Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 10(10): 23067–23083.
DOI: 10.3934/math.20251024
Received: 17 June 2025
Revised: 22 September 2025
Accepted: 23 September 2025
Published: 11 October 2025

Research article

Multi-factor disentangled graph neural networks for session-based new item recommendation

Xinning Li[1,2,3], **Qian Gao**[1,2,3,*], **Jun Fan**[4] **and Lujie Feng**[1,2,3]

[1] Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, Shandong, China

[2] Shandong Engineering Research Center of Big Data Applied Technology, Faculty of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, Shandong, China

[3] Shandong Provincial Key Laboratory of Industrial Network and Information System Security, Shandong Fundamental Research Center for Computer Science, Jinan 250014, Shandong, China

[4] China Telecom Digital Intelligence Technology Co, Ltd, No.1999, Shunhua road, Jinan 250101, Shandong, China

* **Correspondence:** Email: gq@qlu.edu.cn.

Abstract: Recent studies have shown that graph neural networks for session-based recommendation systems typically recommend old items, making it difficult to recommend new items to users, leading to the phenomenon of the "information cocoon". To address this issue, this paper introduces a Multi-Factor Disentangled Graph Neural Network for Session-Based New Item Recommendation (MFD-GNN), which considers both the embedding of new items and user intent from a multi-factor perspective. First, item embeddings from sessions are generated across multiple factors using a disentangled network. By leveraging item classification and attribute information, new item embeddings are inferred through zero-shot learning. Attention weights are assigned to each factor to capture user intent across different factors, enabling reasonable recommendations for new items. Experiments are conducted on two publicly available datasets, and the results are compared with those of leading recommendation models. The findings demonstrate that the proposed method surpasses current models in performance. These experimental outcomes confirm the approach's effectiveness and its advantages over existing methods.

Keywords: session-based recommendation; graph neural network; disentangled representation learning; new item recommendation
Mathematics Subject Classification: 68T07, 68T20

1. Introduction

Recommendation systems are key tools for managing information overload and personalizing user experiences [1]. They are extensively used in areas like e-commerce, streaming services, and social media [2], providing personalized suggestions by analyzing user interests and behaviors [3]. Session-based recommendation (SBR) is a common task in recommendation systems [4]. SBR aims to offer personalized recommendations by analyzing the real-time behavior sequences of users during their interactions with the system. It is a technique that dynamically generates recommendation lists based on the user's current session behavior [5]. Initially, session-based recommender systems (SBR) relied on Markov chain models [6] and frequent sequential patterns [7] for predictions. Although these methods were effective, they had limitations, including high computational costs and challenges in capturing long-term dependencies [8]. With the rise of deep learning, recurrent neural networks (RNNs) began to be applied to SBR. The GRU4Rec [9] model, for instance, used gated recurrent units to capture the sequential features of user interaction data. However, RNNs face limitations when handling short-term user interactions, thus hindering the accurate representation of user preferences [10].

The use of Graph Neural Networks (GNN) in session-based recommendation systems has attracted growing interest due to their ability to capture the intricate transition relationships between items in a session [11]. By modeling sessions as graph structures, GNN methods represent items as nodes and transitions between them as edges, allowing for a more effective understanding of user behavior patterns and preferences. Currently, models such as SR-GNN [12], GCE-GNN [4], and GCSAN [13] have constructed historical session graphs and utilized GNNs to learn user intent and item representations, thereby predicting the probability of user preference for items [14]. These methods solve the problem of traditional recommender systems failing to effectively model the complex relationships within sessions, significantly improving recommendation performance [15].

However, Graph Neural Networks (GNNs) learn item and user intent representations based on historical session graphs to compute user-item matching scores. As a result, this approach may result in a higher probability of old items, which have already appeared in previous sessions, being recommended in the next interaction, while items that have not been encountered in historical sessions (new items) are less likely to be suggested. As shown in Figure 1, in the session shown in (a), old items v_1, v_2, v_3, and v_4 receive higher recommendation scores, while the new items v_5 and v_6 receive a lower score. This occurs because new items do not have prior interaction data, which makes it challenging for the neural network to capture their relevant features accurately. As a result, recommending new items to users becomes challenging, potentially leading to reduced user engagement and the emergence of the "information cocoon". To address this issue, NirGNN proposed a new challenge called 'GNN-based Session-based New Item Recommendation (GSNIR)' [16], and introduced a new session-based method for new item recommendation. NirGNN utilizes a dual-intent network that learns the user's decision-making process using a soft attention mechanism and a β-distribution mechanism. It also uses zero-shot learning to infer the new item embeddings based on item attributes. However, the overall item embedding method adopted by NirGNN has a fundamental defect of "semantic entanglement". The features of different aspects are compressed into an uninterpretable vector, which not only leads to poor interpretability, but more importantly, the preferences of different aspects will interfere with and cancel each other when calculating the user-item matching degree. Secondly, NirGNN only relies on attributes for inference, which faces the risk of "embedding collision". In real-world scenarios, different items may share the same attribute values, and the inference mechanism of NirGNN will map them to the same embedding vector, resulting in the model being unable to distinguish them, seriously damaging the recommendation accuracy.

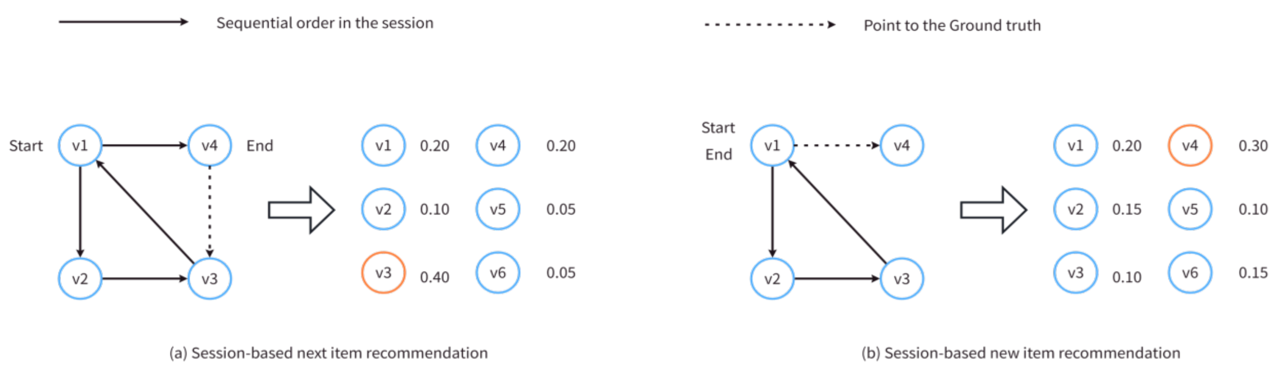

Figure 1. In (a), the session is $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_1 \rightarrow v_4$, and the new items are v_5, v_6; in (b), the session is $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_1$, and the new items are v_4, v_5, v_6.

To address the issues mentioned above, we propose a multi-factor disentangled network model. To solve the first problem, we consider that users may have different levels of attention to various factors of an item. Thus, we use a decoupling network to split the item embedding into several chunks, where each chunk represents the embedding of the item under a different factor. Additionally, we employ a distance function to ensure the independence between factors. We also assign attention weights to each factor and use aggregation operations to capture the user's intent across different factors. For the second problem, we utilize the classification and attribute information of items to infer new item embeddings based on zero-shot learning theory. This approach increases item diversity and helps reduce the situation where different candidate items (new items) have the same embeddings. Meanwhile, we apply a factor-wise zero-shot learning method to learn embeddings of candidate items under different factors, which together form a complete representation of the new item. By leveraging the user intent learned across multiple factors and the new item embeddings, we can recommend new items. The key contributions of this paper can be summarized as follows:

- This paper employs a decoupled network to consider item embeddings at a finer granularity. Each item is represented by embeddings across multiple factors, and user attention to these factors is captured through aggregation operations. This allows us to generate factor-specific embeddings for new items, facilitating more accurate new item recommendations.

- This paper uses item classification and attribute information as auxiliary data and applies zero-shot learning approaches to derive the embeddings of new items. This approach reduces embedding redundancy for new items and enhances the diversity of their embeddings.

- This paper performs comprehensive experiments on two conversational new item recommendation datasets and evaluates our model against state-of-the-art methods. The results highlight the advantages of our model.

2. Related work

2.1. Recommender systems

Session-based recommender systems focus on providing personalized recommendations within a single user interaction [3]. Session-based recommendation predicts the next item by considering the sequence of user interactions, combining Markov chain and neural network models, such as GRU4Rec [9], to improve recommendation accuracy. In addition, graph neural networks have also been applied to SBR, such as SR-GNN [12] and GCE-GNN [4], to capture complex inter-item relationships. In order to enhance the interpretability of the final recommendation, KCLM [17] innovatively combines the session recommendation with the keyword generation task. The model aligns the recommendation results with the generated keywords through comparative learning, thereby improving the accuracy of the recommendation and the interpretability of the decision-making process. Furthermore, KAMVG [18] takes advantage of richer attributes and keyword side information and constructs a multi-view graph neural network to simulate the session-item relationship so as to learn multiple user intentions. These studies strongly prove the effectiveness of introducing side information (such as keywords and attributes) and multi-level user intents for understanding complex user intents and improving recommendation performance. Inspired by the ideas of "multi-intention learning" and "using auxiliary information", we propose to use multi-factor decoupling to learn multi-user intentions in new item recommendation. At the same time, we use auxiliary information (hierarchical classification and attribute) as a bridge to learn multi-factor embedding representations of new items in zero-shot learning, aiming to achieve more accurate new item recommendations.

2.2. Disentangled representation learning

Disentangled representation learning is a machine learning method with the goal of separating potentially independent factors from complex data to obtain a clearer and more interpretable representation [19]. This method has received extensive attention in fields such as image and text representation learning [20] and has shown its unique value in the field of recommendation systems, especially in simulating users' diverse recommendation intentions [21]. DGNN [22] has studied the diversity of users' intentions on items, and proposed a model based on a graph convolutional network. The distance correlation method is used to enhance the independence of different intentions in embedding learning so as to more effectively capture the different motives and preferences of users when adopting items. DMICF [23] proposed a dual-view decoupling multi-intent recommendation model, which integrates structural context from the perspective of users and items. Fine-grained intent alignment is enhanced by a dual-view fusion design and an intention-aware scoring mechanism. DMICF focuses on multi-intention mining in general recommendation scenarios, while in the new item recommendation system, we innovatively introduce the disentangled representation learning technique. On this basis, the factor-level zero-shot reasoning is innovatively implemented, which specifically solves the cold start problem of new items and opens up a new technical path.

2.3. Zero-shot learning

Zero-shot learning is a technique that enables machine learning models to recognize and process categories unseen in the training phase and to transfer cross-category information through attribute or semantic description [24]. Zero-shot learning has been widely used in computer vision, natural language processing, and graph neural networks and can simulate human recognition ability for new things [25]. NirGNN [16] first introduced zero-shot learning into session recommendation to recommend new items to users and obtained the inferred item embedding by using the attribute information of the items. However, there may be two different candidate items with the same attributes when inferencing only based on the attribute information. In this case, the inferred item embeddings may be the same, which may affect the recommendation results. Discovering and utilizing diverse data is the core of auxiliary information-driven session recommendation [26]. Therefore, we propose to integrate the attribute information and category information of items to jointly shape the embedding vector of new items. This comprehensive embedding strategy allows us to capture the multi-dimensional features of items more comprehensively. At the same time, the high-level information provided by the classification information is used to enhance the discrimination of item features.

3. Materials and methods

The user's anonymous historical session $S = [v_1, v_2, ..., v_n]$ consists of items (old items) that the user has interacted with previously. Historical sessions are modeled as a directed session graph $G = (V, E)$, where $(v_i, v_j) \in E$ indicates that the user chose to interact with item v_j after interacting with item v_i . Suppose there is a set of new items, denoted as $C = [c_1, c_2, ..., c_m]$. The goal of GSNIR is to design a GNN-based model to recommend new items to users in a reasonable way. Specifically, the GNN-based new item recommendation model captures user preferences via the session graph. As the GNN cannot directly learn embeddings for new items, these models often leverage auxiliary information to generate simulated embeddings for such items, recommending the top-k highest-scoring new items from the candidate set to users. Each item has its corresponding classification and attribute information. For instance, in a food recommendation system, cola could be classified into three levels: 'Beverages', 'Bottled Beverages', and 'Carbonated Beverages', while the attribute information for cola could include brand and price. For an item v_i, we use a_i and t_i to denote its attribute information and classification information, respectively. For a new item c_i, we use $a_{c,i}$ and $t_{c,i}$ to denote its attribute and classification information.

3.1. Our model

The model we proposed is shown in Figure 2. In part (a), the session items are passed through a disentangled network to obtain the embeddings of the items under multiple factors. This study uses multiple GGNN networks to separately learn the representations of the items under each factor and employs a distance function to ensure the independence between factors. In part (b), this study considers the embedding of new items from a multi-factor perspective while utilizing both classification and attribute information to jointly infer the embedding of the new item, which can increase the diversity of new item embeddings to some extent. Part (c) focuses on user intent learning and prediction. The study assigns attention weights to each factor of the items, and uses an aggregation operation to capture the user's preferences across various factors, then combines it with the embeddings of the new item under various factors to compute the recommendation score.

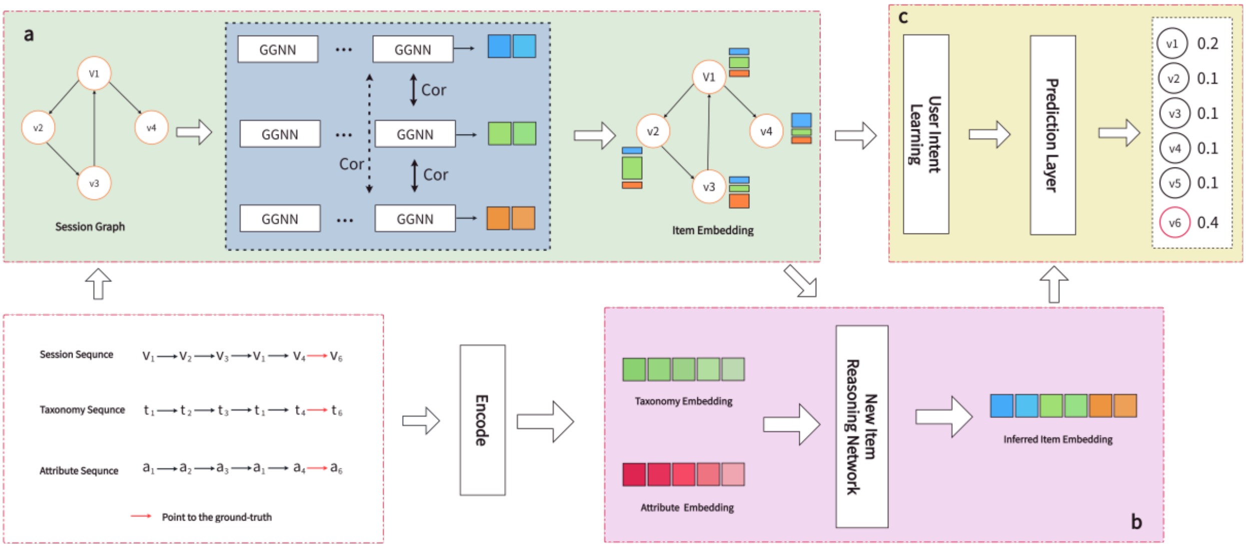

Figure 2. The overall framework of the proposed MFD-GNN.

3.2. *Initialization*

Due to the diversity of user intent when clicking on an item, which is influenced by various factors of the target item, we divide the embedding of each item into K chunks. Each chunk represents the embedding of the item under a specific factor, while ensuring the independence of these factors from each other. Formally, for a session $S = [v_1, v_2, ..., v_n]$, the embedding of item v_i is represented as $e_i \in R^d$, and the entire session is represented as $E_s = [e_1, e_2, ..., e_n]$. The item embedding e_i is composed of K factor blocks, denoted as $e_i = [h_{i,1}, h_{i,2}, ..., h_{i,K}] \in R^{\frac{d}{K}}$, where the embedding calculation process for each factor block is given by [27]:

$$h_{i,k} = \frac{\sigma(W_k^T \cdot e_i) + b_k}{\|\sigma(W_k^T \cdot e_i) + b_k\|_2}, \tag{3.1}$$

$\mathrm{W_k} \in R^{d \times \frac{d}{K}}$ and $b_k \in R^{\frac{d}{K}}$ are the parameters of the k-th factor block. σ denotes a nonlinear activation function. L2 regularization is used to avoid overfitting. The initial session embedding is denoted as $E_s^{(0)} = \left[e_1^{(0)}, e_2^{(0)}, \ldots, e_n^{(0)}\right]$, where $e_i^{(0)} = \left[h_{i,1}^{(0)}, h_{i,2}^{(0)}, \ldots, h_{i,K}^{(0)}\right] \in R^{\frac{d}{K}}$. To measure the degree of association between different items on a specific factor, we establish a similarity matrix based on factors. Specifically, for session S, the k-th factor embedding of each item is initialized as $h_{s,k} = \left[h_{1,k}^{(0)}, h_{2,k}^{(0)}, \ldots, h_{n,k}^{(0)}\right]$, and the similarity between two adjacent items v_i and v_j is calculated as:

$$w_{i,j}^k = h_{i,k}^T \times h_{j,k}, \tag{3.2}$$

where $h_{i,k}, h_{j,k} \in R^{\frac{d}{K}}$ represents the embeddings of the i-th and j-th interacted items in the session on the k-th factor, respectively. $w_{i,j}^k$ represents the similarity score between items v_i and v_j on factor k. We treat the similarity between items as the weight of the edge, and then normalize the similarity scores for the outgoing and incoming edges of each item node as follows:

$$\hat{w}_{i,j}^{k} = \frac{w_{i,j}^{k}}{\sum_{j' \in N_i^o}^{N_i^o} w_{i,j'}^{k}}, \tag{3.3}$$

where N_i^o is the set of outgoing neighbors of item v_i. The in-degree matrix $A_{s,k}^{in}$ and out-degree matrix $A_{s,k}^{out}$ are constructed by using the normalized similarity scores between the item and its neighboring items.

3.3. Multi-factor disentangled network

In the previous sections, we divided the item embeddings into K components and computed the factor-based similarity matrix. Unlike traditional methods, this study learns item embeddings based on the relationships of items across different factors. The factor-based similarity matrix is input into a disentangled network, where a multi-layer gated graph neural network (GGNN) is used to learn the embedding representation of each item under different factors. Additionally, this study employs a distance-based correlation function to ensure the learned factors maintain independence from each other. The model uses T layers of GGNN and considers K factors. We use t $(1 \le t \le T)$ to represent the t-th layer of GGNN and k $(1 \le k \le K)$ to represent the k-th factor. In the following equation, we take the propagation of factor embeddings in the t-th layer of GGNN as an example [21]:

$$\begin{aligned} a_{i,k}^{t} = \text{Concat}\Big(& H_{in} h_{s,k}^{t-1} \left(A_{s,k,i:}^{in}\right)^{T} + b_{in}, \\ & H_{out} h_{s,k}^{t-1} \left(A_{s,k,i:}^{out}\right)^{T} + b_{out}\Big), \end{aligned} \tag{3.4}$$

$$z_{i,k}^{t} = \sigma \left(W_z a_{i,k}^{t} + U_z h_{i,k}^{t-1}\right), \tag{3.5}$$

$$r_{i,k}^{t} = \sigma \left(W_r a_{i,k}^{t} + U_r h_{i,k}^{t-1}\right), \tag{3.6}$$

$$\widetilde{h_{i,k}^{t}} = \tanh\left(W_o a_{i,k}^{t} + U_o \left(r_{i,k}^{t} \odot h_{i,k}^{t-1}\right)\right), \tag{3.7}$$

$$h_{i,k}^{t} = \left(1 - z_{i,k}^{t}\right) \odot h_{i,k}^{t-1} + z_{i,k}^{t} \odot \widetilde{h_{i,k}^{t}}, \tag{3.8}$$

Here, $A_{s,k,i}^{in}$ and $A_{s,k,i}^{out}$ are the i-th row of the similarity matrices $A_{s,k}^{in}$ and $A_{s,k}^{out}$, respectively. $H_{in}, H_{out} \in R^{\frac{d}{K} \times \frac{d}{K}}$ represents the weight matrix to be learned. $h_{s,k}^{t-1} = \left[h_{1,k}^{t-1}, h_{2,k}^{t-1}, ..., h_{n,k}^{t-1}\right]$ denotes the output of the $(t-1)$-th layer of the GGNN, which represents the k-th factor embedding of all items in the session. $Z_{i,k}^{t}$ and $r_{i,k}^{t}$ are the update gate and reset gate; σ is the sigmoid activation function; $\odot$ denotes element-wise multiplication. After passing through all GGNN layers, the embedding of item v_i is obtained as $e_i^T = \left[h_{i,1}^T, h_{i,2}^T, ..., h_{i,K}^T\right]$. For ease of representation, we denote $e_i = [h_{i,1}, h_{i,2}, ..., h_{i,K}]$ as the final learned item embedding. To ensure the independence among latent factors, the model employs a distance correlation function as a regularization term:

$$\mathcal{L}_{dec1} = \sum_{k=1}^{K} \sum_{k'=k+1}^{K} dCor\left(h_{s,k}, h_{s,k'}\right), \tag{3.9}$$

here, $H_{s,k} = [h_{1,k}, h_{2,k}, ..., h_{n,k}]$ represents the representation of each item in session S under the k-th factor. The distance correlation function is denoted as dCor(·), which is expressed as:

$$dCor\left(h_{s,k}, h_{s,k'}\right) = \frac{dCov\left(h_{s,k}, h_{s,k'}\right)}{\sqrt{dVar\left(h_{s,k}\right) \cdot dVar\left(h_{s,k'}\right)}}, \tag{3.10}$$

In the formula, dCov(·) denotes the distance covariance between two matrices, and dVar(·) denotes the distance covariance of the matrix itself.

3.4. User intent learning

In previous approaches, item embeddings were represented as a single holistic embedding. In contrast, this study represents items through embeddings of K distinct factors, thus requiring the learning of user intents for each factor separately. To achieve this objective, the proposed method assigns attention weights to each factor of the items and employs an aggregation operation to capture user intents across different factors. Additionally, since the last item interacted with by the user can reflect their local intent, this work combines the local intent with the global intent inferred from the entire session to derive the user's comprehensive intent. We can see the detailed network structure in Figure 3.

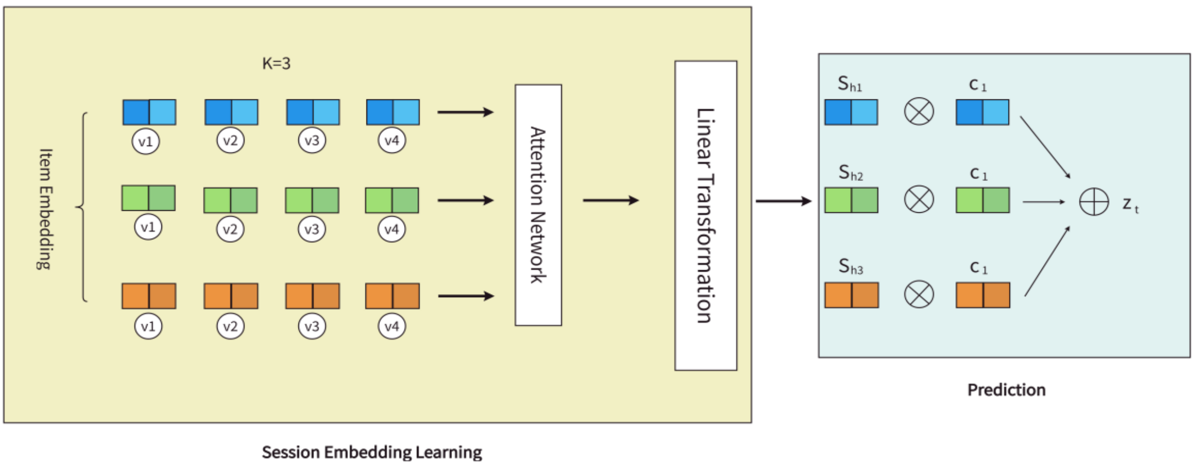

Figure 3. User intent learning network.

The specific operations are as follows: Let the final learned embedding of item v_i be $e_i = [h_{i,1}, h_{i,2}, ..., h_{i,K}]$. For a session $S = [v_1, v_2, ..., v_n]$, the embedding of the last item e_n reflects the local intent I_l, i.e., $I_l = e_n$. The global intent is obtained by aggregating all items in the session. Let the global intent be $I_g = \left[I_{g,1}, I_{g,2}, ...I_{g,K}\right]$, where $I_{g,k}$ denotes the user's global intent on the k-th factor. The computation method is defined as:

$$\mathbf{I}_{g,k} = \sum_{i=1}^{n} \alpha_{i,k}\mathbf{h}_{i,k}, \tag{3.11}$$

in the formula, $a_{i,k}$ represents the attention weight of item v_i on the k-th factor. The calculation method is as follows:

$$\alpha_{i,k} = \mathbf{q}^{\top}\sigma\left(\mathbf{W}_1\mathbf{h}_{n,k} + \mathbf{W}_2\mathbf{h}_{i,k} + \mathbf{b}\right), \tag{3.12}$$

$W_1, W_2 \in R^{\frac{d}{K}\times\frac{d}{K}}$, are the attention parameters that need to be learned by the neural network. The local intent and global intent are linearly transformed to obtain the final user intent representation. The user's intent across the k-th factors is calculated as follows:

$$\mathbf{I}_{h,k} = \mathbf{W}_3\left[\mathbf{I}_{l,k}; \mathbf{I}_{g,k}\right], \tag{3.13}$$

$W_3 \in R^{\frac{d}{K}\times\frac{2d}{K}}$ is a transformation matrix, and $I_h = [I_{h,1}, I_{h,2}, ...I_{h,K}]$ represents the user's final intent representation.

3.5. New item reasoning network

As new items are absent from historical sessions, embeddings derived only from encoding their IDs do not effectively capture their true representation. Based on zero-shot learning theory, we infer the embeddings of new items using their category and attribute information. Given a new item c_i with attribute information $a_{c,i}$ and category information $t_{c,i}$, we train a spatial transformation function θ to project these auxiliary features into the item embedding space. Specifically, leveraging the learned embeddings of existing items $e_i = [h_{i,1}, h_{i,2}, ..., h_{i,K}]$, along with their attribute information a_i and category information t_i, the auxiliary information of an item is represented as p_i:

$$p_i = W_4\left[a_i; t_i\right], \tag{3.14}$$

here, $W_4 \in R^{d\times 2d}$. Using the auxiliary information p_i of existing item v_i, we derive the inferred embeddings $e_i^* = \left[h_{i,1}^*, h_{i,2}^*, ..., h_{i,K}^*\right]$ through K distinct spatial transformation functions θ. Taking the k-th factor as an example, the inferred embedding $h_{i,k}^*$ is computed as $h_{i,k}^* = \theta_k\left(p_i\right)$. To ensure the inferred embeddings align closely with the true embeddings, we minimize the distance between them via a loss function:

$$\mathcal{L}_{dec2}\left(e_i, e_i^*\right) = MSE\left(e_i, e_i^*\right), \tag{3.15}$$

where MSE(·) denotes the mean squared error. Based on the auxiliary information of items, the transformation function is used to infer the embeddings of new items. Taking the new item c_i as an example, its embedding is $c_i = [c_{i,1}, c_{i,2}, ..., c_{i,K}]$, where $c_{i,k} = \theta_k\left(p_{c,i}\right)$ represents the embedding of the new item under the k-th factor, and $p_{c,i}$ denotes the auxiliary information of the new item c_i.

3.6. Prediction

After obtaining the user intent and the embeddings of new items, we calculate the recommendation score for each new item as follows:

$$\hat{\mathbf{z}}_{\mathbf{i}} = \mathrm{softmax}(\mathrm{I_h}^{\top} \cdot \mathbf{c_i}) = \mathrm{softmax}(\sum_{k=1}^{K} \mathbf{I_{h,k}}^{\top} \cdot \mathbf{c_{i,k}}), \tag{3.16}$$

where I_h and c_i represent the user intent embedding and the embedding of the new item, respectively. We aggregate similarities between the user's intent on each factor and the new item's embedding on the corresponding factor. The difference between the predicted scores for new items and the actual interactions is quantified using the Cross-Entropy loss function. (L_{ce}). The final total loss combines L_{ce}, L_{dce1} and L_{dce2}:

$$L = L_{ce}\left(Z, \widehat{Z}\right) + \lambda_1 L_{\mathrm{dec1}} + \lambda_2 \left(\sum_{i=1}^{n} L_{\mathrm{dec2}}\left(v_i, \theta(p_i)\right) \right), \tag{3.17}$$

Here, Z denotes the true labels, and $\widehat{Z}$ represents the predicted recommendation scores for all new items. L_{dce1} is the disentangled learning regularization loss, while L_{dce2} is used to measure the learning ability of the new item embedding network. λ_1 and λ_2 are used to balance the ratio between L_{dce1} and L_{dce2} . Since L_{ce}(CrossEntropy loss) serves as the primary recommendation learning loss, this study does not assign an additional weight to it.

4. Results

In this experiment, the MFD-GNN model was implemented using PyTorch. All experiments were performed on an NVIDIA RTX 1650 GPU. Through grid search key super parameters in the model, in particular, we explore within set $[32, 64, 128, 256]$ the best embedding dimension, in a set $[0.1, 0.05, 0.01, 0.005, 0.001]$ in vector to find the best, For GGNN levels we look for the best case in the set $[1, 2, 3, 4, 5]$. The final setting is as follows: embedding dimension d=128; Training bath size=100; Adam optimizer was used with an initial learning rate of 0.005 and a learning rate decay strategy. The number of GGNN layers is three.

4.1. Datasets and metrics

We utilized two datasets provided by prior research: Amazon G&GF and Cell Phones and Accessories (CellPhones). Both datasets are subsets of the Amazon dataset and include user-item interaction sessions, attribute information of items, and hierarchical category information. The detailed information of the dataset is shown in Table 1.

We adopt MRR (Mean Reciprocal Rank) and precision, two metrics widely used in session-based recommendation. Specifically, P@10 and P@20 measure the accuracy of new item recommendations, while MRR is a performance evaluation metric commonly used for recommendation and information retrieval systems. It computes the average of the reciprocal ranks of the correct items over a set of queries.

Table 1. Dataset statistics.

Dataset	Amazon G&GF	CellPhones
items	18,889	13,672
taxonomy	1,350	40
attributes	7,077	2,948
sessions	89,771	123,602
Avg length	8,394	3,677

4.2. Evaluation results

4.2.1. Comparison of models

SR-GNN [12] constructs a graph of item transitions within sessions and uses a graph neural network (GNN) to capture dynamic user interests through node embeddings. However, it struggles to recommend new items that lack interaction history.

GCE-GNN [4] builds on this by enhancing the architecture with graph-level contrastive learning, improving recommendation accuracy and robustness. Yet, it still faces difficulties in recommending unseen items, as it heavily relies on historical interaction data.

COTREC [28] employs a transformer-based approach with self-attention mechanisms to capture long-range dependencies and session contexts, introducing context-overlapping techniques for better recommendation quality. Despite these improvements, it still faces challenges in recommending new items, with low recommendation probabilities, much like the issues seen in SR-GNN and GCE-GNN.

LS-TGNN [29] adopts a novel temporal session graph to aggregate neighborhood information, introduces an item-granularity method to distinguish long-term and short-term interests, and models the long-term and short-term interests of users through long-term and short-term encoders.

NirGNN [16] is the first GNN-based model specifically designed for new item recommendation. It proposes a dual-intent learning framework, introducing a category tree for each item and determining user intent directions via attention mechanisms and temporal distributions over category data. NirGNN pioneers zero-shot learning in session-based recommendation, inferring embeddings for unseen items using attribute information to enable new item recommendations. However, NirGNN's reliance solely on attributes may fail when distinct items share identical attributes, and its simplistic embedding generation approach limits recommendation accuracy.

TSGNN [30], on the basis of NirGNN, TSGNN further considers the time sensitivity, proposes a time sensitivity graph enhanced network, fully expresses the time relationship between items, and analyzes the time sensitivity weight coefficient, which is a further improvement of NirGNN.

This experiment compares MFD-GNN with five advanced session-based GNN recommendation methods. Table 2 shows the detailed experimental results. It is evident that MFD-GNN achieves significant improvements compared to the baseline. Compared to the SOTA baseline TSGNN, our method shows improvements in all metrics on the Amazon G&GF dataset, with a 6.36% increase in P@20, a 6.21% improvement in P@10, and a 6.89% and 7.89% increase in MRR@20 and MRR@10, respectively. In addition, on the CellPhones dataset, MFD-GNN outperforms NirGNN by 4.07% in P@20, 8.55% in P@10, 4.83% in MRR@20, and 3.50% in MRR@10, fully validating the superiority of MFD-GNN.

Table 2. Experiments on CellPhones and Amazon G&GF. "*" indicates the statistically significant improvements (i.e., two-sided t-test with $p < 0.05$) over the best baseline.

Methods	**Amazon G&GF**				**CellPhones**			
	P@20	**P@10**	**MRR@20**	**MRR@10**	**P@20**	**P@10**	**MRR@20**	**MRR@10**
SR-GNN	1.438	0.086	0.38	0.35	1.89	1.22	0.54	0.44
GCE-GNN	1.85	1.25	0.58	0.51	1.83	1.16	0.57	0.52
COTREC	2.41	1.078	0.65	0.61	2.35	1.23	0.58	0.53
LS-TGNN	2.48	1.72	0.79	0.68	2.62	1.40	0.60	0.55
NirGNN	2.44	1.75	0.85	0.74	2.63	1.43	0.61	0.56
TSGNN	2.67	1.93	0.87	0.76	2.70	1.52	0.62	0.57
MFD-GNN(ours)	**2.84***	**2.05***	**0.93***	**0.82***	**2.81***	**1.65***	**0.65***	**0.59***
Improve(%)	**6.36%**	**6.21%**	**6.89%**	**7.89%**	**4.07%**	**8.55%**	**4.83%**	**3.50%**

4.2.2. Ablation study

We performed ablation experiments to evaluate the contribution and effectiveness of each component in our proposed model. Based on the original MFD-GNN model, we built three variants: MFD-GNN-io, MFD-GNN-attr, and MFD-GNN-tax. Figure 4 shows the experimental results of MFD-GNN and its three variants on Amazon G&GF and CellPhones. It is clearly evident that, across four metrics, the prediction accuracy of all three variants is lower than that of the original MFD-GNN model.

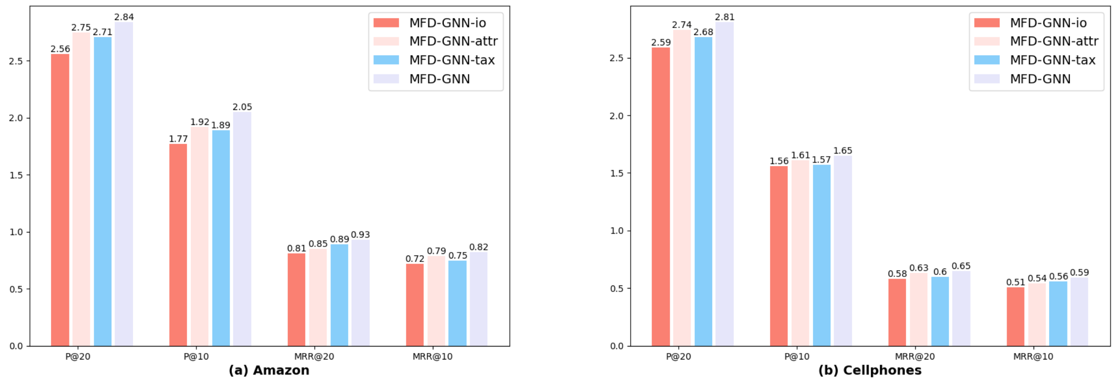

Figure 4. Ablation study on Amazon G&GF and CellPhones.

Specifically, the MFD-GNN-attr and MFD-GNN-tax variants use attribute and category information, respectively, to infer new item embeddings. However, it is inevitable that different items may share the same attribute or category information. In such cases, the inferred new item embeddings might be identical, resulting in a reduction in recommendation accuracy. This proves that jointly using both attribute and category information as auxiliary data can better infer new item embeddings. On the other hand, MFD-GNN-io replaces the factor-based similarity matrix with the original adjacency matrix. This means that, when calculating the original item embeddings, it cannot consider the neighbor relationships on each factor, leading to a drop in model performance.

We utilize the Friedman test to evaluate the robustness of the MFD-GNN model on two datasets and four metrics [31]. Due to the limitation of the number of cases, considering that a small number of cases may lead to insufficient test power. Here, we will no longer analyze the ranking of variants on each indicator separately but integrate them, and the variant method with the best performance is assigned rank 1, and so on. Table 3 shows the average ranking of MFD-GNN and its three variants, which is calculated as follows: $R_i = \frac{1}{n}\sum R_k$. Where n=8 is the number of cases and k=4 is the number of methods. The Friedman test (significance level a=0.05) was performed based on the average ranking of the four methods as follows:

$$X_F^2 = \frac{12n}{i(i+1)}\left[\sum_{j=1} R_j^2 - \frac{i(i+1)^2}{4}\right], \tag{4.1}$$

The test results show that the X_F^2 statistic is 22.2 (degrees of freedom = 3), which is much larger than the critical value $X_{0.05,3}^2$ (7.815) at the 0.05 significance level, and the p-value is 5.7e-10, much less than 0.05, so the null hypothesis is rejected. The above results show that the difference in performance between different model variants is statistically significant. According to the average rank ranking, the average rank of MFD-GNN is significantly lower than that of other variants, which indicates that MFD-GNN exhibits the best performance and the strongest robustness across datasets and across evaluation metrics.

Table 3. The average rank of different variant methods.

Method	MFD-GNN-io	MFD-GNN-attr	MFD-GNN-tax	MFD-GNN
Average rank	4.00	2.25	2.75	1.00

4.2.3. Parameter sensitivity

(1) The impact of parameter K.

We first conducted a sensitivity experiment on the number of factors, K. We performed experiments on the Amazon G&GF dataset, keeping the item embedding dimension d fixed at 80, while progressively modifying the value of K, with each factor embedding having a dimension of d/K. In Figure 5. It can be observed that for the P@20, MRR@20, and MRR@10 metrics, the prediction accuracy is highest when the number of factors is 5. For P@10, the prediction accuracy is also close to the peak when the number of factors is 5. Overall, as the number of factors increases, model performance shows an upward trend, reaching a peak before starting to decline. This indicates that it is necessary to consider items from a multi-factor perspective. However, as the number of factors increases, the performance starts to deteriorate. This may be due to the fixed item embedding dimension d; as the number of factors increases, the dimension of each factor embedding decreases, which limits the representational capacity of the factor embeddings.

(2) The Impact of Parameter λ.

Next, we conducted a sensitivity analysis on the hyperparameter λ. Since the final loss function is made up of the prediction loss (L_{ce}), the factor independence loss (L_{dce1}), and the zero-shot learning loss (L_{dce2}). We investigated the impact of λ on two datasets. We need to adjust the value of λ to achieve the best model performance. We studied its impact on two datasets. The experimental results are shown in Figure 6, where λ_1 and λ_2 take values of 0.1, 0.3, 0.5, 0.7, and 0.9. It is clearly observed that the model performs best when λ_1 and λ_2 are around 0.5, where the weights of decoupled learning regularization loss and new task embedding learning loss are balanced. This indicates that both parts of the loss contribute to the model performance to a certain extent.

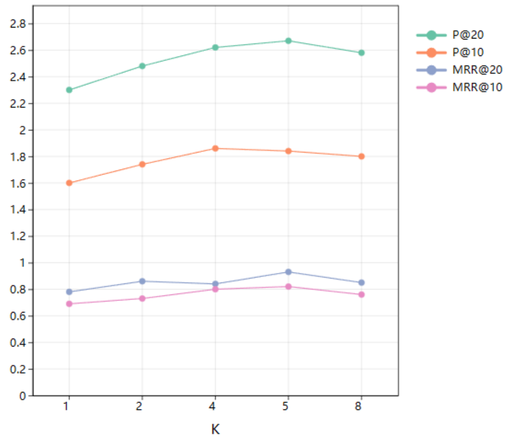

Figure 5. The Impact of Parameter K.

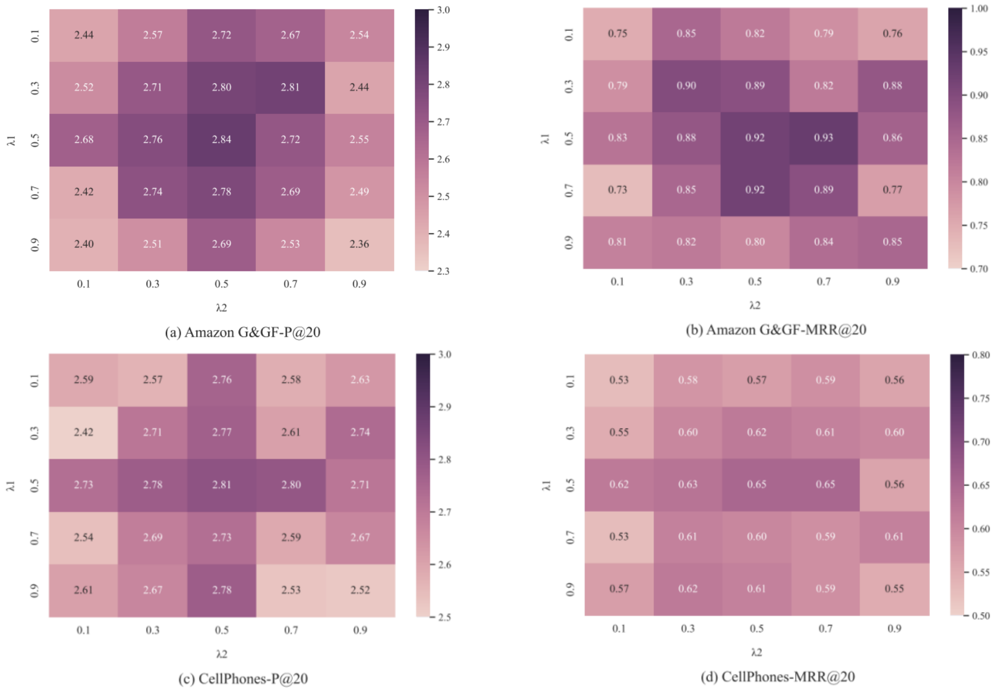

Figure 6. Impact of Parameter λ_1 and λ_2.

5. Conclusions

This paper proposes a disentangled network for session-based new item recommendation. By decoupling items into representations under different factors, we can better learn item embeddings and user intent. For item embeddings, we learn the embeddings of items under different factors using factor-based similarity matrices and aggregate the user's intent across these factors. The importance of the last interacted item is also emphasized. Finally, the inferred embeddings of the new item under each factor are generated based on auxiliary information of the new item, which is then used to predict the recommendation for the new item to the user. We tested on two datasets, and the results demonstrate the effectiveness of MFD-GNN. Looking ahead, our goal is to enhance the inference method for new item embeddings and further improve the understanding of user intent across various factors.

Author contributions

Xinning Li wrote the main manuscript text, contributed main part of the model's contruction, performed the experiments and analyzed the results. Qian Gao provided guidance on the construction of the model, experimental setup, and analysis. She also proposed revision suggestions in the course of writing the manuscript. Jun Fan engaged in some of the model's contruction and experimental environment construction. Lujie Feng participated in the implementation of some experiments.

Use of Generative-AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was funded by the project ZR2022MF333 supported by Shandong Provincial Natural Science Foundation.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. R. Patel, P. Thakkar, V. Ukani, CNNRec: Convolutional neural network based recommender systems– A survey, *Eng. Appl. Artif. Intel.*, **133** (2024), 108062. https://doi.org/10.1016/j.engappai.2024.108062
2. X. Li, L. Sun, M. Ling, Y. Peng, A survey of graph neural network based recommendation in social networks, *Neurocomputing*, **549** (2023), 126441. https://doi.org/10.1016/j.neucom.2023.126441
3. S. Wang, Q. Zhang, L. Hu, X. Zhang, Y. Wang, C. Aggarwal, Sequential/session-based recommendations: Challenges, approaches, applications and opportunities, In: *SIGIR '22: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, 3425–3428. https://doi.org/10.1145/3477495.3532685
4. Z. Wang, W. Wei, G. Cong, X. L. Li, X. L. Mao, M. Qiu, Global context enhanced graph neural networks for session-based recommendation, In: *SIGIR '20: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, 169–178. https://doi.org/10.1145/3397271.3401142

5. S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. Orgun, D. Lian, A survey on session-based recommender systems, 2019, arXiv: 1902.04864. https://doi.org/10.48550/arXiv.1902.04864
6. S. Rendle, C. Freudenthaler, L. Schmidt-Thieme, Factorizing personalized Markov chains for next-basket recommendation, In: *WWW '10: Proceedings of the 19th international conference on World wide web*, 2010, 811–820. https://doi.org/10.1145/1772690.1772773
7. G. Bonnin, D. Jannach, Automated generation of music playlists: Survey and experiments, *ACM Comput. Surv.*, 47 (2014), 1–35 https://doi.org/10.1145/2652481
8. P. Tong, Z. Zhang, Q. Liu, Y. Wang, R. Wang, CARE: Context-aware attention interest redistribution for session-based recommendation, *Expert Syst. Appl.*, **255** (2024), 124714. https://doi.org/10.1016/j.eswa.2024.124714
9. B. Hidasi, A. Karatzoglou, L. Baltrunas, D. Tikk, Session-based recommendations with recurrent neural networks, 2016, arXiv: 1511.06939v4. https://doi.org/10.48550/arXiv.1511.06939
10. B. Peng, S. Parthasarathy, X. Ning, Intention enhanced mixed attentive model for session-based recommendation, *Data Min. Knowl. Disc.*, **38** (2024), 2032–2061. https://doi.org/10.1007/s10618-024-01023-0
11. T. R. Gwadabe, Y. Liu, Improving graph neural network for session-based recommendation system via non-sequential interactions, *Neurocomputing*, **468** (2022), 111–122. https://doi.org/10.1016/j.neucom.2021.10.034
12. S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, T. Tan, Session-based recommendation with graph neural networks, *Proceedings of the AAAI Conference on Artificial Intelligence*, **33** (2019), 346–353. https://doi.org/10.1609/aaai.v33i01.3301346
13. C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, et al., Graph contextualized self-attention network for session-based recommendation, In: *IJCAI'19: Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, 3940–3946.
14. J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, J. Ma, Neural attentive session-based recommendation, In: *CIKM '17: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, 1419–1428 https://doi.org/10.1145/3132847.3132926
15. L. Yin, P. Chen, G. Zheng, Session-enhanced graph neural network recommendation model (SE-GNNRM), *Appl. Sci.*, **12** (2022), 4314 https://doi.org/10.3390/app12094314
16. D. Jin, L. Wang, Y. Zheng, G. Song, F. Jiang, X. Li, et al., Dual intent enhanced graph neural network for session-based new item recommendation, In: *WWW '23: Proceedings of the ACM Web Conference 2023*, 2023, 684–693. https://doi.org/10.1145/3543507.3583526
17. L. Chen, G. Zhu, W. Liang, J. Cao, Y. Chen, Keywords-enhanced contrastive learning model for travel recommendation, *Informa. Process. Manag.*, **61** (2024), 103874. https://doi.org/10.1016/j.ipm.2024.103874
18. L. Chen, X. Zhu, G. Zhu, Exploiting attributes and keywords for session-based recommendation with multi-view graph neural network, *Expert Syst. Appl.*, **296** (2026), 128990. https://doi.org/10.1016/j.eswa.2025.128990
19. Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, 2014, arXiv: 1206.5538v3. https://doi.org/10.48550/arXiv.1206.5538

20. Y. Yuan, F. Xu, H. Cao, G. Zhang, P. Hui, Y. Li, et al., Persuade to click: Context-aware persuasion model for online textual advertisement, *IEEE T. Knowl. Data En.*, **35** (2023), 1938–1951. https://doi.org/10.1109/TKDE.2021.3110724
21. Y. Li, D. Tarlow, M. Brockschmidt, R. Zemel, Gated graph sequence neural networks, 2016, arXiv: 1511.05493. https://doi.org/10.48550/arXiv.1511.05493
22. X. Wang, H. Jin, A. Zhang, X. He, T. Xu, T. S. Chua, Disentangled graph collaborative filtering, In: *SIGIR '20: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, 1001–1010. https://doi.org/10.1145/3397271.3401137
23. S. Zhang, Y. Lin, Y. Rao, C. Zhang, Dual-perspective disentangled multi-intent alignment for enhanced collaborative filtering, 2025, arXiv: 2506.11538v2. https://doi.org/10.48550/arXiv.2506.11538
24. S. Chen, W. Wang, B. Xia, Q. Peng, X. You, F. Zheng, et al., Free: Feature refinement for generalized zero-shot learning, In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, 122–131.
25. H. Xu, A. Woicik, H. Poon, R. B. Altman, S. Wang, Multilingual translation for zero-shot biomedical classification using BioTranslator, *Nat. Commun.*, **14** (2023), 738. https://doi.org/10.1038/s41467-023-36476-2
26. X. Zhang, B. Xu, C. Li, B. He, H. Lin, C. Ma, et al., A survey on side information-driven session-based recommendation: from a data-centric perspective, *IEEE T. Knowl. Data En.*, **37** (2025), 4411–4431. https://doi.org/10.1109/TKDE.2025.3569549
27. A. Li, Z. Cheng, F. Liu, Z. Gao, W. Guan, Y. Peng, Disentangled graph neural networks for session-based recommendation, *IEEE T. Knowl. Data En.*, 35 (2023), 7870–7882. https://doi.org/10.1109/TKDE.2022.3208782
28. X. Xia, H. Yin, J. Yu, Y. Shao, L. Cui, Self-supervised graph co-training for session-based recommendation, In: *CIKM '21: Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, 2180–2190. https://doi.org/10.1145/3459637.3482388
29. Z. Ou, X. Zhang, Y. Zhu, S. Lyu, J. Liu, T. Ao, LS-TGNN: Long and short-term temporal graph neural network for session-based recommendation, In: *Proceedings of the AAAI Conference on Artificial Intelligence*, **39** (2025), 12426–12434. https://doi.org/10.1609/aaai.v39i12.33354
30. L. Wang, D. Jin, A time-sensitive graph neural network for session-based new item recommendation, *Electronics*, **13** (2024), 223. https://doi.org/10.3390/electronics13010223
31. L. Chen, G. Zhu, Self-supervised contrastive learning for itinerary recommendation, *Expert Syst. Appl.*, **268** (2025), 126246. https://doi.org/10.1016/j.eswa.2024.126246