

The Best is Yet to Come: Graph Convolution in the Testing Phase for Multimodal Recommendation

Jinfeng Xu
jinfeng@connect.hku.hk
The University of Hong Kong
HongKong SAR, China

Zheyu Chen
zheyu.chen@connect.polyu.hk
The Hong Kong Polytechnic
University
HongKong SAR, China

Shuo Yang
shuoyang.ee@gmail.com
The University of Hong Kong
HongKong SAR, China

Jinze Li
lijinze-hku@connect.hku.hk
The University of Hong Kong
HongKong SAR, China

Edith C. H. Ngai*
chngai@eee.hku.hk
The University of Hong Kong
HongKong SAR, China

Abstract

The efficiency and scalability of graph convolution networks (GCNs) in training recommender systems remain critical challenges, hindering their practical deployment in real-world scenarios. In the multimodal recommendation (MMRec) field, training GCNs requires more expensive time and space costs and exacerbates the gap between different modalities, resulting in sub-optimal recommendation accuracy. This paper critically points out the inherent challenges associated with adopting GCNs during the training phase in MMRec, revealing that GCNs inevitably create unhelpful and even harmful pairs during model optimization and isolate different modalities. To this end, we propose FastMMRec, a highly efficient multimodal recommendation framework that deploys graph convolutions exclusively during the testing phase, bypassing their use in training. We demonstrate that adopting GCNs solely in the testing phase significantly improves the model's efficiency and scalability while alleviating the modality isolation problem often caused by using GCNs during the training phase. We conduct extensive experiments on three public datasets, consistently demonstrating the performance superiority of FastMMRec over competitive baselines while achieving efficiency and scalability.

CCS Concepts

• Information systems → Recommender systems;

Keywords

Recommender System, Multimedia

ACM Reference Format:

Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, and Edith C. H. Ngai. 2025. The Best is Yet to Come: Graph Convolution in the Testing Phase for Multimodal

**Corresponding authors

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

MM '25, October 27–31, 2025, Dublin, Ireland

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-2035-2/2025/10
<https://doi.org/10.1145/3746027.3755781>

Recommendation. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25)*, October 27–31, 2025, Dublin, Ireland. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3746027.3755781>

1 Introduction

Multimodal recommendation (MMRec) plays a pivotal role in e-commerce and content-sharing platforms, encompassing a amount of web multimedia content, including descriptions and images [37, 56]. Such capabilities allow them to discern users' preferences across different modalities accurately. Several recent studies incorporate multimodal content into multimedia recommendation systems. For example, VBPR [14] expands the matrix decomposition framework to accommodate item modality features. ACF [4] innovates with a hierarchically structured attention network designed to discern user preferences at the component level. Improving the performance of recommendation models with Graph Convolutional Networks (GCNs) has gained widespread attention [15, 33–35, 38]. More recently, models such as MMGCN [31] and GRCN [30] employ GCNs to integrate modality information into message-passing processes, thereby enhancing the inference of user and item representations. To further explore the rich multimodal information of items, LATTICE [48] and FREEDOM [56] construct item-item graphs to aggregate semantically similar items. Despite the notable advancements in graph-based MMRec models, they encounter fundamental challenges [12, 43] related to efficiency and scalability. These challenges stem primarily from the computationally intensive message-passing mechanisms of graph convolution, which are integral to the prevailing training paradigms of graph-based recommendation systems. The deployment of these models on large-scale graphs in real-world applications further exacerbates these challenges, as both time and computational complexity increase exponentially with the growing number of users and items. To make models scalable for real-world deployment, research focuses on two perspectives:

- **Perspective 1:** Extensive studies have been devoted to designing GCNs with complexity that is approximately linear or sublinear for the size of the data [2, 25], with a wide range of research focusing on sampling methods. Sampling-based methods lower the computation and memory requirements of GCNs by using a mini-batch training strategy on GCNs, which samples a limited number of neighbors for target nodes in a node-wise [5, 9, 12],

layer-wise [3, 58], or subgraph-wise [46, 47] manner. However, sampling-based methods inevitably omit a large number of neighbors for aggregation, resulting in large random errors.

- **Perspective 2:** Extensive studies show that simple MLPs as the initialization of graph model [13, 40] or trained with contrastive learning [16, 40, 41], knowledge distillation [50] demonstrate competitive performance compared with GCN models as long as they share an equivalent weight space.

Perspective 1 verifies the importance of a complete graph structure for GCN, and **Perspective 2** explores viable alternatives to GCNs. Therefore, we naturally raise a meaningful and significant question:

What do GCNs actually do during Training?

To answer this problem, we analyze the impact of GCNs on the model during training in Section 3. Then we point out two major challenges posed by employing GCNs in the training phase, including **GCNs inevitably create unhelpful or even harmful positive and negative pairs during model optimization.** and **C2. GCNs isolate different modalities, resulting in sub-optimal recommendation performance.** We further empirically validate these observations. Drawing on our investigation, we contend that the aggregation of neighbor nodes facilitates the representational enhancement attributed to GCNs. However, this aggregation process inherently introduces challenges **C1** and **C2** during model training. Consequently, we critically point out that only adopting GCNs during the testing phase can enjoy the representational enhancement capabilities of GCNs and effectively circumvent the associated training challenges.

Based on the above findings, we propose an efficient MMRec framework, FastMMRec, which deploys graph convolutions exclusively during the testing phase, bypassing their use in training. FastMMRec can effectively address the scalability problem caused by deploying GCNs during the training phase. Specifically, adopting GCNs exclusively during the testing phase not only prevents constructing useless and even harmful positive and negative pairs and prevents isolation between modalities, but it also retains the representation enhancement capabilities of GCNs through the aggregation of neighboring nodes. To achieve satisfactory performance, we adopt a tailored item-item graph enhancement during the training phase and provide a theoretical analysis to verify that adopting item-item graph enhancement will not lead to the same challenges as the GCNs. We detail the training phase and testing phase implementation of our FastMMRec in Section 4.

2 Preliminary

We conceptualize the user-item interaction graph as $\mathcal{G} = (\mathcal{U}, \mathcal{I}, \mathcal{E})$, where \mathcal{U} and \mathcal{I} denote the collections of users and items, respectively, and \mathcal{E} represents the set of interactions. An edge $(u, i) \in \mathcal{E}$ indicates a user u has interacted with an item i . The number of edges is denoted by $|\mathcal{E}|$. To enrich the user-item interaction graph \mathcal{G} with diverse modalities, we introduce modality-specific item embedding i^m for each item i belonging to the set of modalities \mathcal{M} . For user and item embedding, $\mathbf{E}_{u^m} \in \mathbb{R}^{d \times |\mathcal{U}|}$ represents the user's randomly initialized embedding, and $\mathbf{E}_{i^m} \in \mathbb{R}^{d \times |\mathcal{I}|}$ represents item initialized embedding with modality m , extracted by pre-trained encoders. d signifies the dimensionality of these features. Formally,

given an MMRec model denoted as $f(\cdot)$:

$$s_{u,i} = f(e_u, \{e_{i^m} | m \in \mathcal{M}\} | \Theta), \quad (1)$$

where $\Theta \in \mathbb{R}^d$ denotes the model parameters of $f(\cdot)$. Here, e_u and e_{i^m} denote embeddings of user u and item i (with modality m), respectively. The predicted score $s_{u,i}$ indicates user u 's preference for item i , with higher scores reflecting greater interest.

3 Investigation

In this section, we first investigate the impact of graph convolution during the training phase, observing that it spreads the optimization of each node in the loss function to its neighboring nodes. We then identify the first challenge of GCNs that has been overlooked in prior MMRec work: **C1. GCNs inevitably create unhelpful or even harmful positive and negative pairs during model optimization.** Additionally, we empirically reveal a second challenge in MMRec scenarios: **C2. GCNs isolate different modalities, resulting in sub-optimal recommendation performance.**

3.1 What do GCNs actually do during Training?

Existing studies [26, 31, 48, 53, 56] in the MMRec use LightGCN, a lightweight GCN that removes the activation functions and feature transformations of vanilla GCN for each modality m . Formally:

$$e_{u^m}^{(l)} = \sum_{\tilde{i} \in N(u)} \frac{e_{\tilde{i}^m}^{(l-1)}}{\sqrt{|N(u)||N(\tilde{i})|}}, \quad e_{i^m}^{(l)} = \sum_{\tilde{u} \in N(i)} \frac{e_{\tilde{u}^m}^{(l-1)}}{\sqrt{|N(i)||N(\tilde{u})|}}, \quad (2)$$

where $N(\cdot)$ refers to the set of items or users that interact with user u and item i , l is the layer number. For the basic Matrix Factorization (MF) model, the similarity $s_{u,i}^m$ for modality m between any user u and item i can be defined as:

$$s_{u,i}^m = e_{u^m}^\top e_{i^m}. \quad (3)$$

For a one-layer LightGCN, we unfold the calculation of the similarity $s_{u,i}^m$ for modality m between any user u and item i as follows:

$$\begin{aligned} s_{u,i}^m &= (e_{u^m} + \sum_{\tilde{i} \in N(u)} \frac{e_{\tilde{i}^m}}{\sqrt{|N(u)||N(\tilde{i})|}})^\top (e_{i^m} + \sum_{\tilde{u} \in N(i)} \frac{e_{\tilde{u}^m}}{\sqrt{|N(i)||N(\tilde{u})|}}) \\ &= \underbrace{e_{u^m}^\top e_{i^m}}_{\text{Node with Node}} + \underbrace{\sum_{\tilde{u} \in N(i)} \frac{e_{u^m}^\top e_{\tilde{u}^m}}{\sqrt{|N(i)||N(\tilde{u})|}} + \sum_{\tilde{i} \in N(u)} \frac{e_{i^m}^\top e_{\tilde{i}^m}}{\sqrt{|N(u)||N(\tilde{i})|}}}_{\text{Node with Neighbors}} \\ &\quad + \underbrace{\sum_{\tilde{i} \in N(u)} \sum_{\tilde{u} \in N(i)} \frac{e_{\tilde{i}^m}^\top e_{\tilde{u}^m}}{\sqrt{|N(u)||N(\tilde{i})||N(i)||N(\tilde{u})|}}}_{\text{Neighbors with Neighbors}}, \end{aligned} \quad (4)$$

where the final score $s_{u,i}$ is aggregated by $s_{u,i} = \text{Aggr}(s_{u,i}^m | m \in \mathcal{M})$. We divide this function into three parts: **Node with Node**, **Node with Neighbors**, and **Neighbors with Neighbors**. The first part, 'Node with Node' corresponds to the basic interaction mechanism in matrix factorization (MF)-based models. The other two parts, 'Node with Neighbors' and 'Neighbors with Neighbors' reflect the effects of GCNs.

To further analyze what happened in the model optimization process, we first briefly introduce Bayesian Personalized Ranking

(BPR) loss [22]. Essentially, BPR aims to widen the predicted preference margin between positive and negative items for each triplet $(u, p, n) \in \mathcal{D}$, where \mathcal{D} denotes the training set. The positive item p refers to the one with which the user u has interacted, while the negative item n has been randomly chosen from the set of items that the user u has not interacted with. Formally:

$$\mathcal{L}_{bpr} = \sum_{(u,p,n) \in \mathcal{D}} -\log(\sigma(s_{u,p} - s_{u,n})), \quad (5)$$

where $s_{u,p}$ and $s_{u,n}$ are the ratings of user u to the positive item p and negative item n . $\sigma(\cdot)$ is the Sigmoid function. For the MF-based models, the BPR loss function directly pulls close the positive pairs, while pushing away the negative pairs, formally:

$$\mathcal{L}_{bpr_{mf}} = \sum_{(u,p,n) \in \mathcal{D}} -\text{Aggr}(e_{u^m}^\top e_{p^m} - e_{u^m}^\top e_{n^m}), \quad (6)$$

where we simplify the Sigmoid function and log calculator. For GCN-based models, beyond merely focusing on node pairs, the BPR loss function additionally brings each node and its neighbors closer to the neighbors of its positive node while pushing each node and its neighbors away from the neighbors of its negative node.

We mathematically divided the loss function as:

$$\begin{aligned} \mathcal{L}_{bpr_{gcn}} = & \sum_{(u,p,n) \in \mathcal{D}} -\text{Aggr}(e_{u^m}^\top e_{p^m} - e_{u^m}^\top e_{n^m}) \\ & \underbrace{\quad}_{\text{Node } u \text{ with Node } i} \\ & + \underbrace{\sum_{\tilde{u}_p \in N(p)} \frac{e_{u^m}^\top e_{\tilde{u}_p^m}}{\sqrt{|N(p)||N(\tilde{u}_p)|}} - \sum_{\tilde{u}_n \in N(n)} \frac{e_{u^m}^\top e_{\tilde{u}_n^m}}{\sqrt{|N(n)||N(\tilde{u}_n)|}}}_{\text{Node } u \text{ with Neighbors of } i} \\ & + \underbrace{\sum_{\tilde{i} \in N(u)} \frac{e_{p^m}^\top e_{\tilde{i}^m}}{\sqrt{|N(u)||N(\tilde{i})|}} - \sum_{\tilde{i} \in N(u)} \frac{e_{n^m}^\top e_{\tilde{i}^m}}{\sqrt{|N(u)||N(\tilde{i})|}}}_{\text{Node } i \text{ with Neighbors of } u} \\ & + \underbrace{\sum_{\tilde{i} \in N(u)} \sum_{\tilde{u}_p \in N(p)} \frac{e_{\tilde{i}^m}^\top e_{\tilde{u}_p^m}}{\sqrt{|N(u)||N(\tilde{i})||N(p)||N(\tilde{u}_p)|}}}_{\text{Neighbors of } u \text{ with Neighbors of } p} \\ & - \underbrace{\sum_{\tilde{i} \in N(u)} \sum_{\tilde{u}_n \in N(n)} \frac{e_{\tilde{i}^m}^\top e_{\tilde{u}_n^m}}{\sqrt{|N(u)||N(\tilde{i})||N(n)||N(\tilde{u}_n)|}}}_{\text{Neighbors of } u \text{ with Neighbors of } n}. \end{aligned} \quad (7)$$

This equation clearly describes how GCN impacts the model optimization. We detailed each part as follows:

- **P1. Node u with Node i :** this part directly pulls close the positive pairs while pushing away the negative pairs.
- **P2. Node u with Neighbors of i :** this part pulls user closer to other users who interact with its positive item while pushing away user with other users who interact with its negative item.
- **P3. Node i with Neighbors of u :** this part pulls positive item closer to other items that interact with its user while pushing away negative item with other items that interact with its user.

- **P4. Neighbors of u with Neighbors of p :** this part pulls items that user interacts with closer to users with whom positive item interacts.
- **P5. Neighbors of u with Neighbors of n :** this part pushes items the user interacts with away from users with whom negative item interacts.

P1 reflects the basic assumptions of the BPR loss in the recommendation system. Based on real-world observations, we argue that **P2-P5** introduce assumptions that are not always beneficial for model optimization. For **P2**, users $\tilde{u}_n \in N(n)$ who have purchased item n , which user u has not bought, do not necessarily have completely different preferences from u . For **P3**, the fact that user u has not purchased item n does not mean that n has attributes completely dissimilar to the items $\tilde{i} \in N(u)$ that u has purchased. For **P4**, users $\tilde{u}_p \in N(p)$ who have purchased item p , which user u also bought, do not necessarily tend to purchase other items $\tilde{i} \in N(u)$ that u has purchased. For **P5**, users $\tilde{u}_n \in N(n)$ who purchased item n , which u has not bought, are not necessarily disinclined to purchase other items $\tilde{i} \in N(u)$ that u has bought.

We further validate our observation through empirical experiments. Specifically, we select MMGCN [31], a widely used GCN-based MMRc model, as the backbone for our study¹, and design two variants: MMGCN_{train} and MMGCN_{test}. MMGCN_{train} continues to use GCN during the training phase, whereas MMGCN_{test} only adopts GCN during the testing phase, which aims to preserve the powerful representations learned from neighbor aggregation while avoiding the negative influence of bad pairs during the model optimization. We conduct comprehensive experiments² on these two variants, testing different GCN layers across three widely used datasets. As Figure 1 shows, MMGCN_{test} outperforms MMGCN_{train} across all datasets and significantly reduces training time. Therefore, we empirically confirm the negative influence of challenge **C1** and demonstrate that using GCNs exclusively during the testing phase can effectively address this challenge.

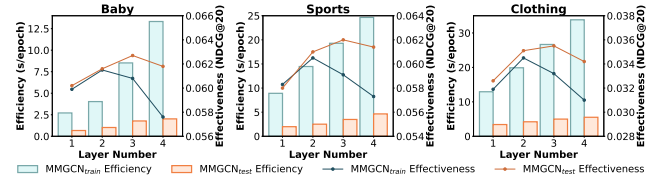


Figure 1: Effectiveness and Efficiency study.

In addition to the challenge **C1** posed by GCNs within each modality, we further examine how GCNs affect the similarity between different modalities. To analyze this, we revisit Equations 6 and 7. For MF in Equation 6, the model learns specific weights for each user's modalities by directly optimizing the nodes within each modality. Conversely, for GCN in Equation 7, the aggregation of neighbor nodes limits the model's ability to effectively learn specific weights for each modality. Aggregating too many neighbor nodes

¹To better support our investigation, we provide additional experiments on other advanced MMRc models in Appendix A.2 in supplementary materials.

²We use NDCG@20 to evaluate performance and seconds per epoch (s/epoch) to measure efficiency. Details of all metrics and datasets are provided in Section 5.

inevitably reduces the node's unique information, which in turn diminishes personalization [19, 51, 55]. From the overall distribution perspective, this could result in each modality being rigidly tied to its inherent features, consequently leading to modality isolation [1, 42].

Therefore, we point out the second challenge faced by GCNs in MMRec: **C2. GCNs isolate different modalities, resulting in sub-optimal recommendation performance.**

We also validate this observation empirically. Specifically, we measure the similarity between different modalities as follows:

$$S = \sum_{o \in (\mathcal{U} \cup \mathcal{I})} \frac{S^o}{|\mathcal{U}| + |\mathcal{I}|}, \quad S^o = \frac{(e_o^v)^\top e_o^t}{\|e_o^v\| \|e_o^t\|}, \quad (8)$$

where e_o^v and e_o^t are visual and textual representations for node o . We conduct experiments on three public datasets using the MMGCN variants MMGCN_{train} and MMGCN_{test} to analyze their performance and modality alignment. As shown in Table 1, the similarity between the visual and textual modalities in MMGCN_{train} is significantly lower than that in MMGCN_{test} . This indicates that adopting GCNs during training inevitably isolates different modalities, leading to sub-optimal recommendation performance. This finding provides strong evidence for the negative influence of challenge C2. To further support our findings, we report the similarity S between the visual and textual embeddings for other advanced MMRec models in Appendix A.2 in supplementary materials.

Table 1: Similarity S between visual and textual embeddings.

Variants	Baby	Sports	Clothing
MMGCN_{train}	0.2207	0.2008	0.2129
MMGCN_{test}	0.3722	0.3261	0.3204

Compared to MMGCN_{train} , MMGCN_{test} achieves significant improvements in both efficiency and effectiveness, successfully mitigating these two challenges. However, the state-of-the-art model has a more complex architecture than MMGCN. Consequently, we propose an efficient and high-performing MMRec model, FastMMRec, which adopts GCNs exclusively during the testing phase to achieve superior performance compared to competitive models.

4 FastMMRec

In this section, we detail our FastMMRec for both the training and testing phases. The architecture is depicted in Figure 2.

4.1 Training Phase

To further exploit the rich modality information of items, item-item graphs have been widely used in MMRec [28, 45, 53, 56] to aggregate and explore relationships and commonalities among items, achieving satisfactory recommendation performance. We first construct modality-specific item-item graphs using the raw features of each modality (e.g., visual and textual) and then build a unified item-item graph by aggregating all modality-specific graphs. Inspired by previous work [56], we freeze the similarity graphs during the training phase to reduce computational costs. The pairwise similarity

between all items for each modality is calculated as follows:

$$S_{i,j}^m = \frac{(e_{im})^\top e_{jm}}{\|e_{im}\| \|e_{jm}\|}. \quad (9)$$

We retain only the top- k neighbors with the highest similarity scores to capture the most relevant features:

$$\tilde{S}_{i,j}^m = \begin{cases} S_{i,j}^m & \text{if } S_{i,j}^m \in \text{top-}k(S_{i,p}^m | p \in \mathcal{I}) \\ 0 & \text{otherwise} \end{cases}, \quad (10)$$

where $\tilde{S}_{i,j}^m$ denotes the edge weight between item i and item j within modality m . $S_{i,p}^m | p \in \mathcal{I}$ represents the neighbor scores for the item i . To mitigate the issues of gradient explosion or vanishing, we normalize the similarity adjacency matrices as follows:

$$\tilde{S}^m = (\mathcal{D}^m)^{-\frac{1}{2}} \tilde{S}^m (\mathcal{D}^m)^{-\frac{1}{2}}, \quad (11)$$

where \mathcal{D}^m is the diagonal degree matrix of \tilde{S}^m . Then, we further build a unified item-item graph by aggregating all modality-specific item-item graphs:

$$\tilde{S} = \sum_{m \in \mathcal{M}} \alpha_m \tilde{S}^m, \quad (12)$$

where α_m is a trainable weighted parameter. A unified item-item graph for all modalities can better extract latent relationships across different modalities. Then, we attentively fuse representations of all modalities of users and items, respectively:

$$\mathbf{E}_u = \text{Con}(\alpha_m \mathbf{E}_{u^m} | m \in \mathcal{M}), \quad \mathbf{E}_i = \text{Con}(\alpha_m \mathbf{E}_{i^m} | m \in \mathcal{M}), \quad (13)$$

where $\text{Con}(\cdot)$ denotes concatenation operation. Then, we aggregate multi-hop neighbors to enhance item representations.

$$\mathbf{E}_i = \mathbf{E}_i + \mathbf{E}_i(\tilde{S})^{L_i}, \quad (14)$$

where L_i is the number of aggregation hop. To preserve the personalization of each item's representation, we add the enhanced representation to the original representation. This strategy ensures that the personalization of each item is maintained while enhanced by aggregating neighbors' representations.

For model optimization, we compute the inner product of user and item representations to calculate predicted scores and adopt the BPR loss function:

$$\mathcal{L}_{bpr} = \sum_{(u,p,n) \in \mathcal{D}} -\log(\sigma(e_u^\top e_p - e_u^\top e_n)) + \lambda \|\Theta\|_2^2, \quad (15)$$

where σ is the Sigmoid function. λ is a balancing hyper-parameter for regularization terms and Θ denotes model parameters.

Analysis: Is item-item graph inevitably build useless and even harmful positive and negative pairs during optimization?

We unroll the calculation of the similarity $s_{u,i}$ between any user u and item i with the one-hop item-item graph enhancement:

$$s_{u,i} = (e_u)^\top \left(e_i + \underbrace{\sum_{j | \tilde{S}_{i,j} \neq 0} \frac{e_j}{k}}_{\text{Node with Node } i} \right) = \underbrace{e_u^\top e_i}_{\text{Node } u \text{ with Node } i} + \underbrace{\sum_{j | \tilde{S}_{i,j} \neq 0} \frac{e_u^\top e_j}{k}}_{\text{Node } u \text{ with Neighbors of } i}. \quad (16)$$

We divided this function into two parts: **interactions between Node u and Node i** and **interactions between Node u and the neighbors of Node i** . For item-item graph enhancement, the neighbors of item i are other items rather than users, which distinguishes

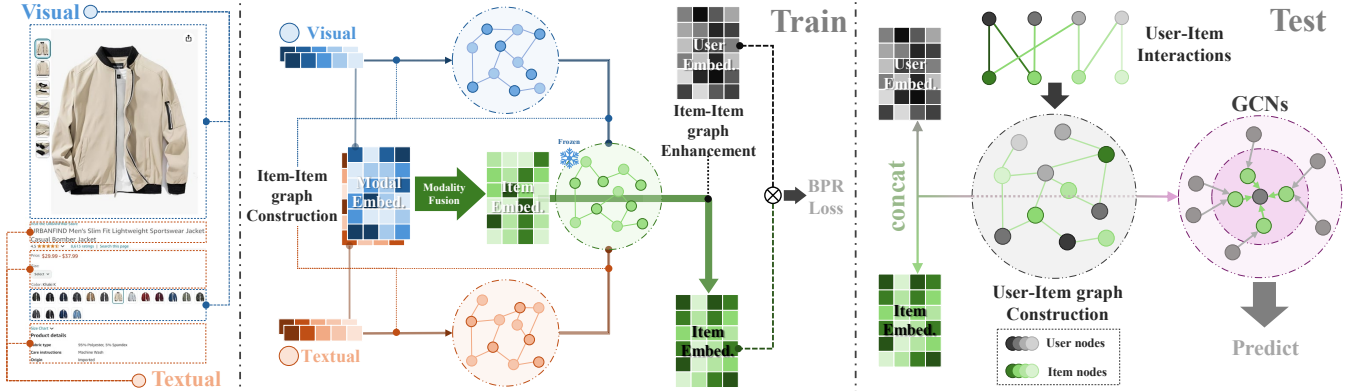


Figure 2: The overall framework of the proposed multimodal recommendation model (FastMMRec).

this approach from traditional GCNs. Next, we mathematically analyze the impact of item-item graph enhancement on BPR loss:

$$\begin{aligned} \mathcal{L}_{bprFastMMRec} = & \sum_{(u,p,n) \in \mathcal{D}} -(\underbrace{e_u^\top e_p - e_u^\top e_n}_{\text{Node } u \text{ with Node } i}) \\ & + \underbrace{\sum_{\hat{p} | S_{p,\hat{p}}} \frac{e_u^\top e_{\hat{p}}}{k} - \sum_{\hat{n} | S_{n,\hat{n}}} \frac{e_u^\top e_{\hat{n}}}{k}}_{\text{Node } u \text{ with Neighbors of } i}. \end{aligned} \quad (17)$$

This equation clearly describes how GCN impacts model optimization. We detailed each part as follows:

- **P1. Node u with Node i :** this part directly pulls close the positive pairs while pushing away the negative pairs.
- **P2. Node u with Neighbors of i :** this part pulls the item closer to other items that are semantically similar to positive items, while pushing the item away from other items that are semantically similar to negative items.

Since the neighbors in an item-item graph are semantically related and consist solely of items, they do not encounter the inherent semantic discrepancies between items and users that are observed in GCNs. As a result, using an item-item graph during training does not create irrelevant or harmful positive and negative pairs during model optimization. Furthermore, it leverages the rich multimodal information of items to enhance the model's robustness. We empirically validate this observation in Section 5.3.

4.2 Testing Phase

Based on the analysis in Section 3, we only adopt GCNs during the testing phase to address the challenges associated with employing GCNs during the training phase and to enhance model efficiency. During the testing phase, we utilize GCNs, and the predicted score $s_{u,i}$ between user u and item i is calculated as:

$$e_u^{(l)} = \sum_{\tilde{i} \in N(u)} \frac{e_{\tilde{i}}^{(l-1)}}{\sqrt{|N(u)||N(\tilde{i})|}}, \quad e_i^{(l)} = \sum_{\tilde{u} \in N(i)} \frac{e_{\tilde{u}}^{(l-1)}}{\sqrt{|N(i)||N(\tilde{u})|}}, \quad (18)$$

$$\tilde{e}_u = \sum_{l=1}^L e_u^{(l)}, \quad \tilde{e}_i = \sum_{l=1}^L e_i^{(l)}, \quad s_{u,i} = \tilde{e}_u^\top \tilde{e}_i. \quad (19)$$

Adopting GCNs in the testing leverages neighbor representations while avoiding the challenges of using GCNs during training.

Table 2: Statistics of three experimented datasets with multimodal item Visual(V) and Textual(T) contents.

Dataset	Baby		Sports		Clothing	
' Modality	V	T	V	T	V	T
Embed Dim	4096	384	4096	384	4096	384
User	19445		35598		39387	
Item	7050		18357		23033	
Interaction	160792		296337		278677	
Sparsity	99.88%		99.95%		99.97%	

5 Evaluation

5.1 Experiment Settings

5.1.1 Datasets. The experiments are conducted on three real-world datasets from the Amazon [21]: Baby, Sports, and Clothing, each encompassing visual and textual modalities for every item. Consistent with most previous studies [33, 56, 57], we apply the 5-core setting to filter users and items within each dataset. We follow the same setting mentioned in [54], which extracts 4096-dimensional visual features and 384-dimensional textual features via pre-trained encoders. Table 2 presents the statistics of these datasets. For each dataset, we randomly split the historical interactions using an 8:1:1 ratio for training, validation, and testing.

5.1.2 Baselines. To verify the effectiveness of our proposed FastMMRec, we compare FastMMRec with a variety of baselines, including conventional recommendation models (**MF-BPR** [22], **LightGCN** [15], **SimGCL** [44], and **LayerGCN** [55]) and multimodal recommendation models (**VBPR** [14], **MMGCN** [31], **DualGNN** [26], **LATTICE** [48], **FREEDOM** [56], **SLMRec** [23], **BM3** [57], **MMSSL** [27], **LGMRec** [11], and **DiffMM** [17]). Details of baselines are presented in Appendix A.1 in supplementary materials.

5.1.3 Metrics. To evaluate the top-K recommendation task performance fairly, we adopt two widely-used metrics: Recall and NDCG. We report the average metrics of all users in the test dataset under Recall@10 (R@10), Recall@20 (R@20), NDCG@10 (N@10), and NDCG@20 (N@20).

Table 3: Performance comparison of baselines on different datasets in terms of Recall@K and NDCG@K.

Baseline	Baby				Sports				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
MF-BPR	0.0357	0.0575	0.0192	0.0249	0.0432	0.0653	0.0241	0.0298	0.0187	0.0279	0.0103	0.0126
LightGCN	0.0479	0.0754	0.0257	0.0328	0.0569	0.0864	0.0311	0.0387	0.0340	0.0526	0.0188	0.0236
SimGCL	0.0513	0.0804	0.0273	0.0350	0.0601	0.0919	0.0327	0.0414	0.0356	0.0549	0.0195	0.0244
LayerGCN	0.0529	0.0820	0.0281	0.0355	0.0594	0.0916	0.0323	0.0406	0.0371	0.0566	0.0200	0.0247
VBPR	0.0423	0.0663	0.0223	0.0284	0.0558	0.0856	0.0307	0.0384	0.0281	0.0415	0.0158	0.0192
MMGCN	0.0378	0.0615	0.0200	0.0261	0.0370	0.0605	0.0193	0.0254	0.0218	0.0345	0.0110	0.0142
DualGNN	0.0448	0.0716	0.0240	0.0309	0.0568	0.0859	0.0310	0.0385	0.0454	0.0683	0.0241	0.0299
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	<u>0.0992</u>	0.0330	0.0424	0.0717	<u>0.1089</u>	0.0385	<u>0.0481</u>	<u>0.0628</u>	<u>0.0941</u>	<u>0.0341</u>	<u>0.0420</u>
SLMRec	0.0529	0.0775	0.0290	0.0353	0.0663	0.0990	0.0365	0.0450	0.0452	0.0675	0.0247	0.0303
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
MMSSL	0.0613	0.0971	0.0326	0.0420	0.0673	0.1013	0.0380	0.0474	0.0531	0.0797	0.0291	0.0359
LGMRec	<u>0.0639</u>	0.0989	<u>0.0337</u>	<u>0.0430</u>	<u>0.0719</u>	0.1068	<u>0.0387</u>	0.0477	0.0555	0.0828	0.0302	0.0371
DiffMM	0.0623	0.0975	0.0328	0.0411	0.0671	0.1017	0.0377	0.0458	0.0522	0.0791	0.0288	0.0354
FastMMRec	0.0667	0.1034	0.0357	0.0453	0.0768	0.1151	0.0415	0.0517	0.0674	0.0992	0.0366	0.0447
<i>p</i> -value	$2.21e^{-4}$	$8.33e^{-5}$	$1.01e^{-4}$	$1.85e^{-4}$	$4.60e^{-4}$	$3.11e^{-4}$	$5.28e^{-4}$	$5.51e^{-4}$	$4.94e^{-4}$	$2.89e^{-4}$	$5.42e^{-4}$	$4.67e^{-4}$
Improv.	4.38%	4.23%	5.93%	5.35%	6.82%	5.89%	7.24%	7.48%	7.32%	5.42%	7.33%	6.43%

5.1.4 Implementation Details. To ensure a fair comparison, we implement our FastMMRec and all the baselines using the MMRec library [54]. Specifically, all models are implemented in PyTorch, using the Adam optimizer [18] and Xavier initialization [10] with default parameters. We perform a complete grid search for all baselines to determine their optimal settings as described in their published papers. As for hyper-parameter settings on our FastMMRec, we perform a grid search on the item-item graph hop number L_i in $\{1, 2, 3\}$, the k of top- k item-item graph in $\{5, 10, 15, 20\}$, and the layer number L of GCN in $\{1, 2, 3, 4\}$. We empirically fix the learning rate with $1e^{-4}$ and regularization weight λ with $1e^{-3}$. To avoid the over-fitting problem, we set 20 as the early stopping epoch number. Following previous studies [37, 54], we utilize Recall@20 on the validation dataset as a metric to update the best record. Note that all models are evaluated on an RTX 3090 with 24GB memory.

5.2 Performance Comparison

Table 3 presents the evaluation results of the performance comparison. In this table, we highlight the optimal results in bold and underline the sub-optimal results for easy identification. We have the following key observations:

- **Performance superiority of our FastMMRec.** Our FastMMRec consistently outperforms all baselines across diverse datasets. This advantage is attributable to the fact that we adopt GCNs only during the testing phase. This approach avoids creating irrelevant or harmful positive and negative pairs during model optimization and prevents isolation between modalities.
- **Effectiveness of item-item graph enhancement.** Utilizing item-item graph enhancement significantly improves the performance of recommender systems. Models such as LATTICE, FREEDOM, MMSSL, LGMRec, and FastMMRec benefit significantly from this approach. This improvement is due to item-item graphs, which enhance item representations by aggregating semantically related neighbors and extracting rich multimodal information.

- **Importance of multimodal information.** Most multimodal recommendation models outperform conventional ones, highlighting the importance of incorporating multimodal information to learn user preferences and item properties.

Table 4: Ablation study on key components of FastMMRec in terms of Recall@20 and NDCG@20.

Dataset	Baby		Sports		Clothing	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
<i>w/o-item</i>	0.0947	0.0402	0.1035	0.0461	0.0899	0.0404
<i>test-item</i>	0.1013	0.0434	0.1119	0.0502	0.0963	0.0431
FastMMRec	0.1034	0.0453	0.1151	0.0517	0.0992	0.0447

5.3 Ablation Study

To validate the effectiveness of FastMMRec, we conduct experiments to justify the importance of key components. We design the following variants: 1) *w/o-item*, which removes the item-item graph entirely, and 2) *test-item*, which applies the item-item graph enhancement only during the testing phase instead of the training phase. The results in Table 4 provides following key conclusions:

- The performance of *w/o-item* drops significantly compared to FastMMRec, demonstrating the effectiveness of the item-item graph enhancement component.
- Comparing FastMMRec with *test-item*, we observe a clear performance advantage, reflecting the benefits of using item-item graph enhancement during model optimization.

5.4 Sparsity Study

We validate the effectiveness of FastMMRec under different levels of data sparsity. To assess its performance, we conduct experiments on sub-datasets derived from all three datasets, each with varying levels of data sparsity. We compare FastMMRec’s performance with

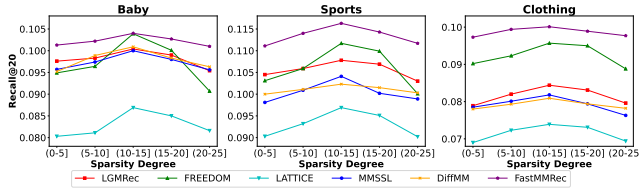
Table 5: Comparison of computational complexity on graph-based multimodal models.

Stage	MMGCN	LATTICE	MMSSL	FastMMRec
Graph Convolution	$O(2 \mathcal{M} L \mathcal{E} d/B)$	$O(2L \mathcal{E} d/B)$	$O(2 \mathcal{M} L \mathcal{E} d/B)$	-
Feature Mapping	$O(\sum_{m \in \mathcal{M}} I (d_m + d)d_h)$	$O(I ^3 + \sum_{m \in \mathcal{M}} I ^2 d_m + k I \log(I))$	$O(\sum_{m \in \mathcal{M}} I d_m d)$	$O(I d^2)$
Loss	$O(2dB)$	$O(2dB)$	$O((2 + \mathcal{M} \mathcal{U} I + 2 \mathcal{M})dB + \mathcal{M} \mathcal{U} I d_m B)$	$O(2dB)$

d_h denotes the dimension of the hidden layer in a two-layer MLP and k is the value of top- k neighbors in the item-item graph.

Table 6: Comparison of our FastMMRec against state-of-the-art baselines on model efficiency.

Dataset	Metrics	VBPR	MMGCN	DualGNN	LATTICE	FREEDOM	SLMRec	BM3	MMSSL	LGMRec	DiffMM	FastMMRec
Baby	Time (s/epoch)	0.55	4.09	5.63	3.20	2.57	2.07	1.93	6.31	4.19	9.45	0.61
	Memory (GB)	1.89	2.69	2.05	4.53	2.13	2.08	2.11	3.77	2.41	4.23	1.93
Sports	Time (s/epoch)	0.97	14.93	11.59	11.07	5.65	5.39	3.82	14.67	8.38	18.61	1.01
	Memory (GB)	2.71	3.91	2.81	19.93	3.34	3.04	3.58	5.34	3.67	5.99	2.79
Clothing	Time (s/epoch)	1.34	17.48	14.19	16.53	6.29	6.02	5.25	17.04	9.72	23.85	1.39
	Memory (GB)	3.02	4.24	3.02	28.22	4.15	3.40	4.13	5.81	4.81	6.54	3.11

**Figure 3: Sparsity degree analysis on three datasets.**

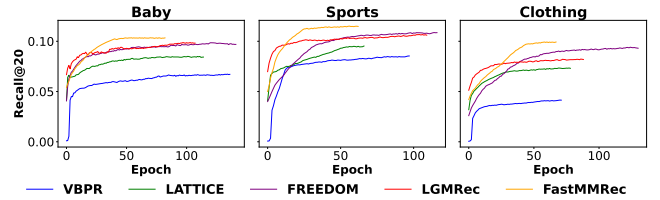
five competitive baselines: LATTICE, FREEDOM, MMSSL, LGMRec, and DiffMM. We categorize user groups based on the number of interactions in the training set, such as users with 0–5 interacted items in the first group. Figure 3 shows that FastMMRec consistently outperforms all baselines across datasets, confirming its robustness under different sparsity levels.

5.5 Efficiency Study

Our FastMMRec achieves surprising efficiency improvements compared to previous studies. We analyze its efficiency by complexity, convergence, and training time.

5.5.1 Complexity. Our FastMMRec achieves significant efficiency improvements over previous studies. We analyze the efficiency of our FastMMRec in terms of complexity, convergence, and training time. We analyze the computational complexity of FastMMRec and compare it with three advanced graph-based MMRec models (MMGCN, LATTICE, and MMSSL) in Table 5. We divide computational complexity into three major components: **Graph Convolution**, **Feature Mapping**, and **Loss**. **1) Graph Convolution.** MMGCN and MMSSL adopt LightGCN for each modality, with a computational complexity of $O(2|\mathcal{M}|L|\mathcal{E}|d/B)$, where $|\mathcal{M}|$ is the number of modalities, L is the number of layers in LightGCN, B is the batch size, d is the embedding dimension, and $|\mathcal{E}|$ is the number of edges in the graph. LATTICE adopts a single LightGCN for the fused modality, with a computational complexity of $O(2L|\mathcal{E}|d/B)$. For FastMMRec, we only adopt GCNs in the testing phase, eliminating all computational costs for graph convolution. **2) Feature Mapping.** MMGCN uses a two-layer MLP feature projection for each modality, with a complexity of $O(\sum_{m \in \mathcal{M}} |I|(d_m + d)d_h)$, where d_h is the hidden dimension and $|I|$ is the number of items. LATTICE

constructs an item-item graph from multimodal features, which involves $O(\sum_{m \in \mathcal{M}} |I|^2 d_m)$ to build the similarity matrix, $O(|I|^3)$ to normalize the matrix, and $O(k|I| \log(|I|))$ to retrieve the top- k most similar items, where k is the number of neighbors per item. MMSSL freezes the item-item graph during training, with a complexity of $O(|I|d_m d)$ per modality, resulting in a total complexity of $O(\sum_{m \in \mathcal{M}} |I|d_m d)$. FastMMRec also freezes the item-item graph during training and uses a single fused item-item graph, resulting in a total complexity of $O(|I|d^2)$. **3) Loss.** MMGCN, LATTICE, and FastMMRec use the vanilla BPR loss, with a complexity of $O(2dB)$. MMSSL, in addition to vanilla BPR loss ($O(2dB)$), includes generator loss ($O(|\mathcal{M}||\mathcal{U}||I|dB)$), discriminator loss ($O(|\mathcal{M}||\mathcal{U}||I|d_m B)$), and contrastive learning loss ($O(2|\mathcal{M}|dB)$). **FastMMRec entirely avoids the complex Graph Convolution module, exhibits lower complexity in the Feature Mapping module compared to existing methods, and only incurs the complexity of the vanilla BPR loss in the Loss module.**

**Figure 4: Convergence study in terms of Recall@20.**

5.5.2 Convergence. Figure 4 shows the training curves of our FastMMRec and the compared models (VBPR, LATTICE, FREEDOM, and LGMRec) on all three datasets as the number of iterations and epochs increases. We have the following findings:

- The faster convergence speed of FastMMRec is clearly evident, highlighting its advantage in training efficiency while maintaining superior recommendation accuracy. This suggests that adopting GCNs only during the testing phase facilitates faster convergence during model training.
- FastMMRec and VBPR achieve faster convergence speeds than graph-based models (LATTICE and FREEDOM), further confirming that GCNs often construct irrelevant or even harmful positive

and negative pairs during model optimization. By adopting GCNs only during the testing phase, FastMMRec addresses this challenge and leverages the representational benefits of GCNs.

5.5.3 Training time. We report the training time and memory usage of FastMMRec and baselines in Table 6. We make the following observations:

- **Training time:** FastMMRec demonstrates faster training speeds, while other graph-based models show a rapid increase in training time as dataset size grows. In contrast, FastMMRec scales approximately linearly with dataset size. This efficiency is due to the exclusive adoption of GCNs during the testing phase, effectively addressing the scalability challenges of graph-based models in real-world applications.
- **Memory:** FastMMRec uses less memory than all other graph-based models, attributable to FastMMRec constructing a single item-item graph and freezing it during the training phase.

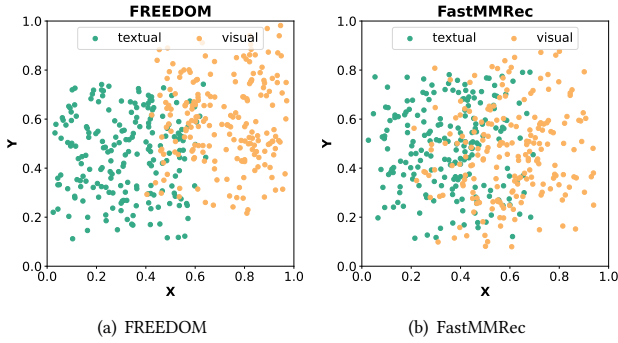


Figure 5: Distribution of visual and textual representations obtained by FREEDOM and FastMMRec on the Baby dataset.

5.6 Visualization

To further validate the advantages of FastMMRec in preventing the modality isolation problem, we perform the following analysis: We randomly select 200 items from the Baby dataset and apply the t-SNE [24] to project the item representations of FREEDOM and FastMMRec into a two-dimensional space. Upon analyzing the 2D feature distributions in Figure 5, we observe that the visual and textual feature distributions in FastMMRec are more similar to each other compared to FREEDOM. This similarity suggests that FastMMRec effectively mitigates the modality isolation problem.

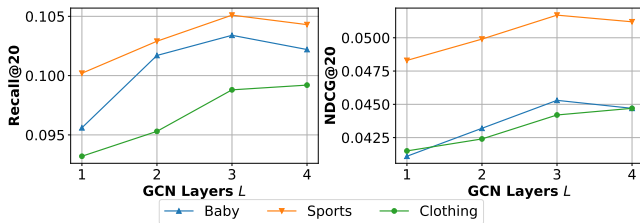


Figure 6: Effect of GCN layers L .

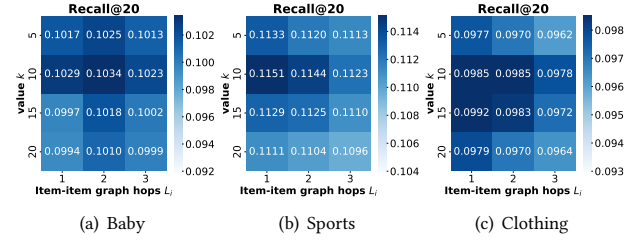


Figure 7: Effect of item-item graph hops L_i and k value.

5.7 Hyper-parameter Study

We examine the sensitivity of several important hyper-parameters of FastMMRec across different datasets.

- **GCN layers L :** We first investigate the impact of GCN depth by varying the number of message-passing layers L in $\{1, 2, 3, 4\}$. Figure 6 shows FastMMRec achieves its best performance with $L = 3$ or 4 . Note that existing MMRec models suffer from performance deterioration due to the over-smoothing problem when the number of GCN layers reaches 2 or 3. FastMMRec mitigates this problem by adopting GCNs only during the testing phase.
- **Item-item graph hops L_i and k value:** We empirically analyze the impact of the item-item graph structure by varying the number of hops L_i and the k value. Figure 7 shows the results of FastMMRec under different hops L_i and the k value on the Baby, Sports, and Clothing datasets. The suggested hops L_i are 2, 1, and 1 for the Baby, Sports, and Clothing datasets, respectively. The suggested k values are 10, 10, and 5 for the Baby, Sports, and Clothing datasets, respectively.

6 Related Work

Due to page limitations, we review recent works and their contributions in Appendix A.3.

7 Testing Phase Efficiency

Due to page limits, we provide an efficiency analysis of FastMMRec in the testing phase in Appendix A.4 in supplementary materials.

8 Conclusion

In this work, we reveal the inevitable challenges associated with employing GCNs during the training phase in MMRec. We propose a surprisingly efficient multimodal recommendation framework for adopting graph convolution in the testing phase (FastMMRec). We conduct extensive experiments on three public datasets, consistently demonstrating the effective and efficient superiority of FastMMRec over competitive baselines. This work not only provides novel and powerful paradigms but also pinpoints potentially new research directions for efficient and large-scale real-world MMRec.

Acknowledgments

This work was supported by the Hong Kong UGC General Research Fund no. 17203320 and 17209822, and the project grants from the HKU-SCF FinTech Academy.

References

- [1] Feiyu Chen, Junjie Wang, Yinwei Wei, Hai-Tao Zheng, and Jie Shao. 2022. Breaking isolation: Multimodal graph fusion for multimedia recommendation by edge-wise modulation. In *Proceedings of the 30th ACM International Conference on Multimedia*. 385–394.
- [2] Hao Chen, Yuanchen Bei, Qijie Shen, Yue Xu, Sheng Zhou, Wenbing Huang, Feiran Huang, Senzhang Wang, and Xiao Huang. 2024. Macro graph neural networks for online billion-scale recommender systems. In *Proceedings of the ACM on Web Conference 2024*. 3598–3608.
- [3] Jie Chen, Tengfei Ma, and Cao Xiao. 2018. FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling. In *International Conference on Learning Representations*.
- [4] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 335–344.
- [5] Jianfei Chen, Jun Zhu, and Le Song. 2018. Stochastic Training of Graph Convolutional Networks with Variance Reduction. In *International Conference on Machine Learning*. PMLR, 942–950.
- [6] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2019. Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 765–774.
- [7] Zheyu Chen, Jinfeng Xu, and Haibo Hu. 2025. Don't Lose Yourself: Boosting Multimodal Recommendation via Reducing Node-neighbor Discrepancy in Graph Convolutional Network. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.
- [8] Zheyu Chen, Jinfeng Xu, Yutong Wei, and Ziyue Peng. 2025. Squeeze and Excitation: A Weighted Graph Contrastive Learning for Collaborative Filtering. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2769–2773.
- [9] Weilin Cong, Rana Forsati, Mahmut Kandemir, and Mehrdad Mahdavi. 2020. Minimal variance sampling with provable guarantees for fast training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1393–1403.
- [10] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.
- [11] Zhiqiang Guo, Jianjun Li, Guohui Li, Chaoyang Wang, Si Shi, and Bin Ruan. 2024. LGMRec: Local and Global Graph Learning for Multimodal Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8454–8462.
- [12] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).
- [13] Xiaotian Han, Tong Zhao, Yozen Liu, Xia Hu, and Neil Shah. 2023. MLPInit: Embarrassingly Simple GNN Training Acceleration with MLP Initialization. In *The Eleventh International Conference on Learning Representations*.
- [14] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.
- [15] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.
- [16] Yang Hu, Haoxuan You, Zhecan Wang, Zhicheng Wang, Erjin Zhou, and Yue Gao. 2021. Graph-mlp: Node classification without message passing in graph. *arXiv preprint arXiv:2106.04051* (2021).
- [17] Yangqin Jiang, Lianghao Xia, Wei Wei, Da Luo, Kangyi Lin, and Chao Huang. 2024. DiffMM: Multi-Modal Diffusion Model for Recommendation. (2024).
- [18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [19] Meng Liu, Hongyang Gao, and Shuiwang Ji. 2020. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 338–348.
- [20] Shang Liu, Zhenzhong Chen, Hongyi Liu, and Xinghai Hu. 2019. User-video co-attention network for personalized micro-video recommendation. In *The world wide web conference*. 3020–3026.
- [21] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.
- [22] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
- [23] Zhulin Tao, Xiaohao Liu, Yewei Xia, Xiang Wang, Lifang Yang, Xianglin Huang, and Tat-Seng Chua. 2022. Self-supervised learning for multimedia recommendation. *IEEE Transactions on Multimedia* (2022).
- [24] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).
- [25] Lin Wang, Wenqi Fan, Jiatong Li, Yao Ma, and Qing Li. 2024. Fast graph condensation with structure-based neural tangent kernel. In *Proceedings of the ACM on Web Conference 2024*. 4439–4448.
- [26] Qifan Wang, Yinwei Wei, Jianhua Yin, Jianlong Wu, Xueming Song, and Liqiang Nie. 2021. Dualgnn: Dual graph neural network for multimedia recommendation. *IEEE Transactions on Multimedia* (2021).
- [27] Wei Wei, Chao Huang, Lianghao Xia, and Chuxu Zhang. 2023. Multi-Modal Self-Supervised Learning for Recommendation. In *Proceedings of the ACM Web Conference 2023*. 790–800.
- [28] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Limrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 806–815.
- [29] Wei Wei, Jiabin Tang, Lianghao Xia, Yangqin Jiang, and Chao Huang. 2024. Promptmm: Multi-modal knowledge distillation for recommendation with prompt-tuning. In *Proceedings of the ACM on Web Conference 2024*. 3217–3228.
- [30] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-refined convolutional network for multimedia recommendation with implicit feedback. In *Proceedings of the 28th ACM international conference on multimedia*. 3541–3549.
- [31] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM international conference on multimedia*. 1437–1445.
- [32] Jinfeng Xu, Zheyu Chen, Jinze Li, Shuo Yang, Hewei Wang, Yijie Li, Mengran Li, Puzhen Wu, and Edith CH Ngai. 2025. MDVT: Enhancing Multimodal Recommendation with Model-Agnostic Multimodal-Driven Virtual Triplets. *arXiv preprint arXiv:2505.16665* (2025).
- [33] Jinfeng Xu, Zheyu Chen, Jinze Li, Shuo Yang, Hewei Wang, and Edith CH Ngai. 2024. AlignGroup: Learning and Aligning Group Consensus with Member Preferences for Group Recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 2682–2691.
- [34] Jinfeng Xu, Zheyu Chen, Jinze Li, Shuo Yang, Wei Wang, Xiping Hu, and Edith C-H Ngai. 2024. FourierKAN-GCF: Fourier Kolmogorov-Arnold Network—An Effective and Efficient Feature Transformation for Graph Collaborative Filtering. *arXiv preprint arXiv:2406.01034* (2024).
- [35] Jinfeng Xu, Zheyu Chen, Zixiao Ma, Jiye Liu, and Edith CH Ngai. 2024. Improving Consumer Experience With Pre-Purify Temporal-Decay Memory-Based Collaborative Filtering Recommendation for Graduate School Application. *IEEE Transactions on Consumer Electronics* (2024).
- [36] Jinfeng Xu, Zheyu Chen, Wei Wang, Xiping Hu, Sang-Wook Kim, and Edith CH Ngai. 2025. COHESION: Composite Graph Convolutional Network with Dual-Stage Fusion for Multimodal Recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1830–1839.
- [37] Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, Hewei Wang, and Edith CH Ngai. 2025. Mentor: multi-level self-supervised learning for multimodal recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 12908–12917.
- [38] Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, Hewei Wang, Wei Wang, Xiping Hu, and Edith Ngai. 2025. NLGCL: Naturally Existing Neighbor Layers Graph Contrastive Learning for Recommendation. *arXiv preprint arXiv:2507.07522* (2025).
- [39] Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, Wei Wang, Xiping Hu, Steven Hoi, and Edith Ngai. 2025. A Survey on Multimodal Recommender Systems: Recent Advances and Future Directions. *arXiv preprint arXiv:2502.15711* (2025).
- [40] Liangwei Yang, Zhiwei Liu, Chen Wang, Mingdai Yang, Xiaolong Liu, Jing Ma, and Philip S Yu. 2023. Graph-based alignment and uniformity for recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4395–4399.
- [41] Wenjie Yang, Shengzhong Zhang, Jiaxing Guo, and Zengfeng Huang. 2024. Your Graph Recommender is Provably a Single-view Graph Contrastive Learning. *arXiv preprint arXiv:2407.17723* (2024).
- [42] Zixuan Yi and Iadh Ounis. 2024. A unified graph transformer for overcoming isolations in multi-modal recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 518–527.
- [43] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.
- [44] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR*

- conference on research and development in information retrieval. 1294–1303.
- [45] Penghang Yu, Zhiyi Tan, Guanming Lu, and Bing-Kun Bao. 2023. Multi-view graph convolutional network for multimedia recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 6576–6585.
 - [46] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. 2019. GraphSAINT: Graph Sampling Based Inductive Learning Method. In *International Conference on Learning Representations*.
 - [47] Jiahao Zhang, Rui Xue, Wenqi Fan, Xin Xu, Qing Li, Jian Pei, and Xiaorui Liu. 2024. Linear-Time Graph Neural Networks for Scalable Recommendations. In *Proceedings of the ACM on Web Conference 2024*. 3533–3544.
 - [48] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. 2021. Mining latent structures for multimedia recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3872–3880.
 - [49] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Mengqi Zhang, Shu Wu, and Liang Wang. 2022. Latent structure mining with contrastive modality fusion for multimedia recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 9 (2022), 9154–9167.
 - [50] Shichang Zhang, Yozen Liu, Yizhou Sun, and Neil Shah. 2022. Graph-less Neural Networks: Teaching Old MLPs New Tricks Via Distillation. In *International Conference on Learning Representations*.
 - [51] Lingxiao Zhao and Leman Akoglu. 2020. PairNorm: Tackling Oversmoothing in GNNs. In *International Conference on Learning Representations*.
 - [52] Hongyu Zhou, Xin Zhou, Zhiwei Zeng, Lingzi Zhang, and Zhiqi Shen. 2023. A comprehensive survey on multimodal recommender systems: Taxonomy, evaluation, and future directions. *arXiv preprint arXiv:2302.04473* (2023).
 - [53] Hongyu Zhou, Xin Zhou, Lingzi Zhang, and Zhiqi Shen. 2023. Enhancing dyadic relations with homogeneous graphs for multimodal recommendation. In *ECAI 2023*. IOS Press, 3123–3130.
 - [54] Xin Zhou. 2023. MMRec: Simplifying Multimodal Recommendation. *arXiv preprint arXiv:2302.03497* (2023).
 - [55] Xin Zhou, Donghui Lin, Yong Liu, and Chunyan Miao. 2023. Layer-refined graph convolutional networks for recommendation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 1247–1259.
 - [56] Xin Zhou and Zhiqi Shen. 2023. A tale of two graphs: Freezing and denoising graph structures for multimodal recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 935–943.
 - [57] Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. 2023. Bootstrap latent representations for multi-modal recommendation. In *Proceedings of the ACM Web Conference 2023*. 845–854.
 - [58] Difan Zou, Ziniu Hu, Yewen Wang, Song Jiang, Yizhou Sun, and Quanquan Gu. 2019. Layer-dependent importance sampling for training deep and large graph convolutional networks. *Advances in neural information processing systems* 32 (2019).

Table 7: Similarity S between visual and textual embeddings.

Models	DualGNN	LATTICE	FREEDOM	SLMRec	BM3	MMSSL	LGMRec	DiffMM	FastMMRec
Baby	0.2007	0.2020	0.2731	0.3403	0.2636	0.3519	0.2591	0.2482	0.3848
Sports	0.1923	0.1950	0.2909	0.2988	0.2594	0.3020	0.2818	0.2619	0.3399
Clothing	0.2005	0.2031	0.2678	0.3009	0.2657	0.3055	0.2833	0.2377	0.3400

Table 8: Performance comparison of baselines on different datasets in terms of Time (s) and Memory (GB).

Baseline	Baby		Sports		Clothing	
	Time (s)	Memory (GB)	Time (s)	Memory (GB)	Time (s)	Memory (GB)
DualGNN	$7.12e^{-5}$ s	1.80GB	$1.33e^{-4}$ s	4.84GB	$1.32e^{-4}$ s	5.42GB
LATTICE	$7.14e^{-5}$ s	1.87GB	$1.34e^{-4}$ s	4.96GB	$1.37e^{-4}$ s	5.58GB
FREEDOM	$7.12e^{-5}$ s	1.83GB	$1.33e^{-4}$ s	4.88GB	$1.32e^{-4}$ s	5.52GB
SLMRec	$7.12e^{-5}$ s	1.80GB	$1.33e^{-4}$ s	4.84GB	$1.32e^{-4}$ s	5.42GB
BM3	$7.12e^{-5}$ s	1.80GB	$1.33e^{-4}$ s	4.84GB	$1.32e^{-4}$ s	5.42GB
MMSSL	$7.21e^{-5}$ s	1.85GB	$1.39e^{-4}$ s	4.92GB	$1.40e^{-4}$ s	5.56GB
LGMRec	$7.25e^{-5}$ s	1.89GB	$1.41e^{-4}$ s	5.03GB	$1.43e^{-4}$ s	5.64GB
DiffMM	$7.27e^{-5}$ s	1.90GB	$1.44e^{-4}$ s	5.05GB	$1.47e^{-4}$ s	5.67GB
FastMMRec	$7.17e^{-5}$ s	1.80GB	$1.37e^{-4}$ s	4.84GB	$1.37e^{-4}$ s	5.42GB

A Appendix

A.1 Baseline

In this section, we provide detailed introductions to all baseline models. 1) Conventional recommendation models:

- **MF-BPR** [22] leverages BPR loss to optimize the traditional collaborative filtering approach by learning representations of users and items through matrix factorization.
- **LightGCN** [15] streamlines the graph convolutional network (GCN) components unnecessary for collaborative filtering, enhancing its suitability for recommendations.
- **SimGCL** [44] proposes a graph contrastive learning that incorporates random noise directly into the feature representations.
- **LayerGCN** [55] employs residual connections to construct a layer-refined GCN, addressing the over-smoothing problem.

2) Multimodal recommendation models:

- **VBPR** [14] combines visual and textual features with ID embeddings as side information for each item, effectively achieving multimodal matrix factorization.
- **MMGCN** [31] applies a GCN for each modality to learn modality-specific features and then integrates all user-predicted ratings across modalities to produce the final rating.
- **DualGNN** [26] introduces a user-user graph to uncover hidden preference patterns among users.
- **LATTICE** [48] develops an item-item graph to detect semantically correlated signals among items.
- **FREEDOM** [56] refines LATTICE by freezing the item-item graph and reducing noise in the user-item graph.
- **SLMRec** [23] proposes a self-supervised learning framework for multimodal recommendations, establishing a node self-discrimination task to reveal hidden multimodal patterns of items.
- **BM3** [57] simplifies SLMRec by replacing the random negative example sampling mechanism with a dropout strategy.
- **MMSSL** [27] designs a modality-aware interactive structure learning paradigm via adversarial perturbations, and proposes

a cross-modal comparative learning method to disentangle the common and specific features among modalities.

- **LGMRec** [11] integrates local embeddings, which capture local topological nuances, with global embeddings, which consider hypergraph dependencies.
- **DiffMM** [17]: This method introduces a well-designed modality-aware graph diffusion model to improve modality-aware user representation learning.

Table 9: Performance comparison of different strategies on all datasets in terms of NDCG@20 (N@20) and s/EPOCH (#T).

Baseline	Baby		Sports		Clothing	
	N@20	#T	N@20	#T	N@20	#T
DualGNN _{train}	0.0309	5.63	0.0385	11.59	0.0299	14.19
DualGNN _{test}	0.0323	3.57	0.0410	7.57	0.0308	8.82
LATTICE _{train}	0.0370	3.20	0.0421	11.07	0.0330	16.53
LATTICE _{test}	0.0383	2.28	0.0427	6.01	0.0339	8.56
FREEDOM _{train}	0.0424	2.57	0.0481	5.65	0.0420	6.29
FREEDOM _{test}	0.0430	1.89	0.0487	3.79	0.0429	4.07
SLMRec _{train}	0.0353	2.07	0.0450	5.39	0.0303	6.02
SLMRec _{test}	0.0359	1.52	0.0459	4.28	0.0310	4.88
BM3 _{train}	0.0383	1.93	0.0438	3.82	0.0281	5.25
BM3 _{test}	0.0390	1.48	0.0451	2.99	0.0287	4.01
MMSSL _{train}	0.0420	6.31	0.0474	14.67	0.0359	17.04
MMSSL _{test}	0.0431	4.61	0.0482	8.39	0.0372	9.59

A.2 More Experiments for Investigation

To further support our investigation (Section 3), we conduct additional experiments on other advanced MMRec models, including DualGNN, LATTICE, FREEDOM, SLMRec, BM3, and MMSSL. (LGMRec and DiffMM are excluded as they utilize different structures—Hypergraph and Diffusion models, respectively.) As shown in Table 9, adopting GCN in the test phase enhances both the efficiency and performance of these models.

We also provide the similarity score S between visual and textual embeddings for other advanced MMRec models, including DualGNN, LATTICE, FREEDOM, SLMRec, BM3, MMSSL, LGMRec, and

DiffMM. As shown in Table 7, our FastMMRec achieves the highest similarity. While SLMRec, BM3, and MMSSL employ multiple self-supervised tasks to align visual and textual modalities, their similarity remains lower than ours (even without SSL tasks) at around 0.3, due to the limitations of GCN. Furthermore, as highlighted in the survey [52], models with low similarity scores, such as DualGNN and LATTICE, often perform better when relying on single-modal information rather than multimodal information. This further validates the effectiveness of deploying graph convolutions exclusively during the testing phase, bypassing their use in training.

A.3 Related Work

Many recent studies incorporate multimodal information to alleviate the data sparsity problem. VBPR [14] utilizes visual content in conjunction with matrix factorization techniques [22] to mitigate data sparsity issues. Subsequent studies [6–8, 20, 32, 36, 39, 45] have further enhanced the representation of items by incorporating both visual and textual modalities, thereby further mitigating the data sparsity problem. In an evolution of traditional recommendation system architectures, MMGCN [31] employs GCN to construct a bipartite graph that extracts latent information from user-item interactions. Building on this, GRCN [30] refines the approach by pruning false-positive edges, thus reducing noise within the bipartite graph. To explicitly explore commonalities in user preferences, DualGNN [26] introduces an additional user co-occurrence graph. Furthermore, LATTICE [48] implements an item semantic graph to capture latent correlative signals between items, while FREEDOM [56] stabilizes these representations by freezing the item semantic graph. In a novel approach, MMSSL [27] and MICRO [49] employ

contrastive self-supervised learning to align modalities and collaborative signals to enhance recommendation. Additionally, BM3 [57] and PromptMM [29] investigate inter-modal relationships to further improve recommendation accuracy and the quality of modal fusion. LGMRec [11] and DiffMM [17] explore the potential of hyper-graph structures and diffusion models in enhancing the effectiveness of multimodal recommendation systems, respectively. However, the complexity of model architectures and graph learning challenges are notably amplified in MMRec. Our FastMMRec model presents a viable solution by demonstrating that adopting GCNs during the testing phase not only enhances performance relative to existing methods but also significantly boosts efficiency. Our work provides a solution for deploying MMRec in large-scale real-world scenarios.

A.4 Testing Phase Efficiency

Inference time is critical for real-world applications. We evaluate FastMMRec and advanced baselines on per-user recommendation time and overall model memory usage. In fact, FastMMRec’s graph convolution in the test phase can be converted to using graph convolution to reconstruct the user and item representations after training, without having to repeat the calculation each time in the testing phase. As shown in Table 8, most models require similar memory usage, as they primarily store user/item embeddings, except for those with specialized structures (LGMRec, MMSSL, and DiffMM), which demand more memory. In terms of inference time, models with complex structures (LGMRec, MMSSL, and DiffMM) exhibit higher costs. FastMMRec and FREEDOM incur approximately 1% extra inference time compared to the fastest baseline (SLMRec).