Balancing User-Item Structure and Interaction with Large Language Models and Optimal Transport for Multimedia Recommendation

Haodong Li[1,2], **Lianyong Qi**[1,2,3*], **Weiming Liu**[4], **Xiaolong Xu**[5], **Wanchun Dou**[3], **Yang Cao**[6,7,8], **Xuyun Zhang**[9], **Amin Beheshti**[9] and **Xiaokang Zhou**[10,11]
[1]College of Computer Science and Technology, China University of Petroleum (East China)
[2]Shandong Key Laboratory of Intelligent Oil and Gas Industrial Software
[3]State Key Lab. for Novel Software Technology, Nanjing University
[4]College of Computer Science and Technology, Zhejiang University
[5]School of Computer and Software, Nanjing University of Information Science and Technology
[6]School of Computing and Information Technology, Great Bay University
[7]Great Bay Institute for Advanced Study, Great Bay University
[8]Shenzhen International Graduate School, Tsinghua University
[9]School of Computing, Macquarie University
[10]Faculty of Business and Data Science, Kansai University
[11]RIKEN Center for Advanced Intelligence Project
lhd_hfut@163.com, lianyongqi@upc.edu.cn, 21831010@zju.edu.cn, xlxu@nuist.edu.cn, douwc@nju.edu.cn, charles.cao@ieee.org, {xuyun.zhang, amin.beheshti}@mq.edu.au, zhou@kansai-u.ac.jp

Abstract

The rapid growth of multimedia content has driven the development of recommender systems. Most previous work focuses on uncovering latent relationships among items to learn better representations. However, this approach does not sufficiently account for user affinities, potentially leading to an imbalance in the structure modeling of users and items. Moreover, the sparsity and imbalance of user-item interactions further hinder effective representation learning. To address these challenges, we propose a framework called BLAST, which BaLances structures and interActions via large language modelS and optimal Transport for multimodal recommendation. Specifically, we utilize large language models to summarize side information and generate user profiles. Based on these profiles, we design an intra- and inter-entity structure balancing module to capture item-item and user-user relationships, integrating these affinities into the final representations. Furthermore, we impose constraints on negative sample selection, augment the training data with false negative items and the optimal transport algorithm, thereby leading to smoother interactions. We evaluate BLAST on three real-world datasets, and the results demonstrate that our method significantly outperforms state-of-the-art baselines, which validates the superiority and effectiveness of BLAST.

*Corresponding author

1 Introduction

In the context of the internet age and the resulting information overload, the importance of recommendation systems is becoming progressively more evident. The emergence of multimodal data (e.g., text, images, audio, video, etc.) provides new views to understand user preferences and behavior patterns. Multimodal recommendation addresses the limitations of single-modal information, offering more efficient solutions for applications such as e-commerce and social media.

Graph-based recommendation methods have shown significant performance by modeling higher-order relationships between users and items [Wang *et al.*, 2019; He *et al.*, 2020; Liu *et al.*, 2022a]. In light of these studies, some methods attempted to explicitly learn the latent item-item semantic graphs [Zhang *et al.*, 2021; Zhou and Shen, 2023]. However, user-side information is difficult to access due to privacy policies, which makes the modeling of semantic relationships between users a challenging task, leading to **Structure imbalance**, as illustrated in Figure 1 (a). To address this issue, some studies have introduced Large Language Models (LLMs) to generate user profiles [Ren *et al.*, 2024; Ma *et al.*, 2024], while others have integrated social graphs from content platforms into graph-based recommendation models to enrich interest patterns [Wu *et al.*, 2022; Wei *et al.*, 2024]. However, simply incorporating user profiles generated by LLMs into ID embeddings fails to capture higher-order user-user relationships. Moreover, inconsistencies between social and behavioral representations may lead to noise propagation [Xiao *et al.*, 2023].

Besides structure imbalance, recommendation systems also suffer from the **Interaction imbalance** [Liu *et al.*,

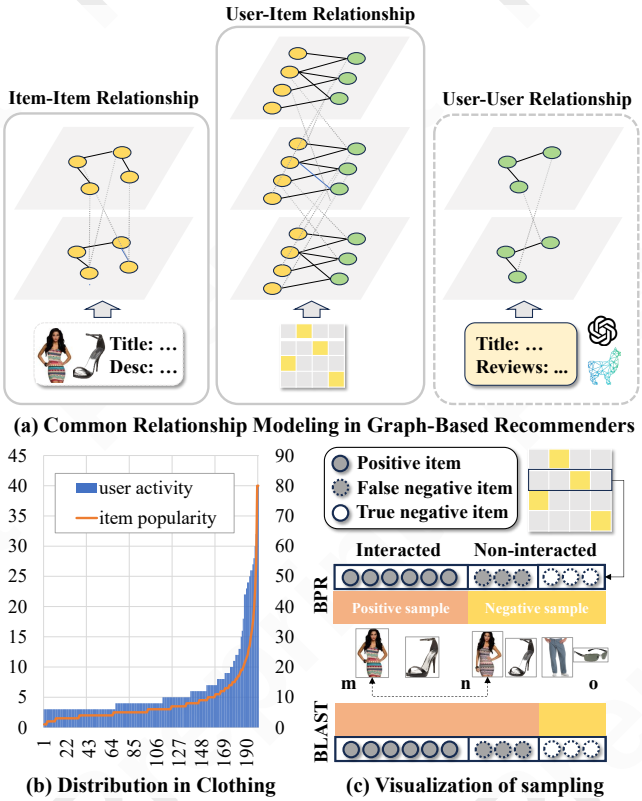

Figure 1: (a) Visualization of relationship modeling in multimodal recommendation models. (b) Distribution of user activity and item popularity in the Clothing dataset across 200 samples. (c) Visualization of sampling from interaction data.

2022b; Wahab *et al.*, 2022; Liu *et al.*, 2023c], which refers to the uneven distribution of user activity and item popularity within interaction data. Here, activity indicates the number of items a user interacts with, while popularity denotes the number of users interacting with an item. As shown in Figure 1 (b), the disparity in user activity and item popularity is so pronounced that many users and items suffer from data sparsity, hindering the learning of high-quality user and item representations [Liu *et al.*, 2021; Xu *et al.*, 2024]. To enrich the behavioral history of sparsely-interacted entities (users or items), more potential items should be uncovered. However, they are often obscured by false negative noise [Li *et al.*, 2021; Ren *et al.*, 2024]. Suppose a female user has not been exposed to an item n that she might like, which is similar to a previously purchased item m, as illustrated in Figure 1 (c). During training, item n may be treated as a negative sample, which can lead the recommender to suppress content the user is actually interested in. This not only wastes potential items but also misguides the modeling of user preferences.

In light of the above limitations and challenges, we propose a novel framework called BLAST that BaLances structures and interActions via large language modelS and optimal Transport for multimodal recommendation. To rectify structure imbalance, we first generate user profiles using large language models. Then, we design an intra- and inter-entity structure balancing module, which converts multimodal features into user-user and item-item graphs, bridging the semantic gap between multimodal information and collaborative signals. Finally, we perform graph convolution on three graphs (the user-item graph, the user-user graph, and the item-item graph) and fuse their outputs to derive a comprehensive representation.

To mitigate interaction imbalance, we first constrain the selection of negative samples when generating the training set to avoid interference from false negatives. Furthermore, to enrich entities with few interactions, we augment the training data with false negative items, then refine the augmented dataset by solving an Optimal Transport (OT) problem, thus providing a solid foundation for efficient representation learning. Note that our OT Augmentation is model-agnostic, which means that it can serve as a plug-and-play tool for a wide range of recommendation models to boost performance.

Our main contributions is summarized as follows:

- We propose a novel framework called BLAST to rectify imbalances in structure and interaction for multimedia recommendation, which can effectively fuse multimodal information and collaborative signals to improve item and user representations.
- We highlight the importance of false negative items and propose an effective data augmentation strategy that can be seamlessly incorporated into different recommendation models to enhance their performance.
- Extensive experiments on three real-world datasets demonstrate the superiority and effectiveness of BLAST in multimedia recommendation.

To foster reproducible research, our code is made publicly available at: https://github.com/UPCLHD/BLAST-master.

2 Related Work

2.1 Multimodal Recommendation

Multimodal recommendation leverages multimedia content to model user preferences and item attributes. Previous work tends to treat multimedia features as side information to enrich item representations [He and McAuley, 2016; Liu *et al.*, 2017]. However, due to the semantic gap between multimodal information and collaborative signals, directly fusing the two may not be optimal. To this end, several graph-based recommendation methods [Zhang *et al.*, 2021; Guo *et al.*, 2024] have attempted to model item-item relationships by converting multimodal features into bipartite graphs. Nevertheless, the user-item interaction graphs in these methods are susceptible to noise introduced by popularity bias or accidental clicks. To address this issue, recent research has explored cropping redundant edges [Rong *et al.*, 2020; Zhou *et al.*, 2023b; Liu *et al.*, 2023a]. In this work, we argue that user-user relationships are as important as item-item ones, and thus we devise a user-item balanced structure.

2.2 Large Language Models for Recommendation

Recently, several studies have attempted to introduce LLMs into recommendation tasks as inference models [Bao *et al.*,

2023]. Leveraging the powerful text comprehension and inference capabilities of LLMs, the cross-domain recommendation task has been extended [Vajjala *et al.*, 2024], where the behavior history of the same user in one domain can be applied to other domains to alleviate the cold start problem. RLMRec [Ren *et al.*, 2024] utilizes LLMs to refine valuable text to learn informative representations. XRec [Ma *et al.*, 2024] builds an explainable recommendation system and generates data by LLMs for supervised training. To enhance interactivity, Chat-Rec [Gao *et al.*, 2023] designs a conversational system which generates recommendations by LLMs. However, there are some disadvantages of using LLMs directly as recommenders, such as the illusion issue and slow inference speed. Therefore, in this paper, we only employ LLMs as a preprocessor to help us refine the textual information of items as well as to summarize user preferences.

3 Preliminaries

In our recommendation scenario, we denote the set of users to be $\boldsymbol{U}(|\boldsymbol{U}| = M)$, the set of items to be $\boldsymbol{I}(|\boldsymbol{I}| = N)$, and the user-item interaction matrix to be $\boldsymbol{R} \in \mathbb{R}^{M \times N}$, where $\boldsymbol{R}_{ui} = 1$ if user $u \in \boldsymbol{U}$ has interacted with item $i \in \boldsymbol{I}$, otherwise $\boldsymbol{R}_{ui} = 0$. $\boldsymbol{z}_u, \boldsymbol{z}_i \in \mathbb{R}^d$ is the ID embedding of u and i, respectively, where d is the embedding dimension. $\mathcal{N}_u$ indicates the set of items that user u has interacted with (positive items). We define the triple (u, i, j) for training, where $i \in \mathcal{N}_u$, $j \in I$, but $j \notin \mathcal{N}_u$. j is selected from negative items at random. Each element of $\boldsymbol{R}$ with a value of 1 ($\boldsymbol{R}_{ui} = 1$) corresponds to a triple (u, i, j), all of which make up the entire training set $\mathcal{D}$. We adopt the BPR loss to train models:

$$L_{BPR} = -\sum_{u \in \boldsymbol{U}} \sum_{i \in \mathcal{N}_u} \sum_{j \notin \mathcal{N}_u} ln(y_{ui} - y_{uj}), \tag{1}$$

where y_{ui} denotes the predicted preference score of user u for item i. The purpose of the recommendation system is to offer the top-k uninteracted items with the highest score for user u.

In this paper, we set the multimodal features of item i as $\boldsymbol{x}_i^m \in \mathbb{R}^{d_m}$, where $m \in \{v, t\}$ is the modality and d_m is the dimension of the features over modality m. v and t denote the visual and textual modalities, respectively.

4 Method

In this section, we describe the core components of BLAST: User and Item Profile Generator, Intra- and Inter-entity Structure Balancing Module, and OT Augmentation. Figure 2 shows the overall architecture.

4.1 User and Item Profile Generator

Following [Ren *et al.*, 2024; Ma *et al.*, 2024], We extracted user and item profiles by feeding prompts and raw data into the LLM. Specifically, the item profile P_i generated by LLM can be described as follows:

$$P_i = \begin{cases} LLM(M_i, l_i, d_i), & \text{if } d_i \text{ exists,} \\ LLM(M_i, l_i, c_i, r_{u,i}), & \text{otherwise,} \end{cases} \tag{2}$$

where M_i denotes the pre-defined item prompt, l_i, d_i, and c_i are title, description and categories of item i, respectively, and $r_{u,i}$ represents the review of user u on item i. To extract deep contextual insights from LLMs, M_i is designed to illustrate the data format, task details, and expected output.

Sampled from items that user u has interacted with, the user profile is generated as follows:

$$P_u = LLM(M_u, l_i, P_i, r_{u,i} | i \in \mathcal{N}_u), \tag{3}$$

where M_u represents the pre-defined user prompt. M_u elaborates the input and output formats for the LLM and asks it to analyze user preferences. Finally, combining the side information with collaborative signals, we obtain the brief but informative item profile P_i and user profile P_u.

To capture the semantic relationships in the text, P_u and P_i need to be encoded to fixed-length representations:

$$\boldsymbol{y}_u = T(P_u), \boldsymbol{x}_i^t = T(P_i), \tag{4}$$

where T denotes the text encoder. $\boldsymbol{x}_i^t \in \mathbb{R}^d$ and $\boldsymbol{y}_u \in \mathbb{R}^d$ are the text-modality (t) feature of item i and user u, respectively.

4.2 Intra- and Inter-entity Structure Balancing Module

Item-Item and User-User Graph Constructing

Following [Zhou and Shen, 2023], we construct an initial modality-aware item-item graph $\boldsymbol{S}^m$ by features of each modality m. $\boldsymbol{S}_{ij}^m$ is the element of the matrix $\boldsymbol{S}^m \in \mathbb{R}^{N \times N}$ in row i, column j, which is calculated by the cosine similarity between the features of item i and item j in modality m:

$$\boldsymbol{S}_{ij}^m = \frac{(\boldsymbol{x}_i^m)^\top \boldsymbol{x}_j^m}{\|\boldsymbol{x}_i^m\| \left\|\boldsymbol{x}_j^m\right\|}, (i, j \in I). \tag{5}$$

Then, kNN sparsification [Chen *et al.*, 2009] is conducted. For each item i, only the top-k similar edges are retained:

$$\widehat{\boldsymbol{S}}_{ij}^m = \begin{cases} 1, & \boldsymbol{S}_{ij}^m \in top-k(\boldsymbol{S}_i^m), \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

We set k to 10, and $\widehat{\boldsymbol{S}}_{ij}^m$ is normalized to $\widetilde{\boldsymbol{S}}^m$, following [He *et al.*, 2020]. Then, information from all modalities are aggregated into the final item-item graph $\boldsymbol{S}_I = \sum_{m \in M} \alpha_m \widetilde{\boldsymbol{S}}^m$.

Beyond the item-item graph $\boldsymbol{S}_I$, the user-user graph $\boldsymbol{S}_U$ is constructed from textual features $\boldsymbol{y}_u$ extracted from user profiles. Similarly, after getting the user similarity matrix, we perform kNN sparsification:

$$\boldsymbol{S}_{ij}^U = \frac{(\boldsymbol{y}_i)^\top \boldsymbol{y}_j}{\|\boldsymbol{y}_i\| \|\boldsymbol{y}_j\|}, (i, j \in \boldsymbol{U}), \tag{7}$$

$$\widehat{\boldsymbol{S}}_{ij}^U = \begin{cases} 1, & \boldsymbol{S}_{ij}^U \in top-k(\boldsymbol{S}_i^U), \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

$\widehat{\boldsymbol{S}}_{ij}^U$ is then normalized to $\boldsymbol{S}_U$. Since there is no image for users, $\boldsymbol{S}_U$ is the final user-user graph.

Item and User Representation Constructing

Graph-based recommendation methods usually aggregate and propagate information between nodes to model higher-order relationships in user interactions [Zhang *et al.*, 2021; Liu *et al.*, 2023b; Guo *et al.*, 2024]. Inspired by these studies, we

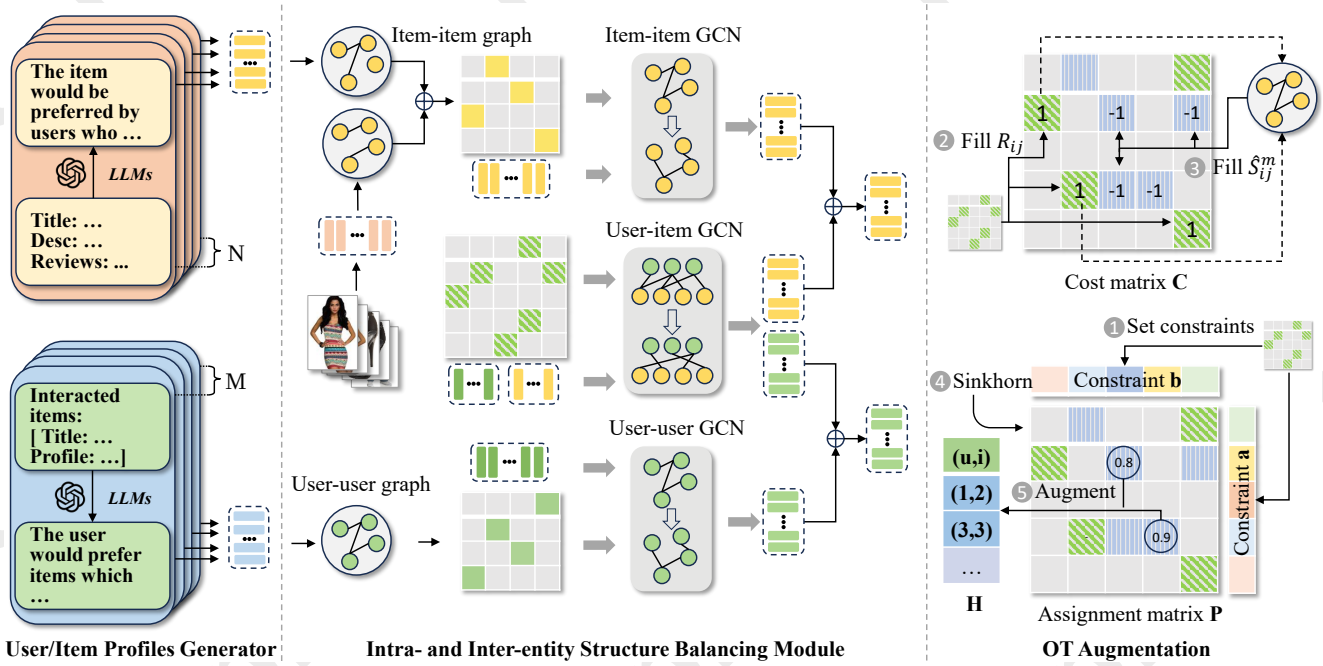

Figure 2: The overall framework of our proposed structure and interaction balanced model BLAST.

conduct graph convolutions on the item-item graph $\boldsymbol{S}_I$, the user-user graph $\boldsymbol{S}_U$, and the user-item graph $\boldsymbol{A}$, respectively.

User-Item graph convolution. Following [Zhou and Shen, 2023], we conduct the edge pruning and normalization on $\boldsymbol{A} = \begin{pmatrix} 0 & \boldsymbol{R} \\ \boldsymbol{R}^\top & 0 \end{pmatrix}$ to get $\boldsymbol{A}'$. Then, we perform the graph convolution:

$$\boldsymbol{H}_{UI}^{(l+1)} = \boldsymbol{A}'\boldsymbol{H}_{UI}^{(l)}, l \in [0, L_{UI}], \tag{9}$$

where $\boldsymbol{H}_{UI}^{(l)} \in \mathbb{R}^{(M+N)\times d}$ is the l-th layer embedding matrix, $\boldsymbol{H}_{UI}^{(0)}$ is initialized by ID embedding $\boldsymbol{z}_u, \boldsymbol{z}_i$, and L_{UI} is the number of layers of the user-item graph convolution. The embedding matrices obtained at each layer are combined to form the representation:

$$\boldsymbol{H}_{UI} = \frac{1}{L_{UI}+1}\sum_{l=0}^{L_{UI}} \boldsymbol{H}_{UI}^{l}. \tag{10}$$

Item-Item and User-User graph convolution. In addition to modeling the relationship between users and items, we also mine the relevance within items and users. Therefore, we conduct graph convolution on the item-item graph $\boldsymbol{S}_I$:

$$\boldsymbol{H}_I^{(l+1)} = \boldsymbol{S}_I\boldsymbol{H}_I^{(l)}, l \in [0, L_{II}], \tag{11}$$

where $\boldsymbol{H}_I^{(l)} \in \mathbb{R}^{N\times d}$ is the l-th layer item embedding matrix, $\boldsymbol{H}_I^{(0)}$ is initialized by item ID embedding $\boldsymbol{z}_i (i \in \boldsymbol{I})$, and L_{II} is the number of layers of the item-item graph convolution. Analogously, the user embedding matrix $\boldsymbol{H}_U^{(l)} \in \mathbb{R}^{M\times d}$ is obtained by graph convolution on the user-user graph $\boldsymbol{S}_U$:

$$\boldsymbol{H}_U^{(l+1)} = \boldsymbol{S}_U\boldsymbol{H}_U^{(l)}, l \in [0, L_{UU}]. \tag{12}$$

Embedding fusion. To integrate relationships from multiple graph convolutions, we fuse the user-item, item-item, and user-user embeddings to obtain the final representation:

$$\boldsymbol{f}_i = \boldsymbol{f}_i^I + \boldsymbol{f}_i^{UI}, i \in [1, N], \tag{13}$$

$$\boldsymbol{h}_u = \boldsymbol{h}_u^U + \boldsymbol{h}_u^{UI}, u \in [1, M], \tag{14}$$

where $\boldsymbol{f}_i^I$ and $\boldsymbol{h}_u^U$ are the embedding in $\boldsymbol{H}_I^{(L_{II}+1)}$ and $\boldsymbol{H}_U^{(L_{UU}+1)}$, respectively, $\boldsymbol{f}_i^{UI}$ and $\boldsymbol{h}_u^{UI}$ are the embedding in $\boldsymbol{H}_{UI}$. Then, the predicted preference scores are obtained by calculating the inner product of user and item representations:

$$y_{ui} = \boldsymbol{h}_u^T\boldsymbol{f}_i. \tag{15}$$

4.3 OT Augmentation

As illustrated in Figure 1 (c), given a user u, the items which have interacted with u are called positive items ($\mathcal{N}_u$), otherwise negative items. BPRloss (Eqs. 1) encourages the score of a positive pair (user u, positive item m) to be higher than a negative pair (user u, negative item n), where m and n sampled from positive items and negative items, respectively. A further distinction can be made among negative items: true negative items T_u, which are those that user u indeed dislikes, such as o, and false negative items F_u, which are those that user u has not seen but is interested in, such as n. In our study, false negative items are defined based on multimodal features. Specifically, for a positive item m, items connected to it in either the visual item-item graph ($\widehat{\boldsymbol{S}}^v_{mn} = 1$) or the textual item-item graph ($\widehat{\boldsymbol{S}}^t_{mn} = 1$) are regarded as its false negative items.

Negative Pairs Sampling Restriction

If user u has purchased item m, she may also be interested in item n because of their similar appearance or function. How-

Method	Aug ratio	R@10	N@10
Original	-	0.0613	0.0332
Add 1x	100%	0.0641	0.0346
Add to 10	62%	0.0640	0.0348
OT Aug.	10%	**0.0651**	**0.0351**

Table 1: Performance under different augmentation strategies on the Clothing dataset. Aug ratio indicates the ratio of increased data to original data.

ever, if item n is mistakenly included in a negative pair, the model will be trained to assign it a lower prediction score, which may lead the model to misrepresent the user's true preferences. Therefore, false negative items should be excluded from negative pairs. To this end, we impose a constraint on negative pair sampling in the BPR loss. Specifically, we restrict negative samples to be drawn only from true negative items T_u, as shown in Figure 1 (c).

Effect of False Positive Items

Furthermore, we believe that positive samples can also be taken from false negative items. To verify this, we conduct an experiment. Specifically, we build positive pairs using false negative items to augment the Clothing dataset and train a classical model FREEDOM under different ugmentation strategies. The results are demonstrated in Table 1 (higher is better). The experimental settings are as follows: (1) The original dataset without any augmentation. (2) Augment positive pairs by sampling from the false negative items until the number of training data doubles. (3) Augment positive pairs by sampling from the false negative items until each user interacts with 10 items at least. (4) Augment positive pairs with guidance of OT. We can clearly observe that false negative items can be used to supplement samples for items and users that lack interactions, thus compensating for the imbalance of interaction data and improving recommendation accuracy.

Data Augmentation via Sinkhorn

In this work, we reformulate the task of sampling positive pairs from false negative items for data augmentation as an OT problem. The objective of OT is to find an assignment matrix $\boldsymbol{P}$ that minimizes the cost $T = \sum_{i,j} \boldsymbol{C}_{ij}\boldsymbol{P}_{ij}$, while satisfying the constraint $\sum_j \boldsymbol{P}_{ij} = a_i$ and $\sum_i \boldsymbol{P}_{ij} = b_j$. $\boldsymbol{C}$ is the cost matrix. By converting $\boldsymbol{C}$ into $-\boldsymbol{C}$, the cost becomes $T = -\sum_{i,j}(-\boldsymbol{C}_{ij})\boldsymbol{P}_{ij}$. The new objective is to maximize $\sum_{i,j}(-\boldsymbol{C}_{ij})\boldsymbol{P}_{ij}$, which is equivalent to finding a $\boldsymbol{P}$ that maximizes the inner product with $-\boldsymbol{C}$. As illustrated in Figure 2, we set a_i, b_j as the average user activity and item popularity of the dataset, respectively, and fill $\boldsymbol{C}$ with the user-item interaction matrix $\boldsymbol{R}$ and the item-item graph $\widehat{\mathbf{S}}_{ij}^m$:

$$\boldsymbol{C}_{ij} = \begin{cases} 1, \ if \ \boldsymbol{R}_{ij} = 1, \\ -1, \ if \ j \in \mathcal{G}_i, \\ 0, \ otherwise, \end{cases} \tag{16}$$

where $\mathcal{G}_i$ denotes the set of items that item i has interacted with in graph $\widehat{\boldsymbol{S}}_{ij}^m$ $(m \in \{v, t\})$, which includes items most visually or textually similar to item i. To maximize the inner product, the elements of $\boldsymbol{P}$ and $-\boldsymbol{C}$ tend to align. Therefore, at positions where $\boldsymbol{R}_{ij} = 1$ $(-\boldsymbol{C}_{ij} = -1)$, $\boldsymbol{P}_{ij}$ is encouraged to take a negative value, thereby reducing redundancy between the augmented and original data. In contrast, when j is a false negative item $(-\boldsymbol{C}_{ij} = 1)$, $\boldsymbol{P}_{ij}$ tends to be assigned a positive value, thus increasing its likelihood of being selected as part of a positive pair. Furthermore, the row sum (a_i) and column sum (b_j) of $\boldsymbol{P}$ are restricted to match the average user activity and item popularity of the dataset, respectively. This ensures that users and items with fewer interactions are provided with more augmented data.

Following [Sarlin *et al.*, 2020; Izquierdo and Civera, 2024], we use the Sinkhorn algorithm to get $\boldsymbol{P}$. Finally, we construct a set $\boldsymbol{H} = \{(u, i) | \boldsymbol{P}_{ui} \in top - q(\boldsymbol{P})\}$ by collecting the indices of the top q largest elements in the assignment matrix $\boldsymbol{P}$, where $q = r \times |\mathcal{D}|$. r is the augmentation ratio, and $|\mathcal{D}|$ is the size of the original training set. We add $\boldsymbol{H}$ to the original dataset to obtain the augmented training set $\mathcal{D}_H = \{(u, i, j) | (u, i) \in \boldsymbol{H} \ or \ \boldsymbol{R}_{ui} = 1, j \in T_u\}$, which is used to train recommendation models later.

The computational complexity of OT Augmentation is $\mathcal{O}(MN)$. Both OT Augmentation and LLM-based profile generation are conducted as preprocessing steps before training, and thus only need to be executed once per dataset, with the results reused throughout the training process. Moreover, BLAST performs LightGCN-style graph convolutions on three graphs using PyTorch's sparse matrix operations for efficiency. Its computational complexity is $\mathcal{O}(d \cdot (\|\boldsymbol{R}\|_0 + M + N))$, where $\|\boldsymbol{R}\|_0$ denotes the number of non-zero entries in the user-item interaction matrix $\boldsymbol{R}$.

5 Experiment

To validate the effectiveness of our proposed method, we conduct experiments on three widely used datasets to answer the following research questions:

- RQ1: How does BLAST perform compared with state-of-the-art multimodal recommendation models?
- RQ2: How does OT Augmentation perform when applied to other models?
- RQ3: How do different components of BLAST contribute to the overall performance?
- RQ4: How sensitive is BLAST to perturbations in hyperparameters?

5.1 Experiments Settings

Datasets. Following [Zhang *et al.*, 2021; Zhou and Shen, 2023], we conduct experiments on three subsets of the Amazon dataset [McAuley *et al.*, 2015]: (1) Baby, (2) Sports and Outdoors, (3) Clothing, Shoes and Jewelry. We refer to them simply as Baby, Sports and Clothing. Each dataset provides both visual and textual information. The visual information consists of 4096-dimensional features extracted from item images. The textual information includes the title, description, categories, and brand of the items. Following [Ren *et al.*, 2024; Zhang *et al.*, 2023], we encode all textual data (including user and item profiles) by a pre-trained text encoder [Izacard *et al.*, 2022].

model	Baby				Sport				Clothing			
	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
BPR	0.0379	0.0607	0.0202	0.0261	0.0452	0.069	0.0252	0.0314	0.0211	0.0315	0.0118	0.0144
LightGCN	0.0479	0.0754	0.0257	0.0328	0.0569	0.0864	0.0311	0.0387	0.0361	0.0544	0.0197	0.0243
BM3	0.0564	0.0883	0.0301	0.0383	0.0656	0.0980	0.0355	0.0438	0.0422	0.0621	0.0231	0.0281
LATTICE	0.0547	0.0850	0.0292	0.0370	0.0620	0.0953	0.0335	0.0421	0.0492	0.0733	0.0268	0.0330
FREEDOM	0.0627	0.0992	0.0330	0.0424	0.0717	0.1089	0.0385	0.0481	0.0629	0.0941	0.0341	0.0420
MGCN	0.0620	0.0964	0.0339	0.0427	0.0729	0.1106	0.0397	0.0496	0.0641	0.0945	0.0347	0.0428
LGMRec	0.0644	0.1002	0.0349	0.0440	0.0720	0.1068	0.0390	0.0480	0.0555	0.0828	0.0302	0.0371
BLAST	**0.0767**	**0.1171**	**0.0426**	**0.0529**	**0.0781**	**0.1186**	**0.0422**	**0.0526**	**0.0681**	**0.1004**	**0.0372**	**0.0453**
Improv.	19.10%	16.87%	22.06%	20.23%	7.13%	7.23%	6.30%	6.05%	6.24%	6.24%	7.20%	5.84%

Table 2: Overall performances of BLAST and baselines on three datasets. The best results are bolded and the second best results are underlined. Improv. indicates the improvement of BLAST on second best results. R@K and N@K denote Recall@K and NDCG@K.

Model	Metric	LayerGCN			VBPR			SLMRec			BM3			FREEDOM			MGCN		
		Base	OT Aug.	Imprv.	Base	OT Aug.	Imprv.	Base	OT Aug.	Imprv.	Base	OT Aug.	Imprv.	Base	OT Aug	Imprv.	Base	OT Aug.	Imprv.
Baby	R@20	0.0829	**0.0900**	8.56%	0.0664	**0.0830**	25.00%	0.0806	**0.0895**	11.04%	0.0865	**0.0945**	9.25%	0.0986	**0.1013**	2.74%	0.0938	**0.0991**	5.65%
	N@20	0.0359	**0.0391**	8.91%	0.0285	**0.0350**	22.81%	0.0356	**0.0391**	9.83%	0.0372	**0.0402**	8.06%	0.0421	**0.0445**	5.70%	0.0412	**0.0430**	4.37%
Sport	R@20	0.0940	**0.1043**	10.96%	0.0830	**0.0980**	18.07%	0.1005	**0.1084**	7.86%	0.0973	**0.1030**	5.86%	0.1078	**0.1104**	2.41%	0.1135	**0.1145**	0.88%
	N@20	0.0415	**0.0464**	11.81%	0.0367	**0.0419**	14.17%	0.0450	**0.0483**	7.33%	0.0433	**0.0461**	6.47%	0.0478	**0.0489**	2.30%	0.0509	**0.0516**	1.38%
Clothing	R@20	0.0557	**0.0770**	38.24%	0.0411	**0.0792**	92.70%	0.0692	**0.0856**	23.70%	0.0626	**0.0785**	25.40%	0.0926	**0.0965**	4.21%	0.0963	**0.0998**	3.63%
	N@20	0.0249	**0.0341**	36.95%	0.0190	**0.0352**	85.26%	0.0310	**0.0380**	22.58%	0.0283	**0.0352**	24.38%	0.0411	**0.0431**	4.87%	0.0436	**0.0450**	3.21%

Table 3: Recommendation performance improvement of several backbone methods on three datasets with OT augmentation.

Baselines. We compare BLAST with representative state-of-the-art models, including collaborative filters (BPR [Rendle *et al.*, 2009], LightGCN [He *et al.*, 2020], and LayerGCN [Zhou *et al.*, 2023b]) and multimodal recommenders (VBPR [He and McAuley, 2016], LATTICE [Zhang *et al.*, 2021], SLMRec [Tao *et al.*, 2023], BM3 [Zhou *et al.*, 2023c], FREEDOM [Zhou and Shen, 2023], MGCN [Yu *et al.*, 2023], and LGMRec [Guo *et al.*, 2024]).

Evaluation Protocols. To ensure fair comparisons, we adopt the same evaluation settings as those used in [Zhang *et al.*, 2021; Zhou and Shen, 2023]. Specifically, we randomly split all user-item interactions into training, validation and testing sets with a ratio of 8:1:1, and evaluate model performance on the testing set using two widely used metrics: Recall@K and NDCG@K (K = 10 or 20).

Implementation details. We adopt Llama 3.1[1] to generate item and user profiles. The profiles are encoded into 768-dimensional textual features $\boldsymbol{x}_i^t$ and $\boldsymbol{y}_u$ using Contriever [Izacard *et al.*, 2022]. We train BLAST using the Adam optimizer [Kingma and Ba, 2015], with a batch size of 2048 and a learning rate of 1e-3. Following [Zhou and Shen, 2023], we perform edge pruning on the user-item graph with pruning ratios of 0.8 or 0.9. The optimal hyperparameters, such as L_{UI}, L_{UU}, and L_{UU}, are selected via grid search. The embedding parameters are initialized using the Xavier method [Glorot and Bengio, 2010]. Early stopping is applied when no improvement is observed on the validation set for 20 consecutive epochs. Visual feature ratio α_v is set to 0.1.

5.2 Performance Comparison

Performance of BLAST (RQ1). To demonstrate the effectiveness of our proposed method, we compare BLAST with several state-of-the-art models. The results are presented in Table 2. All baseline results are directly cited from their original papers. We make the following key observations:

(1) BLAST significantly outperforms other models, including both traditional collaborative filtering and graph-based approaches, demonstrating the effectiveness of our method. Specifically, compared with the strongest baseline, BLAST achieves improvements of 22.06%, 6.30%, and 7.20% on NDCG@10 for the Baby, Sport, and Clothing datasets, respectively. These results suggest that complementing the graph structure with user affinities and balancing interactions through data augmentation are beneficial for enhancing recommendation performance.

(2) BLAST achieves the largest improvement on the Baby dataset, outperforming LGMRec by 19.10% and 22.06% in Recall@10 and NDCG@10, respectively. This may be attributed to the Baby dataset containing more comprehensive attributes and richer item descriptions, resulting in higher-quality item profiles. In BLAST, the item-item graph $\boldsymbol{S}_I$ is constructed from the item profiles P_i. Hence, the item text not only directly affects the graph convolution operation but also indirectly influences OT Augmentation through the selection of false negative items.

Performance of OT Augmentation (RQ2). To demonstrate the effectiveness of OT Augmentation, we integrate it into several recommendation models. Table 3 shows the consistent performance gains achieved by augmented interaction data. Note that all results reported in Table 3 are obtained from MMRec [Zhou *et al.*, 2023a], and may differ from those in Table 2. To ensure a fair comparison, we keep the basic hyperparameter settings consistent for both baselines and their augmented variants. We can observed that OT Augmentation, as a model-agnostic method, significantly enhances the performance of baselines. This highlight both the potential of false negative items and the effectiveness of complementing entities with fewer interactions.

[1]https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

Model	Baby		Sport		Clothing	
	R@20	N@20	R@20	N@20	R@20	N@20
FREEDOM	0.0992	0.0424	0.1089	0.0481	0.0941	0.0420
$\text{BLAST}_{\text{LLM}}$	0.1009	0.0432	0.1097	0.0487	0.0914	0.0407
BLAST_{FN}	0.1013	0.0445	0.1104	0.0489	0.0965	0.0431
$\text{BLAST}_{\text{USER}}$	0.1120	0.0512	0.1155	0.0517	0.0973	0.0444
BLAST	0.1171	0.0529	0.1186	0.0526	0.1004	0.0453

Table 4: Comparing performance of BLAST and its variants.

5.3 Ablation Studies (RQ3)

In this section, we evaluate the contribution of each component in BLAST to the overall effectiveness. We choose FREEDOM as the baseline and design several variants:

- $\text{BLAST}_{\text{LLM}}$: The LLM summarizes item texts to generate item profiles and learn item-item relationships.
- BLAST_{FN}: Negative sample selection is restricted, and false negative items are utilized to augment the data.
- $\text{BLAST}_{\text{USER}}$: User-user relationships are modeled with user profiles to enhance user representations.

The results are shown in Table 4, from which we have the following observations:

(1) Each component of BLAST has a positive impact on the improvement of performance. The most significant gain in performance comes from $\text{BLAST}_{\text{USER}}$, highlighting the effectiveness of incorporating user-user relationships to balance the user and item structure. Notably, these relationships are derived from the LLM, demonstrating its strong contextual understanding and reasoning abilities.

(2) On the Clothing dataset, item profiles generated by the LLM result in decreased performance. We attribute this to the limited availability of item descriptions in the dataset. An observation of the profiles suggests that the lack of sufficient information may lead the LLM to make inaccurate inferences.

(3) BLAST_{FN} gains consistent improvements over baseline on three datasets, which validates the effectiveness of restrictions and augmentations for false negative items. Constraining negative sampling helps prevent false negatives from disrupting the modeling of user interests. Mining potential interactions balances the training data and supports the representation learning of low-interaction entities.

5.4 Sensitivity Analysis (RQ4)

Augmentation Ratio. To explore the optimal augmentation ratio r, We conduct experiments on three datasets, As illustrated in Figure 3 (a). We observe that BLAST generally achieves optimal performance when r is small (0.1-0.3), and its performance declines as r increases. We infer that with the continuous expansion of the training data, some true negative items are introduced, which encourages BLAST to increase the prediction scores on items that users dislikes, thus decreasing the recommendation accuracy.

Visual Feature Ratio & Edge Pruning Ratio. As illustrated in Figure 3 (b), we evaluate the performance of BLAST under different modality ratios and pruning rates. We observe that BLAST achieves the best performance when the visual feature ratio is low and the edge pruning ratio is high. We

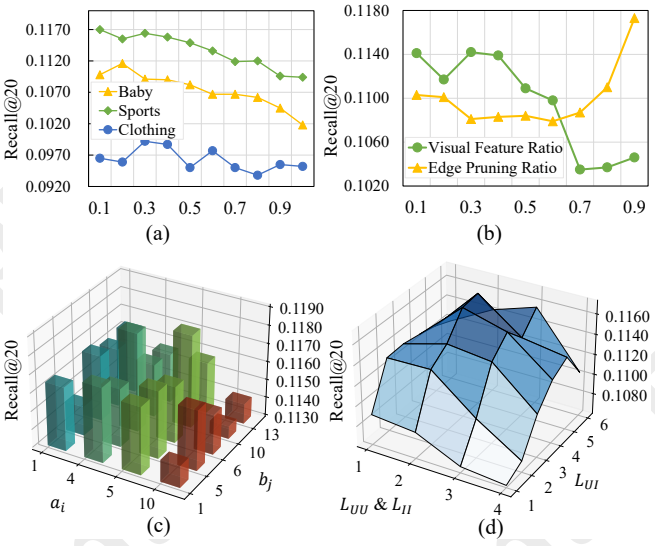

Figure 3: Performance of BLAST under different hyperparameters.

speculate that, compared to visual features, textual features are more informative and thus more effective in capturing latent item-item relationships.

Assignment Matrix Constraints. Figure 3 (c) illustrates the performance of BLAST under different OT constraints (a_i, b_j) on the Sports dataset. We observe that BLAST benefits from appropriate constraint values, particularly when a_i and b_j are close to the average user activity and item popularity (Sports: 6.14, 11.90). By taking average activity as the objective, OT Augmentation encourages low-interaction entities to engage with more entities, thereby facilitating better representation learning.

Number of Graph Convolutional Layers. As illustrated in Figure 3 (d), we evaluate the performance of BLAST under different combinations of L_{UI}, L_{UU}, and L_{II} on the Baby dataset. We observe that BLAST achieves optimal performance with the following number of layers: $L_{UI} = 5$, $L_{UU} = 2$, and $L_{II} = 2$. This indicates that graph structures built on semantic similarity can effectively model high-order relationships, highlighting the strength of BLAST.

6 Conclusion

In this paper, we propose a model called BLAST that balance user-item structure and interaction for multimedia recommendation. Specifically, we utilize LLMs to generate user profiles, and devise an intra- and inter-entity structure balancing module that model the item-item and user-user relationship simultaneously, integrating their affinities into final representations. Afterwards, to compensate for the user or item with few interactions, we augment the training data guided by the optimal transport and restrict negative sampling to exclude the interference of false negative items. Finally, extensive experiments are conducted on three real-world datasets to demonstrate the effectiveness of our proposed method.

Acknowledgements

This work is partially supported by the Natural Science Foundation of Shandong Province (ZR2023MF007), State Key Lab. for Novel Software Technology (KFKT2024B50, KFKT2024A03).

References

[Bao *et al.*, 2023] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, pages 1007–1014, 2023.

[Chen *et al.*, 2009] Jie Chen, Haw-ren Fang, and Yousef Saad. Fast approximate *k*nn graph construction for high dimensional data via recursive lanczos bisection. *J. Mach. Learn. Res.*, 10:1989–2012, 2009.

[Gao *et al.*, 2023] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *CoRR*, abs/2303.14524, 2023.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org, 2010.

[Guo *et al.*, 2024] Zhiqiang Guo, Jianjun Li, Guohui Li, Chaoyang Wang, Si Shi, and Bin Ruan. Lgmrec: Local and global graph learning for multimodal recommendation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 8454–8462, 2024.

[He and McAuley, 2016] Ruining He and Julian J. McAuley. VBPR: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 144–150, 2016.

[He *et al.*, 2020] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 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, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 639–648, 2020.

[Izacard *et al.*, 2022] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022, 2022.

[Izquierdo and Civera, 2024] Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 17658–17668, 2024.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[Li *et al.*, 2021] Xiucheng Li, Jin Yao Chin, Yile Chen, and Gao Cong. Sinkhorn collaborative filtering. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 582–592, 2021.

[Liu *et al.*, 2017] Qiang Liu, Shu Wu, and Liang Wang. Deepstyle: Learning user preferences for visual recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 841–844, 2017.

[Liu *et al.*, 2021] Weiming Liu, Jiajie Su, Chaochao Chen, and Xiaolin Zheng. Leveraging distribution alignment via stein path for cross-domain cold-start recommendation. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 19223–19234, 2021.

[Liu *et al.*, 2022a] Weiming Liu, Xiaolin Zheng, Mengling Hu, and Chaochao Chen. Collaborative filtering with attribution alignment for review-based non-overlapped cross domain recommendation. In *Proceedings of the ACM web conference 2022*, pages 1181–1190, 2022.

[Liu *et al.*, 2022b] Weiming Liu, Xiaolin Zheng, Jiajie Su, Mengling Hu, Yanchao Tan, and Chaochao Chen. Exploiting variational domain-invariant user embedding for partially overlapped cross domain recommendation. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, pages 312–321, 2022.

[Liu *et al.*, 2023a] Weiming Liu, Chaochao Chen, Xinting Liao, Mengling Hu, Jianwei Yin, Yanchao Tan, and Longfei Zheng. Federated probabilistic preference distribution modelling with compactness co-clustering for privacy-preserving multi-domain recommendation. In *IJCAI*, pages 2206–2214, 2023.

[Liu *et al.*, 2023b] Weiming Liu, Xiaolin Zheng, Chaochao Chen, Jiajie Su, Xinting Liao, Mengling Hu, and Yanchao Tan. Joint internal multi-interest exploration and external domain alignment for cross domain sequential recommendation. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 383–394. ACM, 2023.

[Liu *et al.*, 2023c] Weiming Liu, Xiaolin Zheng, Jiajie Su, Longfei Zheng, Chaochao Chen, and Mengling Hu. Contrastive proxy kernel stein path alignment for cross-domain

cold-start recommendation. *IEEE Trans. Knowl. Data Eng.*, 35(11):11216–11230, 2023.

[Ma *et al.*, 2024] Qiyao Ma, Xubin Ren, and Chao Huang. Xrec: Large language models for explainable recommendation. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 391–402, 2024.

[McAuley *et al.*, 2015] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 43–52, 2015.

[Ren *et al.*, 2024] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. Representation learning with large language models for recommendation. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pages 3464–3475, 2024.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, pages 452–461, 2009.

[Rong *et al.*, 2020] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[Sarlin *et al.*, 2020] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 4937–4946, 2020.

[Tao *et al.*, 2023] Zhulin Tao, Xiaohao Liu, Yewei Xia, Xiang Wang, Lifang Yang, Xianglin Huang, and Tat-Seng Chua. Self-supervised learning for multimedia recommendation. *IEEE Trans. Multim.*, 25:5107–5116, 2023.

[Vajjala *et al.*, 2024] Ajay Krishna Vajjala, Dipak Falgun Meher, Ziwei Zhu, and David S. Rosenblum. Cross-domain recommendation meets large language models. *CoRR*, abs/2411.19862, 2024.

[Wahab *et al.*, 2022] Omar Abdel Wahab, Gaith Rjoub, Jamal Bentahar, and Robin Cohen. Federated against the cold: A trust-based federated learning approach to counter the cold start problem in recommendation systems. *Information Sciences*, 601:189–206, 2022.

[Wang *et al.*, 2019] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 165–174. ACM, 2019.

[Wei *et al.*, 2024] Chuyuan Wei, Chuanhao Hu, Chang-Dong Wang, and Shuqiang Huang. Time-aware multi-behavior contrastive learning for social recommendation. *IEEE Trans. Ind. Informatics*, 20(4):6424–6435, 2024.

[Wu *et al.*, 2022] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Trans. Knowl. Data Eng.*, 34(10):4753–4766, 2022.

[Xiao *et al.*, 2023] Xuanji Xiao, Huaqiang Dai, Qian Dong, Shuzi Niu, Yuzhen Liu, and Pei Liu. Incorporating social-aware user preference for video recommendation. In *International Conference on Web Information Systems Engineering*, volume 14306, pages 544–558. Springer, 2023.

[Xu *et al.*, 2024] Yuanbo Xu, En Wang, Yongjian Yang, and Hui Xiong. GS-RS: A generative approach for alleviating cold start and filter bubbles in recommender systems. *IEEE Trans. Knowl. Data Eng.*, 36(2):668–681, 2024.

[Yu *et al.*, 2023] Penghang Yu, Zhiyi Tan, Guanming Lu, and Bing-Kun Bao. Multi-view graph convolutional network for multimedia recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, pages 6576–6585, 2023.

[Zhang *et al.*, 2021] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. Mining latent structures for multimedia recommendation. In *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*, pages 3872–3880, 2021.

[Zhang *et al.*, 2023] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Mengqi Zhang, Shu Wu, and Liang Wang. Latent structure mining with contrastive modality fusion for multimedia recommendation. *IEEE Trans. Knowl. Data Eng.*, 35(9):9154–9167, 2023.

[Zhou and Shen, 2023] Xin Zhou and Zhiqi Shen. A tale of two graphs: Freezing and denoising graph structures for multimodal recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, pages 935–943, 2023.

[Zhou *et al.*, 2023a] Hongyu Zhou, Xin Zhou, Zhiwei Zeng, Lingzi Zhang, and Zhiqi Shen. A comprehensive survey on multimodal recommender systems: Taxonomy, evaluation, and future directions. *CoRR*, abs/2302.04473, 2023.

[Zhou *et al.*, 2023b] Xin Zhou, Donghui Lin, Yong Liu, and Chunyan Miao. Layer-refined graph convolutional networks for recommendation. In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023*, pages 1247–1259, 2023.

[Zhou *et al.*, 2023c] Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. Bootstrap latent representations for multi-modal recommendation. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 845–854, 2023.