Knowledge-Based Systems 317 (2025) 113496

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Discrepancy Learning Guided Hierarchical Fusion Network for Multi-modal Recommendation

Yuzhuo Dang [a], Zhiqiang Pan [a], Xin Zhang [a], Wanyu Chen [b], Fei Cai [a,*], Honghui Chen [a]

[a] National Key Laboratory of Information Systems Engineering, National University of Defense Technology, Changsha, Hunan, China
[b] College of Electronic Countermeasures, National University of Defense Technology, Hefei, Anhui, China

ARTICLE INFO

Keywords:
Multi-modal recommendation
Discrepancy learning
Popularity denoising
Hierarchical fusion

ABSTRACT

Multi-modal recommender systems aim to generate accurate recommendations for users by leveraging the user behaviors and the modal information of items, which is typically achieved using a multi-modal fusion mechanism that is mainly categorized into early fusion and late fusion strategies. The former strategy occurs in the feature extraction stage and the latter occurs after a decision is made based on each modality. However, these strategies are effective only in limited scenarios because of the discrepancies between modalities. In addition, existing methods for extracting user behavior features generally ignore noisy interactions prompted by user's unintentional behaviors and popularity trends, which leads to suboptimal recommendations. To address the aforementioned issues, we propose a novel framework called **D**iscrepancy learning **G**uided **H**ierarchical fusion **Net**work (**DGHNet**), that comprises four components, i.e., a cross-modal alignment fusion, a multi-modal discrepancy learning, a multi-graph construction, and a hierarchical fusion network. In detail, we first apply an early fusion strategy to generate cross-modal features. Then, we quantify the discrepancies between different modalities using multi-modal discrepancy learning. After that, we construct multiple graphs to discover implicit structural information and capture user behavior preferences using a popularity-aware pruning strategy. Finally, we design a hierarchical fusion network to learn comprehensive representations of users and items using the quantified results and constructed multi-graph. Extensive experiments on three public datasets, i.e., *Baby*, *Sports*, and *Clothing*, validate the superiority of DGHNet over state-of-the-art baselines, where the improvements are particularly noticeable on the large-scale dataset.

1. Introduction

Recommender systems have emerged as an essential tool for helping users filter out products or contents that do not match their personal preferences from many candidates [1–5]. Because online platforms are flooded with large amounts of multi-modal data (e.g., image, text, and audio) revealing user preferences at a modal fine-grained level, researchers are actively exploring ways to integrate such information into recommender systems to accurately model user preferences [6–9]. Unlike traditional recommendation methods, multi-modal recommender systems (MMRS) enhance the efficacy of representation learning of items by combining the multi-modal information of items with the historical user–item interactions.

Mainstream methods in multi-modal recommendation initially extract modal features using pre-trained models, followed by a series of operations for modal fusion [10,11]. The fused modal features are then integrated with the user interactive behavior to form the final representations of items. On the basis of the timing of considering the modal fusion effect, existing methods are categorized as early fusion and late fusion strategies. Early fusion [6,12,13] occurs in the feature extraction stage, where different uni-modal features of each item are combined directly into a unified feature representation, which is then inputted to the model for training. By contrast, the late fusion [14–18] occurs following the decision based on each modality and can dynamically integrate the modal features based on learned outcomes.

Despite the success achieved by existing approaches based on multi-modal fusion, the following issues remain. (1) Existing methods overlook the complementarity between early fusion and later fusion strategies in addressing various modal discrepancies when extracting multi-modal information for items. Specifically, the early fusion is prone to overfitting as a result of redundant information between different modalities when the discrepancies are small, and the late fusion is easily affected by the noise interference due to large discrepancies [17,19]. Especially, in real-world scenarios, modal discrepancy fluctuates

* Corresponding author.
E-mail addresses: dangyuzhuo@nudt.edu.cn (Y. Dang), panzhiqiang@nudt.edu.cn (Z. Pan), zhangxin16@nudt.edu.cn (X. Zhang), wanyuchen@nudt.edu.cn (W. Chen), caifei08@nudt.edu.cn (F. Cai), chenhonghui@nudt.edu.cn (H. Chen).

https://doi.org/10.1016/j.knosys.2025.113496
Received 21 May 2024; Received in revised form 4 December 2024; Accepted 3 April 2025
Available online 12 April 2025

significantly, leading to evident shortcomings in both strategies. (2) The noise in the user–item interaction graph causes transition smoothing in the graph convolution process, which potentially obscures valuable user behaviors. Existing methods use random edge dropout and graph convolution norm adjustment strategies [20,21], but the effects are not significant. They lack the ability to recognize interactions triggered by users' unintentional behaviors or popularity trends, resulting in the continued propagation of popularity-related noise throughout the graph convolution process.

To address the above issues, we propose a novel framework called **D**iscrepancy learning **G**uided **H**ierarchical fusion **Net**work (**DGHNet**) for multi-modal recommendation. Specifically, we first design an early fusion method that maps the original uni-modal features into the same semantic space and then concatenate them to obtain the cross-modal features of items. In addition, we calculate the discrepancy scores between the original uni-modal features in the multi-modal difference learning module to guide the hierarchical fusion network in adaptively fusing uni-modal and cross-modal features. Then, we design a multi-graph construction module, in which we adopt the top-n sparsity to construct the visual and textual item–item graphs from the original uni-modal features and the cross-modal one from the generated cross-modal features. In addition, we obtain a purified user–item graph by designing a popularity-aware pruning strategy. Finally, we progressively fuse various item embeddings obtained by convolving the randomly initialized ID embeddings on three item–item graphs and one user–item graph, using the calculated discrepancy scores in the proposed hierarchical fusion network.

Experiments on three public datasets, i.e., *Baby*, *Sport*, and *Clothing*, demonstrate that our DGHNet outperforms the state-of-the-art baselines in terms of both Recall@K and NDCG@K. In addition, the ablation experiments validate the effectiveness of our popularity denoising strategy in filtering out noisy edges from the user–item graph, and the robustness of our fusion method that combines the complementary information between early fusion and late fusion. In summary, our contributions can be summarized in the following three parts:

- We design a novel framework that integrates the early fusion and late fusion strategies of multi-modal information seamlessly, which enables the complementary advantages of both approaches, to obtain the robust and comprehensive multi-modal representations of items.
- We propose a popularity-aware pruning strategy to remove interactions triggered by user's unintentional behaviors and popularity trends. In addition, this strategy is capable of flexible application in arbitrary user–item graphs to capture accurate user behavior information.
- We evaluate our proposal on three public datasets and find that DGHNet obviously outperforms the state-of-the-art baselines in terms of recommendation accuracy.

2. Related works

In this section, we review related studies from two aspects, i.e., GNN-based recommendation in Section 2.1 and multi-modal recommendation in Section 2.2.

2.1. GNN-based recommendation

In recent years, the rapid advancement of Graph Neural Networks (GNN) technology has enhanced the efficient representation learning of structured data in complex systems [22–25]. In the field of recommender systems, GNN has emerged as a crucial tool for learning user preferences from historical interactions that can typically be structured as a user–item graph [26–28].

In an early study, van den Berg et al. [29] propose a graph convolutional matrix completion (GC-MC) framework for addressing the sparse matrix completion problem in recommender systems. However, multi-layer graph convolutions lead to overfitting, thus Pan et al. [26] design star graph neural networks with highway networks (SGNN-HN). It captures information from non-adjacent items in the current session and uses a highway network to adaptively select item features. Several studies [30,31] have pointed out that the introduction of weight transformation matrices and nonlinear activation functions in GCN adds unnecessary parameters, which may trigger overfitting. To address this issue, He et al. [32] propose a simplifying and powering graph convolution network (LightGCN) that captures higher-order co-signals at a light GCN without a weight transformation matrix and nonlinear activation functions.

To integrate time series information with graph structure information, Zhou et al. [33] construct a daily graph to model the relationships between items. Building on this, Zhang et al. [34] design a diffusion convolutional recurrent neural network (DCRNN) to address dependency on complex spaces, which inputs the spatial information collected by GNN into a recurrent neural network (RNN). Although these methods emphasize dynamic information, they ignore the static intentions of users. Therefore, Zhang et al. [35] integrate both the spatial and temporal aspects of graphs simultaneously using a dynamic and static intention-integrated graph neural network (DSGNN). Further, Yang et al. [36] integrate the long short term memory (LSTM) and graph attention networks (GAT) [37], where LSTM is initially used to extract users' short-term interests, and GAT is subsequently applied to aggregate the influence of various friends in the relationship network.

Inspired by these antecedent studies, our proposed DGHNet employs GCN as core architecture. By incorporating high-level information, the proposal is capable of capturing the intricate relationships among disparate modalities, thereby yielding improved performance in its task-specific applications.

2.2. Multi-modal recommendation

MMRS is a content enhancement recommendation that integrates user preferences obtained from various modalities into the traditional collaborative filtering (CF) paradigm [32].

In early studies, He and McAuley [38] propose the visual bayesian personalized ranking (VBPR), which extends the bayesian personalized ranking (BPR) [39] method using visual features that extracted a pre-trained model. Subsequently, researchers delve into the study of multi-modal user preference. Wei et al. [14] design a multi-modal graph convolution network (MMGCN), that applies GCN to the interaction graph of various modalities to derive the user preferences for each modality. However, multi-view graph learning results in longer training times. Xu et al. [40] propose a deep users' multimodal preferences-based recommendation (UMPR) approach, that captures textual and visual preferences through multimodal matching. Considering the varying impacts of different modal information on recommendation systems, some studies have introduced attention mechanisms into MMRS. For instance, Liu et al. [13] propose a multimodal attentional metric learning (MAML) method to integrate multi-modal data with different attentional weights into a metric-based learning. Furthermore, to achieve optimal results when using GNN, Tao et al. [41] design a multi-modal graph attention network (MGAT), that employs a weighted information propagation scheme based on MMGCN.

To further extract the hidden semantic information between items, Zhang et al. [6] design a latent structure mining scheme (LATTICE), that constructs the item–item graphs for each modality. However, Zhou and Shen [42] emphasize that updating the item–item graph can increase model complexity and can introduce noise. Consequently, they advocate for freezing the graph for an effective and efficient recommendation process. To address the problem of sparse data, Tao et al. [17] design a framework of self-supervised learning-guided multi-media recommendation (SLMRec), that uses data augmentation techniques mediated by adversarial perturbation. Unfortunately, data augmentation

Table 1
Main notations employed in this paper.

Notation	Description
$\mathcal{U}$	A set of all users
$\mathcal{I}$	A set of all items
$\boldsymbol{e}_u, \boldsymbol{e}_i$	The ID embeddings of users and items
$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$	The observed historical interactions graph between $\mathcal{U}$ and $\mathcal{I}$
$\tilde{\mathcal{G}} = \{\tilde{\mathcal{V}}, \tilde{\mathcal{E}}\}$	The denoised historical interactions graph between $\mathcal{U}$ and $\mathcal{I}$
$\mathcal{M}$	A set of modalities
$\boldsymbol{e}_i^m$	The modal features of items
$\mathcal{G}_m$	The constructed item–item relationships graphs under modality
$q(\boldsymbol{z}_i^m \mid \boldsymbol{e}_i^m)$	The variational posterior probability distribution of unimodal observation in different modalities of items
d_i	The discrepancy scores between visual and textual modalities of items
n	The number of edges in graph $\mathcal{G}_m$
p_k	A pruning probability of the edge $(v_u, v_i) \in \mathcal{E}$ in the graph $\mathcal{G}$
ρ	A certain pruning percentage of edges in $\mathcal{E}$
L_{ui}, L_{ii}	The layers of convolution in graphs $\tilde{\mathcal{G}}$ and $\mathcal{G}_m$
$\boldsymbol{h}_u, \boldsymbol{h}_i$	The interaction embeddings of users and items obtained by graph convolution of $\boldsymbol{e}_u$ and $\boldsymbol{e}_i$ on graph $\tilde{\mathcal{G}}$
$\boldsymbol{h}_i^m$	The modal embeddings of items obtained by graph convolution of $\boldsymbol{e}_i$ on graph $\mathcal{G}_m$
α	A trade-off parameter in late fusion
$\mathcal{L}, \mathcal{L}_{bpr}, \mathcal{L}_{cmbpr}, \mathcal{L}_{vae}$	The joint, BPR, cross-modal BPR and VAE loss
λ_1, λ_2	The trade-off parameters for balancing $\mathcal{L}_{cmbpr}$ and $\mathcal{L}_{vae}$
K	The number of items recommended to the user

can introduce noise and can increase computational costs. Thus, Zhou et al. [18] propose a bootstrapped multi-modal model (BM3) that has three contrastive losses and eliminates the requirement for auxiliary graphs to avoid noise from random sampling. In addition, Yan et al. [43] argue that denoising cannot completely eliminate noise in multi-modal contents and may restrict the ability to capture personalized user preferences. Therefore, they design a trustworthy sequential recommendation method via noisy user-generated multi-modal content (TrurhSR), which proposes a trustworthy decision strategy to mitigate noise interference.

However, existing multi-modal recommendation methods fail to adequately explore the relationship between uni-modal and cross-modal features, which compromises robustness of models. To address this, we propose a hierarchical fusion network that integrates the complementary information of uni-modal and cross-modal features, thereby providing a more comprehensive representation of item features.

3. Approach

In this section, we first describe the task definition and major notations of multi-modal recommendation in Section 3.1. Subsequently, Section 3.2 outlines an overview of the DGHNet framework. We then introduce the cross-modal alignment fusion and multi-modal discrepancy learning in Sections 3.3 and 3.4, respectively. After that, we detail the multi-graph construction and the hierarchical fusion network in Sections 3.5 and 3.6, respectively. Building upon the preceding subsections, we generate the top-K recommendations and conduct model optimization in Section 3.7.

3.1. Task definition and notations

Let $\mathcal{U}$ and $\mathcal{I}$ denote a user set and an item set, respectively. The input ID embeddings generated by an embedding layer for user $u \in \mathcal{U}$ and item $i \in \mathcal{I}$ are $\boldsymbol{e}_u, \boldsymbol{e}_i \in \mathbb{R}^d$, where d is the embedding dimension. User-item historical interaction records are presented as a bipartite graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{\mathcal{U} \cup \mathcal{I}\}$ denotes a set of nodes and $\mathcal{E}_{u,i} \in \{0, 1\}$ indicates the presence of an edge between user u and item i. An edge $\mathcal{E}_{u,i}$ exists and is assigned the value 1 if user u has interacted with item i; otherwise, $\mathcal{E}_{u,i} = 0$. In addition to user–item interactions, the multi-modal features are provided as additional contents for items. We present the modal features of item i as $\boldsymbol{e}_i^m \in \mathbb{R}^{d_m}$ by the pre-trained model, where $m \in \mathcal{M}$ denotes modality and d_m is the dimension of the features. We focus on the original visual and textual modalities as well as the cross-modality obtained in Section 3.3, which we collectively refer to as a set of modalities $\mathcal{M} = \{v, t, c\}$. Notably, in Sections 3.3 and 3.4, $\mathcal{M} = \{v, t\}$. The main notations used are listed in Table 1.

Our task in multi-modal recommendation is to capture user preference by merging the uni-modal and cross-modal information. In particular, we construct three item–item graphs $\mathcal{G}_m$ for each modal $m \in \{v, t, c\}$ and a purified user–item graph $\tilde{\mathcal{G}}$. Subsequently, we leverage these graph structures to optimize the ID embeddings of users and items, which enhance the accuracy of ranking user's items according to the predicted preference scores $\hat{y}_{ui}$.

3.2. Overall framework

The framework of DGHNet is presented in Fig. 1, consisting of four major components, i.e., a cross-modal alignment fusion, a multi-modal discrepancy learning, a multi-graph construction, and a hierarchical fusion network. The original uni-modal features $\boldsymbol{e}_i^v$ and $\boldsymbol{e}_i^t$ of the corresponding image and text are obtained using the pre-trained models, which are subsequently passed to the subsequent modules directly.

Cross-modal Alignment Fusion is an early fusion strategy that aligns the semantic gap between the pre-trained models and the recommendation task. We design a loss $\mathcal{L}_{cmbpr}$ based on the Bayesian Personalized Ranking (BPR) loss [39] to optimize MLPs, resulting in more accurate cross-modal features.

Multi-modal Discrepancy Learning employs the concept of Variational Auto Encoder (VAE) [44] to propose a method for calculating the discrepancy scores between different modalities. These scores are used to facilitate the adaptive integration of early fusion and late fusion information. To make the scores more accurate, we design a loss $\mathcal{L}_{vae}$ to optimize the variational autoencoders.

Multi-graph Construction constructs three item–item graphs $\mathcal{G}_m$ using the uni-modal and cross-modal features of items, and generates a purified user–item graph $\tilde{\mathcal{G}}$ by removing noisy interactions. These graphs are constructed to reveal the similarities and latent structural connections between items across modalities, facilitating the process of generating accurate user preference.

Hierarchical Fusion Network performs graph convolution operations of ID embeddings $\boldsymbol{e}_u$ and $\boldsymbol{e}_i$ on graphs $\tilde{\mathcal{G}}$ and $\mathcal{G}_m$ to obtain high-level representations of users and items. A multi-step late fusion approach is then applied to adaptively fuse the embeddings of items from the cross-modal and uni-modal graphs, generating the final item representation. Finally, the loss $\mathcal{L}_{bpr}$ is designed to optimize $\boldsymbol{e}_u$ and $\boldsymbol{e}_i$.

3.3. Cross-modal alignment fusion

According to Zhang et al. [6] and Zhou et al. [18], the features from diverse modalities obtained using pre-trained models exhibit an obvious semantic gap. To address this issue, we initially employ the Multi-layer Perceptrons (MLPs) to perform a dimensionality reduction on the original uni-modal features $\boldsymbol{e}_i^m$ extracted from the pre-trained models, ensuring that all modal features are fixed in a uniform dimension. This can be formulated as follows:

$$\boldsymbol{x}_i^m = \sigma_1\left(\boldsymbol{e}_i^m \boldsymbol{W}^m + \boldsymbol{b}^m\right), \tag{1}$$

where $\boldsymbol{W}^m \in \mathbb{R}^{d_m \times d}$ and $\boldsymbol{b}^m \in \mathbb{R}^d$ denote the transformation matrix and bias vector in the MLPs, respectively. The function $\sigma_1(\cdot)$ is the LeakyRelu activation function.

Unlike traditional additive or averaging fusion strategies [6,45], we concatenate each modal features after the dimensionality reduction process, which adequately preserves the original modal information. Consequently, the concatenated composite features are inputted to a linear layer for deep fusion, generating the cross-modal fusion features $\boldsymbol{e}_i^c$. This can be formulated as follows:

$$\boldsymbol{e}_i^c = \left([\boldsymbol{x}_i^v, \boldsymbol{x}_i^t]\right) \boldsymbol{W}^c + \boldsymbol{b}^c, \tag{2}$$

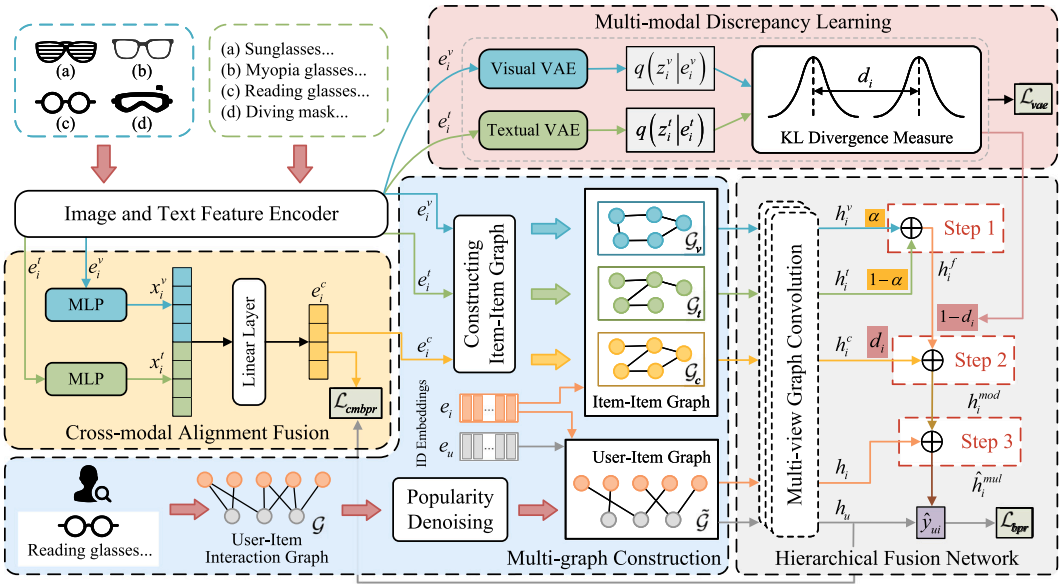


Fig. 1. Overview of the DGHNet framework.

where $\boldsymbol{x}_i^v$ and $\boldsymbol{x}_i^t$ represent the visual and textual features calculated from Eq. (1), respectively. $[\cdot,\cdot]$ denotes the concatenation operation, while $\boldsymbol{W}^c \in \mathbb{R}^{2d\times d}$ and $\boldsymbol{b}^c \in \mathbb{R}^d$ denote the transformation matrix and bias vector in the linear layer, respectively.

3.4. Multi-modal discrepancy learning

To understand the distribution of fixed uni-modal features for each input term, we adopt a variational autoencoder (VAE)-based approach [44], which can capture the underlying structure of uni-modal features from a generative perspective. To this end, we use the Kullback–Leibler (KL) divergence [46] in the feature space to quantify the distribution discrepancies between different modal features.

First, the mean value $\mu(\boldsymbol{e}_i^m)$ and variance $\sigma(\boldsymbol{e}_i^m)^2$ of the unimodal distribution are obtained using a variational autoencoder tailored to the original uni-modality. Then, we model the variational posterior probability distribution $q(\boldsymbol{z}_i^m \,|\boldsymbol{e}_i^m)$ using

$$q(\boldsymbol{z}_i^m \,|\boldsymbol{e}_i^m) = \mathcal{N}\left(\boldsymbol{z}_i^m \,\middle|\mu(\boldsymbol{e}_i^m), \sigma(\boldsymbol{e}_i^m)^2\right). \tag{3}$$

where $\boldsymbol{z}_i^m$ is a latent variable that follows the prior distribution $p(\boldsymbol{z}_i^m) = \mathcal{N}(0, \boldsymbol{I})$ and $\boldsymbol{I}$ denotes an identity matrix.

Then, the discrepancy scores between the modalities of item i can be quantified using the mean KL divergence between their respective unimodal distributions. Because of the asymmetry of the KL divergence, we get two discrepancy scores $d_{i,1}$ and $d_{i,2}$:

$$d_{i,1} = \left(\frac{D_{KL}\left(q(\boldsymbol{z}_i^v \,|\boldsymbol{e}_i^v) \,\middle\|\, q(\boldsymbol{z}_i^t \,|\boldsymbol{e}_i^t)\right)}{D_{KL}\left(q(\boldsymbol{z}^v) \,\|\, q(\boldsymbol{z}^t)\right)}\right), \tag{4}$$

$$d_{i,2} = \left(\frac{D_{KL}\left(q(\boldsymbol{z}_i^t \,|\boldsymbol{e}_i^t) \,\middle\|\, q(\boldsymbol{z}_i^v \,|\boldsymbol{e}_i^v)\right)}{D_{KL}\left(q(\boldsymbol{z}^t) \,\|\, q(\boldsymbol{z}^v)\right)}\right), \tag{5}$$

where the function $D_{KL}(\cdot)$ is the KL divergence and $q(\boldsymbol{z}^m \,|\boldsymbol{e}^m) = \mathcal{N}\left(\boldsymbol{z}^m \,\middle|\mu(\boldsymbol{e}^m), \sigma(\boldsymbol{e}^m)^2\right)$ denotes the distribution of the overall dataset for each modality.

By averaging $d_{i,1}$ and $d_{i,2}$, the symmetric KL divergence is obtained as the discrepancy scores:

$$d_i = \mathrm{Tanh}\left(\frac{1}{2}\left(d_{i,1} + d_{i,2}\right)\right), \tag{6}$$

where $\mathrm{Tanh}(\cdot)$ is the Tanh activation function that is employed to map the discrepancy scores to a range between 0 and 1. The calculated score d_i is used to adaptively control the contribution of early and late fusion in Step 2 of Section 3.6.

3.5. Multi-graph construction

As structured information can obviously improve the effectiveness of multi-modal recommendation [47–49], we construct three item–item graphs and generate a purified user–item graph by removing noisy interactions.

3.5.1. Constructing item–item graph

Following Zhang et al. [6], we employ top-n sparsity to construct item–item graphs $\mathcal{G}_m$. First, we evaluate the semantic relationship between items i_a and i_b by calculating their cosine similarity score $\boldsymbol{S}_{a,b}^m$:

$$\boldsymbol{S}_{a,b}^m = \frac{(\boldsymbol{e}_{i_a}^m)^{\mathrm{T}} \boldsymbol{e}_{i_b}^m}{\left\|\boldsymbol{e}_{i_a}^m\right\| \cdot \left\|\boldsymbol{e}_{i_b}^m\right\|}, \tag{7}$$

where $m \in \{v, t, c\}$. $\boldsymbol{e}_i^c$ denotes the cross-modal fusion feature obtained in Section 3.3, while $\boldsymbol{e}_i^v$ and $\boldsymbol{e}_i^t$ are directly generated by the pre-trained model.

The generation of fully-connected graphs requires substantial computational resources, and the presence of unnecessary connections can introduce some noise. To address these issues, we apply top-n sparsity to the fully connected graphs, maintaining only top-n edges with the highest similarity for each item i:

$$\tilde{\boldsymbol{S}}_{a,b}^m = \begin{cases} 1, & \boldsymbol{S}_{a,b}^m \in \mathrm{top} - n(\boldsymbol{S}_{a,\cdot}^m), \\ 0, & \mathrm{otherwise}. \end{cases}, \tag{8}$$

where $\boldsymbol{S}_{a,\cdot}^m$ denotes all elements of row a in matrix $\boldsymbol{S}^m$ and the value of $\tilde{\boldsymbol{S}}^m \in \mathbb{R}^{|I|\times|I|}$ is either 0 or 1, with 1 indicating a potential connection between two items. As a result, we obtain three item–item graphs $\mathcal{G}_m$ corresponding to the original visual, textual, and the cross-modal features.

3.5.2. Popularity denoising user–item graph

Different from [21], which uses the popularity-aware graph Laplacian norm in the graph convolution, we design an edge pruning strategy based on the concept of model simplification [20,50], which enables personalized popularity denoising on the observed user–item graph $\mathcal{G}$.

Specifically, in the bipartite graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, for each existing edge $\langle v_u, v_i \rangle \in \mathcal{E}$ that denotes the link between nodes v_u and v_i ($v \in \mathcal{V}$), we define a pruning probability:

$$p_{(v_u, v_i)} = \frac{1}{1 + exp\left(-d\left(v_u\right) \cdot d\left(v_i\right)\right)}, \tag{9}$$

where $d(\cdot)$ denotes the degree of node v in graph $\mathcal{G}$ and the function $exp(\cdot)$ is the exponential operation.

We then sample from the entire pruning probability distribution $p = \{p_1, p_2, \ldots, p_{|\mathcal{E}|-1}\}$. Simply pruning a fixed number of edges or removing edges below a certain threshold can result in an excessive or insufficient number of remaining edges, failing to dynamically adjust the number of pruned edges according to the scale and complexity of the graph. To ensure that the algorithm adapts to datasets of different scales and effectively eliminates noisy edges while maintaining the connectivity and structural information of the graph, we define a certain pruning percentage ρ of the edges. In other words, we need to remove $\lfloor \rho|\mathcal{E}| \rfloor$ edges, where $\lfloor \cdot \rfloor$ denotes the downward rounding. The purified graph of the user–item interaction obtained after pruning is $\tilde{\mathcal{G}} = \{\tilde{\mathcal{V}}, \tilde{\mathcal{E}}\}$, where $|\tilde{\mathcal{V}}| = |\mathcal{U}| + |\mathcal{I}|$ and $|\tilde{\mathcal{E}}| = |\mathcal{E}| - \lfloor \rho|\mathcal{E}| \rfloor$, and the corresponding matrix is $\tilde{\boldsymbol{G}}$.

3.6. Hierarchical fusion network

First, we initialize all nodes in the item–item graph and user–item graph by mapping all user and item ID embeddings to a dense vector representation as follows:

$$\mathbf{E}_{\mathcal{U}}^{id(0)} = \left\{ \boldsymbol{e}_{u_1}, \boldsymbol{e}_{u_2}, \ldots, \boldsymbol{e}_{u_{|\mathcal{U}|}} \right\}, \tag{10}$$

$$\mathbf{E}_{\mathcal{I}}^{id(0),m} = \mathbf{E}_{\mathcal{I}}^{id(0)} = \left\{ \boldsymbol{e}_{i_1}, \boldsymbol{e}_{i_2}, \ldots, \boldsymbol{e}_{i_{|\mathcal{I}|}} \right\}, \tag{11}$$

where $\mathbf{E}_{\mathcal{U}}^{id(0)} \in \mathbb{R}^{|\mathcal{U}| \times d}$ and $\mathbf{E}_{\mathcal{I}}^{id(0),m} = \mathbf{E}_{\mathcal{I}}^{id(0)} \in \mathbb{R}^{|\mathcal{I}| \times d}$ denote the ID embeddings matrices of the users and items at the 0-layer, respectively, and d is the embedding dimension.

Subsequently, we proceed with the convolution operation on three item–item graphs and the purified user–item graph. Following Zhang et al. [6], Zhou and Shen [42] and Zhou et al. [18], we employ LightGCN [32] to propagate and aggregate information from the neighboring nodes, thereby enriching the features of each node by identifying the commonalities among similar nodes.

Specifically, the convolution operation on three item–item graphs is defined as follows:

$$\mathbf{E}_{\mathcal{I}}^{id(l),m} = \left((\boldsymbol{D}^m)^{-\frac{1}{2}} \tilde{\boldsymbol{S}}^m (\boldsymbol{D}^m)^{-\frac{1}{2}} \right) \mathbf{E}_{\mathcal{I}}^{id(l-1),m}, \tag{12}$$

where $\boldsymbol{D}^m \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{I}|}$ is the diagonal degree matrix of $\tilde{\boldsymbol{S}}^m$ and $\boldsymbol{D}_{aa}^m = \sum_b \tilde{\boldsymbol{S}}_{ab}^m$. The embeddings on the lth layer are updated solely from the $(l-1)$th layer to perform a linear aggregation through the transfer matrix $(\boldsymbol{D}^m)^{-\frac{1}{2}} \tilde{\boldsymbol{S}}^m (\boldsymbol{D}^m)^{-\frac{1}{2}}$. Because there is a self-loop for each item node in the item–item graph obtained by top-n sparsity, each layer retains the information of the previous layer. We directly use the output of the L_{ii}-layer as the final representation $\boldsymbol{h}_i^m$. This can be formulated as follows:

$$\boldsymbol{h}_i^m = \boldsymbol{e}_i^{id(L_{ii}),m}. \tag{13}$$

Similarly, we perform a convolution on the user–item graph. To improve computational efficiency, we concatenate $\mathbf{E}_{\mathcal{U}}^{id(0)}$ and $\mathbf{E}_{\mathcal{I}}^{id(0)}$ into a single matrix $\mathbf{E}_{\mathcal{U},\mathcal{I}}^{id(l)} \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{I}|) \times d}$ and create a symmetric adjacency matrix as follows:

$$\boldsymbol{M} = \begin{bmatrix} 0 & \tilde{\boldsymbol{G}} \\ \tilde{\boldsymbol{G}}^T & 0 \end{bmatrix}, \tag{14}$$

where $\boldsymbol{M} \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{I}|) \times (|\mathcal{U}|+|\mathcal{I}|)}$. Then, the graph convolution operation at lth on graph $\tilde{\mathcal{G}}$ is formulated as follows:

$$\mathbf{E}_{\mathcal{U},\mathcal{I}}^{id(l)} = \left((\boldsymbol{D})^{-\frac{1}{2}} \boldsymbol{M} (\boldsymbol{D})^{-\frac{1}{2}} \right) \mathbf{E}_{\mathcal{U},\mathcal{I}}^{id(l-1)}, \tag{15}$$

where $\boldsymbol{D} \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{I}|) \times (|\mathcal{U}|+|\mathcal{I}|)}$ is the diagonal degree matrix of $\boldsymbol{M}$ and $\boldsymbol{D}_{aa} = \sum_b \boldsymbol{M}_{ab}$. Unlike the item–item graph, the nodes in the user–item graph do not preserve their self information from the previous layer of the same node during the convolution process. Therefore, after the L_{ui}-layer process, we integrate the embeddings learned at each layer to construct the final representation $\boldsymbol{h}_u$ and $\boldsymbol{h}_i$ of users and items:

$$\boldsymbol{h}_u = \frac{1}{L_{ui}+1} \sum_{l=0}^{L_{ui}} \boldsymbol{e}_u^{id(l)}, \quad \boldsymbol{h}_i = \frac{1}{L_{ui}+1} \sum_{l=0}^{L_{ui}} \boldsymbol{e}_i^{id(l)}. \tag{16}$$

On the basis of the above embeddings obtained from multiple information sources, to effectively fuse this information, we design a multi-step hierarchical fusion method.

Step 1. Late fusion. The visual and textual embeddings are combined to produce the uni-modal fusion embeddings $\boldsymbol{h}_i^f$:

$$\boldsymbol{h}_i^f = \alpha \boldsymbol{h}_i^v + (1-\alpha) \boldsymbol{h}_i^t, \tag{17}$$

where $\boldsymbol{h}_i^v$ and $\boldsymbol{h}_i^t$ represent the visual and textual embeddings calculated from Eq. (13), respectively. α denotes a hyperparameter indicating the relative importance of the visual feature versus textual feature.

Step 2. Hybrid fusion. The uni-modal fusion embeddings $\boldsymbol{h}_i^f$ and cross-modal embeddings $\boldsymbol{h}_i^c$ are fused as follows:

$$\boldsymbol{h}_i^{mod} = d_i \boldsymbol{h}_i^c + (1-d_i) \boldsymbol{h}_i^f, \tag{18}$$

where $\boldsymbol{h}_i^{mod}$ denotes the modal embeddings of items and d_i is obtained from Eq. (6) in Section 3.4.

Step 3. Global fusion. The resulting item modal embeddings $\boldsymbol{h}_i^{mod}$ and the item interaction embeddings $\boldsymbol{h}_i$ learned from the user–item graph are integrated to derive the final item representations $\hat{\boldsymbol{h}}_i^{mul}$:

$$\hat{\boldsymbol{h}}_i^{mul} = \boldsymbol{h}_i^{mod} + \boldsymbol{h}_i. \tag{19}$$

So far, the final representations of users $\boldsymbol{h}_u$ and items $\hat{\boldsymbol{h}}_i^{mul}$ are produced.

3.7. Recommendation and optimization

3.7.1. Top-K recommendation

Following Eqs. (16) and (19), we employ the inner product operation to calculate the preference scores $\hat{y}_{ui}$ of user u to the candidate items i:

$$\hat{y}_{ui} = (\boldsymbol{h}_u)^{\mathrm{T}} \hat{\boldsymbol{h}}_i^{mul}. \tag{20}$$

A high score indicates a strong preference that the user holds for the item. We rank the scores in a descending order and select the top-K items to recommend to the corresponding user.

3.7.2. Model optimization

In the model training phase, we employ a multi-task learning strategy to concurrently optimize DGHNet, involving the following tasks:

(i) cross-modal alignment fusion task $\mathcal{L}_{cmbpr}$.

To train the MLPs to obtain the cross-modal features accurately, we construct a triplet set D, which consists of triplets (u, i, j), where user u has an interaction with item i but no interaction with a randomly sampled item j. Then, we use the BPR loss $\mathcal{L}_{cmbpr}$ to assign a larger score to the positive pairs (u, i) than the negative pairs (u, j).

$$\mathcal{L}_{cmbpr} = -\sum_{(u,i,j) \in D} \log \sigma_2 \left((\boldsymbol{h}_u)^{\mathrm{T}} \boldsymbol{e}_i^c - (\boldsymbol{h}_u)^{\mathrm{T}} \boldsymbol{e}_j^c \right), \tag{21}$$

where $\boldsymbol{h}_u$ and $\boldsymbol{e}_i^c$ are obtained from Eqs. (16) and (2), respectively, and $\sigma_2(\cdot)$ is the Sigmoid activation function.

(ii) multi-modal discrepancy learning task $\mathcal{L}_{vae}$.

To improve the accuracy of generating the discrepancy scores, we employ a loss function $\mathcal{L}_{BCE}^m$:

$$\mathcal{L}_{BCE}^m = -\sum_{i \in I} \left(\hat{\boldsymbol{e}}_i^m \cdot \log(\boldsymbol{e}_i^m) + (1 - \hat{\boldsymbol{e}}_i^m) \cdot \log(1 - \boldsymbol{e}_i^m) \right), \tag{22}$$

Algorithm 1 Training DGHNet.

Input: Adjacency matrix $\boldsymbol{G} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$ obtained by the observed historical interactions graph $\mathcal{G}$. Original uni-modal features $\boldsymbol{e}_i^v$ and $\boldsymbol{e}_i^t$ of the image and text obtained by the pre-trained models, respectively.

Output: User–item preference scores $\hat{y}_{ui}$.

1: Initialize: Xavier initialized $\boldsymbol{e}_u$, $\boldsymbol{e}_i$, etc.
2: $\tilde{\mathcal{G}} = \{\tilde{\mathcal{V}}, \tilde{\mathcal{E}}\} \leftarrow$ PopularityDenoising $(\mathcal{G})$;
3: **for** epoch in range(epochs) **do**
4: **for** (u, i) in $\mathcal{G}$ **do**
// Cross-modal alignment fusion.
5: $\boldsymbol{e}_i^c$ = EarlyFusion($\boldsymbol{e}_i^v$, $\boldsymbol{e}_i^t$) based on Eq. (1) and (2);
// Multi-modal discrepancy learning.
6: d_i = DiscrepancyLearning($\boldsymbol{e}_i^v$, $\boldsymbol{e}_i^t$) based on Eq. (6);
// Constructing item–item graph.
7: $\mathcal{G}_m \leftarrow$ GraphConstruct($\boldsymbol{e}_i^v$, $\boldsymbol{e}_i^t$, $\boldsymbol{e}_i^c$) based on Eq. (7) and (8);
// Hierarchical fusion network.
8: $\boldsymbol{h}_i^m \leftarrow$ GraphConvolve($\mathcal{G}_m$, $\boldsymbol{e}_i$) based on Eq. (12) and (13);
9: $\boldsymbol{h}_u, \boldsymbol{h}_i \leftarrow$ GraphConvolve($\tilde{\mathcal{G}}$, $\boldsymbol{e}_u$, $\boldsymbol{e}_i$) based on Eq. (15) and (16);
10: $\boldsymbol{h}_i^f$ = LateFusion($\boldsymbol{h}_i^v, \boldsymbol{h}_i^t$) based on Eq. (17);
11: $\boldsymbol{h}_i^{mod}$ = HybridFusion($\boldsymbol{h}_i^c, \boldsymbol{h}_i^f, d_i$) based on Eq. (18);
12: $\hat{\boldsymbol{h}}_i^{mul}$ = GlobalFusion($\boldsymbol{h}_i^m, \boldsymbol{h}_i$) based on Eq. (19);
// Top-K recommendation.
13: $\hat{y}_{ui}$ = CosineSimilarity($\boldsymbol{h}_u$, $\hat{\boldsymbol{h}}_i^{mul}$) based on Eq. (20);
// Model optimization.
14: Joint training loss based on Eq. (25):
$\mathcal{L} = \mathcal{L}_{bpr}$ (Eq. (24)) + $\lambda_1 \mathcal{L}_{cmbpr}$ (Eq. (21)) + $\lambda_2 \mathcal{L}_{vae}$ (Eq. (23));
15: **end for**
16: Optimize the MLPs, VAEs, $\boldsymbol{e}_u$, and $\boldsymbol{e}_i$ with backpropagation;
17: **end for**
18: **return** $\hat{y}_{ui}$.

where $\hat{\boldsymbol{e}}_i^m$ denotes the output of the decoder in VAEs. In addition, to ensure the generative ability of the VAEs, it is necessary to align the output $q(\boldsymbol{z}_i^m \,|\boldsymbol{e}_i^m)$ of the encoder with the standard normal distribution [44]. Therefore, the loss for optimizing the VAEs is as follows:

$$\mathcal{L}_{vae} = \sum_{m\in\{v,t\}} \left(\mathcal{L}_{BCE}^m + D_{KL}\left(q(\boldsymbol{z}_i^m \,|\boldsymbol{e}_i^m) \,\|p(\boldsymbol{z})\right)\right), \tag{23}$$

where $p(\boldsymbol{z}) = \mathcal{N}(0\,|1)$, and the function $D_{KL}(\cdot)$ returns the KL divergence.

(iii) hierarchical fusion task $\mathcal{L}_{bpr}$.

To produce an accurate preference score, we also employ the BPR loss to optimize the ID embeddings of users and items.

$$\mathcal{L}_{bpr} = -\sum_{(u,i,j)\in D} \log\sigma_2\left(\hat{y}_{ui} - \hat{y}_{uj}\right), \tag{24}$$

where $\hat{y}_{u\cdot}$ is obtained from Eq. (20).

In summary, the joint loss for training our modal DGHNet is as follows:

$$\mathcal{L} = \mathcal{L}_{bpr} + \lambda_1 \mathcal{L}_{cmbpr} + \lambda_2 \mathcal{L}_{vae}, \tag{25}$$

where λ_1 and λ_2 are the hyperparameters that control the weights of $\mathcal{L}_{cmbpr}$ and $\mathcal{L}_{vae}$, respectively.

To clearly show the training process of DGHNet, we detail it in Algorithm 1. Using the historical interaction graph $\mathcal{G}$ and modal features $\boldsymbol{e}_i^v$ and $\boldsymbol{e}_i^t$ obtained using the pre-trained models, we first construct a purified user–item graph $\tilde{\mathcal{G}}$, which removes the popularity noise in line 2. Then, for each user–item pair (u, i) in graph $\mathcal{G}$, we obtain the cross-modal features in line 5 and calculate the discrepancy score d_i between $\boldsymbol{e}_i^v$ and $\boldsymbol{e}_i^t$ in line 6. Besides, we construct the item–item graphs $\mathcal{G}_m$ based on the visual, textual and cross-modal features in line 7. After that, the graph convolution operations are performed in lines 8 and 9 to obtain the high-level representations of users and items. Next, we fuse different types of item representations in lines 10, 11, and 12, and then compute the similarity between $\hat{\boldsymbol{h}}_i^{mul}$ and $\boldsymbol{h}_u$ to obtain the user–item preference scores in line 13. Finally, we design a joint loss in line 14 and optimize the model parameters with backpropagation in line 16.

Table 2
Statistics of the experimental datasets.

Datasets	#Users	#Items	#Interactions	Density
Baby	19,445	7,050	160,792	0.117%
Sports	35,598	18,357	296,337	0.045%
Clothing	39,387	23,033	278,677	0.031%

4. Experiments

In this section, we first present the research questions that guide the experiments in Section 4.1. Then, we provide detailed information on the datasets and evaluation metrics employed for the experiments in Section 4.2. Subsequently, we summarize the baselines in Section 4.3. Finally, Section 4.4 presents the experimental configuration and setup.

4.1. Research questions

The following research questions are proposed to guide our experiments in this study:

- **RQ1**: Can DGHNet achieve better performance than various state-of-the-art baselines on the multi-modal recommendation task? (See Section 5.1)
- **RQ2**: What is the efficiency of DGHNet in terms of computational complexity and memory cost? (See Section 5.2)
- **RQ3**: How is the contribution of each key component in DGHNet to the recommendation performance? (See Section 5.3.1)
- **RQ4**: How do different types of item embeddings affect the performance of DGHNet? (See Section 5.3.2)
- **RQ5**: How does the hyperparameter configuration affect the performance of DGHNet? (See Section 5.4)
- **RQ6**: Why does our fusion strategy achieve better recommendation performance? (See Section 5.5)

4.2. Datasets and evaluation metrics

Following Zhang et al. [6] and Zhou et al. [18], we conduct experiments on three categories of the public Amazon Reviews datasets[1]: (a) *Baby*, (b) *Sports and Outdoors*, and (c) *Clothing, Shoes and Jewelry*. For brevity, we refer to them as *Baby*, *Sports*, and *Clothing*, respectively. For each user or item, we use a 5-core setting so that it is associated with at least five interactions [6]. The filtering results are shown in Table 2. Following Zhou et al. [18], a visual pre-trained model [51] is employed to extract 4096-dimensional visual features for images and a language pre-trained model is employed [52] to extract 384-dimensional textual features for the title, description, category, and brand of each item.

To ensure a fair comparison, we adhere to the same evaluation protocols as described in the baseline BM3 [18]. Specifically, we randomly select 80% and 10% of the historical interactions for training and validation, respectively, and the remaining 10% as the test set. During the recommendation phase, all items that do not interact with the given user are considered candidates. In addition, we use Recall@K and NDCG@K as metrics to evaluate the recommendation performance of different models. Specifically:

[1] http://jmcauley.ucsd.edu/data/amazon/links.html.

(i) Recall@K: It measures the ratio of the number of relevant items in the top-K recommendations to the total number of relevant items. The calculation formula is as follows:

$$\text{Recall@}K = \frac{n_{\text{rel}}}{N_{\text{rel}}}, \tag{26}$$

where n_{rel} denotes the number of items that match the user's actual clicks in the top K recommendations and N_{rel} indicates the total number of items clicked by the user in the test set.

(ii) NDCG@K: It is the normalized discounted cumulative gain metric that considers the ranking quality of the items in the recommendation list, The formula is as follows:

$$\text{NDCG@}K = \frac{\text{DCG@}K}{\text{IDCG@}K}, \tag{27}$$

where,

$$\text{DCG@}K = \sum_{i=1}^{K} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)}, \tag{28}$$

and rel_i denotes the relevance score of the ith recommended item. IDCG@K refers to the DCG@K of the optimal ranking, which is the DCG@K of the sequence obtained by sorting in descending order according to the rel_i values.

In this work, we report the results under $K = 10$ and $K = 20$. To simplify the representation, we rename Recall@K and NDCG@K as R@K and N@K, respectively.

4.3. Baseline summary

To verify the effectiveness of DGHNet, we compare it with several state-of-the-art recommendation models, which can be categorized into the following two groups:

(i) General models comprise two models based on CF that are trained using only user–item interactions.

- **BPR** [39] proposes a loss called bayesian personalized ranking to train the representations of users and items with the framework of Matrix Factorization (MF) [53].
- **LightGCN** [32] removes the layers responsible for nonlinear activation and feature transformation in graph convolutional networks, thereby simplifying and streamlining the message passing between nodes in the user–item graph.

(ii) Multi-modal models comprise seven multi-modal recommendation models that leverage both multi-modal information and user–item interactions for recommendation.

- **VBPR** [38] employs a pre-trained model to extract visual features for each item and integrates them with ID embeddings of items to expand the BPR approach.
- **MMGCN** [14] applies GCN to user–item graphs of various modalities to derive the user preferences for each modality, which are then aggregated with the user behaviors to form the embeddings of users and items.
- **GRCN** [15] leverages multi-modal features to detect and prune fake interactions in the user–item graphs, and subsequently performs information aggregation and propagation within the refined graph structure.
- **DualGNN** [16] designs a dubbed dual graph neural network that constructs user co-occurrence graphs to mine hidden information among users with the same preferences.
- **LATTICE** [6] constructs item–item graphs that can efficiently extract the hidden semantic information between items to enhance item representations for recommendation.
- **SLMRec** [17] introduces self-supervised learning into multi-modal recommendation using three data augmentation techniques and contrastive learning for modeling optimization.
- **BM3** [18] processes the ID embeddings obtained by convolving on the user–item graph using dropout and MLPs, and refines the results by optimizing in the absence of negative samples using three contrastive losses.

4.4. Hyperparameter settings

We use the integrated multi-modal recommendation framework MMRec [54] to implement the proposed model and baselines. We set the embedding dimension of all models to 64 and initialize the model parameters using the Xavier initialization [55]. All models are implemented in Pytorch [56] and run on a server equipped with an NVIDIA RTX3090Ti GPU with 24 GB of memory. We optimize all models using the Adam [57] optimizer with a batch size of 2048. As noted in [49], adopting too many convolutional layers in graphs may lead to over-smoothing and may capture noisy features. Therefore, we set the number of GCN layers L_{ui} and L_{ii} in the user–item graph and item–item graphs to 2 and 1, respectively. Following Zhang et al. [6], we set n to 10 in Eq. (8). In addition, we empirically determine the hyperparameters λ_1 and λ_2 as 0.1 and 0.001, respectively. We tune the visual feature ratios α in $\{0.1, 0.2, 0.3, 0.4\}$ and search the ratios of popularity-sensitive edge pruning ρ in $\{0.6, 0.7, 0.8, 0.9\}$. We set the maximum number of epochs to 1000 and apply the early stopping strategy. Training terminates when the R@20 metric on the validation set does not increase for 20 consecutive epochs. To facilitate the reproducibility of our research findings and ensure the reliability and accuracy of the experimental results, we share the experimental code and data at https://github.com/Yuzhuo-Dang/DGHNet.

5. Results and discussion

In this section, we discuss the overall performance of DGHNet and baselines in Section 5.1. Then, we discuss modeling complexity in Section 5.2. Next, we explore the contributions of different modules and item embeddings in Section 5.3. Moreover, the effects of the hyperparameter settings on the performance of DGHNet are discussed in Section 5.4. Finally, the effectiveness of the model is confirmed through the visualizations presented in Section 5.5.

5.1. Overall performance

To answer RQ1, we assess our proposed DGHNet and the baselines by comparing their performances on the three datasets. Their overall performances in terms of R@10, R@20, N@10, and N@20 are presented in Table 3.

First, from the results of the baselines, it is evident that incorporating the multi-modal information of items enhances the recommendation performance compared with traditional CF methods. For instance, on the *Clothing* dataset, VBPR outperforms BPR by 48.39% in terms of R@20 by incorporating the item visual features extracted from the pre-trained model into the BPR model. Similarly, the corresponding improvements achieved by DualGNN and LATTICE above LightGCN are 28.33% and 39.16%, respectively.

Moreover, compared with the classical MF models (i.e., BPR and VBPR), the graph-based multi-modal models generally exhibit superior performance in most cases. However, MMGCN stands out as an exception, because it integrates information from all modalities, which makes it challenging to discern the precise influence of each modality individually. And GRCN simplifies the graph convolution process with the pruning strategies and achieves excellent performance on dense datasets. However, assigning low weights to low-similarity edges in sparse data environments limits the information propagation capability of GCN. In addition, DualGNN and LATTICE enhance user and item representation learning with auxiliary information, while LATTICE exhibits superior performance because of its effective capacity to weigh the importance of multi-modal features of items. Furthermore, SLMRec

Table 3
Overall performance achieved by different methods for Recall and NDCG. The global best and second-best results are highlighted in bold and underlined, respectively. The percentage improvement (impov.) is calculated as the ratio of the performance increase from the best baseline to DGHNet. To validate the stability of our method, we conducted experiments under five different randomized seeds, and the statistical significance of the pairwise differences in DGHNet relative to the strongest baseline was confirmed with a t-test p-value < 0.01(1e−2).

Model	*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
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
VBPR	0.0423	0.0663	0.0223	0.0284	0.0558	0.0856	0.0307	0.0384	0.0280	0.0414	0.0159	0.0193
MMGCN	0.0378	0.0615	0.0200	0.0261	0.0370	0.0605	0.0193	0.0254	0.0197	0.0328	0.0101	0.0135
GRCN	0.0532	0.0824	0.0282	0.0358	0.0559	0.0877	0.0306	0.0389	0.0424	0.0650	0.0225	0.0283
DualGNN	0.0513	0.0803	0.0278	0.0352	0.0588	0.0899	0.0324	0.0404	0.0452	0.0675	0.0242	0.0298
LATTICE	0.0545	0.0849	0.0289	0.0368	0.0618	0.0950	0.0334	0.0419	0.0491	0.0732	0.0266	0.0327
SLMRec	0.0538	0.0809	0.0284	0.0357	0.0660	0.0988	0.0359	0.0441	0.0441	0.0658	0.0238	0.0293
BM3	0.0563	0.0881	0.0298	0.0382	0.0654	0.0978	0.0351	0.0434	0.0420	0.0618	0.0229	0.0282
DGHNet	**0.0620**	**0.0985**	**0.0332**	**0.0425**	**0.0696**	**0.1070**	**0.0372**	**0.0468**	**0.0619**	**0.0927**	**0.0334**	**0.0412**
p-Value	4.01e−5	4.26e−5	9.87e−6	8.4e−6	1.05e−4	2.54e−6	6.91e−4	5.5e−5	7.25e−6	7.77e−6	2.84e−6	1.35e−6
improv.	10.05%	11.76%	11.28%	11.31%	5.39%	8.32%	3.62%	6.21%	26.15%	26.69%	25.41%	25.87%

Table 4
Efficiency comparison between DGHNet and the baselines.

Dataset	Metric	General model		Multi-modal model							
		BPR	LightGCN	VBPR	MMGCN	GRCN	DualGNN	LATTICE	SLMRec	BM3	DGHNet
Baby	Time (s/epoch)	0.46	0.97	0.55	3.41	2.05	4.11	0.98	1.65	0.86	1.78
	Memory (GB)	1.56	1.64	1.77	2.48	2.72	1.93	4.31	1.96	1.98	1.91
Sports	Time (s/epoch)	0.94	2.22	0.89	10.76	5.89	8.65	6.08	4.33	2.62	7.21
	Memory (GB)	1.96	2.21	2.58	3.71	4.25	2.59	17.80	2.80	3.02	4.30
Clothing	Time (s/epoch)	0.95	2.89	0.96	10.87	5.23	9.39	11.26	3.89	2.79	8.72
	Memory (GB)	2.12	2.43	2.80	3.88	4.32	2.80	23.89	3.10	3.51	5.60

constructs the data enhancement tasks using a self-supervised learning approach, obtaining better results than DualGNN and LATTICE on the *Sports* dataset. Last, BM3 employs dropout and MLPs for data augmentation and uses contrastive learning without negative samples to avoid the noise that may be introduced by randomly sampling negative samples in the BPR loss. This approach achieves superior performance while consuming few resources.

Next, we compare the baselines with DGHNet. Obviously, DGHNet outperforms both the general recommendation models and the multi-modal recommendation methods across all datasets, demonstrating the efficacy of our proposed method. Overall, DGHNet outperforms the best baseline in terms of R@20 by 11.76%, 8.32%, and 26.69% on *Baby*, *Sports*, and *Clothing*, respectively. This advantage is because DGHNet fuses cross-modal and uni-modal embeddings while designing the multi-modal discrepancy learning to adaptively learn their weights. That is, our designed fusion strategy effectively identifies and leverages complementary information between early fusion and late fusion strategies. Furthermore, denoising the user–item graph and constructing the item–item graphs can respectively mitigate the influence of popularity noise and excavate the intrinsic information of item modalities.

Moreover, it can be observed that as the number of recommendations increases from 10 to 20, the proportion of DGHNet's performance improvements consistently increases. This suggests that the learned accurate user and item embeddings enable DGHNet to be more effective in long recommendation lists compared to the traditional graph learning models (i.e., LightGCN, MMGCN, and GCRN). Next, compared with LATTICE, which also mines hidden information between items by constructing an item–item graph, DGHNet not only constructs a cross-modal item–item graph, but also preserves original uni-modal item–item graphs to uncover their hidden information, thus achieving performance improvement. Although some self-supervised learning methods (e.g., SLMRec and BM3) mitigate the impact of modal noise to a certain degree, our adaptive multi-modal discrepancy learning approach is capable of achieving superior results when calculating modal contributions. In particular, our approach exhibits better performance in recommendation scenarios with a large number of user–item interactions (i.e., *Clothing* dataset). This advantage can be attributed to our precise construction of item–item graphs and denoised user–item graphs, which enables the graph structure information to more closely align with the intrinsic connections in the actual data. It is also worth noting that the performance improvement on the *Sports* dataset is relatively minor. We conjecture that one potential reason is that the *Sports* dataset contains relatively less multi-modal information. We also observe that based on LightGCN, the performance improvement of GRCN and DualGNN that introduce multi-modal information is not significant on the *Sports* dataset.

5.2. Computational efficiency

In addition to evaluating the precision of the recommendations, we record the runtime and memory cost of DGHNet and the baselines during each training epoch to further analyze model complexity. The results for all models running on a 24 GB NVIDIA RTX3090Ti GPU are listed in Table 4.

We can observe that multi-modal models typically consume more time and memory than general models, as they need to process not only interaction information but also the multi-modal features of items. Moreover, GNN-based models exhibit higher time and memory consumption. Specifically, MMGCN performs convolutions on each modality-specific interaction graph, resulting in a long training time. LATTICE dynamically updates the item–item graph and requires more memory, but it shows quicker runtime on smaller datasets. In addition, self-supervised learning methods like SLMRec and BM3 do not require the dependency on graph learning, thereby reducing the consumption of memory space, but this is usually at the expense of recommendation accuracy. In contrast, DGHNet aims at bridging the gap between early and late fusion, thereby employing more graph convolutional modules to mine uni-modal and cross-modal features, which inevitably increases computational time and memory requirements. However, the designed denoising module in DGHNet does not lead to an exponential increase in its consumption which remains within the support range of current hardware technology. Therefore, the increased complexity of

Table 5
Performance comparison between DGHNet with its different variants. ↓ denotes the performance decrease percentage of the variants compared to DGHNet.

Model variant	*Baby*		*Sports*		*Clothing*	
	R@20	N@20	R@20	N@20	R@20	N@20
DGHNet w/o CA	$0.0923_{\downarrow 6.2\%}$	$0.0396_{\downarrow 6.8\%}$	$0.1042_{\downarrow 2.6\%}$	$0.0457_{\downarrow 2.4\%}$	$0.0904_{\downarrow 2.5\%}$	$0.0402_{\downarrow 2.3\%}$
DGHNet w/o DL	$0.0968_{\downarrow 1.7\%}$	$0.0419_{\downarrow 1.4\%}$	$0.1035_{\downarrow 3.3\%}$	$0.0453_{\downarrow 3.4\%}$	$0.0848_{\downarrow 8.6\%}$	$0.0376_{\downarrow 8.6\%}$
DGHNet w/o PD	$0.0895_{\downarrow 9.1\%}$	$0.0387_{\downarrow 9.1\%}$	$0.0981_{\downarrow 8.4\%}$	$0.0427_{\downarrow 8.8\%}$	$0.0848_{\downarrow 8.5\%}$	$0.0377_{\downarrow 8.4\%}$
DGHNet-RD	$0.0921_{\downarrow 6.5\%}$	$0.0397_{\downarrow 6.6\%}$	$0.1009_{\downarrow 5.7\%}$	$0.0440_{\downarrow 6.1\%}$	$0.0874_{\downarrow 5.8\%}$	$0.0388_{\downarrow 5.7\%}$
DGHNet-CD	$0.0956_{\downarrow 2.9\%}$	$0.0412_{\downarrow 3.1\%}$	$0.1046_{\downarrow 2.3\%}$	$0.0457_{\downarrow 2.4\%}$	$0.0904_{\downarrow 2.5\%}$	$0.0402_{\downarrow 2.3\%}$
DGHNet	**0.0985**	**0.0425**	**0.1070**	**0.0468**	**0.0927**	**0.0412**

DGHNet is justified, and its enhancement in multi-modal recommendation performance is obvious, especially when dealing with large-scale datasets.

5.3. Ablation study

To answer RQ2 and RQ3, we conduct an exhaustive ablation study on the different modules in DGHNet and the item embeddings contained in DGHNet.

5.3.1. Effects of the modules

To assess the impact of key components in DGHNet on the recommendation performance, we construct the following model variants:

- **DGHNet w/o CA**, which removes the MLPs in the cross-modal alignment fusion module to prevent the semantic alignment of visual and textual features extracted using pre-trained models.
- **DGHNet w/o DL**, which abandons the multi-modal discrepancy learning module, i.e., early fusion and late fusion are considered to be of equal importance in the hybrid fusion stage.
- **DGHNet w/o PD**, which excludes the popularity denoising module, i.e., the influence of popular nodes in the user–item graph on recommendations is not considered.
- **DGHNet-RD**, which replaces the popularity denoising strategy in DGHNet with a random edge dropout strategy [20].
- **DGHNet-CD**, which replaces the popularity denoising strategy in DGHNet with a graph convolution denoising strategy [21].

Table 5 presents the experimental results of DGHNet and the model variants on three datasets, from which we can obtain the following conclusions:

(1) By comparing DGHNet w/o CA and DGHNet, we can see that removing the CA module leads to a performance decrease. This suggests that bridging the semantic gap between the pre-trained model and the recommendation task is necessary, which can improve recommendation accuracy. In addition, the semantic gap has a more prominent influence on the model's effectiveness on the *Baby* dataset, which has a smaller data size.

(2) By comparing DGHNet w/o DL and DGHNet, we can observe that DL has an obvious effect on recommendation performance, indicating that early fusion and late fusion are not of equal importance. Moreover, as the size of the dataset increases, the discrepancies between modal features may become more intricate and varied. Therefore, effectively calculating these discrepancies will become more crucial, which can further enhance recommendation performance.

(3) By comparing DGHNet w/o PD, DGHNet exhibits improved recommendation performance, demonstrating the efficacy of the popularity denoising module. This could be explained by that users tend to frequently interact with popular items due to the factors such as conformity or unintentional clicks, which do not indeed reflect their real preferences. The denoising strategy is particularly effective on the *Baby* dataset, where interactions are more dense as observed from Table 2.

(4) By observing the performance of DGHNet-RD that employs the random edge dropout strategy and DGHNet-CD that uses the graph convolution denoising strategy, we can see that both of them show improvements over DGHNet w/o PD. This indicates the presence of noise in the user–item graph, so designing a suitable strategy for denoising can improve the recommendation performance. However, DGHNet-RD and DGHNet-CD show suboptimal results compared with DGHNet, illustrating that our proposed popularity denoising strategy is more effective in cleaning the user–item graph from noise.

5.3.2. Effects of the different types of item embeddings

To thoroughly investigate the influence of different modalities and fusion strategies on the recommendation accuracy, we introduce different types of item modal embeddings to interaction embeddings $\boldsymbol{h}_i$. In Fig. 2, **Visual** includes visual embeddings $\boldsymbol{h}_i^v$, **Textual** includes textual embeddings $\boldsymbol{h}_i^t$, **Early Fusion** includes cross-modal embeddings $\boldsymbol{h}_i^c$, and **Late Fusion** includes uni-modal fusion embeddings $\boldsymbol{h}_i^f$.

As shown in Fig. 2, **DGHNet** outperforms all other models with different item embeddings. In addition, by comparing **Visual** and **Textual**, we can observe that the influence of textual embeddings is more obvious. We think the cause of this phenomenon is that textual descriptions usually reflect the essential features of an item more accurately and visual information may be contaminated by less critical or even irrelevant elements [45]. Current visual models are not yet adept at identifying and emphasizing the relevant parts of images that are of interest to the user. Thus text assumes a more prominent role in the recommendation process. This observation is corroborated by the experimental findings presented in Section 5.4.1.

Comparing the different fusion strategies shows that, **Late Fusion** typically exhibits superior performance over **Early Fusion** on the *Baby* and *Sports* datasets. This is because the late fusion strategy is capable of effectively capturing the user's personalized preferences across different modalities. However, on the *Clothing* dataset, where the **Visual** and **Textual** results exhibit pronounced differences, **Early Fusion** outperforms **Late Fusion**. This outcome aligns with our earlier analysis in Section 1, which suggests that robust early fusion can effectively mitigate the discrepancies between modalities and enhances the overall prediction performance. In conclusion, **DGHNet** outperforms **Late Fusion** and **Early Fusion** on all three datasets, validating the efficacy of our hierarchical fusion network in integrating the information of the early fusion and late fusion strategies.

5.4. Hyperparameter sensitivity analysis

To answer RQ4, we perform the sensitivity experiments on the visual feature ratio α, the edge pruning ratio ρ, and the loss weights λ_1 and λ_2.

5.4.1. Effects of the visual feature ratio α

In the late fusion stage, the visual feature ratio α is employed to modulate the proportion of the visual and textual features. Specifically, if the ratio is set to 0.0, the item modal features are determined entirely by the textual and cross-modal features. When the ratio value is set to 1, the item features are based solelyon the visual and cross-modal features.

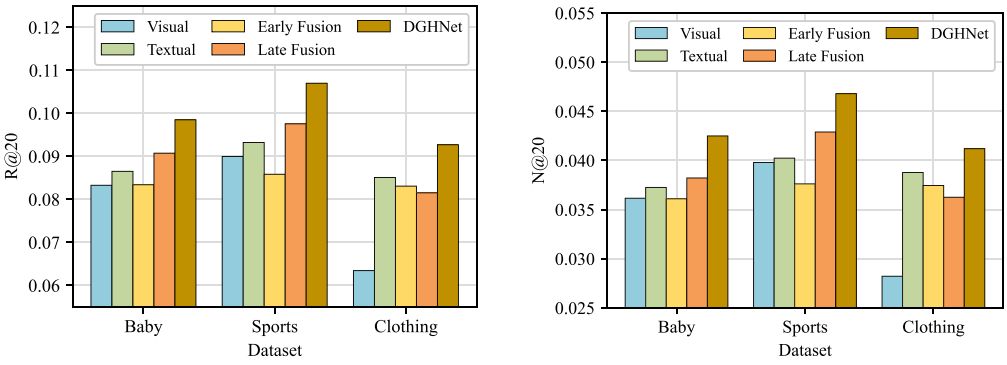


Fig. 2. Performance comparison of utilizing different types of item embeddings.

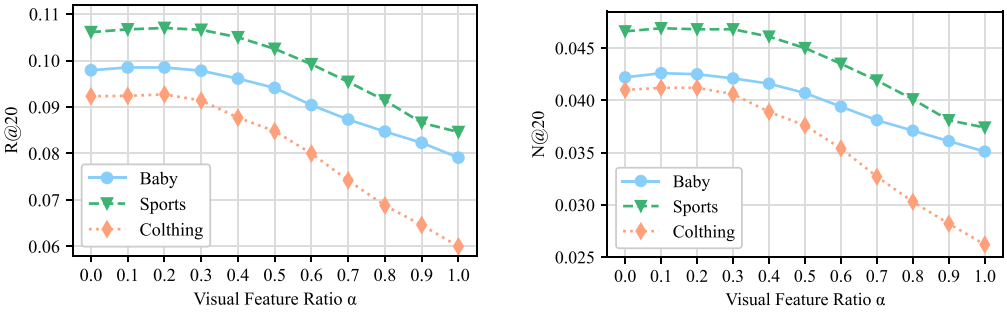


Fig. 3. Model performance under the visual feature ratios ranging from 0.0 to 1.0.

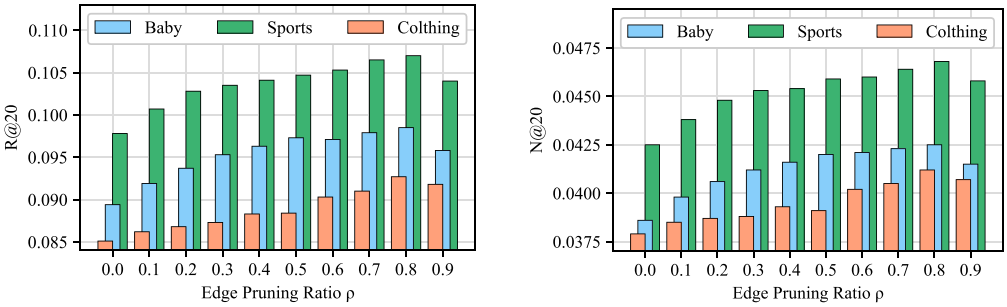


Fig. 4. Model performance under the edge pruning ratios ranging from 0.0 to 0.9.

The experimental results obtained on the three datasets are shown in Fig. 3. By comparing the performance across different visual feature ratios, we observe that the optimal results are achieved when α is set to 0.2. We can get that textual features are more obvious than visual features and contribute more to improving model performance. As the value of α exceeds 0.2 and continues to increase, model performance continuously degrades, with this effect being particularly pronounced on the *Clothing* dataset. This phenomenon can be attributed to the fact that certain images in the *Clothing* dataset may contain inaccurate descriptions or visual noise, which could mislead potential purchases, thereby reducing the overall contribution of visual features to the model's performance.

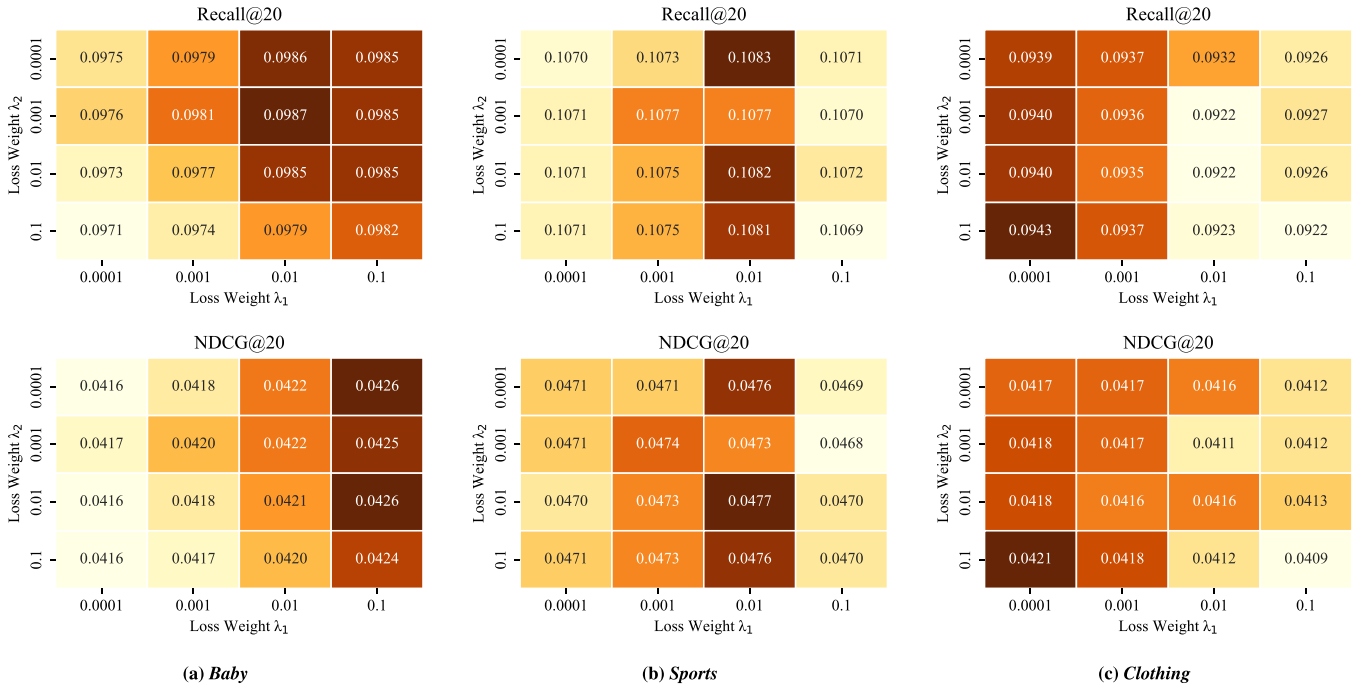
5.4.2. Effects of the edge pruning ratio ρ

In the popularity denoising user–item graph step, the edge pruning ratio ρ controls the number of edges in the purified user–item graph. Considering the feasibility of graph computation and the authenticity of user–item interactions, this purified graph cannot be an empty graph, which means that ρ cannot be equal to 1. Therefore, we adjust the value of ρ in the range of [0, 0.9].

Fig. 4 shows the model performance achieved by DGHNet on the three datasets as the edge pruning ratio varies. The results suggest that in a certain range from 0.0 to 0.8, an increased edge pruning ratio benefits the model performance. However, when the ratio exceeds 0.8, it cuts off essential connections for model training, leading to performance degradation, although it may eliminate unnecessary noisy edge information. In addition, the beneficial impact of improving the edge pruning ratio on the performance of DGHNet is more obvious on the *Baby* dataset, where user–item interactions are more abundant and the data density is higher, especially under the N@20 evaluation metric. This phenomenon can be attributed to the substantial amount of potentially invalid or noisy edge information present in such datasets. By implementing the popularity denoising step, the structure of the user–item graph can be effectively purified and optimized, thus ensuring that the model can extract a more accurate representation of the features during subsequent GNN processing. In summary, this refinement improves recommendation accuracy.

5.4.3. Effects of the loss weights λ_1 and λ_2

In the joint loss (Eq. (25)), λ_1 and λ_2 control the weights of cross-modal BPR loss and VAE loss, respectively, influencing the optimization direction of the whole model. We analyze the effects by tuning the values of both λ_1 and λ_2 in [0.1, 0.01, 0.001, 0.0001].



(a) *Baby* (b) *Sports* (c) *Clothing*

Fig. 5. Model performance under the loss weights in [0.1, 0.01, 0.001, 0.0001].

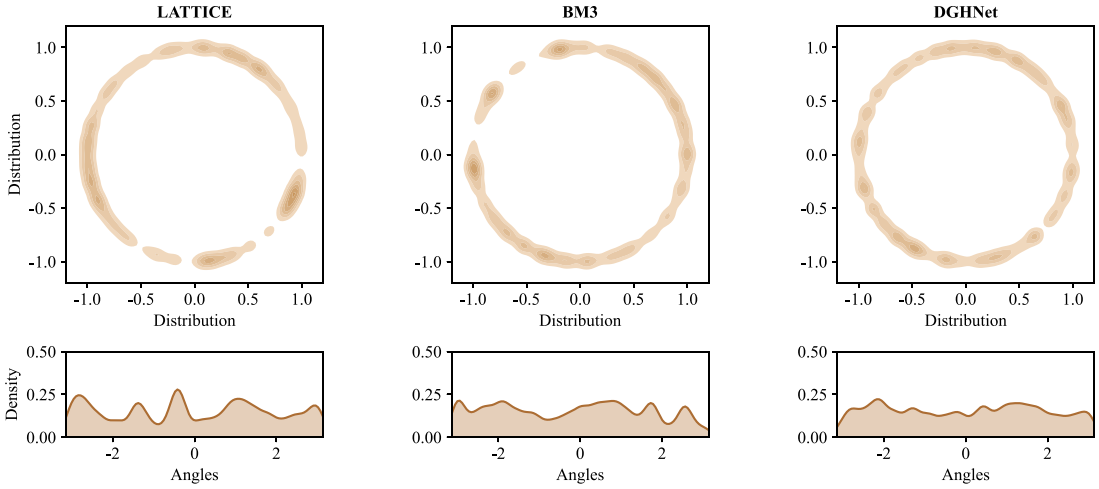


Fig. 6. The distribution of learned item representations on the *Baby* dataset from different models. The upper figures show the spatial distribution, where darker colors indicate a higher concentration of points in a particular region; and the lower figures depict the distribution density of points across different angular directions.

Fig. 5 displays the performance of DGHNet on the three datasets corresponding to the parameter combinations of different λ_1 and λ_2 values. We can observe that on the *Baby* dataset, DGHNet performs best with λ_1 set to 0.1 or 0.01. On the *Sports* dataset, the model exhibits superior performance with λ_1 of 0.01. On the *Clothing* dataset, the model achieves its optimal performance when λ_1 is set to 0.0001 and λ_2 to 0.1. These findings suggest the importance of adjusting the weight assignment of each parameter in the loss to match the characteristics of different datasets. This observation aligns with the earlier finding presented in Section 5.3.1, which shows the varying importance of the CA and DL modules across different datasets. Taking the *Baby* and *Clothing* datasets as examples, the CA module is found to be more crucial than the DL module in the *Baby* dataset and the DL module assumes greater prominence in the *Clothing* dataset. This differentiation is mirrored in the selection of loss parameter weights: increasing the weight of the cross-modal BPR loss (i.e., λ_1) for the CA module increases model performance on the *Baby* dataset, whereas amplifying the weight of the VAE loss (i.e., λ_2) for the DL module has a more pronounced effect on the model performance in the *Clothing* dataset.

5.5. Visualization analysis

To answer RQ5, we visualize the final item representations of LATTICE, BM3, and DGHNET on the *Baby* dataset. As shown in Fig. 6, we select 500 item samples from the *Baby* dataset and employ tSNE [58] to reduce the final item representation vectors to 2D space. Subsequently, we apply Gaussian kernel density estimation (KDE) [59] to depict their feature distributions as a visualization of the different effects of the three models when dealing with multi-modal feature fusion.

Previous studies [49,60,61] have demonstrated that a uniform distribution of item representations helps in preserving the intrinsic attributes of nodes and enhances their generalization capabilities, thereby improving recommendation performance. As observed in Fig. 6, DGHNet generates the item features with a more homogeneous distribution

compared with LATTICE and BM3 models, which shows multiple aggregations and inhomogeneities. This suggests that DGHNet excels in integrating the multi-modal features of items, with its advantage primarily stemming from the hierarchical fusion network. This network adaptively combines the complementary information between early fusion and late fusion, thereby enhancing the efficacy of multi-modal information.

6. Conclusion and future work

In this paper, we design a discrepancy learning guided hierarchical fusion network (DGHNet) for multi-modal recommendation. Specifically, we propose a method that integrates early fusion and late fusion information to address the issue of modal discrepancy, using a multi-modal discrepancy learning module to quantify discrepancies and guide the hierarchical fusion network to adaptively fuse cross-modal and uni-modal embeddings. Furthermore, we design a popularity-aware pruning strategy to remove interactions triggered by users' unintentional behaviors and popularity trends, thereby addressing the issue of popularity noise. Experiments conducted on three real-world datasets, i.e., *Baby*, *Sports*, and *Clothing*, verify the superior performance of DGHNet in terms of the evaluation metrics Recall@K and NDCG@K.

Nevertheless, the current DGHNet assumes equal importance between modal and interaction information, failing to distinguish their contributions. Thus, in the future, we are planning to explore the dynamic and adaptive integration between the multi-modal and interaction information to construct more precise item representations, thereby improving recommendation performance.

CRediT authorship contribution statement

Yuzhuo Dang: Writing – original draft, Validation, Methodology, Conceptualization. **Zhiqiang Pan:** Validation, Methodology, Formal analysis. **Xin Zhang:** Formal analysis, Conceptualization. **Wanyu Chen:** Project administration, Investigation. **Fei Cai:** Writing – review & editing, Investigation. **Honghui Chen:** Resources, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the support by the COSTA: complex system optimization team of the College of System Engineering at NUDT. This research was funded by the National Natural Science Foundation of China under Grant No. 62302511, Basic Strengthening Program under Grant No. 2023-JCJQ-QT-048, and Independent Innovation Science foundation project of National University of Defense Technology under Grant No. 23-ZZCX-JDZ-43.

Data availability

Data will be made available on request.

References

[1] L. Lü, M. Medo, C.H. Yeung, Y.-C. Zhang, Z.-K. Zhang, T. Zhou, Recommender systems, Phys. Rep. 519 (1) (2012) 1–49, http://dx.doi.org/10.1016/j.physrep.2012.02.006.

[2] P. Covington, J. Adams, E. Sargin, Deep neural networks for YouTube recommendations, in: RecSys, ACM, 2016, pp. 191–198, http://dx.doi.org/10.1145/2959100.2959190.

[3] Y. Guo, Y. Ling, H. Chen, A neighbor-guided memory-based neural network for session-aware recommendation, IEEE Access 8 (2020) 120668–120678, http://dx.doi.org/10.1109/ACCESS.2020.3006360.

[4] H.J. Jeong, K.H. Lee, M. Kim, DGC: dynamic group behavior modeling that utilizes context information for group recommendation, Knowl.-Based Syst. 213 (2021) 106659, http://dx.doi.org/10.1016/J.KNOSYS.2020.106659.

[5] C. Chang, J. Zhou, Y. Weng, X. Zeng, Z. Wu, C. Wang, Y. Tang, KGTN: knowledge graph transformer network for explainable multi-category item recommendation, Knowl.-Based Syst. 278 (2023) 110854, http://dx.doi.org/10.1016/J.KNOSYS.2023.110854.

[6] J. Zhang, Y. Zhu, Q. Liu, S. Wu, S. Wang, L. Wang, Mining latent structures for multimedia recommendation, in: ACM Multimedia, ACM, 2021, pp. 3872–3880, http://dx.doi.org/10.1145/3474085.3475259.

[7] S. Tao, R. Qiu, Y. Ping, H. Ma, Multi-modal knowledge-aware reinforcement learning network for explainable recommendation, Knowl.-Based Syst. 227 (2021) 107217, http://dx.doi.org/10.1016/J.KNOSYS.2021.107217.

[8] L. Wu, X. He, X. Wang, K. Zhang, M. Wang, A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation, IEEE Trans. Knowl. Data Eng. 35 (5) (2023) 4425–4445, http://dx.doi.org/10.1109/TKDE.2022.3145690.

[9] S. Kim, S. Yun, J. Lee, G. Chang, W. Roh, D. Sohn, J. Lee, H. Park, S. Kim, Self-supervised multimodal graph convolutional network for collaborative filtering, Inf. Sci. 653 (2024) 119760, http://dx.doi.org/10.1016/J.INS.2023.119760.

[10] D. Cao, L. Miao, H. Rong, Z. Qin, L. Nie, Hashtag our stories: Hashtag recommendation for micro-videos via harnessing multiple modalities, Knowl.-Based Syst. 203 (2020) 106114, http://dx.doi.org/10.1016/J.KNOSYS.2020.106114.

[11] H. Zhou, X. Zhou, Z. Zeng, L. Zhang, Z. Shen, A comprehensive survey on multimodal recommender systems: Taxonomy, evaluation, and future directions, 2023, http://dx.doi.org/10.48550/ARXIV.2302.04473, CoRR abs/2302.04473.

[12] F. Zhang, N.J. Yuan, D. Lian, X. Xie, W. Ma, Collaborative knowledge base embedding for recommender systems, in: KDD, ACM, 2016, pp. 353–362, http://dx.doi.org/10.1145/2939672.2939673.

[13] F. Liu, Z. Cheng, C. Sun, Y. Wang, L. Nie, M.S. Kankanhalli, User diverse preference modeling by multimodal attentive metric learning, in: ACM Multimedia, ACM, 2019, pp. 1526–1534, http://dx.doi.org/10.1145/3343031.3350953.

[14] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, T. Chua, MMGCN: multi-modal graph convolution network for personalized recommendation of micro-video, in: ACM Multimedia, ACM, 2019, pp. 1437–1445, http://dx.doi.org/10.1145/3343031.3351034.

[15] Y. Wei, X. Wang, L. Nie, X. He, T. Chua, Graph-refined convolutional network for multimedia recommendation with implicit feedback, in: ACM Multimedia, ACM, 2020, pp. 3541–3549, http://dx.doi.org/10.1145/3394171.3413556.

[16] Q. Wang, Y. Wei, J. Yin, J. Wu, X. Song, L. Nie, DualGNN: Dual graph neural network for multimedia recommendation, IEEE Trans. Multimed. 25 (2021) 1074–1084, http://dx.doi.org/10.1109/TMM.2021.3138298.

[17] Z. Tao, X. Liu, Y. Xia, X. Wang, L. Yang, X. Huang, T. Chua, Self-supervised learning for multimedia recommendation, IEEE Trans. Multimed. 25 (2023) 5107–5116, http://dx.doi.org/10.1109/TMM.2022.3187556.

[18] X. Zhou, H. Zhou, Y. Liu, Z. Zeng, C. Miao, P. Wang, Y. You, F. Jiang, Bootstrap latent representations for multi-modal recommendation, in: WWW, ACM, 2023, pp. 845–854, http://dx.doi.org/10.1145/3543507.3583251.

[19] Z. Mu, Y. Zhuang, J. Tan, J. Xiao, S. Tang, Learning hybrid behavior patterns for multimedia recommendation, in: ACM Multimedia, ACM, 2022, pp. 376–384, http://dx.doi.org/10.1145/3503161.3548119.

[20] Y. Rong, W. Huang, T. Xu, J. Huang, Dropedge: Towards deep graph convolutional networks on node classification, in: ICLR, OpenReview.net, 2020, p. n.p..

[21] S. Li, D. Guo, K. Liu, R. Hong, F. Xue, Multimodal counterfactual learning network for multimedia-based recommendation, in: SIGIR, ACM, 2023, pp. 1539–1548, http://dx.doi.org/10.1145/3539618.3591739.

[22] J. Zheng, F. Cai, Y. Ling, H. Chen, Heterogeneous graph neural networks to predict what happen next, in: COLING, International Committee on Computational Linguistics, 2020, pp. 328–338, http://dx.doi.org/10.18653/V1/2020.COLING-MAIN.29.

[23] C. Chen, F. Cai, X. Hu, W. Chen, H. Chen, HHGN: a hierarchical reasoning-based heterogeneous graph neural network for fact verification, Inf. Process. Manag. 58 (5) (2021) 102659, http://dx.doi.org/10.1016/J.IPM.2021.102659.

[24] D. Xu, Y. Chen, N. Cui, J. Li, Towards multi-dimensional knowledge-aware approach for effective community detection in LBSN, World Wide Web (WWW) 26 (4) (2023) 1435–1458, http://dx.doi.org/10.1007/S11280-022-01101-7.

[25] Z. Pan, F. Cai, X. Liu, H. Chen, Distance-aware learning for inductive link prediction on temporal networks, IEEE Trans. Neural Netw. Learn. Syst. (2023).

[26] Z. Pan, F. Cai, W. Chen, H. Chen, M. de Rijke, Star graph neural networks for session-based recommendation, in: CIKM, ACM, 2020, pp. 1195–1204, http://dx.doi.org/10.1145/3340531.3412014.
[27] S. Wang, L. Hu, Y. Wang, X. He, Q.Z. Sheng, M.A. Orgun, L. Cao, F. Ricci, P.S. Yu, Graph learning based recommender systems: A review, in: IJCAI, ijcai.org, 2021, pp. 4644–4652, http://dx.doi.org/10.24963/IJCAI.2021/630.
[28] H. Yang, J. Wang, R. Duan, C. Yan, DCOM-GNN: a deep clustering optimization method for graph neural networks, Knowl.-Based Syst. 279 (2023) 110961, http://dx.doi.org/10.1016/J.KNOSYS.2023.110961.
[29] R. van den Berg, T.N. Kipf, M. Welling, Graph convolutional matrix completion, 2017, CoRR abs/1706.02263. URL: https://doi.org/10.48550/arXiv.1706.02263.
[30] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, J. Tang, Session-based social recommendation via dynamic graph attention networks, in: WSDM, ACM, 2019, pp. 555–563, http://dx.doi.org/10.1145/3289600.3290989.
[31] L. Chen, L. Wu, R. Hong, K. Zhang, M. Wang, Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach, in: AAAI, AAAI Press, 2020, pp. 27–34, http://dx.doi.org/10.1609/AAAI.V34I01.5330.
[32] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, LightGCN: Simplifying and powering graph convolution network for recommendation, in: SIGIR, ACM, 2020, pp. 639–648, http://dx.doi.org/10.1145/3397271.3401063.
[33] H. Zhou, Q. Tan, X. Huang, K. Zhou, X. Wang, Temporal augmented graph neural networks for session-based recommendations, in: SIGIR, ACM, 2021, pp. 1798–1802, http://dx.doi.org/10.1145/3404835.3463112.
[34] J. Zhang, C. Ma, X. Mu, P. Zhao, C. Zhong, A. Ruhan, Recurrent convolutional neural network for session-based recommendation, Neurocomputing 437 (2021) 157–167, http://dx.doi.org/10.1016/J.NEUCOM.2021.01.041.
[35] C. Zhang, Q. Liu, Z. Zhang, DSGNN: a dynamic and static intentions integrated graph neural network for session-based recommendation, Neurocomputing 468 (2022) 222–232, http://dx.doi.org/10.1016/J.NEUCOM.2021.10.028.
[36] L. Yang, S. Wang, Y. Tao, J. Sun, X. Liu, P.S. Yu, T. Wang, Dgrec: Graph neural network for recommendation with diversified embedding generation, in: WSDM, ACM, 2023, pp. 661–669, http://dx.doi.org/10.1145/3539597.3570472.
[37] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: ICLR (Poster), OpenReview.net, 2018, p. n.p. URL: https://openreview.net/forum?id=rJXMpikCZ.
[38] R. He, J.J. McAuley, VBPR: visual Bayesian personalized ranking from implicit feedback, in: AAAI, AAAI Press, 2016, pp. 144–150, http://dx.doi.org/10.1609/AAAI.V30I1.9973.
[39] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in: UAI, AUAI Press, 2009, pp. 452–461.
[40] C. Xu, Z. Guan, W. Zhao, Q. Wu, M. Yan, L. Chen, Q. Miao, Recommendation by users' multimodal preferences for smart city applications, IEEE Trans. Ind. Inform. 17 (6) (2021) 4197–4205, http://dx.doi.org/10.1109/TII.2020.3008923.
[41] Z. Tao, Y. Wei, X. Wang, X. He, X. Huang, T. Chua, MGAT: multimodal graph attention network for recommendation, Inf. Process. Manag. 57 (5) (2020) 102277, http://dx.doi.org/10.1016/J.IPM.2020.102277.
[42] X. Zhou, Z. Shen, A tale of two graphs: Freezing and denoising graph structures for multimodal recommendation, in: ACM Multimedia, ACM, 2023, pp. 935–943, http://dx.doi.org/10.1145/3581783.3611943.
[43] M. Yan, H. Huang, Y. Liu, J. Zhao, X. Gao, C. Xu, Z. Guan, W. Zhao, TruthSR: Trustworthy sequential recommender systems via user-generated multimodal content, 2024, http://dx.doi.org/10.48550/ARXIV.2404.17238, CoRR abs/2404.17238.
[44] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, in: ICLR, 2014, n.p..
[45] K. Liu, F. Xue, D. Guo, P. Sun, S. Qian, R. Hong, Multimodal graph contrastive learning for multimedia-based recommendation, IEEE Trans. Multimed. 25 (2023) 9343–9355, http://dx.doi.org/10.1109/TMM.2023.3251108.
[46] Y. Chen, D. Li, P. Zhang, J. Sui, Q. Lv, T. Lu, L. Shang, Cross-modal ambiguity learning for multimodal fake news detection, in: WWW, ACM, 2022, pp. 2897–2905, http://dx.doi.org/10.1145/3485447.3511968.
[47] M. Wang, Y. Lin, G. Lin, K. Yang, X. Wu, M2GRL: a multi-task multi-view graph representation learning framework for web-scale recommender systems, in: KDD, ACM, 2020, pp. 2349–2358, http://dx.doi.org/10.1145/3394486.3403284.
[48] K. Mao, J. Zhu, X. Xiao, B. Lu, Z. Wang, X. He, UltraGCN: Ultra simplification of graph convolutional networks for recommendation, in: CIKM, ACM, 2021, pp. 1253–1262, http://dx.doi.org/10.1145/3459637.3482291.
[49] P. Yu, Z. Tan, G. Lu, B. Bao, Multi-view graph convolutional network for multimedia recommendation, in: ACM Multimedia, ACM, 2023, pp. 6576–6585, http://dx.doi.org/10.1145/3581783.3613915.
[50] J. Zhu, G. Mao, C. Jiang, DII-GCN: dropedge based deep graph convolutional networks, Symm. 14 (4) (2022) 798, http://dx.doi.org/10.3390/SYM14040798.
[51] J. Ni, J. Li, J.J. McAuley, Justifying recommendations using distantly-labeled reviews and fine-grained aspects, in: EMNLP/IJCNLP (1), Association for Computational Linguistics, 2019, pp. 188–197, http://dx.doi.org/10.18653/V1/D19-1018.
[52] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT-networks, in: EMNLP/IJCNLP (1), Association for Computational Linguistics, 2019, pp. 3980–3990, http://dx.doi.org/10.18653/V1/D19-1410.
[53] Y. Koren, R.M. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Comput. 42 (8) (2009) 30–37, http://dx.doi.org/10.1109/MC.2009.263.
[54] X. Zhou, MMRec: Simplifying multimodal recommendation, in: MMAsia (Workshops), ACM, 2023, pp. 6:1–6:2, http://dx.doi.org/10.1145/3611380.3628561.
[55] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: AISTATS, in: JMLR Proceedings, vol. 9, JMLR.org, 2010, pp. 249–256.
[56] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E.Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: NeurIPS, 2019, pp. 8024–8035.
[57] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR (Poster), 2015, n.p..
[58] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).
[59] G.R. Terrell, D.W. Scott, Variable kernel density estimation, Ann. Statist. (1992) 1236–1265.
[60] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, Q.V.H. Nguyen, Are graph augmentations necessary?: Simple graph contrastive learning for recommendation, in: SIGIR, ACM, 2022, pp. 1294–1303, http://dx.doi.org/10.1145/3477495.3531937.
[61] C. Wang, Y. Yu, W. Ma, M. Zhang, C. Chen, Y. Liu, S. Ma, Towards representation alignment and uniformity in collaborative filtering, in: KDD, ACM, 2022, pp. 1816–1825, http://dx.doi.org/10.1145/3534678.3539253.