

# NFARec: A Negative Feedback-Aware Recommender Model

Xinfeng Wang  
g22dtsa7@yamanashi.ac.jp  
University of Yamanashi  
Kofu, Japan

Fumiyo Fukumoto  
fukumoto@yamanashi.ac.jp  
University of Yamanashi  
Kofu, Japan

Jin Cui  
g22dtsa5@yamanashi.ac.jp  
University of Yamanashi  
Kofu, Japan

Yoshimi Suzuki  
ysuzuki@yamanashi.ac.jp  
University of Yamanashi  
Kofu, Japan

Dongjin Yu  
yudj@hdu.edu.cn  
Hangzhou Dianzi University  
Hangzhou, China

## ABSTRACT

Graph neural network (GNN)-based models have been extensively studied for recommendations, as they can extract high-order collaborative signals accurately which is required for high-quality recommender systems. However, they neglect the valuable information gained through negative feedback in two aspects: (1) different users might hold opposite feedback on the same item, which hampers optimal information propagation in GNNs, and (2) even when an item vastly deviates from users' preferences, they might still choose it and provide a negative rating. In this paper, we propose a negative feedback-aware recommender model (NFARec) that maximizes the leverage of negative feedback. To transfer information to multi-hop neighbors along an optimal path effectively, NFARec adopts a feedback-aware correlation that guides hypergraph convolutions (HGCs) to learn users' structural representations. Moreover, NFARec incorporates an auxiliary task - predicting the feedback sentiment polarity (i.e., positive or negative) of the next interaction - based on the Transformer Hawkes Process. The task is beneficial for understanding users by learning the sentiment expressed in their previous sequential feedback patterns and predicting future interactions. Extensive experiments demonstrate that NFARec outperforms competitive baselines. Our source code and data are released at <https://github.com/WangXFng/NFARec>.

## CCS CONCEPTS

• Information systems → Recommender systems.

## KEYWORDS

Recommendation, Negative Feedback, HyperGraph Convolution, Transformer Hawkes Process

## ACM Reference Format:

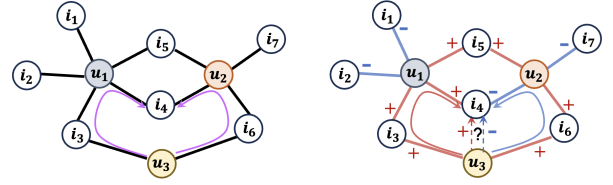
Xinfeng Wang, Fumiyo Fukumoto, Jin Cui, Yoshimi Suzuki, and Dongjin Yu. 2024. NFARec: A Negative Feedback-Aware Recommender Model. In *Proceedings of the 47th International ACM SIGIR Conference on Research and*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SIGIR '24, July 14–18, 2024, Washington, DC, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0431-4/24/07.  
<https://doi.org/10.1145/3626772.3657809>



(a) Message Passing in Vanilla GCNs (b) Message Passing in Feedback-Guided GCNs

**Figure 1: Illustration of motivation. In (a), vanilla GCN operators transmit messages from  $u_3$  to  $i_4$  through their interactive connections. However, the optimal message-passing path, either  $(u_3 \rightarrow i_3 \rightarrow u_1 \rightarrow i_4)$  or  $(u_3 \rightarrow i_6 \rightarrow u_2 \rightarrow i_4)$ , is determined by the user,  $u_1$  or  $u_2$ , with greater similarity to  $u_3$  in (b).**

*Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3626772.3657809>*

## 1 INTRODUCTION

With the advent of the internet era, effective recommender systems bring considerable benefits to both service supporters and customers in various fields, such as social service, movie, and book recommendations [25, 34, 47, 50]. Graph convolution networks (GCNs) have been widely employed to capture user-item interactive signals for recommendations due to their effectiveness in capturing high-order structural features [6, 8, 11]. Several authors have incorporated self-supervised learning [44, 46], hypergraph correlations [37, 38, 46], and contrastive augmentations [1, 14, 43] into neural graph learning to boost the representation capability of node embeddings. However, they do not pay attention to the fact that individual users may give different or conflicting feedback for the same item, making GCNs generate flawed node representations. For instance, in Fig. 1, both user  $u_1$  and  $u_2$  exhibit similarity with user  $u_3$ . This is because  $u_1$  and  $u_3$  provide positive feedback on item  $i_3$ , and both  $u_2$  and  $u_3$  give positive feedback on item  $i_6$ . However, there is a divergence in their preferences when it comes to item  $i_4$ :  $u_1$  provides positive feedback, while  $u_2$  provides negative feedback. Typically, recommender algorithms assess whether user  $u_1$  or  $u_2$  is more similar to  $u_3$  to predict the preferences of user  $u_3$  for item  $i_4$ . If  $u_3$  expresses a positive sentiment score toward item  $i_4$ , the path  $(u_3 \rightarrow i_3 \rightarrow u_1 \rightarrow i_4)$  should be the optimal message-passing than the path  $(u_3 \rightarrow i_6 \rightarrow u_2 \rightarrow i_4)$ . The observations show that when

treating neighboring nodes equally in GCNs, conducting convolution along the connections between nodes results in a lack of distinction among nodes. This suggests that user feedback is an important factor to guide message-passing in GCNs.

Negative feedback from interactive events, such as low ratings [9, 27], clicking dislikes [5, 33, 54], or skipping content [5, 9], have been utilized to generate negative representations for recommendations. However, their approaches learn the negative and positive representations independently. This results in their methods failing to effectively capture behavioral features across users' interacted items. Even though these items receive diverse positive and negative feedback, there still exists some connection among them that leads to interactions with the same user. Several recent studies either exclude negative instances, such as ratings below 4 out of 5 [44, 45, 47], or regard all interaction records as positive feedback. These approaches either simplify the task, which deviates from real-world scenarios, or overlook crucial feedback features to enrich user representations, leading to a performance decrease. In contrast, our solution is to utilize interaction feedback to guide GCN operators to learn both behavioral features and user feedback.

User sequential patterns, such as time-aware patterns [4], long- and short-term user behaviors [2, 26], and spatiotemporal relationships [15, 21], are extensively employed to improve sequential recommendations [11, 22, 24, 39, 41]. More recently, several efforts [29, 37, 38] have been made to integrate sequential and graph-based representations for recommendations. Despite their inspiring results, they overlook the fact that even though an item significantly diverges from a user's preferences, the user may still try it and provide negative feedback. The issue prevents the model from understanding user behavioral characteristics. For instance, some customers prefer products with lower prices over reputations, while others may prefer to be cautious and only engage with highly rated items after a thorough evaluation. Hence, it is beneficial for understanding users by learning the sentiment expressed in their previous sequential patterns of feedback, together with predicting whether they will interact with an item or not.

Motivated by the aforementioned issues, we propose a negative feedback-aware recommender model (NFARec) that maximizes the use of negative feedback in sequential and structural patterns. For sequential patterns, the NFARec incorporates an auxiliary task - predicting the feedback sentiment polarity of the next interaction - based on the Transformer Hawkes Process (THP). This is motivated by the advantage of the THP to model the influence of past events on the current event, whose impact diminishes over time. As such, the task prompts the NFARec to recognize the items that the user interacted with but may not prefer and learn the emotional relatedness between those interacted items to understand users.

To enhance the modeling of structural patterns, we propose a two-phase hypergraph convolution (HGC) approach that leverages high-order feedback relations between users and items. In the initial stage, we employ HGCs to capture correlations beyond pairwise interactions among users and items via a user hypergraph, in which a hyperedge denotes a user and nodes in the hyperedge refer to the items interacted by the user. Then, we guide the HGC operator to efficiently exchange messages between neighboring nodes during convolutions through a high-order feedback-aware path. The main contributions of our work can be summarized as follows:

- We propose a negative feedback-aware recommender model that maximizes the leverage of negative feedback in sequential and structural patterns.
- We adopt an auxiliary task to learn the sentiment relatedness between the user's interacted items based on THP. We empirically find that the task promotes the NFARec to identify items the user has interacted with but may not prefer.
- We propose a two-phase HGC approach that leverages feedback relations between users and items to construct an optimal path for message-passing in HGC.
- Extensive experiments on five public real-world datasets demonstrate that our NFARec model outperforms SOTA recommender methods.

## 2 RELATED WORK

**Sequential Neural Network for Recommendation.** Early efforts [12, 13] concentrate on modeling user sequential trajectories for recommendations. For instance, Wang et al. [26] propose to mine periodic and short-term patterns from sequential behaviors. Luo et al. [15] exploit spatiotemporal information of the historical check-ins. Yang et al. [41] present a user-agnostic global trajectory flow map to learn discontinuous interactions. Since then, attention-based algorithms [3, 22, 30, 35] have been explored to learn representative items from historical interactions. Furthermore, attempts [2, 37, 38, 52] to incorporate sequential and spatial representations gained inspiring results. Another attempt has been made to improve training strategies, such as predicting masked items [20] and intention disentanglement [16]. More recently, contrastive learning (CL) draws intensive attention to derive self-supervision signals from user behavior sequences [4, 39]. However, they neglect that a user may still provide negative feedback on an item that significantly diverges from the user's preferences.

**Graph Neural Network for Recommendation.** Many efforts [8, 18, 25, 31, 32, 34, 47] in GCNs have been made to capture high-order signals from the user-item interactive graph. Several authors perform theoretical strategies to enhance GCNs, including pre-training GCNs [6], low-pass collaborative filter networks [50], multi-view intent disentangle graph networks [53]. Furthermore, contrastive augmentations uniformize graph embeddings by ensuring the consistency between representations derived from contrastive graph views [1, 14, 43, 44]. These approaches are further improved to mine rich information against data sparsity by stacking Transformers behind GCN layers [29, 37, 38]. However, few of them address the different feedback for the same item in GCN that results in defective node representations.

**Negative Feedback Learning for Recommendation.** Recent work exploits interactive events that provide negative feedback, such as low ratings [9, 27], dislikes [5, 33, 54], or skipped content [5, 9], to create negative representations for recommendations. They treat negative and positive representations independently, failing to capture behavioral features across users' interacted items. Some attempts exclude negative instances (e.g., ratings below 4 [44, 45, 47]) or regard all interaction records as positive feedback. They either make the task deviate from real-world scenarios or lose feedback features, resulting in a performance drop.

### 3 PRELIMINARIES

#### 3.1 Definitions

**(User-Item Feedback Graph).** The user-item feedback graph  $\mathcal{G} = (\mathcal{U}, \mathcal{I}, \mathcal{E})$  shows the network of feedback correlations between users and items, where  $\mathcal{U}$  and  $\mathcal{I}$  denote a set of users and items, respectively.  $\mathcal{E} = \{\zeta_{u,i} \mid u \in \mathcal{U}, i \in \mathcal{I}\}$  denotes the set of edges. Each element  $\zeta_{u,i}$  equals 1 if user  $u$  provides positive feedback for item  $i$ , -1 if user  $u$  gives negative feedback for item  $i$ , and 0 otherwise.

**(Item Recommendation).** Given a user historical interaction set  $\mathcal{S}$ , where  $\mathcal{S}^{(u)} = \{i_k\}_{k=1}^{|\mathcal{S}^{(u)}|}$  denotes the item sequence of user  $u$ , and  $|\mathcal{S}^{(u)}|$  refers to the number of items in  $\mathcal{S}^{(u)}$ , the goal is to generate a list of candidate items with the probabilities.

#### 3.2 Hawkes Process

The basic concept of the Hawkes Process [7] is to model the impact of past events on the occurrence of the current event, and the influence decreases over time. The original proposal suggests using exponential and power-law functions for the intensity function. Thereafter, due to the capacity of recurrent neural networks (RNNs) in modeling hidden states of sequential events, Mei and Eisner [17] propose a neural Hawkes process (NHP) to replace the intensity function with the learning process of RNNs. More recently, since the Transformer [23] can capture long-range dependencies between every two events in the sequence, Zuo et al. [55] propose THP that adopts the self-attention mechanism as the core of its intensity function. Motivated by this, we utilize the THP to infer the feedback sentiment polarity of the next interactions, to capture underlying sentiment towards users' interacted items.

#### 3.3 Enhanced HGC with Feedback

HGCs have been extensively employed to mine user-item collaborative signals in recommendations that go beyond pairwise user-item connections [10, 37, 38, 40, 46] since each hyperedge in the hypergraph can connect more than two nodes [48, 49]. They typically regard nodes equally weighted in each hyperedge to perform convolution, while neglecting that various users may give opposite feedback for the same items. As shown in Fig. 1 (b), the feedback sentiment of items  $i_3$  and  $i_4$  given by user  $u_1$  are both positive, indicating that the HGC operator should pass information from  $i_3$  to  $i_4$ . In contrast, there should be fewer messages from  $i_6$  to  $i_4$  as user  $u_2$  gives different negative feedback on these two items. To address the issue, we introduce a two-phase HGC approach that leverages feedback relations between users and items to perform more efficient information propagation during convolutions.

### 4 METHODOLOGY

The framework of NFARec is illustrated in Fig. 2, which consists of sequential and structural representation learning.

#### 4.1 Sequential Representation Learning

Given a matrix  $\mathcal{V} \in \mathbb{R}^{|\mathcal{I}| \times d_m}$  representing item embeddings, where  $d_m$  is the dimension size, the embedding  $v_i$  for an item  $i$  is obtained by using the one-hot vector  $\mathbf{v}'_i \in \mathbb{R}^{|\mathcal{I}|}$  as a slicing operation that selects corresponding item embedding vectors on  $\mathcal{V}$ :  $v_i = \mathcal{V}\mathbf{v}'_i$ .

Since each of the past interactions has an impact on the current interaction occurrence, we employ the Transformer blocks to capture long-term sequential dependencies through the self-attention scores as shown in the THP of Fig. 2. Considering  $\mathcal{T}_t = \{(t_j, z_j) : t_j < t\}$  as a user's historical interactions up to time  $t$  with the feedback sentiment polarity  $z_j \in \{\text{positive}, \text{negative}\}$ , we feed the interaction sequence embedding  $\mathcal{V}[\mathcal{T}_t] = \mathcal{V} \cdot \text{Concat}([\mathbf{v}'_1, \mathbf{v}'_2, \dots, \mathbf{v}'_t])$  as the input,  $\mathbf{e}_t^{(0)}$  into the first layer of the Transformer. Thus, the representation  $\hat{\mathbf{e}}_t^{(l)}$  of historical interactions at time  $t$  in the  $l$ -th layer is obtained by the following equation:

$$\mathbf{e}_t^{(l+1)} = \alpha_1 \left( \sum_{j \in \mathcal{T}_t} \frac{\text{Mask}(\mathbf{e}_t^{(l)} \mathbf{W}_*^q (\mathbf{e}_j^{(l)} \mathbf{W}_*^k)^T)}{\sqrt{d_k}} \right) \mathbf{e}_t^{(l)} \mathbf{W}_*^v, \quad (1)$$

where  $\alpha_1$  denotes softmax activation function and  $\mathbf{W}^q$ ,  $\mathbf{W}^k$  and  $\mathbf{W}^v$  are three weight matrices. The function  $\text{Mask}(\cdot)$  refers to a masking operation [23, 55] using an upper triangular matrix. This ensures that  $\mathbf{e}_t$  is informed by  $\mathbf{e}_{t-1}$  while avoiding  $\mathbf{e}_{t-1}$  from peeking into future interactions.

After stacking the number of  $L_1$  Transformer layers, an average-pooling operation, and an L2 norm normalization, we obtain the hidden representation  $\mathbf{h}(t_j) = \mathbf{H}_{j,*}$  for each interaction  $(t_j, z_j)$  in the sequence. The user sequential representations are obtained by considering the hidden states of users' historical interactions:

$$\mathbf{e}_u^{\mathcal{S}} = \text{SumPooling}(\{\mathbf{h}(t_j) : j \in |\mathcal{T}_t|\}). \quad (2)$$

To further enhance user sequential representations by ensuring sentiment consistency among interacted items, we introduce an auxiliary task, i.e., predicting the feedback sentiment polarity of the next interaction. Specifically, let  $\lambda(t|\mathcal{T}_t)$  be the conditional intensity function to model the influence of the past interactions up to time  $t$ . It accumulates the influences of both types of feedback in the past interactions as follows:

$$\lambda(t|\mathcal{T}_t) = \sum_z \lambda_z(t|\mathcal{T}_t), \quad (3)$$

where the intensity ( $\lambda_z(t|\mathcal{T}_t)$ ) at time  $t$  is determined by the historical hidden state  $\mathbf{h}(t_j)$ , along with the current state denoted as  $\alpha_z \frac{t-t_j}{t_j}$  (where  $t_j$  is the timestamp of the last interaction before  $t$ ):

$$\lambda_z(t|\mathcal{T}_t) = \text{softplus} \left( \alpha_z \frac{t-t_j}{t_j} + \text{MLP}(\mathbf{h}(t_j)) \right), \quad (4)$$

where  $\text{softplus}(x) = \beta_z \log(1 + \exp(x/\beta_z))$  is the softplus function with a trainable softness parameter  $\beta_z$  and  $\alpha_z$  indicates a learnable parameter. The goal of the task is to maximize the intensity of the feedback sentiment polarity for every next interaction, i.e.,  $j \in |\mathcal{T}_t|$ , which is given by:

$$\hat{z}_{j+1} = \arg\max_z \frac{\lambda_z(t_{j+1}|\mathcal{T}_t)}{\lambda(t_{j+1}|\mathcal{T}_t)}. \quad (5)$$

It is noteworthy that the hidden representation  $\mathbf{h}(t)$  by NFARec is enhanced throughout the auxiliary task by predicting the feedback polarity of every next item. As such, NFARec explores the sentiment relatedness among the items in the interaction sequences. This enables NFARec to learn to predict items that the user will interact with, even if the user may not necessarily prefer them.

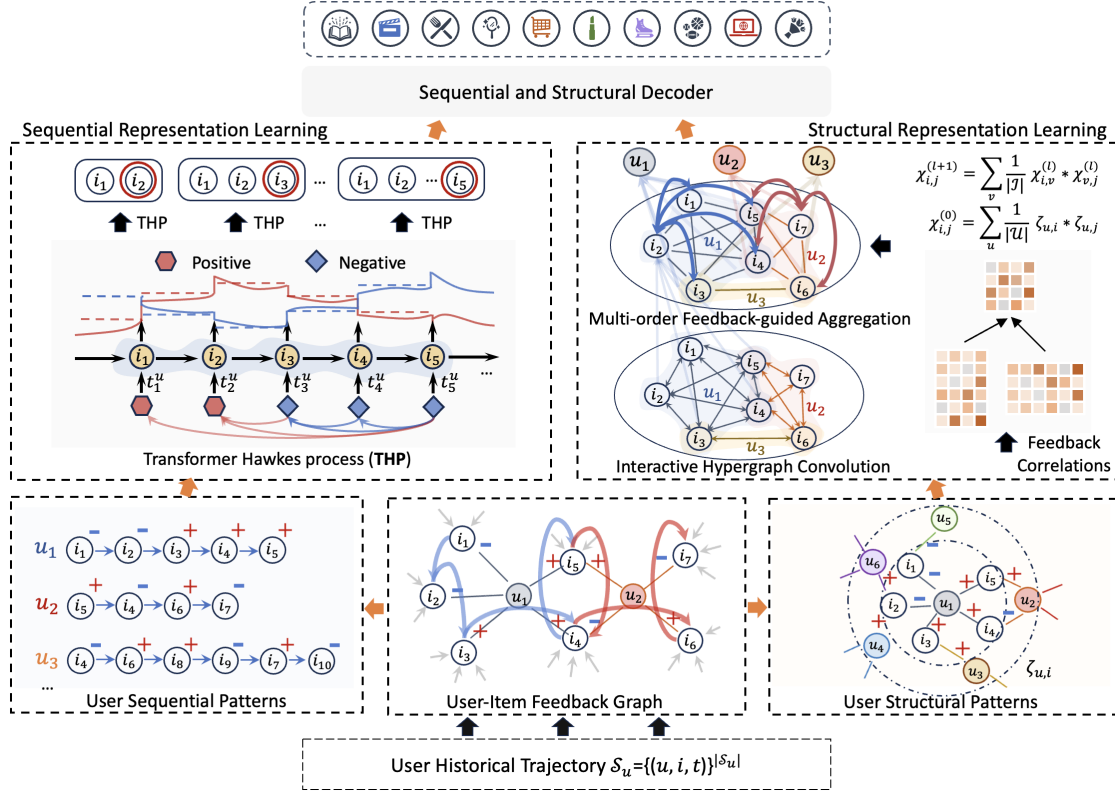


Figure 2: Architecture of our NFARec framework. NFARec learns negative feedback in both sequential and structural patterns.

## 4.2 Structural Representation Learning

Motivated by HGCs [29, 37, 38, 40, 42], we propose a two-parse HGC paradigm to exploit the rich feedback features. As illustrated in Fig. 2, the paradigm includes an interactive hypergraph convolution and a multi-order feedback-aware aggregation.

**Interactive Hypergraph Convolution.** Following the works [8, 29] that simplify GCNs and make them more appropriate for recommendations, we removed the global weight matrix and added a linear feature transformation ( $\mathbf{W}_1 \in \mathbb{R}^{d_m \times d_m}$ ). Given the global item embeddings  $\mathcal{V}$  as the input  $\Lambda^{(0)}$  of the first layer and the user hyperedge  $\mathcal{H}^{(u)} \in \mathbb{R}^{1 \times |\mathcal{I}|}$ , the matrix equivalence form of informative diffusion in the HGC is as follows:

$$\Lambda^{(l+1)} = \mathcal{H}^{(u)} \hat{\mathcal{A}} \mathcal{H}^{(u)\top} \alpha_2(\Lambda^{(l)} \mathbf{W}_1), \quad (6)$$

where  $\Lambda^{(l)} \in \mathbb{R}^{|\mathcal{I}| \times d_m}$  indicates the output of the  $l$ -th HGC layer,  $\alpha_2$  is the ELU activation function, and  $\hat{\mathcal{A}}$  denotes the symmetrically normalized adjacency matrix:

$$\hat{\mathcal{A}} = \mathbf{D}^{-\frac{1}{2}} \mathcal{A} \mathbf{D}^{-\frac{1}{2}}, \quad (7)$$

where  $\mathcal{A} \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{I}|}$  denotes the global adjacent matrix for the item-item graph, and  $a_{ij} = 1$ , if item  $i$  and  $j$  have interacted with the same user, otherwise 0.  $\mathbf{D} \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{I}|}$  is a diagonal matrix in which each entry  $d_{ij}$  denotes the number of nonzero entries in the  $i$ -th row vector of the adjacency matrix  $\mathcal{A}$ .

After repeating the number of  $L_2$  HGCs and an average pooling operation, the user structural representations are obtained by

aggregating the representations of users' interacted items:

$$e_u^{\mathcal{H}_1} = \text{AvgPooling}(\Lambda^{(L_2)}). \quad (8)$$

**Feedback-aware Aggregation.** To spread the information through an optimal path during convolution, we exploit user feedback to enhance HGC operators. There are two common scenarios among adjacent nodes, which are illustrated in Fig. 3: (a) opposite feedback, where user  $u_j$  provides positive and negative feedback on  $i_1$  and  $i_2$ , respectively; (b) multiple feedback, where three users —  $u_j$ ,  $u_k$ , and  $u_l$  — offer varying feedback on  $i_1$  and  $i_2$ . Hence, we construct a feedback correlation matrix  $\mathcal{Z} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$ . Each element  $\zeta_{i,j}$  of  $\mathcal{Z}$  is set to 1 if user  $u_i$  provides positive feedback for item  $v_j$ , -1 if user  $u_i$  gives negative feedback for item  $v_j$ , and 0 otherwise. Subsequently, we create the 1st-order feedback correlation matrix  $\chi^{(0)} \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{I}|}$  by incorporating feedback among users interacting with the same items:

$$\chi^{(0)} = \begin{bmatrix} \chi_{1,1}^{(0)} & \chi_{1,2}^{(0)} & \cdots & \chi_{1,|\mathcal{I}|}^{(0)} \\ \chi_{2,1}^{(0)} & \chi_{2,2}^{(0)} & \cdots & \chi_{2,|\mathcal{I}|}^{(0)} \\ \vdots & \vdots & \ddots & \vdots \\ \chi_{|\mathcal{I}|,1}^{(0)} & \chi_{|\mathcal{I}|,2}^{(0)} & \cdots & \chi_{|\mathcal{I}|,|\mathcal{I}|}^{(0)} \end{bmatrix}, \chi_{i,j}^{(0)} = \sum_{u \in \mathcal{U}} \frac{\zeta_{i,u} \cdot \zeta_{u,j}}{|\mathcal{U}|}. \quad (9)$$

To model the multi-hop feedback correlations, such as the relationship between item  $i_1$  and  $i_3$  in Fig. 3 (c), we distill the high-order

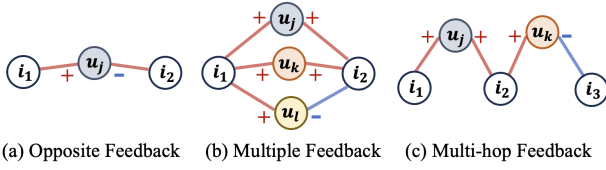


Figure 3: Illustration of various feedback.

relationships from the 1st-order feedback correlations. The correlation matrices denoted by  $\chi^{(l)}$  are computed layer by layer:

$$\chi^{(l+1)} = \begin{bmatrix} \chi_{1,1}^{(l+1)} & \chi_{1,2}^{(l+1)} & \cdots & \chi_{1,|I|}^{(l+1)} \\ \chi_{2,1}^{(l+1)} & \chi_{2,2}^{(l+1)} & \cdots & \chi_{2,|I|}^{(l+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \chi_{|I|,1}^{(l+1)} & \chi_{|I|,2}^{(l+1)} & \cdots & \chi_{|I|,|I|}^{(l+1)} \end{bmatrix}, \chi_{i,j}^{(l+1)} = \sum_{v \in I} \frac{\chi_{i,v}^{(l)} \cdot \chi_{v,j}^{(l)}}{|I|}. \quad (10)$$

To integrate the multi-order correlations across different orders, we define the correlation matrix by  $\hat{\chi} = (\chi^{(0)} + \chi^{(1)} + \cdots)$ . Similar to Eq. (1), we utilize the masking technique to avoid message-passing between conflicting nodes. Specifically, if an element in the mask is below 0, it is masked; otherwise, it remains unchanged. The feedback-aware user representation is given as follows:

$$e_u^{\mathcal{H}_2} = \text{AvgPooling}(\mathcal{H}^{(u)} \text{Mask}(\hat{\chi}) \Lambda^{(L_2)}). \quad (11)$$

Recall that the HGC paradigm models the interactions between users and items in the first stage, and utilizes the feedback to guide HGC operators in the second stage, enabling the NFARec to learn both behavioral features and user feedback for recommendations.

### 4.3 Sequential and Structural Decoder

We employ the inner products between user sequential and structural embeddings, i.e.,  $e_u^S$  and  $e_u^{\mathcal{H}_1}$ , and the item embeddings, i.e.,  $\mathcal{V}$ , to infer their preferences in candidate items. In addition, the model leverages the feedback-aware structural feature,  $e_u^{\mathcal{H}_2}$ , to enhance its robustness to user feedback. We define the sequential, structural, and feedback-aware structural features as Seq, Gra<sub>1</sub>, and Gra<sub>2</sub>, respectively. The final rating scores over all candidate items are given by:

$$\mathbf{r}_{u,*} = \alpha_3 (\underbrace{e_u^S \mathcal{V}^\top}_{\text{Seq}} + \underbrace{e_u^{\mathcal{H}_1} \mathcal{V}^\top}_{\text{Gra}_1} + \underbrace{\delta \text{MLP}(e_u^{\mathcal{H}_2})}_{\text{Gra}_2}), \quad (12)$$

where MLP denotes a linear transformer to decode Gra<sub>2</sub> and  $\alpha_3$  indicates the tanh activation function.  $\delta$  refers to the hyperparameter to assign weight to the third term.

### 4.4 Model Optimization

The NFARec consists of a main task, a recommendation, and an auxiliary task, the next feedback sentiment polarity prediction.

**Recommendation.** We adopt a weighted multi-label cross-entropy loss [29] as the main objective function:

$$\mathcal{L}_{\text{main}} = \sum_{(u \in \mathcal{U})} \sum_{(i \in I)} -c_{u,i} * \gamma_i^{(u)} \log(\alpha_4(\hat{\mathbf{r}}_{u,i})), \quad (13)$$

**Table 1: Statistics of datasets. "#Avg." denotes the average count of users' interactions. "Perc.(#Pos/#Neg)" refers to the percentage of positive and negative samples.**

Dataset	#Users	#Items	#Interactions	Perc.(#Pos/#Neg)	#Avg.
Yelp 2023	48,993	34,298	3,415,569	72.4%/27.6%	39.5
Beauty	22,363	12,101	198,502	80.2%/19.8%	8.9
Book	19,804	22,086	1,273,679	80.6%/19.4%	64.3
Recipes	7,452	12,911	311,389	82.5%/17.5%	41.8
MovieLens	6,041	3,955	1,000,209	73.5%/26.5%	165.6

where  $\gamma^{(u)} \in \mathbb{R}^{|I|}$  represents a label vector, in which the element equals 1 if the corresponding item is a ground-truth candidate; otherwise, 0.  $\alpha_4$  denotes the sigmoid activation function.  $c_{u,i}$  is set to  $\beta_1$  when the user has interacted with the item,  $\beta_2$  when the user will interact with the item, and 0 otherwise.

**Next Feedback Sentiment Polarity Prediction.** Given user  $u$ 's interaction sequence  $S_u$  with a time interval of  $[t_1, t_{|S_u|}]$ , the log-likelihood for the task is given as follows:

$$\mathcal{L}_{\text{auxi}} = \sum_{(u \in \mathcal{U})} \left( \underbrace{\sum_{j=1}^{|S_u|} \log \lambda(t_j | \mathcal{T}_t)}_{\text{interactive log-likelihood}} - \underbrace{\int_{t_1}^{t_{|S_u|}} \lambda(t | \mathcal{T}_t) dt}_{\text{continuous log-likelihood}} \right). \quad (14)$$

The task is learned by maximizing interactive and continuous log-likelihood. The final multi-task objective loss is given by:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{main}} + \delta_2 \mathcal{L}_{\text{auxi}}, \quad (15)$$

where  $\delta_2$  is the hyperparameter for the auxiliary task.

To make the continuous log-likelihood, i.e.,  $\Delta = \int_{t_1}^{t_{|S_u|}} \lambda(t | \mathcal{T}_t) dt$ , trainable by back-propagation, we exploit Monte Carlo integration (MCI) [19] following Zuo et al.'s work [55], which leverages random sampling to estimate definite integrals, to approximate  $\Delta$ :

$$\hat{\Delta}_{\text{MCI}} = \sum_{j=2}^{|S_u|} (t_j - t_{j-1}) \left( \frac{1}{N} \sum_{k=1}^N \lambda(x_k) \right), \quad (16)$$

where  $N$  represents the number of sampling points in MCI, and  $x_k \sim \text{uniform}(t_{j-1}, t_j)$  is a uniform distribution such that  $\lambda(x_k)$  can be calculated by back-propagation through the NFARec and the unbiased estimation  $\mathbb{E}[\hat{\Delta}_{\text{MCI}}]$  is equal to  $\Delta$ .

## 5 EXPERIMENT

### 5.1 Experimental Setup

**Datasets and Evaluation Metrics.** To examine user feedback, we collected five public datasets with rating scores, i.e., Yelp 2023<sup>1</sup>, Beauty<sup>2</sup>, Books<sup>3</sup>, Recipes<sup>4</sup>, and MovieLens [14]. Similar to previous works [9, 27, 44, 45, 47], we regard interactions with ratings below 4 as negative feedback; others are positive. We exclude users and items with fewer than  $n$  interactions, where  $n$  is 15, 10, 25, and 5 for Yelp 2023, Recipes, Books, and Beauty, respectively. The statistical

<sup>1</sup><https://www.yelp.com/dataset>

<sup>2</sup><https://www.kaggle.com/datasets/skillsnugger/amazon-ratings/>

<sup>3</sup><https://jmcauley.ucsd.edu/data/amazon/links.html>

<sup>4</sup><https://www.kaggle.com/datasets/shuyangli94/food-com-recipes-and-user-interactions/>

**Table 2: Performance comparison of recommender models. Bold: Best, underline: Second best. “\*” indicates that the improvement is statistically significant (p-value < 0.01) in the 10-trial T-test.**

Dataset	Metric	CF-based		GCN-based			GCL-based			Dif-based	S&G-based			Ours	Improv.	p-value
		DirectAU	InvCF	LightGCN	SGL	HCCF	NCL	LightGCL	XSimGCL	DiffRec	SHT	AutoCF	EEDN			
Yelp 2023	Recall@5	0.0316	0.0321	0.0344	0.0366	0.0299	0.0360	0.0341	0.0376	0.0323	0.0340	0.0391	0.0426	<b>0.0449</b>	+5.6% *	1.4e-07
	NDCG@5	0.0497	0.0607	0.0540	0.0551	0.0426	0.0535	0.0479	0.0566	0.0487	0.0619	0.0614	<u>0.0668</u>	<b>0.0741</b>	+10.9% *	2.4e-09
	Recall@10	0.0517	0.0464	0.0564	0.0581	0.0403	0.0572	0.0502	0.0590	0.0529	0.0476	0.0535	<u>0.0672</u>	<b>0.0707</b>	+5.2% *	2.2e-06
	NDCG@10	0.0507	0.0644	0.0587	0.0603	0.0457	0.0571	0.0498	0.0595	0.0548	0.0673	0.0704	<u>0.0733</u>	<b>0.0778</b>	+6.1% *	4.9e-07
	Recall@20	0.0722	0.0798	0.0752	0.0721	0.0518	0.0795	0.0675	0.0857	0.0809	0.0796	0.0939	<u>0.1031</u>	<b>0.1074</b>	+4.2% *	6.1e-06
	NDCG@20	0.0714	0.0795	0.0715	0.0740	0.0526	0.0743	0.0661	0.0756	0.0713	0.0785	0.0810	<u>0.0858</u>	<b>0.0894</b>	+4.1% *	2.2e-06
MovieLens	Recall@5	0.0726	0.0728	0.0603	0.0665	0.0603	0.0673	0.0690	0.0726	0.0642	0.0698	0.0691	<u>0.0737</u>	<b>0.0789</b>	+7.1% *	1.8e-08
	NDCG@5	0.2969	0.3144	0.2438	0.2767	0.2490	0.2579	0.2556	0.3165	0.2710	0.2822	0.3084	<u>0.3222</u>	<b>0.3716</b>	+15.3% *	1.9e-12
	Recall@10	0.1044	0.1085	0.0863	0.1069	0.0967	0.1029	0.1046	0.1037	0.0923	0.1072	0.0991	<u>0.1136</u>	<b>0.1213</b>	+6.7% *	8.3e-06
	NDCG@10	0.2819	0.2831	0.2476	0.2829	0.2539	0.2497	0.2494	0.2754	0.2670	0.2739	0.2729	<u>0.2878</u>	<b>0.3314</b>	+15.1% *	3.03-13
	Recall@20	0.1504	0.1481	0.1107	0.1432	0.1287	0.1298	0.1313	0.1426	0.1204	0.1509	0.1470	<u>0.1628</u>	<b>0.1736</b>	+6.6% *	3.5e-07
	NDCG@20	0.2592	0.2537	0.2047	0.2541	0.2312	0.2479	0.2540	0.2481	0.2284	0.2400	0.2452	<u>0.2631</u>	<b>0.3007</b>	+14.2% *	5.6e-10
Recipes	Recall@5	0.0235	0.0167	0.0208	0.0265	0.0224	0.0207	0.0164	0.0253	0.0193	0.0249	0.0162	<u>0.0269</u>	<b>0.0292</b>	+8.6% *	1.6e-09
	NDCG@5	0.0359	0.0352	0.0283	0.0365	0.0330	0.0267	0.0257	0.0389	0.0283	0.0377	0.0264	<u>0.0395</u>	<b>0.0418</b>	+9.4% *	1.5e-08
	Recall@10	0.0426	0.0266	0.0333	0.0406	0.0340	0.0330	0.0296	0.0419	0.0324	0.0405	0.0273	0.0401	<b>0.0442</b>	+5.5% *	3.6e-06
	NDCG@10	0.0429	0.0254	0.0304	0.0382	0.0332	0.0293	0.0275	<u>0.0435</u>	0.0305	0.0412	0.0237	0.0429	<b>0.0459</b>	+5.5% *	1.4e-06
	Recall@20	0.0595	0.0376	0.0399	0.0566	0.0386	0.0402	0.0360	<u>0.0595</u>	0.0416	0.0560	0.0331	0.0592	<b>0.0609</b>	+2.4% *	1.4e-03
	NDCG@20	0.0476	0.0227	0.0329	0.0465	0.0346	0.0319	0.0299	<u>0.0481</u>	0.0392	0.0406	0.0186	0.0475	<b>0.0492</b>	+2.3% *	2.3e-04
Books	Recall@5	0.0392	0.0301	0.0329	0.0338	0.0252	0.0359	0.0275	0.0440	0.0254	0.0317	0.0329	<u>0.0473</u>	<b>0.0489</b>	+3.4% *	1.7e-04
	NDCG@5	0.0973	0.0827	0.0905	0.0874	0.0524	0.0912	0.0723	0.1029	0.0654	0.0834	0.0890	<u>0.1082</u>	<b>0.1139</b>	+5.3% *	1.0e-06
	Recall@10	0.0696	0.0590	0.0655	0.0693	0.0474	0.0667	0.0513	0.0712	0.0484	0.0680	0.0516	<u>0.0729</u>	<b>0.0753</b>	+3.3% *	1.1e-05
	NDCG@10	0.0988	0.0804	0.0956	0.0896	0.0635	0.0969	0.0715	0.0990	0.0632	0.0913	0.0727	<u>0.1011</u>	<b>0.1065</b>	+5.3% *	3.1e-07
	Recall@20	0.1029	0.0854	0.0994	0.0981	0.0670	0.0971	0.0672	0.1037	0.0647	0.0828	0.0762	<u>0.1075</u>	<b>0.1112</b>	+3.4% *	7.8e-05
	NDCG@20	0.1024	0.0792	0.1007	0.0992	0.0812	0.0943	0.0645	0.1033	0.0560	0.0799	0.0730	<u>0.1099</u>	<b>0.1141</b>	+3.9% *	1.1e-08
Beauty	Recall@5	0.0518	0.0530	0.0489	0.0522	0.0367	0.0521	0.0468	0.0548	0.0479	0.0520	0.0490	<u>0.0548</u>	<b>0.0585</b>	+6.8% *	5.7e-07
	NDCG@5	0.0447	0.0476	0.0414	0.0467	0.0344	0.0453	0.0417	0.0486	0.0402	0.0415	0.0437	<u>0.0476</u>	<b>0.0514</b>	+8.0% *	7.8e-07
	Recall@10	0.0725	0.0710	0.0704	0.0737	0.0517	0.0688	0.0615	0.0733	0.0657	0.0719	0.0759	<u>0.0779</u>	<b>0.0811</b>	+4.1% *	1.2e-06
	NDCG@10	0.0527	0.0522	0.0499	0.0530	0.0387	0.0502	0.0488	0.0557	0.0476	0.0513	0.0542	<u>0.0558</u>	<b>0.0596</b>	+6.8% *	8.1e-09
	Recall@20	0.0908	0.1022	0.0879	0.0939	0.0672	0.0839	0.0803	0.1007	0.0876	0.0901	0.1034	<u>0.1054</u>	<b>0.1103</b>	+4.6% *	3.0e-05
	NDCG@20	0.0614	0.0613	0.0570	0.0607	0.0413	0.0529	0.0499	0.0640	0.0517	0.0587	0.0625	<u>0.0648</u>	<b>0.0683</b>	+5.4% *	4.1e-05

details are summarized in Table 1. Following [29, 47], we split each dataset into training, validation, and test data with a ratio of 7:1:2. We employed the widely used Recall@K (R@K) and NDCG@K (N@K) with  $K \in \{5, 10, 20\}$  as metrics to evaluate performance.

**Baselines.** We compared the NFARec with the following 12 SOTA recommender models which are classified into five groups:

- **Collaborative Filtering (CF)-based method:**

**DirectAU** [25] optimizes alignment and uniformity of graph embeddings in the hypersphere. **InvCF** [51] discovers disentangled the preference and popularity semantics for recommendations.

- **Graph Convolution Network (GCN)-based method:**

**LightGCN** [8] simplifies GCNs, making them more appropriate for recommendations. **SGL** [34] exploits a self-supervised task to reinforce node representation. **HCCF** [34] proposes hypergraph-enhanced cross-view contrastive learning for recommendations.

- **Graph Contrastive Learning (GCL)-based method:**

**NCL** [14] regards users (or items) and their structural neighbors as contrastive pairs. **LightGCL** [1] utilizes singular value decomposition for refining contrastive learning. **XSimGCL** [47] adds uniform noise to the embedding space to create contrastive views.

- **Diffusion (Dif)-based method:**

**DiffRec** [28] learns the generative process in a denoising manner from personalized information in user interactions.

- **Sequential and Graph (S&G)-based method:**

**SHT** [38] explores the self-attention- and hypergraph-based collaborative relationships. **AutoCF** [36] exerts self-supervised learning to automatically augment data for recommendations. **EEDN** [14] exploits rich latent features between users, items, and interactions between users and items.

**Implementation and Parameter Settings.** The best parameters of the FNGRec were sampled as follows:  $\beta_1$ ,  $\beta_2$ ,  $\delta_1$ , and  $\delta_2$  were set to 0.12, 1.49, 1.2, and 1e-3 for Yelp 2023, 1.47, 3.99, 1.2, and 0.5 for MovieLens, 0.12, 3.81, 1.0 and 1e-5 for Recipes, 0.25, 3.53, 1.2, and 1e-5 for Books, and 0.62, 3.74, 1.2, and 1e-3 for Beauty, respectively.  $d_m$  was 1,024. The numbers  $L_1$  and  $L_2$  of Transformer layers and HGC layers were 1. These hyperparameters were tuned using Optuna<sup>5</sup>. The parameters for the baselines were tuned to attain the best performance or set as proposed by the authors. We implemented our NFARec and experimented with Pytorch on Nvidia GeForce RTX 3090 (24GB memory).

## 5.2 Overall Performance

Table 2 shows the performance comparison between NFARec and SOTA baselines on five datasets. The results highlight the effectiveness of NFARec in making the most of negative feedback for

<sup>5</sup><https://github.com/pfnet/optuna>



recommendations. Specifically, the NFARec surpasses the runner-ups XSimGCL and EEDN by 4.2 ~ 5.6% in Recall@ $k$  and 4.1 ~ 10.9% in NDCG@ $k$  on the Yelp 2023 dataset, 6.6 ~ 7.1% in Recall@ $k$  and 14.2 ~ 15.3% in NDCG@ $k$  on the MovieLens dataset, 2.4 ~ 8.6% in Recall@ $k$  and 2.3 ~ 9.4% in NDCG@ $k$  on the Recipes dataset, 3.3 ~ 3.4% in Recall@ $k$  and 3.9 ~ 5.3% in NDCG@ $k$  on the Books dataset, and 4.1 ~ 6.8% in Recall@ $k$  and 5.4 ~ 8.0% in NDCG@ $k$  on the Beauty dataset. Table. 2 also promotes the following observations and insights:

(1) The S&G-based methods are competitive among baselines, which indicates the effectiveness of both sequential and graph-based features to improve performance. Among them, EEDN achieves the highest performance across all datasets, suggesting that mining implicit features is beneficial to mitigate exposure bias for recommendations. However, their underperformance compared to NFARec shows that they do not take into account negative feedback in modeling user-item interactions.

(2) The GCN- and GCL-based methods capture high-order interactive information between users and items via GCNs. However, they typically treat neighboring nodes as equal in GCN and perform convolution along the connections between nodes, leading to a performance drop. The algorithms SGL, NCL, and XSimGCL outperform others because of their effective graph augmentations on edges or nodes. The lower effectiveness of LightGCL may be due to its susceptibility to conflicting feedback from different users, but this aspect was overlooked.

(3) For the CF-based methods, DirectAU optimizes the perspective of alignment and uniformity in the hypersphere and InvCF disentangles representations of preference and popularity semantics to refine the invariant information. They are dedicated to promoting better user and item representations but neglect the underlying feedback correlations in the user-item interactive network.

(4) DiffRec utilizes generative diffusion models for personalized interactive information learning. Its suboptimal performance suggests potential explorations for integrating negative feedback into conditional diffusion models.

### 5.3 Ablation Study

We conducted an ablation experiment to examine the effectiveness of each module in NFARec. Table. 3 shows the result in NDCG@20. Overall, the sequential features contribute more than the structural features in NFARec. We have the following findings:

(1) The sequential features enhance performance across five datasets, with an improvement ranging from 3.5% to 11.1%. This shows the efficacy of the auxiliary task, which predicts the feedback sentiment polarity of the next interaction.

(2) On the Yelp 2023, Recipes, and Beauty datasets, the graph-based feature (Gra<sub>1</sub>) is effective, but its influence is not drastically on the Books dataset and harms the performance of the MovieLens dataset. This indicates that in several real-world scenarios, features derived from an interactive graph, without addressing conflicting feedback, have limited or even negative effects on performance.

(3) The feedback-aware graph-based feature benefits the recommendation quality for all datasets. It supports our assertion that the enhanced HGC operators effectively convey information along the feedback-aware paths.

**Table 3: Ablation study. “w/o X” denotes the removed parts. “Seq”, “Gra<sub>1</sub>”, and “Gra<sub>2</sub>” indicate the sequential, graph-based, and feedback-aware graph-based features, respectively. Note that on MovieLens, the results of “w/o Seq” and “w/o Gra<sub>2</sub>” are obtained without “Gra<sub>1</sub>”.**

Model	Yelp 2023	MovieLens	Recipes	Books	Beauty
w/o Seq	0.0862 (+3.7%)	0.2741 (+9.7%)	0.0460 (+7.0%)	0.1026 (+11.2%)	0.0654 (+4.4%)
w/o Gra <sub>1</sub>	0.0866 (+3.2%)	<b>0.3007</b> (-5.2%)	0.0462 (+6.5%)	0.1133 (+0.7%)	0.0648 (+5.4%)
w/o Gra <sub>2</sub>	0.0873 (+2.4%)	0.2865 (+5.0%)	0.0466 (+5.6%)	0.1094 (+4.3%)	0.0669 (+2.1%)
Full	<b>0.0894</b>	0.2851	<b>0.0492</b>	<b>0.1141</b>	<b>0.0683</b>

**Table 4: Comparison of sequential encoders. “THP” and “NHP” indicate the Transformer Hawkes process and neural Hawkes process, respectively. “Masking” refers to the masking technology in Eq. (1).**

Encoder	Yelp 2023		MovieLens		Recipes	
	R@20	N@20	R@20	N@20	R@20	N@20
Transformer	0.0960	0.0777	0.1512	0.2543	0.0555	0.0407
NHP	0.0921	0.0793	0.1478	0.2402	0.0587	0.0480
THP	<b>0.1074</b>	<b>0.0894</b>	<b>0.1736</b>	<b>0.3007</b>	<b>0.0609</b>	<b>0.0492</b>
w/o Masking	0.1053	0.0876	0.1608	0.2765	0.0574	0.0463

**Table 5: Comparison of feedback across various orders.**

	Metric	Yelp 2023	MovieLens	Recipes	Books	Beauty
<b>1-Order</b>	R@20	0.1071	0.1725	<b>0.0609</b>	<b>0.1112</b>	0.1098
	N@20	0.0892	0.2980	<b>0.0492</b>	<b>0.1141</b>	0.0678
<b>2-Order</b>	R@20	<b>0.1074</b>	<b>0.1736</b>	0.0605	0.1111	<b>0.1103</b>
	N@20	<b>0.0894</b>	<b>0.3007</b>	0.0486	0.1139	<b>0.0683</b>
<b>3-Order</b>	R@20	0.1071	0.1713	0.0602	0.1105	0.1101
	N@20	0.0893	0.2961	0.0482	0.1133	0.0681
<b>4-Order</b>	R@20	0.1068	0.1709	0.0603	0.1103	0.1083
	N@20	0.0884	0.2956	0.0481	0.1131	0.0679

### 5.4 In-Depth Analysis of NFARec

To further understand the NFARec, we conducted further experiments. Due to the space limit, we excluded contents with similar observations, i.e., comparisons of sequential encoders on Books and Beauty datasets, and sensitivity of other hyperparameters.

**Effect of Various Sequential Encoders.** We compared the encoder by NFARec with the Transformer, the neural Hawkes process, and the THP without masking. As illustrated in Table. 4, we have the following findings: (1) the standalone Transformer as the encoder cannot attain the best performance, indicating that the sequential dependencies by Hawkes stochastic process cannot be ignored; (2) a major drawback of the neural Hawkes process is that it inherits the weaknesses of RNNs to capture long-term sequential dependencies; and (3) the encoder without masking will peek into the future and results in a representation degeneration. This indicates that sequentially predicting the sentiment polarity of the next interaction contributes to NFARec for recommendations.

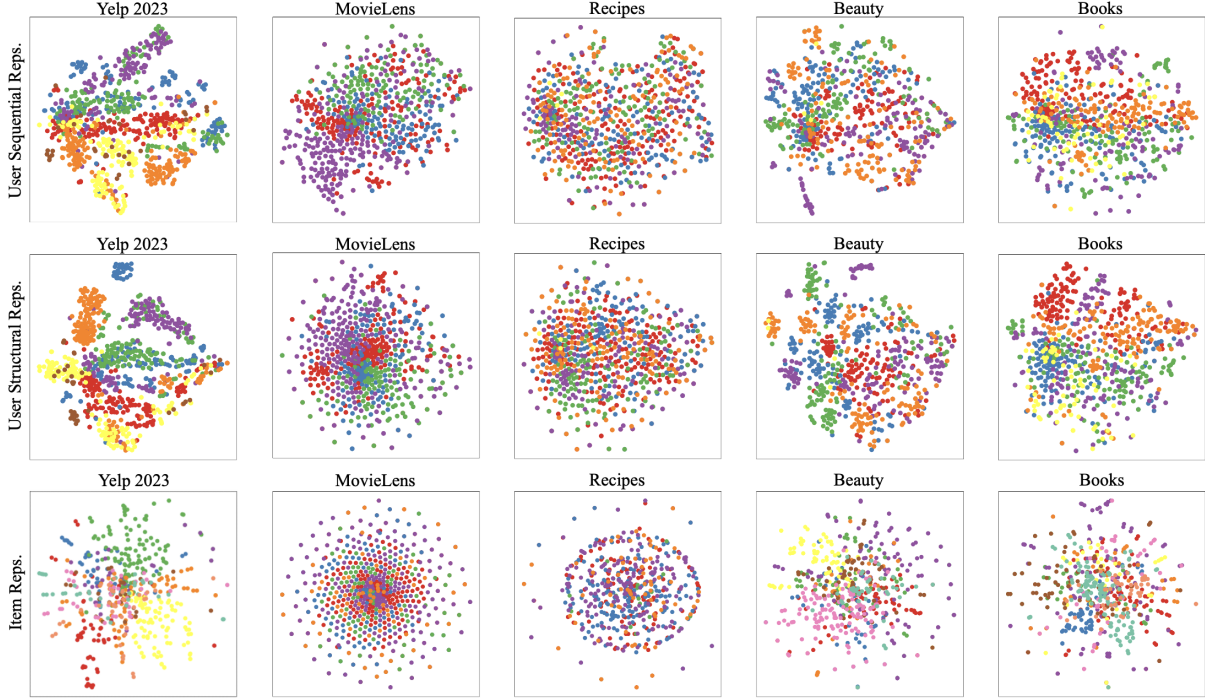


Figure 4: Visualizations of users' sequential and structure representations (Reps.) and item representations.

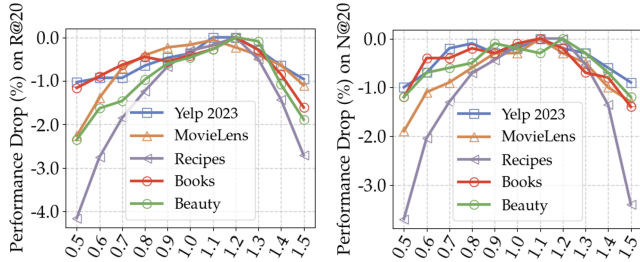


Figure 5: Effect of various  $\delta$  on performance.

**Effect of Various Orders of Feedback.** It is interesting to note how high-order feedback impacts performance. The results in Table 5 provide the following findings: (1) the optimal performance is achieved with a 1-order feedback correlation on the Recipes and Books datasets, while a 2-order correlation works better on other datasets; and (2) excessive higher-order information diminishes the lower-order dependencies, affecting the performance.

**Computational Complexity.** The computational complexity of NFARec relies on three components, i.e., sequential representing learning ( $O(|\tilde{S}|^2)$ , where  $|\tilde{S}|$  denotes the average user interaction count), structure representation learning ( $O(|I|^2)$ ), and the decoder ( $O(|I|)$ ). As  $|I|^2 \gg |\tilde{S}|^2 + |I|$ , the computational complexity of NFARec in a fixed setting is  $O(|I|^2)$ , indicating that considering the number of items is essential in a real deployment.

**Parameter Sensitivity.** Fig. 5 shows the effect of various  $\delta$  on performance decline in terms of R@20 and N@20. We can see that (1) although there exists a slight distinction between Recall@20

and NDCG@20, the optimal value of  $\delta$  is approximately 1.2 across all datasets. and (2) it is more sensitive on the Recipes dataset than others due to their different data distributions.

## 5.5 Representation Visualization

To examine the sequential and structural features, we visualize user and item representations in Fig. 4. Dots with the same color refer to users who interact with the same item or items that interact with the same user. We have the following observations:

(1) In the majority of cases, NFARec effectively identifies the similarities between user and item representations, while the quality of visualization relies heavily on the datasets. Notably, the clustering results on the Yelp 2023, Beauty, and Books datasets outperform others by a significant margin.

(2) Both sequential and structural representations can reveal which users are similar but reflect different aspects. We can see that clusters of the same color in two representations exhibit distinct shapes and distributions on five datasets.

(3) The structural features suggest more cohesion within clusters, while sequential features exhibit greater dispersion. It indicates that the former reinforces the similarity and diversity among users, whereas the latter, by learning more uniform representations, helps alleviate the issue of inactive users becoming isolated nodes.

(4) Interestingly, the dense stacking of users and items on MovieLens is due to the large number of users engaging with the same items, averaging 165.6 shown in Table 1. In contrast, the undesired visualization on the Recipes dataset results from a more challenging data distribution for recommendations. This is evident in the lower Recall and NDCG results in Table 2 compared to other datasets.



Cases	(a) UserID#fr3zIEDdv2nagOdOt1jBOW	(b) UserID#Zp1o52IR9I2dunmsfiT96g	(c) UserID#qlso3Vi6fo7zc5_Us3QCvg	(d) UserID#VOiJrfDciZzuOcK3cSPRWg
Users' Historical Interactions	$i_1$ This place <u>sucks!</u> ....., it took <u>an hour</u> to get my food ....., the employee got <u>rude</u> with me. $i_2$ Good food, <u>good</u> service, and <u>nice</u> place. ...., <u>would go back again</u> . $i_3$ Please avoid!! ....., This place is <u>expensive</u> , and it <u>sucks</u> ....., $i_4$ <u>Excellent</u> French dip and came before the time expected. Sandwich was <u>very good</u> and came with <u>everything expected</u> ....., $i_5$ ... $i_{19}$	$i_1$ Federal Doughnuts ....., <u>who doesn't love</u> some doughnuts and fried chicken? ....., $i_2$ ....., <u>expensive</u> coffee ....., the organization and distribution of drinks <u>could be better</u> ....., $i_3$ Hot diggity dog! See what I did there? I <u>LOVE LOVE LOVE</u> a small business and <u>really hope this place takes off</u> ....., $i_4$ The service is <u>exactly what I expect</u> from a fast food joint -- <u>slow and lousy</u> ....., $i_5$ ... $i_{242}$	$i_1$ ....., This place is <u>so much fun</u> & Alex the bartender was the <u>BEST!</u> ....., <u>sanitized &amp; clean</u> . $i_2$ Good Burger & ....., & we had the orings which were cooked to <u>perfection</u> . Crew was very nice & <u>not really busy</u> . $i_3$ The food & wine are absolutely <u>delicious!!!</u> The server Erin was very <u>attentive &amp; kind</u> ....., $i_4$ ....., our server Raff was very <u>attentive &amp; fun</u> . Took home some <u>great</u> wines & ....., $i_5$ ... $i_{19}$	$i_1$ Brunch-Bloody Mary <u>does not come with bacon as it says</u> on the menu :( olives instead, <u>false advertising</u> ....., $i_2$ <u>Mediocre</u> . <u>Pretty tasteless</u> ....., I hate to say it, but I <u>don't like</u> the pizza at all ....., $i_3$ It is so <u>quiet</u> , and the food is <u>great!</u> ....., <u>Great</u> lunch specials too! $i_4$ ....., It <u>used to be great</u> now it is meh. Served in an ice bowl-translation watery and <u>half the size it used to be</u> . $i_5$ <u>Best</u> pizza in Pinellas county! ....., $i_6$ ... $i_{11}$
Ground-truth Interactions	$i_{19}$ $i_{20}$ $i_{21}$ $i_{22}$ $i_{23}$	$i_{243}$ ... $i_{274}$ ... $i_{285}$ ... $i_{303}$ $i_{304}$	$i_{16}$ $i_{17}$ $i_{18}$ $i_{19}$ $i_{20}$	$i_{12}$ $i_{13}$ $i_{14}$ $i_{15}$ $i_{16}$
EEDN	x x x x x	x x x ✓ x	x x x x x	x x x x x
NFARec	x ✓ x x x	x ✓ ✓ ✓ x	x x x ✓ x	x x ✓ x x
w/o Seq	x <span style="background-color: #90EE90;">x</span> x x x	x ✓ <span style="background-color: #90EE90;">x</span> ✓ x	x x x ✓ x	x x ✓ x x
w/o Gra <sub>1</sub>	x ✓ x x x	x ✓ ✓ ✓ x	x x x ✓ x	x x ✓ x x
w/o Gra <sub>2</sub>	x ✓ x x x	x <span style="background-color: #90EE90;">x</span> ✓ ✓ x	x x x <span style="background-color: #90EE90;">x</span> x	x x <span style="background-color: #90EE90;">x</span> x x

Figure 6: Case Study. ✓ and x indicate that the model recommends correctly and incorrectly, respectively. x denotes that the removed component is the key in each case. We do not show the case study on Gra<sub>1</sub>, as it is not our main focus.

## 5.6 Case Study

We collected several apparent cases where NFARec achieves success, whereas baseline models fall short. We chose the Yelp 2023 dataset due to its explainable reviews.

**Cases for Sequential Representations.** We empirically found two major cases where the sequential representations are effective. **Case 1.** The majority of the ground-truth candidates are with negative feedback. As Case (a) shown in Fig. 6, the user has visited 18 places, and the ground-truth candidates from  $i_{19}$  to  $i_{23}$  were rated negatively by the user. In this scenario, NFARec successfully recommended item  $i_{20}$ , while EEDN incorrectly recommended high-rated items instead of the items that users will interact with.

**Case 2.** A user has a long historical visit sequence. In Case (b), the user has visited 242 places, and NFARec successfully recommended  $i_{285}$ . This is because the THP encoder, attributed to the Transformer, is crucial for capturing sequential correlations among nodes in long sequences to reflect their preferences.

**Cases for Structural Representations.** There are three typical cases to demonstrate the effectiveness of the feedback-aware structural representations.

**Case 1.** A user exclusively visits places with high scores. In Case (c), the user rates all visited and future-visited places with positive scores, indicating that the user has less curiosity about unknown things. NFARec conveys information through an optimal feedback-aware path during HGCs and reduces message-passing between conflicting nodes, which is particularly beneficial for the user who solely entertains places with positive feedback.

**Case 2.** A user actively explores new things. In Case (d), the majority of the user's experiences were with negative feedback, indicating that the user prefers to explore new places. For this user, the

feedback-aware HGC operators in NFARec improve learning correlations between negative interactions, resulting in recommendations that better reflect real-life situations, even if the recommended items may be perceived negatively by the user.

**Case 3.** A user is involved in an intricate interactive network. As depicted in Case (b) in Fig. 6, the user has visited 242 places, signifying a substantial interactive network. The NFARec correctly recommends  $i_{274}$  to the user based on structural representations. This indicates that NFARec can discriminate the user's latent preferences from the complex interactive graph.

## 6 CONCLUSION

In this paper, we proposed an NFARec that maximizes the utilization of negative feedback in sequential and structural patterns for recommendations. For sequential patterns, NFARec predicts the feedback sentiment of the next interaction based on the THP. This prompts the NFARec to identify items that the user has interacted with but may not prefer. For structural patterns, NFARec adopts a two-phase HGC approach to guide HGC operators to efficiently exchange messages along feedback-aware paths. Extensive experiments on five public datasets show that NFARec outperforms SOTA methods. In the future, we intend to (1) explore the impact of implicit negative feedback to improve performance, such as skipping content, and (2) incorporate large language models to make more interpretable recommendations.

## ACKNOWLEDGEMENTS

We would like to thank anonymous reviewers for their thorough comments. This work is supported by the JKA Foundation (No.2024M-557) and China Scholarship Council (No.202208330093).

## REFERENCES

- [1] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *The Eleventh International Conference on Learning Representations*.
- [2] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 378–387.
- [3] Lei Chen, Jie Cao, Youquan Wang, Weichao Liang, and Guixiang Zhu. 2022. Multi-view graph attention network for travel recommendation. *Expert Systems with Applications* 191 (2022), 116234.
- [4] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*. 2172–2182.
- [5] Shansan Gong and Kenny Q Zhu. 2022. Positive, negative and neutral: Modeling implicit feedback in session-based news recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1185–1195.
- [6] Bowen Hao, Jing Zhang, Hongzhi Yin, Cuiping Li, and Hong Chen. 2021. Pre-training graph neural networks for cold-start users and items representation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 265–273.
- [7] Alan G Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.
- [8] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.
- [9] Junjie Huang, Ruobing Xie, Qi Cao, Huawei Shen, Shaoliang Zhang, Feng Xia, and Xueqi Cheng. 2023. Negative can be positive: Signed graph neural networks for recommendation. *Information Processing & Management* 60, 4 (2023), 103403.
- [10] Shuyi Ji, Yifan Feng, Rongrong Ji, Xibin Zhao, Wanwan Tang, and Yue Gao. 2020. Dual channel hypergraph collaborative filtering. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020–2029.
- [11] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Multi-behavior recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 659–668.
- [12] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.
- [13] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*. 322–330.
- [14] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. In *Proceedings of the ACM Web Conference 2022*. 2320–2329.
- [15] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the web conference 2021*. 2177–2185.
- [16] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 483–491.
- [17] Hongyuan Mei and Jason M Eisner. 2017. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems* 30 (2017).
- [18] Shaowen Peng, Kazunari Sugiyama, and Tsunenori Mine. 2022. Less is more: reweighting important spectral graph features for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1273–1282.
- [19] Christian P Robert, George Casella, and George Casella. 1999. *Monte Carlo statistical methods*. Vol. 2. Springer.
- [20] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.
- [21] Xiaoxiao Sun, Boyi Huang, Xinfeng Wang, and Dongjin Yu. 2024. MARAN: Supporting awareness of users' routines and preferences for next POI recommendation based on spatial aggregation. *Expert Systems with Applications* 238 (2024), 121961.
- [22] Wanjie Tao, Yu Li, Liangyue Li, Zulong Chen, Hong Wen, Peilin Chen, Tingting Liang, and Quan Lu. 2022. SMNet: State-Aware Multi-Aspect Interests Representation Network for Cold-Start Users Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8476–8484.
- [23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [24] Chenyang Wang, Weizhi Ma, Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. Sequential recommendation with multiple contrast signals. *ACM Transactions on Information Systems* 41, 1 (2023), 1–27.
- [25] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards Representation Alignment and Uniformity in Collaborative Filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1816–1825.
- [26] Dongjing Wang, Xin Zhang, Zhengzhe Xiang, Dongjin Yu, Guandong Xu, and Shuiguang Deng. 2021. Sequential recommendation based on multivariate Hawkes process embedding with attention. *IEEE transactions on cybernetics* 52, 11 (2021), 11893–11905.
- [27] Wei Wang and Longbing Cao. 2021. Interactive sequential basket recommendation by learning basket couplings and positive/negative feedback. *ACM Transactions on Information Systems (TOIS)* 39, 3 (2021), 1–26.
- [28] Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. 2023. Diffusion Recommender Model. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 832–841.
- [29] Xinfeng Wang, Fumiyo Fukumoto, Jin Cui, Yoshimi Suzuki, Jiyi Li, and Dongjin Yu. 2023. EEDN: Enhanced Encoder-Decoder Network with Local and Global Interactions for POI Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 383–392.
- [30] Xinfeng Wang, Fumiyo Fukumoto, Jiyi Li, Dongjin Yu, and Xiaoxiao Sun. 2023. STaTRL: Spatial-temporal and text representation learning for POI recommendation. *Applied Intelligence* 53, 7 (2023), 8286–8301.
- [31] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 950–958.
- [32] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhengguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the Web Conference 2021*. 878–887.
- [33] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2020. Neural news recommendation with negative feedback. *CCF Transactions on Pervasive Computing and Interaction* 2 (2020), 178–188.
- [34] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.
- [35] Qiong Wu, Yong Liu, Chunyan Miao, Binqiang Zhao, Yin Zhao, and Lu Guan. 2019. PD-GAN: Adversarial learning for personalized diversity-promoting recommendation. In *IJCAI*, Vol. 19. 3870–3876.
- [36] Lianghao Xia, Chao Huang, Chunzhen Huang, Kangyi Lin, Tao Yu, and Ben Kao. 2023. Automated Self-Supervised Learning for Recommendation. In *Proceedings of the ACM Web Conference 2023*. 992–1002.
- [37] Lianghao Xia, Chao Huang, Yong Xu, Jiahu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*. 70–79.
- [38] Lianghao Xia, Chao Huang, and Chuxu Zhang. 2022. Self-supervised hypergraph transformer for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2100–2109.
- [39] Xu Xie, Fei Sun, Zhaoqiang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.
- [40] Xiaodong Yan, Tengwei Song, Yifeng Jiao, Jianshan He, Jiaotuan Wang, Ruopeng Li, and Wei Chu. 2023. Spatio-temporal hypergraph learning for next POI recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 403–412.
- [41] Song Yang, Jiamou Liu, and Kaiqi Zhao. 2022. GETNext: trajectory flow map enhanced transformer for next POI recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on research and development in information retrieval*. 1144–1153.
- [42] Nan Yin, Fuli Feng, Zhigang Luo, Xiang Zhang, Wenjie Wang, Xiao Luo, Chong Chen, and Xian-Sheng Hua. 2022. Dynamic hypergraph convolutional network. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1621–1634.
- [43] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. 2023. XSimGCL: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2023).

- [44] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-aware self-supervised tri-training for recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2084–2092.
- [45] Junliang Yu, Hongzhi Yin, Jundong Li, Min Gao, Zi Huang, and Lizhen Cui. 2020. Enhancing social recommendation with adversarial graph convolutional networks. *IEEE Transactions on knowledge and data engineering* 34, 8 (2020), 3727–3739.
- [46] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *Proceedings of the Web Conference 2021*. 413–424.
- [47] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1294–1303.
- [48] Wenhui Yu and Zheng Qin. 2019. Spectrum-enhanced pairwise learning to rank. In *The world wide web conference*. 2247–2257.
- [49] Wenhui Yu and Zheng Qin. 2020. Graph convolutional network for recommendation with low-pass collaborative filters. In *International Conference on Machine Learning*. PMLR, 10936–10945.
- [50] Wenhui Yu, Zixin Zhang, and Zheng Qin. 2022. Low-Pass Graph Convolutional Network for Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8954–8961.
- [51] An Zhang, Jingnan Zheng, Xiang Wang, Yancheng Yuan, and Tat-Seng Chua. 2023. Invariant Collaborative Filtering to Popularity Distribution Shift. In *Proceedings of the ACM Web Conference 2023*. 1240–1251.
- [52] Mengqi Zhang, Shu Wu, Xueli Yu, Qiang Liu, and Liang Wang. 2022. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 5 (2022), 4741–4753.
- [53] Sen Zhao, Wei Wei, Ding Zou, and Xianling Mao. 2022. Multi-view intent disentangle graph networks for bundle recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4379–4387.
- [54] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1040–1048.
- [55] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. 2020. Transformer Hawkes process. In *International conference on machine learning*. PMLR, 11692–11702.