 ± 0.17	12.68 ± 0.24	4.65 ± 0.10	5.87 ± 0.23	–
A^2-GCN	3.94 ± 0.17	6.55 ± 0.26	2.09 ± 0.09	2.79 ± 0.10	11.02 ± 0.10	16.81 ± 0.12	6.52 ± 0.06	8.12 ± 0.06	6.58 ± 0.15	10.50 ± 0.22	3.69 ± 0.08	4.70 ± 0.20	–
TGCN	4.15 ± 0.18	6.96 ± 0.29	2.23 ± 0.10	2.98 ± 0.12	14.39 ± 0.14	20.65 ± 0.16	8.70 ± 0.08	10.43 ± 0.08	7.73 ± 0.18	11.99 ± 0.26	4.41 ± 0.10	5.57 ± 0.24	–
ExpGCN	**4.92 ± 0.23**	8.10 ± 0.34	2.63 ± 0.12	3.48 ± 0.14	17.38 ± 0.17	24.37 ± 0.20	**10.78 ± 0.11**	12.70 ± 0.10	**9.25 ± 0.22**	14.05 ± 0.31	5.33 ± 0.13	6.65 ± 0.30	– (ref)
CGCLER	4.74 ± 0.26	**8.79 ± 0.44**	**2.69 ± 0.14**	**3.86 ± 0.18**	**18.08 ± 0.22**	**28.36 ± 0.28**	10.71 ± 0.13	**13.44 ± 0.13**	**9.25 ± 0.26**	**15.79 ± 0.41**	**5.36 ± 0.15**	**7.20 ± 0.22**	0.0016 **

Note: Bold values indicate the best performance for each metric. ** denotes statistical significance at $p < 0.01$.

In the Yelp dataset, although CGCLER improves most metrics, Recall@10 (item recommendation) and NDCG@10 (explanation ranking) exhibit slight declines compared to the best baseline. This decrease may be attributed to the higher sparsity of the Yelp dataset relative to the other two datasets. The effectiveness of CGCLER depends on the quality of the learned causal and confounding features; however, the sparsity inherent in the Yelp dataset may limit the diversity of reliable perturbation views.

6.2. Stratified Evaluation for Deconfounding Analysis (RQ 2)

To further evaluate whether the improvement of CGCLER is mainly caused by popular items or high-frequency explanations, we report stratified results by item popularity and explanation frequency in Tables 6 and 7. In each row, the better result between ExpGCN and CGCLER is highlighted in bold. The all group corresponds to the original full test set, while the low-, medium-, and high-frequency groups are diagnostic subsets constructed from training-set occurrence counts.

As shown in Table 6, CGCLER improves the all-group R@20 and N@20 results on all three datasets. More importantly, positive gains are also observed in most low- and medium-popularity groups, indicating that the improvement is not confined to popular items. On Yelp and TripAdvisor, several R@10 and N@10 values fluctuate because of data sparsity, but the more stable R@20 and N@20 results still show overall gains across most popularity groups.

As shown in Table 7, CGCLER improves the all-group R@20 and N@20 results on all three datasets. The gains are also observed in most low- and medium-frequency groups, suggesting that CGCLER does not rely only on high-frequency explanations. These results provide diagnostic empirical evidence for the deconfounding motivation of CGCLER, because low- and medium-frequency subsets are more susceptible to sparse co-occurrence patterns and frequency bias.

Table 6. Stratified item recommendation results by item popularity.

Dataset	Group	ExpGCN				CGCLER				Delta			
		R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
Amazon Movies & TV	Low	0.32	1.60	0.14	0.55	**0.55**	**2.05**	**0.25**	**0.72**	+0.23	+0.45	+0.11	+0.17
Amazon Movies & TV	Medium	1.94	4.95	0.98	2.05	**2.35**	**5.55**	**1.22**	**2.28**	+0.41	+0.60	+0.24	+0.23
Amazon Movies & TV	High	7.53	12.80	4.14	5.55	**7.78**	**13.10**	**4.30**	**5.78**	+0.25	+0.30	+0.16	+0.23
Amazon Movies & TV	All	5.25	8.53	2.86	3.72	**5.43**	**8.93**	**2.98**	**3.90**	+0.18	+0.40	+0.12	+0.18
Yelp	Low	**0.31**	0.80	**0.11**	0.25	0.26	**1.05**	0.09	**0.34**	−0.05	+0.25	−0.02	+0.09
Yelp	Medium	**2.89**	5.80	**1.44**	2.30	2.74	**6.12**	1.41	**2.45**	−0.15	+0.32	−0.03	+0.15
Yelp	High	**6.30**	12.60	**3.39**	5.35	6.05	**13.35**	3.32	**5.70**	−0.25	+0.75	−0.07	+0.35
Yelp	All	**4.92**	8.10	2.63	3.48	4.74	**8.79**	**2.69**	**3.86**	−0.18	+0.69	+0.06	+0.38
TripAdvisor	Low	**0.00**	0.05	**0.00**	0.02	**0.00**	**0.09**	**0.00**	**0.03**	+0.00	+0.04	+0.00	+0.01
TripAdvisor	Medium	**1.50**	1.50	**0.58**	0.58	1.32	**1.82**	0.50	**0.68**	−0.18	+0.32	−0.08	+0.10
TripAdvisor	High	3.71	6.20	2.11	2.88	**3.95**	**6.80**	**2.25**	**3.14**	+0.24	+0.60	+0.14	+0.26
TripAdvisor	All	2.94	4.73	1.70	2.22	**3.33**	**5.31**	**1.96**	**2.53**	+0.39	+0.58	+0.26	+0.31

Note: Bold values indicate the better result between ExpGCN and CGCLER for each metric within each group.

Table 7. Stratified explanation ranking results by explanation frequency.

Dataset	Group	ExpGCN				CGCLER				Delta			
		R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
Amazon Movies & TV	Low	14.35	24.53	9.76	13.82	**17.24**	**27.13**	**11.65**	**15.12**	+2.89	+2.60	+1.89	+1.30
Amazon Movies & TV	Medium	23.34	39.82	17.94	24.52	**27.13**	**42.04**	**20.63**	**26.13**	+3.79	+2.22	+2.69	+1.61
Amazon Movies & TV	High	48.34	59.23	36.23	39.52	**50.23**	**60.43**	**37.22**	**40.25**	+1.89	+1.20	+0.99	+0.73
Amazon Movies & TV	All	40.63	47.75	30.21	32.17	**41.41**	**48.60**	**30.71**	**32.70**	+0.78	+0.85	+0.50	+0.53
Yelp	Low	**1.79**	2.43	**0.78**	0.97	1.65	**2.68**	0.72	**1.06**	−0.14	+0.25	−0.06	+0.09
Yelp	Medium	4.88	7.83	2.65	3.22	**5.34**	**8.27**	**2.82**	**3.39**	+0.46	+0.44	+0.17	+0.17
Yelp	High	20.91	30.53	**12.65**	15.23	**21.24**	**32.24**	12.62	**15.72**	+0.33	+1.71	−0.03	+0.49
Yelp	All	17.38	24.37	**10.78**	12.70	**18.08**	**28.36**	10.71	**13.44**	+0.70	+3.99	−0.07	+0.74
TripAdvisor	Low	**0.37**	0.47	**0.21**	0.23	0.31	**0.53**	0.18	**0.26**	−0.06	+0.06	−0.03	+0.03
TripAdvisor	Medium	**2.43**	2.44	**1.15**	1.17	2.16	**2.72**	1.03	**1.29**	−0.27	+0.28	−0.12	+0.12
TripAdvisor	High	19.62	25.83	12.57	14.22	**20.13**	**26.87**	**12.96**	**14.86**	+0.51	+1.04	+0.39	+0.64
TripAdvisor	All	15.67	21.18	10.79	12.41	**15.97**	**22.03**	**11.04**	**12.82**	+0.30	+0.85	+0.25	+0.41

Note: Bold values indicate the better result between ExpGCN and CGCLER for each metric within each group.

6.3. Representation-Level Explainability Diagnostics (RQ 3)

To further examine whether the learned confounding-related component captured meaningful popularity- and frequency-sensitive signals, we conducted representation-level diagnostic analyses. Specifically, we computed item popularity and explanation frequency from the training split, derived causal and confounding representation statistics from the trained CGCLER model, and analyzed their correlations with the observed counts. We further visualized representative learned representations using PCA.

Table 8 reports the correlation between representation statistics and item popularity or explanation frequency. The causal representation norm is positively correlated with popularity and frequency across all three datasets, indicating that high-frequency nodes tend to form more stable ranking-related representations. In contrast, the confounding share is negatively correlated with popularity and frequency in most cases. This suggests that the confounding-related component should not be interpreted as directly encoding item popularity or explanation frequency themselves. Instead, it is more associated with

sparse, low-frequency, and frequency-sensitive residual signals, which are likely to contain unstable co-occurrence patterns and noise.

Table 8. Representation-level correlation with item popularity and explanation frequency.

Dataset	Node Type	ρ (Count, Causal Norm)	ρ (Count, Conf. Share)	r (Log Count, Causal Norm)	r (Log Count, Conf. Share)
Amazon Movies & TV	Item popularity	0.827	−0.680	0.871	−0.261
Amazon Movies & TV	Explanation frequency	0.785	−0.384	0.824	−0.117
Yelp	Item popularity	0.799	−0.538	0.815	−0.648
Yelp	Explanation frequency	0.783	−0.807	0.844	−0.573
TripAdvisor	Item popularity	0.894	−0.628	0.939	−0.686
TripAdvisor	Explanation frequency	0.787	−0.357	0.820	−0.401

Overall, the positive correlations for causal representation norm and the mostly negative correlations for confounding share indicate that the two representation statistics capture different frequency-sensitive structures.

To examine the global linear structure of the learned representation space, we applied PCA to the concatenated causal and confounding representations of Yelp items. Figure 4 presents the same two-dimensional PCA coordinates using two coloring schemes: item popularity measured by $\log(1 + \text{count})$ and learned confounding share.

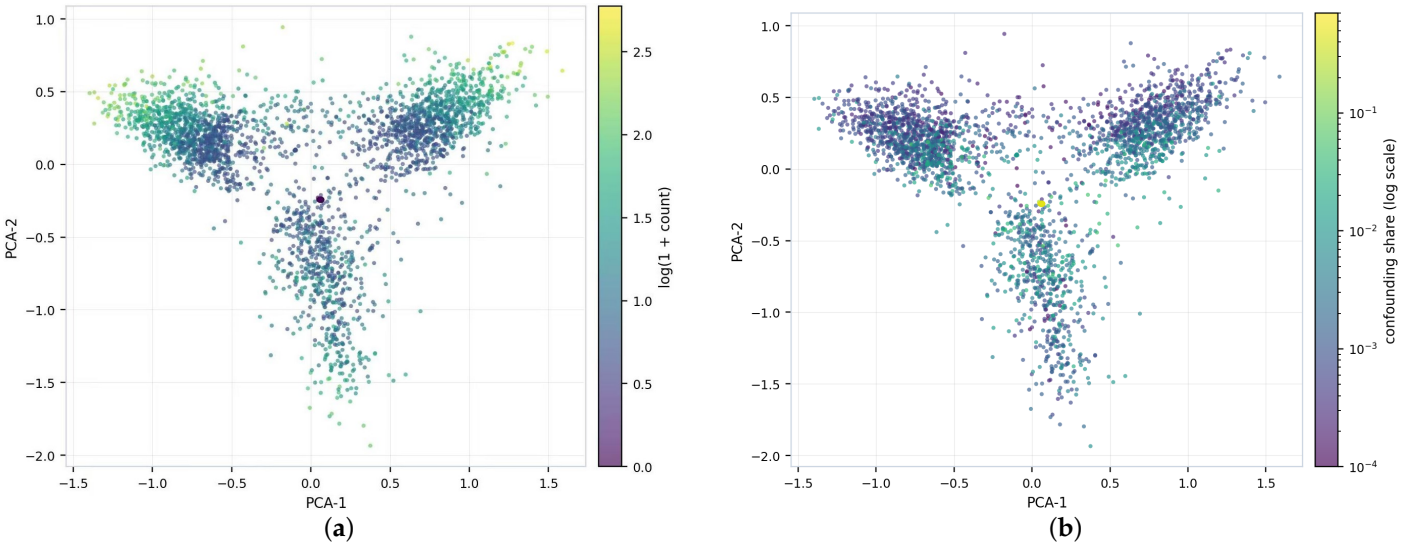

Figure 4. PCA visualization of Yelp item representations using the same two-dimensional coordinates, colored by (**a**) $\log(1 + \text{item popularity})$ and (**b**) learned confounding share.

The PCA projection exhibits a clear global multi-branch geometry. Item popularity varies gradually across this representation space, whereas confounding share is more diffusely distributed and does not align with a single high-popularity direction. The two coloring patterns are therefore related but not identical. This visual evidence is consistent with the correlation results in Table 8: the learned confounding-related component should not be interpreted as directly encoding item popularity but is instead associated with frequency- and sparsity-sensitive residual variation.

To avoid overloading the main text with repetitive visualization panels, we report PCA visualization as the main qualitative evidence and use the correlation results as quantitative evidence across all three datasets. Additional nonlinear visualizations show similar diagnostic patterns and are omitted for brevity.

We do not claim that the confounding branch perfectly recovers all real-world confounders. Instead, the correlation and PCA results provide representation-level diagnostic evidence that the learned confounding-related component is associated with popularity-, frequency-, and sparsity-sensitive residual signals, while the causal component preserves more stable ranking-related information.

6.4. Ablation Study (RQ 4)

To assess the effectiveness of CGCLER and the contribution of its individual components, this section reports ablation experiments conducted on the following variant models of CGCLER:

- **w/o RD:** This variant excludes the random addition strategy of causal and confounding features during the contrast learning view generation, such that only the corresponding confounding features are added to each node's causal features.
- **w/o BICL:** In this configuration, the backdoor-inspired contrastive learning module is entirely omitted from CGCLER.
- **w/o CC:** This variant removes the constraints on confounding features, thereby allowing CGCLER's predictions based on confounding features to be unconstrained.
- **w/o BICL+CC:** This model eliminates constraints on confounding features and the backdoor-inspired contrastive learning module within CGCLER.

Tables 9–11 present the ablation results on all three datasets. The metrics Recall@K and NDCG@K are denoted as R@K and N@K, respectively, with all values expressed as percentages (the percentage symbol is omitted for clarity). Each metric's best-performing model is highlighted in bold. All metrics are reported as mean ± standard deviation over five random seeds. The p-values were computed on GM N@20 using a two-tailed paired t-test, where each ablated variant was compared with the full CGCLER model. *, **, and *** denote $p < 0.05$, $p < 0.01$, and $p < 0.001$, respectively. The results of the ablation experiments lead to several key insights:

Table 9. Ablation results of CGCLER on the Amazon Movies & TV dataset.

Variant	Item Recommendation				Explanation Ranking				Geometric Mean (GM)				p
Metrics	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	GM N@20
CGCLER	**5.43 ± 0.17**	**8.93 ± 0.27**	**2.98 ± 0.09**	**3.90 ± 0.11**	**41.41 ± 0.25**	**48.60 ± 0.24**	**30.71 ± 0.18**	**32.70 ± 0.16**	**15.00 ± 0.24**	**20.83 ± 0.33**	**9.53 ± 0.15**	**11.25 ± 0.15**	–
w/o RD	5.42 ± 0.21	8.61 ± 0.32	2.97 ± 0.11	3.80 ± 0.13	40.95 ± 0.30	47.77 ± 0.29	30.58 ± 0.22	32.47 ± 0.20	14.93 ± 0.29	20.28 ± 0.40	9.52 ± 0.19	11.10 ± 0.18	0.0225 *
w/o BICL	5.28 ± 0.18	8.70 ± 0.28	2.89 ± 0.10	3.79 ± 0.12	40.74 ± 0.26	48.20 ± 0.26	30.09 ± 0.19	32.16 ± 0.17	14.67 ± 0.25	20.47 ± 0.35	9.32 ± 0.16	11.04 ± 0.16	0.0048 **
w/o CC	**5.43 ± 0.23**	8.76 ± 0.35	2.96 ± 0.12	3.81 ± 0.15	41.00 ± 0.32	48.40 ± 0.32	30.41 ± 0.24	32.56 ± 0.21	14.92 ± 0.32	20.59 ± 0.43	9.49 ± 0.20	11.14 ± 0.19	0.0734
w/o BICL+CC	5.40 ± 0.25	8.71 ± 0.38	2.94 ± 0.13	3.80 ± 0.16	40.72 ± 0.35	48.18 ± 0.35	30.02 ± 0.26	32.08 ± 0.23	14.83 ± 0.34	20.49 ± 0.48	9.39 ± 0.22	11.05 ± 0.21	0.0172 *

Note: Bold values indicate the best performance for each metric. * and ** denote statistical significance at $p < 0.05$ and $p < 0.01$, respectively.

Table 10. Ablation results of CGCLER on the TripAdvisor dataset.

Variant	Item Recommendation				Explanation Ranking				Geometric Mean (GM)				p
Metrics	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	GM N@20
CGCLER	**3.33 ± 0.15**	**5.31 ± 0.21**	**1.96 ± 0.08**	**2.53 ± 0.10**	**15.97 ± 0.14**	**22.03 ± 0.18**	**11.04 ± 0.10**	**12.82 ± 0.10**	**7.29 ± 0.16**	**10.82 ± 0.24**	**4.66 ± 0.10**	**5.70 ± 0.18**	–
w/o RD	3.24 ± 0.17	5.08 ± 0.25	1.90 ± 0.10	2.45 ± 0.11	15.48 ± 0.17	21.38 ± 0.21	10.62 ± 0.12	12.38 ± 0.12	7.07 ± 0.19	10.41 ± 0.28	4.51 ± 0.12	5.52 ± 0.22	0.0239 *
w/o BICL	3.29 ± 0.16	5.20 ± 0.22	1.94 ± 0.09	2.50 ± 0.10	15.82 ± 0.15	21.85 ± 0.19	10.92 ± 0.11	12.69 ± 0.11	7.22 ± 0.17	10.69 ± 0.25	4.61 ± 0.11	5.65 ± 0.20	0.3436
w/o CC	3.26 ± 0.19	5.11 ± 0.27	1.91 ± 0.11	2.46 ± 0.12	15.55 ± 0.18	21.47 ± 0.23	10.68 ± 0.13	12.43 ± 0.13	7.10 ± 0.21	10.46 ± 0.30	4.53 ± 0.13	5.54 ± 0.23	0.0452 *
w/o BICL+CC	3.19 ± 0.20	5.00 ± 0.29	1.88 ± 0.11	2.43 ± 0.13	15.30 ± 0.20	21.22 ± 0.25	10.37 ± 0.14	12.22 ± 0.14	6.98 ± 0.22	10.29 ± 0.33	4.47 ± 0.14	5.51 ± 0.26	0.0379 *

Note: Bold values indicate the best performance for each metric. * denotes statistical significance at $p < 0.05$.

1. CGCLER achieves the best overall GM performance on all three datasets. Removing RD, CC, or both BICL and CC generally reduces GM N@20, and the degradation is statistically significant in most cases.

2. The variant **w/o RD** shows diminished performance relative to CGCLER, suggesting that diverse confounding contexts constructed by random feature recombination facilitate the representation learning of CGCLER.
3. The isolated effect of BICL is dataset-dependent: removing BICL produces a significant drop on Amazon Movies & TV, but the degradation is not statistically significant on Yelp and TripAdvisor. In contrast, removing both BICL and CC consistently weakens the overall performance, indicating that the backdoor-inspired contrastive learning module and the confounding-feature constraint provide complementary benefits.

Table 11. Ablation results of CGCLER on the Yelp dataset.

Variant	Item Recommendation				Explanation Ranking				Geometric Mean (GM)				p	
Metrics	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**GM**	**N@20**
CGCLER	**4.74 ± 0.26**	**8.79 ± 0.44**	**2.69 ± 0.14**	**3.86 ± 0.19**	**18.08 ± 0.22**	**28.36 ± 0.28**	**10.71 ± 0.13**	**13.44 ± 0.13**	**9.25 ± 0.26**	**15.79 ± 0.41**	**5.36 ± 0.15**	**7.20 ± 0.22**	–	
w/o RD	4.64 ± 0.31	8.54 ± 0.52	2.62 ± 0.17	3.77 ± 0.22	17.56 ± 0.26	27.65 ± 0.34	10.28 ± 0.15	12.93 ± 0.16	9.02 ± 0.31	15.36 ± 0.49	5.19 ± 0.18	6.99 ± 0.26	0.0253 *	
w/o BICL	4.70 ± 0.27	8.71 ± 0.47	2.67 ± 0.15	3.82 ± 0.20	17.99 ± 0.23	28.21 ± 0.30	10.63 ± 0.14	13.35 ± 0.14	9.19 ± 0.28	15.67 ± 0.44	5.31 ± 0.16	7.15 ± 0.23	0.4166	
w/o CC	4.66 ± 0.33	8.60 ± 0.57	2.63 ± 0.18	3.78 ± 0.24	17.48 ± 0.28	27.58 ± 0.36	10.23 ± 0.16	12.86 ± 0.17	8.96 ± 0.33	15.29 ± 0.52	5.17 ± 0.19	7.01 ± 0.28	0.0451 *	
w/o BICL+CC	4.60 ± 0.36	8.48 ± 0.61	2.60 ± 0.20	3.75 ± 0.26	17.40 ± 0.30	27.54 ± 0.40	10.03 ± 0.17	12.83 ± 0.19	8.93 ± 0.36	15.25 ± 0.57	5.16 ± 0.21	6.99 ± 0.30	0.0472 *	

Note: Bold values indicate the best performance for each metric. * denotes statistical significance at $p < 0.05$.

6.5. Parameter Analysis (RQ 5)

This section investigates the impact of coefficients λ_1 and λ_2 on the performance of CGCLER across two datasets: Yelp and Amazon Movies & TV. Here, λ_1 regulates the influence of $\mathcal{L}_{\text{conf}}$, while λ_2 governs the impact of $\mathcal{L}_{\text{backdoor}}$. The experimental results are illustrated in Figure 5. Based on our observations from this figure, we can draw several key conclusions:

1. λ_2 exhibits greater sensitivity than λ_1, with its optimal value range significantly varying between the datasets. Therefore, careful tuning is necessary. Specifically, the reasonable range for λ_2 on the Yelp dataset is $[0.002, 0.003]$, while for the Amazon Movies & TV dataset, it is $[0.02, 0.18]$. This substantial difference indicates a dataset-dependent sensitivity of the contrastive objective and represents a practical limitation of CGCLER. A possible reason is that the effective strength of $\mathcal{L}_{\text{backdoor}}$ depends on graph sparsity and on the reliability of randomly recombined confounding contexts. In sparser data, a large λ_2 may overemphasize the auxiliary contrastive signal relative to the ranking loss. Therefore, λ_2 should be tuned on a validation set when CGCLER is applied to a new dataset.
2. An excessively high value of λ_1 can lead to performance degradation. In such cases, CGCLER tends to assign the soft mask M_c of causal features as an all-one matrix, while the soft mask M_o of confounding features is set to an all-zero matrix. This configuration hampers CGCLER's ability to effectively distinguish between causal and confounding features at the node representation level.
3. Compared with λ_2, λ_1 shows milder sensitivity within the tested range. A moderately larger λ_1 can help reduce ranking-relevant information encoded in the confounding branch, whereas either too small or excessively large values may weaken the effectiveness of representation-level disentanglement.

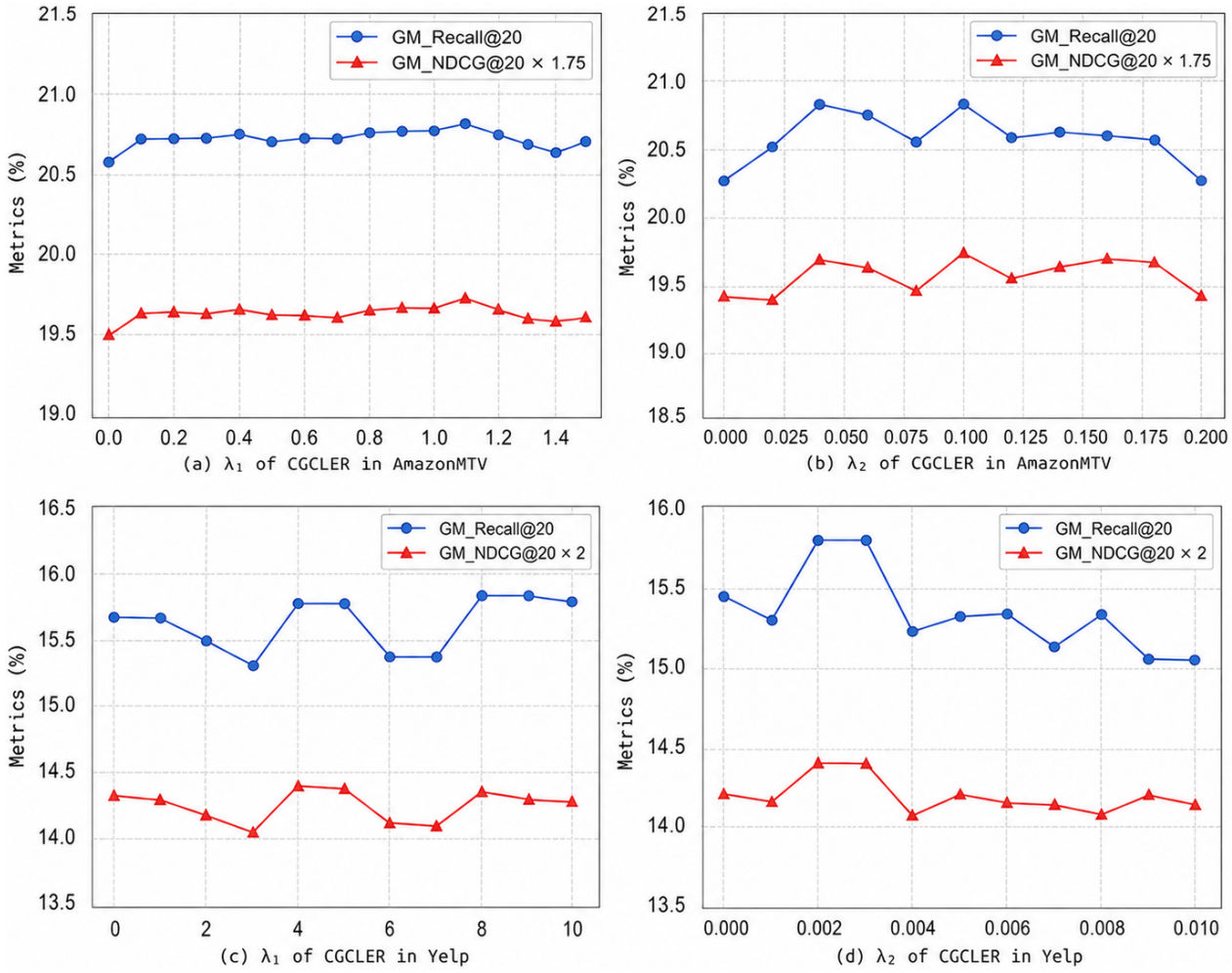

Figure 5. Effect of the loss function coefficients on CGCLER performance. Subfigures (**a**,**c**) vary λ_1, the coefficient of $\mathcal{L}_{\text{conf}}$, while subfigures (**b**,**d**) vary λ_2, the coefficient of $\mathcal{L}_{\text{backdoor}}$. In each subfigure, the x-axis gives the tested coefficient value and the y-axis reports GM-based metrics in percentage; blue curves denote GM_Recall@20, and red curves denote GM_NDCG@20 scaled only for visualization ($\times$1.75 for Amazon Movies & TV and $\times$2 for Yelp).

6.6. Overall Discussion

The results indicate that item–explanation joint ranking can benefit from distinguishing causal and confounding features in graph node representations. Compared with conventional GNN-based joint ranking models, CGCLER does not manually specify a single confounder. Instead, it learns confounding features from node representations and uses them to construct perturbation views for graph contrastive learning. This design is particularly suitable for explainable recommendation, where confounding effects may arise not only from user–item interactions but also from user–explanation and item–explanation co-occurrence patterns.

The ablation results further indicate that the contrastive learning module and the confounding-feature constraint play complementary roles. The former encourages causal features to remain stable under different confounding contexts, while the latter weakens the ranking discrimination of the confounding branch. The added stratified evaluation and representation-level diagnostics further support this interpretation. The stratified results show that the performance gains of CGCLER are not restricted to high-popularity items or high-frequency explanations, while the correlation and PCA analyses indicate that the learned causal and confounding components are associated with different popularity- and frequency-sensitive patterns. Nevertheless, this study still has several limitations.

CGCLER should be interpreted as a causal representation learning framework rather than a fully identifiable causal discovery method. The learned confounding features are latent representations and should not be treated as directly observed confounders or a valid adjustment set. Accordingly, the observed improvement should be interpreted as the effect of representation-level feature separation and stability regularization, rather than as direct evidence that item popularity or review frequency has been explicitly decoupled. In addition, the random addition strategy provides a practical way to construct perturbation views, but it does not model the true distribution of real-world confounders. Future work will also consider fair comparisons with causal baselines adapted to the item–explanation joint ranking protocol.

The present experiments used static random splits, so they did not evaluate stricter old-to-new temporal generalization, explicit popularity shifts, or counterfactual robustness. Under the current transductive setting, a new interaction would update the corresponding user–item, user–explanation, and item–explanation edges, and the affected embeddings would need to be refreshed through incremental fine-tuning or periodic retraining. Future work should further include long-tail performance metrics, chronological splits when timestamped data are available, streaming evaluation, synthetic-confounder experiments, and textual semantic modeling of explanation candidates.

7. Conclusions

In this study, we presented Causality-enhanced Graph Contrastive Learning for Explainable Recommendation (CGCLER), a novel approach for item–explanation joint ranking. To address the challenges posed by confounding factors that can mislead graph model learning, CGCLER differentiates causal and confounding features at the node representation level. The model incorporates user, item, and explanation interactions based on the learned causal features.

To further mitigate the impact of confounding features, CGCLER introduces a backdoor-inspired graph contrastive learning strategy. The model randomly combines causal features with confounding features to construct perturbation views, encouraging stable representations under different confounding contexts. Experiments on three public datasets validated the effectiveness of CGCLER, and ablation analyses on the three datasets together with sensitivity analyses demonstrated the contributions and limitations of both the confounding-feature constraint and the contrastive learning module.

Future research could explore the integration of textual semantic information into the item–explanation joint ranking task or leverage advanced techniques grounded in causal theory to further isolate causal features from confounding factors, thereby enhancing the capabilities of CGCLER.

Author Contributions: Conceptualization, Y.G. and Z.Y.; Methodology, Y.G. and Z.Y.; Validation, Z.Y. and M.D.; Formal Analysis, Y.G. and Z.Y.; Investigation, Z.Y.; Resources, Y.G.; Data Curation, Z.Y. and M.D.; Writing—Original Draft, Z.Y. and M.D.; Writing—Review and Editing, Y.G., Z.Y. and M.D.; Visualization, Z.Y. and M.D.; Project Administration, Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available on GitHub, at https://github.com/lileipisces/EXTRA (accessed on 14 August 2026) or [7].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Shuai, J.; Wu, L.; Zhang, K.; Sun, P.; Hong, R.; Wang, M. Topic-enhanced Graph Neural Networks for Extraction-based Explainable Recommendation. In *SIGIR'23, Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*; ACM : New York, NY, USA, 2023; pp. 1188–1197. [CrossRef]
2. Tan, J.; Xu, S.; Ge, Y.; Li, Y.; Chen, X.; Zhang, Y. Counterfactual Explainable Recommendation. In *CIKM'21, Proceedings of the 30th ACM International Conference on Information & Knowledge Management*; ACM: New York, NY, USA, 2021; pp. 1784–1793. [CrossRef]
3. Zhang, Y.; Chen, X. Explainable Recommendation: A Survey and New Perspectives. *Found. Trends Inf. Retr.* **2020**, *14*, 1–101. [CrossRef]
4. Li, L.; Zhang, Y.; Chen, L. Personalized Prompt Learning for Explainable Recommendation. *ACM Trans. Inf. Syst.* **2023**, *41*, 1–26. [CrossRef]
5. Geng, S.; Fu, Z.; Tan, J.; Ge, Y.; de Melo, G.; Zhang, Y. Path Language Modeling over Knowledge Graphs for Explainable Recommendation. In *WWW'22, Proceedings of the ACM Web Conference 2022*; ACM: New York, NY, USA, 2022; pp. 946–955. [CrossRef]
6. Li, L.; Zhang, Y.; Chen, L. On the Relationship between Explanation and Recommendation: Learning to Rank Explanations for Improved Performance. *ACM Trans. Intell. Syst. Technol.* **2023**, *14*, 1–24. [CrossRef]
7. Li, L.; Zhang, Y.; Chen, L. EXTRA: Explanation Ranking Datasets for Explainable Recommendation. In *SIGIR'21, Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*; ACM: New York, NY, USA, 2021; pp. 2463–2469. [CrossRef]
8. Zhang, Y.; Feng, F.; He, X.; Wei, T.; Song, C.; Ling, G.; Zhang, Y. Causal Intervention for Leveraging Popularity Bias in Recommendation. In *SIGIR'21, Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*; ACM: New York, NY, USA, 2021; pp. 11–20. [CrossRef]
9. He, X.; Zhang, Y.; Feng, F.; Song, C.; Yi, L.; Ling, G.; Zhang, Y. Addressing Confounding Feature Issue for Causal Recommendation. *ACM Trans. Inf. Syst.* **2023**, *41*, 1–23. [CrossRef]
10. Zhu, J.; Wang, Y.; Zhu, F.; Sun, Z. Causal Deconfounding via Confounder Disentanglement for Dual-Target Cross-Domain Recommendation. *ACM Trans. Inf. Syst.* **2025**, *43*, 1–33. [CrossRef]
11. Xu, S.; Ge, Y.; Li, Y.; Fu, Z.; Chen, X.; Zhang, Y. Causal Collaborative Filtering. In *ICTIR'23, Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*; ACM: New York, NY, USA, 2023; pp. 235–245. [CrossRef]
12. Zhu, X.; Zhang, Y.; Feng, F.; Yang, X.; Wang, D.; He, X. Mitigating Hidden Confounding Effects for Causal Recommendation. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 4794–4805. [CrossRef]
13. Guo, Z.; Song, P.; Feng, C.; Yao, K.; Liang, J. Causal inference for alleviating confounding bias in multi-criteria rating recommendation. *Inf. Process. Manag.* **2026**, *63*, 104364. [CrossRef]
14. Ghazimatin, A.; Pramanik, S.; Saha Roy, R.; Weikum, G. ELIXIR: Learning from User Feedback on Explanations to Improve Recommender Models. In *WWW'21, Proceedings of the Web Conference 2021*; ACM: New York, NY, USA, 2021; pp. 3850–3860. [CrossRef]
15. Yang, A.; Wang, N.; Cai, R.; Deng, H.; Wang, H. Comparative Explanations of Recommendations. In *WWW'22, Proceedings of the ACM Web Conference 2022*; ACM: New York, NY, USA, 2022; pp. 3113–3123. [CrossRef]
16. Zhang, Y.; Lai, G.; Zhang, M.; Zhang, Y.; Liu, Y.; Ma, S. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *SIGIR'14, Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*; ACM: New York, NY, USA, 2014; pp. 83–92. [CrossRef]
17. Chen, X.; Chen, H.; Xu, H.; Zhang, Y.; Cao, Y.; Qin, Z.; Zha, H. Personalized Fashion Recommendation with Visual Explanations based on Multimodal Attention Network: Towards Visually Explainable Recommendation. In *Proceedings of the SIGIR'19: 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*; ACM: New York, NY, USA, 2019; pp. 765–774. [CrossRef]
18. Wang, X.; Chen, Y.; Yang, J.; Wu, L.; Wu, Z.; Xie, X. A Reinforcement Learning Framework for Explainable Recommendation. In *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM)*; IEEE: Piscataway, NJ, USA, 2018; pp. 587–596. [CrossRef]
19. Chen, X.; Qin, Z.; Zhang, Y.; Xu, T. Learning to Rank Features for Recommendation over Multiple Categories. In *SIGIR'16, Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*; ACM: New York, NY, USA, 2016; pp. 305–314. [CrossRef]
20. Rendle, S.; Schmidt-Thieme, L. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM'10, Proceedings of the Third ACM International Conference on Web Search and Data Mining*; ACM: New York, NY, USA, 2010; pp. 81–90. [CrossRef]
21. Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.S. Neural Graph Collaborative Filtering. In *SIGIR'19, Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*; ACM: New York, NY, USA, 2019; pp. 165–174. [CrossRef]

22. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR'20, Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*; ACM: New York, NY, USA, 2020; pp. 639–648. [CrossRef]
23. Chen, B.; Guo, W.; Tang, R.; Xin, X.; Ding, Y.; He, X.; Wang, D. TGCN: Tag Graph Convolutional Network for Tag-Aware Recommendation. In *CIKM'20, Proceedings of the 29th ACM International Conference on Information & Knowledge Management*; ACM: New York, NY, USA, 2020; pp. 155–164. [CrossRef]
24. Wei, T.; Chow, T.W.; Ma, J.; Zhao, M. ExpGCN: Review-aware Graph Convolution Network for explainable recommendation. *Neural Netw.* **2023**, *157*, 202–215. [CrossRef] [PubMed]
25. Liu, F.; Cheng, Z.; Zhu, L.; Liu, C.; Nie, L. An Attribute-Aware Attentive GCN Model for Attribute Missing in Recommendation. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 4077–4088. [CrossRef]
26. Xu, S.; Tan, J.; Heinecke, S.; Li, V.J.; Zhang, Y. Deconfounded Causal Collaborative Filtering. *ACM Trans. Recomm. Syst.* **2023**, *1*, 1–25. [CrossRef]
27. Sui, Y.; Wang, X.; Wu, J.; Lin, M.; He, X.; Chua, T.S. Causal Attention for Interpretable and Generalizable Graph Classification. In *Proceedings of the KDD'22: 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2022; pp. 1696–1705. [CrossRef]
28. Zhao, W.; Tang, D.; Chen, X.; Lv, D.; Ou, D.; Li, B.; Jiang, P.; Gai, K. Disentangled Causal Embedding With Contrastive Learning For Recommender System. In *WWW'23 Companion, Proceedings of the Companion ACM Web Conference 2023* ; ACM: New York, NY, USA, 2023; pp. 406–410. [CrossRef]
29. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence UAI'09* ; AUAI Press: Arlington, VA, USA, 2009; pp. 452–461. [CrossRef]
30. Chen, G.; Wang, Y.; Guo, F.; Guo, Q.; Shao, J.; Shen, H.; Cheng, X. Causality and Independence Enhancement for Biased Node Classification. In *CIKM'2023 , Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, Birmingham, UK, 21–25 October 2023*; Frommholz, I., Hopfgartner, F., Lee, M., Oakes, M., Lalmas, M., Zhang, M., Santos, R.L.T., Eds.; ACM: New York, NY, USA, 2023; pp. 203–212. [CrossRef]
31. Pan, D.; Li, X.; Li, X.; Zhu, D. Explainable Recommendation via Interpretable Feature Mapping and Evaluation of Explainability. *arXiv* **2020**, arXiv:2007.06133.
32. Zhao, W.X.; Mu, S.; Hou, Y.; Lin, Z.; Chen, Y.; Pan, X.; Li, K.; Lu, Y.; Wang, H.; Tian, C.; et al. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *CIKM'21, Proceedings of the 30th ACM International Conference on Information & Knowledge Management*; ACM: New York, NY, USA, 2021; pp. 4653–4664. [CrossRef]