Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.085768

ARTICLE

DDGEM: Diffusion Denoising and Generative Enhancement for Multimodal Recommendation

Weiwei Li*, Li Zhao, Chengshan Li and Wenjie Geng

School of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China

*Corresponding Author: Weiwei Li. Email: liww@cqut.edu.cn

Received: 18 May 2026; Accepted: 10 August 2026; Published: 15 September 2026

ABSTRACT: In multimodal recommendation, sparse implicit feedback can lead to noisy collaborative graphs, while long-tail and cold-start items often lack reliable collaborative signals. Textual and visual features provide useful item-side information, but they may also be incomplete or inconsistent across modalities. These issues make robust user and item representation learning difficult. To address them, we propose Diffusion Denoising and Generative Enhancement for Multimodal Recommendation (DDGEM). DDGEM first applies node-wise diffusion denoising in the latent collaborative space to reduce unreliable user–item signals. It then uses relational diffusion to reconstruct adaptive item–item relations instead of relying on fixed semantic similarity. For sparse and cold-start items, a conditional variational autoencoder (CVAE)-based conditional generative module produces complementary item representations from textual and visual features, followed by popularity-aware generation balance. Finally, behavior-aware multimodal fusion dynamically combines collaborative, textual, and visual representations for each user–item pair. Experiments on three public Amazon datasets show that DDGEM outperforms representative collaborative filtering, generative recommendation, and multimodal recommendation baselines. Further significance tests, ablation studies, efficiency comparison, graph-noise robustness analysis, strict cold-start evaluation, modality-conflict analysis, and hyperparameter analysis verify the effectiveness and practical behavior of DDGEM.

KEYWORDS: Multimodal recommendation; diffusion denoising; generative enhancement; behavior-aware fusion

1 Introduction

Recommender systems have become a core component of modern online services, including e-commerce, social media, and digital entertainment. Most recommendation models infer user preference from historical user–item interactions [1,2]. This paradigm is effective when feedback is dense and reliable, but it is fragile under sparse implicit interactions. A click, view, or purchase may reflect real preference, but it may also result from accidental exposure, temporary interest, or popularity bias. Once such unreliable signals are encoded into the collaborative graph, graph-based models may repeatedly propagate them and degrade user and item representations.

Multimodal recommendation introduces item-side content, such as textual descriptions and visual features, to alleviate sparsity and cold-start issues [3,4]. However, additional modalities are not always reliable. Textual and visual features may be noisy, incomplete, or even inconsistent with each other. Existing methods still leave three issues insufficiently addressed. First, matrix factorization methods [2,5], neural collaborative filtering models [6], and graph-based methods [7] are often trained on imperfect feedback, while recent contrastive learning and multimodal denoising methods [8–11] mainly reduce feature-level

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

noise and rarely model the perturbation and recovery of collaborative embeddings. Second, explicit item–item graphs [12,13] are usually constructed by fixed similarity measures before training, making them less adaptive to uncertain semantic relations. Third, long-tail and cold-start items lack reliable collaborative anchors, and existing content–collaboration fusion methods [14] often overlook sparse-item uncertainty and the popularity-dependent balance between content and collaborative signals.

To improve representation reliability under these conditions, this paper proposes **Diffusion Denoising and Generative Enhancement for Multimodal Recommendation**, termed **DDGEM**. Instead of loosely combining several existing techniques, DDGEM follows a structured representation flow. It first applies node-wise diffusion denoising to refine collaborative user and item embeddings. The denoised item representations are then used for relational diffusion graph reconstruction, where candidate item–item relations are adaptively refined rather than fixed by one-shot similarity calculation. For sparse and cold-start items, a CVAE-based conditional generative module produces complementary item representations from textual and visual semantics. Finally, a behavior-aware fusion module dynamically combines collaborative, textual, and visual signals for each user–item pair, allowing unreliable modalities to be down-weighted when modality quality is imbalanced or conflicting.

The main contributions of this paper are summarized as follows:

- A unified multimodal recommendation framework, DDGEM, is proposed to improve representation robustness under sparse interactions, noisy collaborative structures, and modality imbalance.
- Node-wise and relation-level diffusion modules are designed to denoise collaborative embeddings and adaptively reconstruct item–item relations.
- A CVAE-based conditional generative enhancement module with popularity-aware fusion is developed to improve item representations for long-tail and cold-start scenarios.
- A behavior-aware multimodal fusion strategy is introduced to dynamically integrate collaborative, textual, and visual signals. Experiments on three public datasets, together with ablation, significance, efficiency, robustness, cold-start, and modality-conflict analyses, verify the effectiveness of DDGEM.

2 Related Work

2.1 Collaborative Filtering–Based and Graph Neural Network(GNN)–Based Recommendation

Collaborative filtering is a fundamental paradigm in recommender systems. Matrix factorization and pairwise ranking methods infer user preference from historical interactions [15,16], while neural recommendation models enhance representation learning through nonlinear user–item interaction functions [6]. These methods are effective but depend heavily on the quality and density of observed feedback.

Graph neural networks further model high-order collaborative signals by propagating messages over the user–item bipartite graph [17]. NGCF explicitly encodes high-order connectivity [7], and LightGCN simplifies graph convolution by retaining only neighborhood aggregation [18]. Recent graph recommendation methods also incorporate contrastive learning or structural motifs to improve representation robustness [19,20]. A common limitation is that the observed interaction graph is usually treated as reliable. When implicit feedback contains accidental clicks, weak preferences, or popularity bias, graph propagation may also spread noise. This motivates denoising-oriented collaborative representation learning.

2.2 Multimodal Recommendation

Multimodal recommendation uses item-side content, such as images and textual descriptions, to alleviate sparsity and cold-start problems. VBPR introduces visual features into Bayesian personalized ranking [9]; MMGCN models modality-specific user preference through graph convolution [10]; LATTICE

constructs modality-aware item–item graphs, and BM3 simplifies multimodal contrastive learning by bootstrapping latent representations [11,14]. More recent studies investigate multimodal denoising, graph structure learning, and adaptive modality fusion [21–25]. However, multimodal content is not always reliable. Item images may contain background noise, style variation, or domain-specific visual bias, while textual descriptions may emphasize attributes weakly related to user preference. Therefore, static modality fusion may amplify unreliable content signals, while user–item-specific fusion is more suitable for multimodal preference modeling.

2.3 Generative and Diffusion Models for Recommendation

Generative models have been used to improve recommendation under sparse and cold-start scenarios. DropoutNet maps item content features to collaborative embeddings [26]. Variational autoencoders and conditional variational autoencoders model latent preference distributions and content-conditioned item representations [27–31]. GoRec further generates cold-start item representations from multimodal content and collaborative anchors [32]. These methods improve content-based representation learning, but they pay limited attention to noisy collaborative structures and adaptive item–item relations.

Diffusion models have recently been introduced into recommendation. DiffKG filters noisy knowledge graph signals through diffusion-based denoising [33]. DiffRec models user–item interaction generation with a denoising process [34]. Sequential diffusion methods refine user interaction sequences [35–37], and CDiff4Rec uses reviews to guide collaborative diffusion [38]. DiffMM further applies multimodal diffusion to recommendation representation learning [39]. These studies show the value of diffusion models, but most of them focus on sequence generation, knowledge graph denoising, or modality-level enhancement. Jointly modeling collaborative denoising, relation reconstruction, generative item enhancement, and behavior-aware fusion remains underexplored.

Diffusion-based generative augmentation has also been studied in computer vision. GenMix performs diffusion image editing for visual data augmentation [40], while DiffuseMix generates label-preserving augmented images with diffusion models [41]. Architecturally, these methods place diffusion models before the downstream task model and use them to synthesize or edit raw visual samples. DDGEM follows a different design. It does not synthesize images or texts, nor does it use diffusion as an external augmentation module. Instead, DDGEM embeds diffusion inside the recommendation architecture to denoise collaborative embeddings and refine item relations, while the CVAE module generates latent item representations for user–item ranking.

In addition, predictor-corrector aided discretized zeroing neural networks highlight the value of stable iterative modeling in dynamic optimization [42]. Although this line of work is methodologically different from recommendation, it provides a useful perspective on iterative refinement under noisy conditions.

3 Method

3.1 Overall Framework of DDGEM

Let $\mathcal{U}$, $\mathcal{I}$, and $\mathcal{E} \subseteq \mathcal{U} \times \mathcal{I}$ denote the user set, item set, and implicit interaction set. Based on $\mathcal{E}$, we construct a user–item graph $\mathcal{G}_{ui}$. Each user u and item i has an identifier (ID) embedding $\mathbf{e}_u$ and $\mathbf{e}_i$, while each item is also associated with textual and visual features $\mathbf{x}_i^{(t)}$ and $\mathbf{x}_i^{(v)}$.

As shown in Fig. 1, DDGEM follows a sequential representation refinement flow. The graph representation layer first performs K-hop message passing on $\mathcal{G}_{ui}$ to obtain preliminary collaborative embeddings $\mathbf{c}_u$ and $\mathbf{c}_i$. Textual and visual features are projected and propagated on modality-specific graphs to obtain $\mathbf{e}_i^{(t)}$ and $\mathbf{e}_i^{(v)}$. Then, the node-wise diffusion denoiser reconstructs the noisy collaborative embedding $\mathbf{c}_n$ and

outputs $\mathbf{z}_u$ and $\mathbf{z}_i$. The relational diffusion graph reconstructor uses the multimodal semantic representation $\mathbf{m}_i$ and $\mathbf{z}_i$ to rebuild the item–item relation matrix $\mathbf{A}^{(rel)}$ and obtain $\tilde{\mathbf{z}}_i$. The conditional generative module further generates $\mathbf{z}_i^{(g)}$ from $\tilde{\mathbf{z}}_i$ and the masked semantic representation $\tilde{\mathbf{m}}i$. Finally, the behavior-aware multimodal fusion module combines collaborative, textual, and visual signals into $\mathbf{z}_{u,i}^{mm}$ for bayesian personalized ranking (BPR)-based ranking.

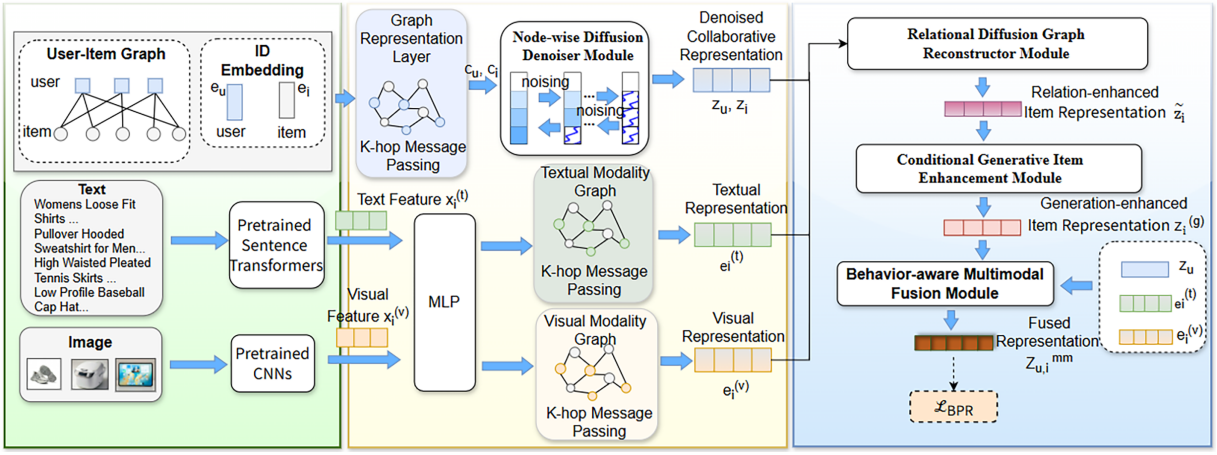


Figure 1: Overall framework of the Diffusion Denoising and Generative Enhancement for Multimodal Recommendation (DDGEM) model. ID, identifier; MLP, multilayer perceptron; CNN, convolutional neural network.

The core representation path of DDGEM is $\mathbf{c}_n \to \mathbf{z}_n \to \tilde{\mathbf{z}}_i \to \mathbf{z}_i^{(g)} \to \mathbf{z}_{u,i}^{mm}$. In this path, node-wise diffusion refines the collaborative representation, relational diffusion reconstructs item–item structures based on denoised and multimodal evidence, conditional generation enhances sparse item representations, and behavior-aware fusion produces user–item-specific multimodal representations for ranking. This design also differs from GenMix and DiffuseMix, which apply diffusion to raw image generation or editing before downstream learning. DDGEM embeds diffusion inside the recommendation architecture and performs denoising and generation in the latent recommendation space.

3.2 Node-Wise Diffusion Denoiser Module

The graph representation layer outputs the preliminary collaborative embedding $\mathbf{c}_n$ for each user or item node $n \in \mathcal{U} \cup \mathcal{I}$. As shown in Fig. 2, DDGEM performs node-wise diffusion denoising on $\mathbf{c}_n$ in the latent collaborative space. A discrete Gaussian diffusion process with T steps is adopted, where the variance schedule is linearly increased from $\beta_{\min}$ to $\beta_{\max}$:

$$\beta_\tau = \beta_{\min} + \frac{\tau - 1}{T - 1}(\beta_{\max} - \beta_{\min}), \quad \alpha_\tau = 1 - \beta_\tau, \quad \bar{\alpha}_\tau = \prod_{s=1}^{\tau} \alpha_s. \tag{1}$$

For diffusion step τ, the forward process generates the noisy embedding $\mathbf{h}_{n,\tau}$ as

$$q(\mathbf{h}_{n,\tau}|\mathbf{c}_n) = \mathcal{N}(\sqrt{\bar{\alpha}_\tau}\mathbf{c}_n, (1 - \bar{\alpha}_\tau)\mathbf{I}), \quad \mathbf{h}_{n,\tau} = \sqrt{\bar{\alpha}_\tau}\mathbf{c}_n + \sqrt{1 - \bar{\alpha}_\tau}\boldsymbol{\varepsilon}, \tag{2}$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The denoiser $\varepsilon_\phi(\cdot)$ follows the structure in Fig. 2 and is implemented as a three-layer multilayer perceptron (MLP) with rectified linear unit (ReLU) activation:

$$\varepsilon_\phi(\mathbf{h}_{n,\tau}, \tau) = \mathbf{W}_3\sigma(\mathbf{W}_2\sigma(\mathbf{W}_1[\mathbf{h}_{n,\tau} \| \mathbf{t}_\tau] + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3, \tag{3}$$

where $\mathbf{t}_\tau = \tau/T$ is the normalized time-step embedding and $[\cdot\|\cdot]$ denotes concatenation. In our implementation, $T = 50$, $\beta_{\min} = 10^{-4}$, and $\beta_{\max} = 0.02$. The node diffusion loss trains the denoiser to predict the injected noise:

$$\mathcal{L}_{node} = \mathbb{E}_{n,\tau,\boldsymbol{\varepsilon}}\left[\left\|\boldsymbol{\varepsilon} - \varepsilon_\phi(\mathbf{h}_{n,\tau}, \tau)\right\|_2^2\right]. \tag{4}$$

After reverse denoising, the reconstructed embedding is denoted as $\tilde{\mathbf{c}}_n$, and the denoised user and item representations are denoted as $\mathbf{z}_u$ and $\mathbf{z}_i$. In implementation, the reconstructed item embedding is injected by

$$\mathbf{c}_i \leftarrow 0.9\mathbf{c}_i + 0.1\tilde{\mathbf{c}}_i, \tag{5}$$

which keeps the original collaborative signal while introducing a denoising prior for the subsequent relation reconstruction and generative item enhancement.

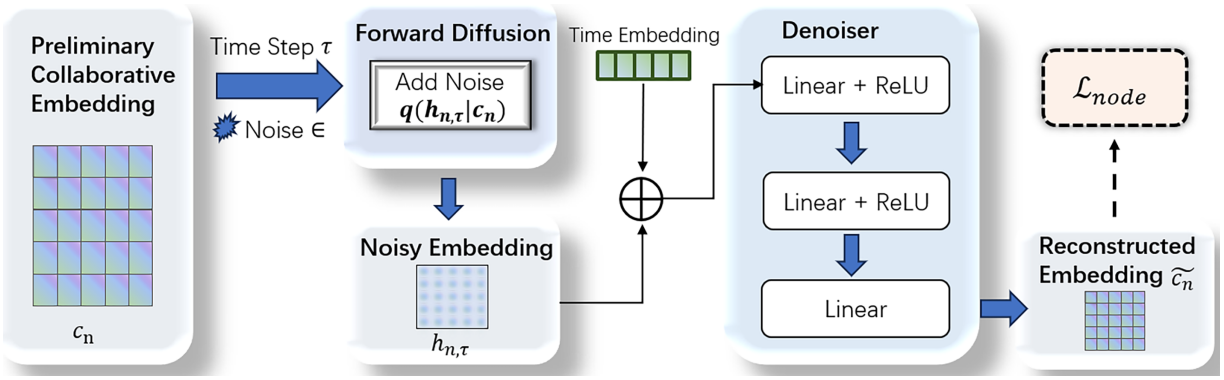


Figure 2: Schematic diagram of the node-wise diffusion denoiser. ReLU, rectified linear unit.

3.3 Relational Diffusion Graph Reconstructor Module

The denoised item representation $\mathbf{z}_i$ captures collaborative signals, but item–item semantic relations are still needed to provide complementary structural evidence. As shown in Fig. 3, DDGEM reconstructs an adaptive item–item graph from multimodal semantic representations. Here, $\mathbf{m}_i$ denotes the multimodal semantic representation obtained from the projected textual and visual features introduced in Section 3.1. For each item i, candidate neighbors are selected by

$$\mathcal{N}_i^{rel} = \mathrm{TopK}_{j\in\mathcal{I}} \cos(\mathbf{m}_i, \mathbf{m}_j). \tag{6}$$

For each candidate edge (i, j), the pair encoder $g_\varphi(\cdot)$ generates the initial relation code

$$p_{ij} = g_\varphi([\mathbf{m}_i\|\mathbf{m}_j]), \tag{7}$$

where $[\cdot\|\cdot]$ denotes concatenation.

Relational diffusion is performed on p_{ij}. At relation step ρ, the noisy relation $r_{ij,\rho}$ is sampled as

$$q(r_{ij,\rho}|p_{ij}) = \mathcal{N}(\sqrt{\bar{\alpha}_\rho}p_{ij}, (1-\bar{\alpha}_\rho)\mathbf{I}), \quad r_{ij,\rho} = \sqrt{\bar{\alpha}_\rho}p_{ij} + \sqrt{1-\bar{\alpha}_\rho}\varepsilon_{ij}, \tag{8}$$

where $\varepsilon_{ij} \sim \mathcal{N}(0,1)$. Following Fig. 3, RelDenoise takes $r_{ij,\rho}$, $[\mathbf{m}_i \| \mathbf{m}_j]$, and ρ as inputs, and is implemented as a three-layer MLP with ReLU activation. The relation diffusion loss is

$$\mathcal{L}_{rel} = \mathbb{E}_{(i,j),\rho,\varepsilon_{ij}} \left[\left\| \varepsilon_{ij} - \varepsilon_\psi(r_{ij,\rho}, [\mathbf{m}_i \| \mathbf{m}_j], \rho) \right\|_2^2 \right]. \tag{9}$$

After reverse sampling, the denoised relation strength $\hat{r}_{ij}$ is used to update the candidate edge weight. The reconstructed item–item matrix is row-normalized as

$$\mathbf{A}_{ij}^{(rel)} = \frac{[\hat{r}_{ij}]_+}{\sum_{k \in \mathcal{N}_i^{rel}} [\hat{r}_{ik}]_+}, \quad j \in \mathcal{N}_i^{rel}, \tag{10}$$

where $[\cdot]_+$ denotes non-negative clipping. The relation-enhanced item representation is then computed by

$$\tilde{\mathbf{z}}_i = \sum_{j \in \mathcal{N}_i^{rel}} \mathbf{A}_{ij}^{(rel)} \mathbf{z}_j. \tag{11}$$

In implementation, $K = 20$ candidate neighbors are used, and the relation diffusion adopts $T = 50$, $\beta_{\min} = 10^{-4}$, and $\beta_{\max} = 0.02$.

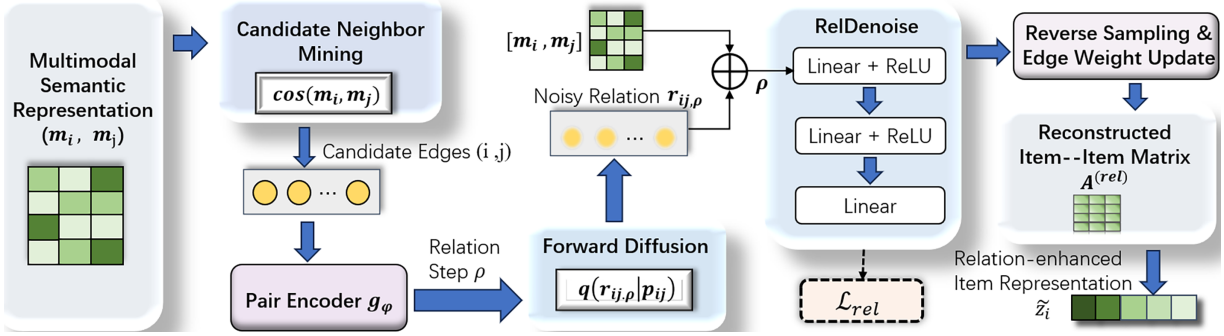


Figure 3: Structure of the relational diffusion graph reconstruction module. ReLU, rectified linear unit.

3.4 Conditional Generative Item Enhancement Module

Sparse and cold-start items may have weak collaborative anchors even after relation-aware propagation. As shown in Fig. 4, DDGEM uses a conditional generative module to generate item representations from multimodal semantics. Given the multimodal semantic representation $\mathbf{m}_i$, the content projection and masking operation produces the masked semantic representation

$$\bar{\mathbf{m}}_i = \mathbf{d}_i \odot f_m(\mathbf{m}_i), \quad \mathbf{d}_i \sim \text{Bernoulli}(1 - r_m), \tag{12}$$

where $f_m(\cdot)$ is the content projection network and r_m is the mask ratio.

For warm items, the posterior encoder and prior encoder are defined as

$$q_\phi(\mathbf{l}_i | \tilde{\mathbf{z}}_i, \bar{\mathbf{m}}_i) = \mathcal{N}(\boldsymbol{\mu}_{\phi,i}, \text{diag}(\boldsymbol{\sigma}_{\phi,i}^2)), \quad p_\theta(\mathbf{l}_i | \bar{\mathbf{m}}_i) = \mathcal{N}(\boldsymbol{\mu}_{\theta,i}, \text{diag}(\boldsymbol{\sigma}_{\theta,i}^2)). \tag{13}$$

The latent representation is sampled by reparameterization and decoded as

$$\mathbf{l}_i = \boldsymbol{\mu}_{\phi,i} + \boldsymbol{\sigma}_{\phi,i} \odot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \hat{\mathbf{z}}_i = G_\chi(\bar{\mathbf{m}}_i, \mathbf{l}_i). \tag{14}$$

In implementation, the posterior encoder, prior encoder, and decoder are implemented as MLPs.

The training objective of this module is

$$\mathcal{L}_{cgie} = \mathcal{L}_{rec} + \lambda_{KL}\mathcal{L}_{KL} + \lambda_{uni}\mathcal{L}_{uni}, \tag{15}$$

where

$$\mathcal{L}_{rec} = \|\hat{\mathbf{z}}_i - \tilde{\mathbf{z}}_i\|_2^2, \quad \mathcal{L}_{KL} = D_{KL}\left(q_\phi(\mathbf{l}_i|\tilde{\mathbf{z}}_i, \bar{\mathbf{m}}_i)\|p_\theta(\mathbf{l}_i|\bar{\mathbf{m}}_i)\right), \tag{16}$$

and the prior uniformity loss over a mini-batch $\mathcal{B}$ is

$$\mathcal{L}_{uni} = \log \frac{1}{|\mathcal{B}|(|\mathcal{B}|-1)} \sum_{i \neq j} \exp\left(-\|\boldsymbol{\mu}_{\theta,i} - \boldsymbol{\mu}_{\theta,j}\|_2^2\right). \tag{17}$$

In implementation, the semantic-cluster prior is obtained by applying K-means to the prior means $\{\boldsymbol{\mu}_{\theta,i}\}$ of warm items. In the cold-start phase, $\tilde{\mathbf{z}}_i$ is unavailable or unreliable. DDGEM samples $\mathbf{l}_i$ from the semantic-conditioned prior with the nearest semantic-cluster prior as an anchor, and generates the cold-start item representation by

$$\mathbf{z}_i^{(g)} = G_\chi(\bar{\mathbf{m}}_i, \mathbf{l}_i), \quad \mathbf{l}_i \sim p_\theta(\mathbf{l}_i|\bar{\mathbf{m}}_i). \tag{18}$$

The generated representation $\mathbf{z}_i^{(g)}$ is then fed into the behavior-aware multimodal fusion module.

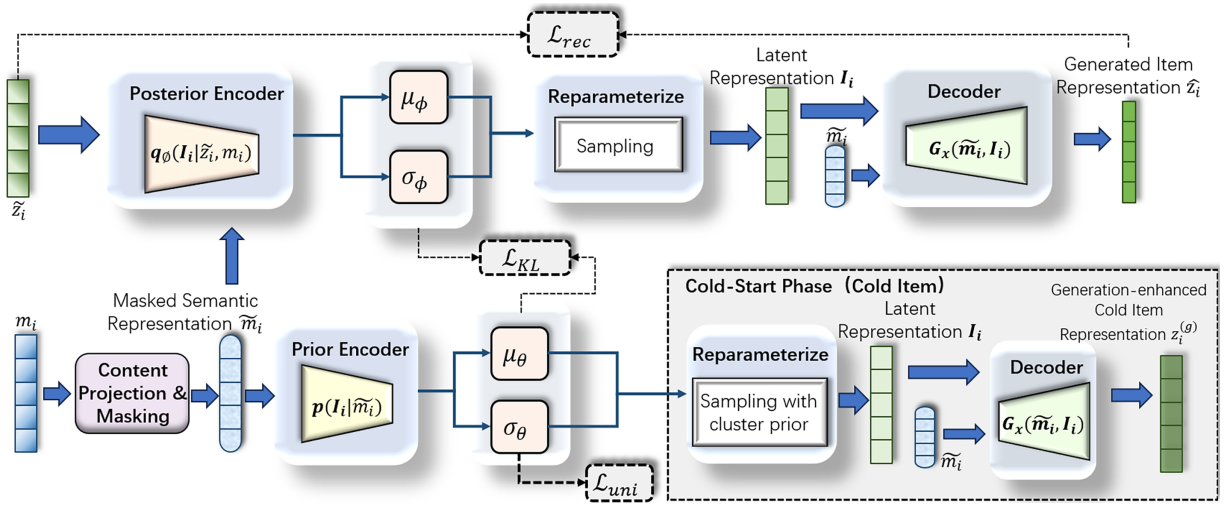


Figure 4: Conditional generative item enhancement and cold-start generation.

3.5 Behavior-Aware Multimodal Fusion Module

Collaborative, textual, and visual signals may contribute differently to different user–item pairs. As shown in Fig. 5, DDGEM uses a behavior-aware gating mechanism to assign pair-specific modality weights before final ranking.

For item i, the generation-enhanced representation is projected as the collaborative modality embedding, while textual and visual representations are mapped into the same latent space:

$$\mathbf{e}_i^{(c)} = \mathbf{W}_c\mathbf{z}_i^{(g)}, \quad \hat{\mathbf{e}}_i^{(t)} = \mathbf{W}_t\mathbf{e}_i^{(t)}, \quad \hat{\mathbf{e}}_i^{(v)} = \mathbf{W}_v\mathbf{e}_i^{(v)}. \tag{19}$$

For each user–item pair (u, i), three gating MLPs take the denoised user representation $\mathbf{z}_u$ and the corresponding modality embedding as input:

$$g_{u,i}^{(c)} = \mathrm{MLP}_c([\mathbf{z}_u \| \mathbf{e}_i^{(c)}]), \quad g_{u,i}^{(t)} = \mathrm{MLP}_t([\mathbf{z}_u \| \hat{\mathbf{e}}_i^{(t)}]), \quad g_{u,i}^{(v)} = \mathrm{MLP}_v([\mathbf{z}_u \| \hat{\mathbf{e}}_i^{(v)}]). \tag{20}$$

The modality weights are obtained by softmax normalization:

$$[\omega_{u,i}^{(c)}, \omega_{u,i}^{(t)}, \omega_{u,i}^{(v)}] = \mathrm{softmax}([g_{u,i}^{(c)}, g_{u,i}^{(t)}, g_{u,i}^{(v)}]). \tag{21}$$

The fused representation is computed by weighted summation:

$$\mathbf{z}_{u,i}^{mm} = \omega_{u,i}^{(c)} \mathbf{e}_i^{(c)} + \omega_{u,i}^{(t)} \hat{\mathbf{e}}_i^{(t)} + \omega_{u,i}^{(v)} \hat{\mathbf{e}}_i^{(v)}. \tag{22}$$

The ranking score is $s(u, i) = \mathbf{z}_u^\top \mathbf{z}_{u,i}^{mm}$.

The ranking objective is optimized by BPR:

$$\mathcal{L}_{BPR} = -\sum_{(u,i,j)} \log \sigma \left(s(u, i) - s(u, j) \right), \tag{23}$$

where i and j are positive and negative items, respectively. Following Fig. 5, DDGEM further uses contrastive and alignment regularization:

$$\mathcal{L}_{con} = -\frac{1}{|\mathcal{B}|} \sum_{(u,i)\in\mathcal{B}} \log \frac{\exp(\mathrm{sim}(\mathbf{z}_u, \mathbf{z}_{u,i}^{mm})/\tau_c)}{\sum_{i'\in\mathcal{B}} \exp(\mathrm{sim}(\mathbf{z}_u, \mathbf{z}_{u,i'}^{mm})/\tau_c)}, \tag{24}$$

$$\mathcal{L}_{align} = \sum_{i\in\mathcal{B}} \left(\|\mathbf{e}_i^{(c)} - \hat{\mathbf{e}}_i^{(t)}\|_2^2 + \|\mathbf{e}_i^{(c)} - \hat{\mathbf{e}}_i^{(v)}\|_2^2 \right). \tag{25}$$

Here, $\mathrm{sim}(\cdot, \cdot)$ denotes cosine similarity, τ_c is the contrastive temperature, and $\mathcal{B}$ is a mini-batch. In implementation, the projection layers are linear transformations, and each gating branch is implemented as an MLP.

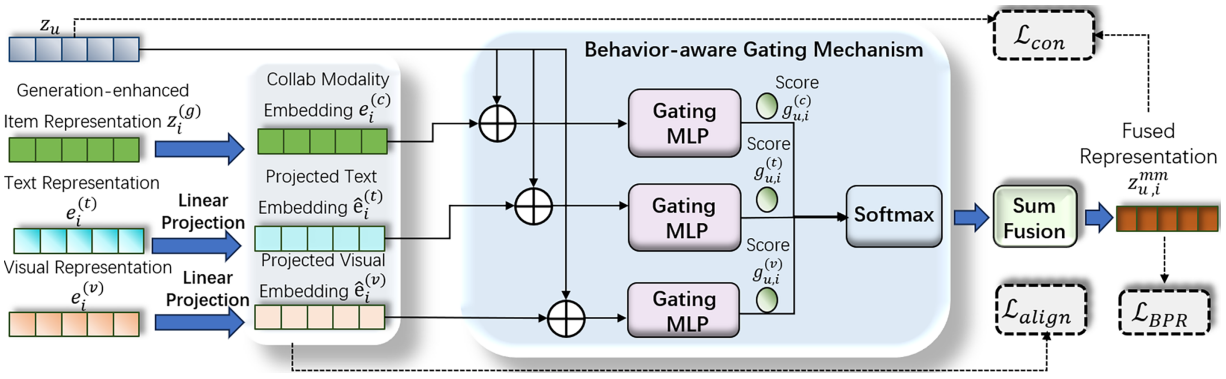


Figure 5: Structure of the behavior-aware multimodal fusion module. MLP, multilayer perceptron.

3.6 Joint Optimization and Complexity Analysis

Let Θ denote all trainable parameters in DDGEM. The final training objective is

$$\mathcal{L}_{DDGEM} = \mathcal{L}_{BPR} + \lambda_{node}\mathcal{L}_{node} + \lambda_{rel}\mathcal{L}_{rel} + \lambda_{cgie}\mathcal{L}_{cgie} + \lambda_{con}\mathcal{L}_{con} + \lambda_{align}\mathcal{L}_{align} + \lambda_2\|\Theta\|_2^2. \tag{26}$$

Here, $\mathcal{L}_{BPR}$ optimizes ranking, $\mathcal{L}_{node}$ and $\mathcal{L}_{rel}$ train the two diffusion denoisers, $\mathcal{L}_{cgie}$ optimizes conditional item generation, and $\mathcal{L}_{con}$ and $\mathcal{L}_{align}$ regularize the fused multimodal representation. These terms are jointly optimized by back-propagation.

DDGEM applies diffusion in the latent recommendation space rather than in the raw image or text space. The forward processes perturb collaborative embeddings and relation codes with Gaussian noise, and the denoisers learn to recover stable latent signals. This gives a denoising interpretation to the node-wise and relational diffusion modules while keeping the generation process inside the recommendation architecture.

Let d be the embedding dimension, B the mini-batch size, K_g the number of graph propagation layers, K_r the number of relation neighbors, and T the number of diffusion steps. The dominant training complexity is

$$\mathcal{O}_{train} = \mathcal{O}\left(K_g|\mathcal{E}|d + TBd^2 + |\mathcal{I}|K_r d + Bd^2\right), \tag{27}$$

where the terms correspond to graph propagation, diffusion denoising, relation-enhanced aggregation, and MLP-based generation/fusion. In implementation, $K_r = 20$ and $T = 50$. The semantic TopK graph can be precomputed before training.

During inference, auxiliary training losses are not optimized, and item-side representations can be precomputed. For a candidate set $\mathcal{C}_u$, the online cost is mainly from the gating MLPs and dot-product scoring:

$$\mathcal{O}_{infer}(u) = \mathcal{O}\left(|\mathcal{C}_u|(3d^2 + d)\right). \tag{28}$$

4 Experiment

We evaluate DDGEM on three public Amazon datasets. The experiments address seven research questions (RQs): **RQ1:** Does DDGEM outperform representative baselines, and are the gains over MIG-GT statistically reliable across random seeds? **RQ2:** How much does each module and sub-design contribute? **RQ3:** What are the training cost, inference cost, memory usage, and parameter size? **RQ4:** How robust is DDGEM to noisy collaborative graphs? **RQ5:** How does DDGEM perform under strict item cold-start settings? **RQ6:** How does DDGEM behave when textual and visual modalities conflict? **RQ7:** How sensitive is DDGEM to key hyperparameters?

4.1 Experimental Settings

4.1.1 Datasets

We use three public Amazon Review datasets [43]: Baby, Sports, and Clothing. Each dataset contains implicit user–item interactions and item-side textual and visual features. Following prior work [44], we use the released 384-dimensional text features and 4096-dimensional visual features, which are fixed during training. The dataset statistics are reported in Table 1.

Table 1: Statistics of the Baby, Sports, and Clothing datasets.

Dataset	Users	Items	Interactions	Sparsity
Baby	19,445	7050	160,792	99.88%
Sports	35,598	18,357	296,337	99.95%
Clothing	39,387	23,033	278,677	99.97%

4.1.2 Evaluation Metrics

We evaluate top-K recommendation using Recall@K and normalized discounted cumulative gain (NDCG)@K, where $K \in \{10, 20\}$. Recall@K measures the proportion of relevant items retrieved in the top-K list, and NDCG@K further considers their ranking positions. During evaluation, items observed in the training set are masked from the candidate list. To examine statistical reliability, DDGEM and the strongest baseline are run with five random seeds, and significant improvements are marked in Table 2 using paired t-tests.

4.1.3 Baseline Models for Comparison

We compare DDGEM with representative methods from three groups: collaborative filtering methods, including MF [2], LightGCN [18], and DirectAU [45]; generative recommendation methods, including GAR [46], GoRec [32], DGVAE [47], and DiffMM [39]; and multimodal recommendation methods, including VBPR [9], MMGCN [10], LATTICE [11], BM3 [14], DGHNet [22], FREEDOM [23], DA-MRS [24], AM2HRec [25], and MIG-GT [44].

4.1.4 Implementation Details

DDGEM is implemented in PyTorch and trained on a single NVIDIA GeForce RTX 3090 graphics processing unit (GPU). The main recommendation parameters are optimized by AdamW, while the node-wise and relation-level denoising networks are optimized by Adam. The batch size is set to 8000, and one negative item is sampled for each positive interaction. The maximum number of training epochs is 600, with early stopping based on validation Recall@20. The embedding size is 64. The learning rate is 10^{-2} for Baby and Sports and 10^{-3} for Clothing. For both node-wise and relation-level diffusion, we use a linear noise schedule with $T = 50$, $\beta_{\min} = 10^{-4}$, and $\beta_{\max} = 0.02$. The number of candidate neighbors in relational diffusion is set to 20. The hidden dimension and latent dimension of the CVAE are 256 and 64, respectively. For efficiency comparison, all reproduced models are evaluated under the same hardware setting.

4.2 Overall Performance Comparison (RQ1)

Table 2 reports the overall performance on the three datasets. The best results are shown in bold, and the second-best results are underlined. The rows "p-value" and "Improv." are computed between DDGEM and MIG-GT over five random seeds.

DDGEM achieves the best results on all datasets and metrics. Compared with MIG-GT, it improves Recall@20 by 1.95%, 0.53%, and 3.96% on Baby, Sports, and Clothing, respectively. The corresponding NDCG@20 improvements are 2.43%, 0.98%, and 5.69%. The paired t-test results are below 0.05 (5.00×10^{-2}), indicating that the gains are statistically reliable across five random seeds.

The improvements on Sports are smaller than those on Baby and Clothing. One possible reason is that MIG-GT already captures strong multimodal signals on this dataset, leaving limited room for further improvement. Nevertheless, DDGEM still gives consistent gains, which supports the usefulness of latent collaborative denoising, relation reconstruction, generative item enhancement, and behavior-aware fusion performance.

Table 2: Performance comparison on the Baby, Sports, and Clothing datasets (**best** in bold, second best underlined).

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**
MF	0.0357	0.0575	0.0192	0.0249	0.0432	0.0653	0.0241	0.0298	0.0206	0.0303	0.0114	0.0138
LightGCN	0.0479	0.0754	0.0257	0.0328	0.0569	0.0864	0.0311	0.0387	0.0361	0.0544	0.0197	0.0243
DirectAU	0.0543	0.0804	0.0300	0.0367	0.0682	0.1017	0.0379	0.0464	0.0443	0.0669	0.0240	0.0298
GAR	0.0299	0.0502	0.0160	0.0216	0.0385	0.0651	0.0224	0.0288	0.0393	0.0634	0.0209	0.0275
GoRec	0.0456	0.0703	0.0272	0.0341	0.0515	0.0828	0.0290	0.0376	0.0575	0.0929	0.0308	0.0395
DGVAE	0.0636	0.1009	0.0340	0.0436	0.0753	0.1127	0.0410	0.0506	0.0619	0.0917	0.0336	0.0412
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
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	0.0421	0.0660	0.0220	0.0282	0.0401	0.0636	0.0209	0.0270	0.0227	0.0361	0.0154	0.0154
LATTICE	0.0547	0.0844	0.0289	0.0366	0.0626	0.0959	0.0337	0.0423	0.0468	0.0688	0.0256	0.0312
BM3	0.0542	0.0862	0.0285	0.0367	0.0619	0.0971	0.0338	0.0429	0.0425	0.0637	0.0232	0.0286
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
FREEDOM	0.0627	0.0992	0.0330	0.0424	0.0717	0.1089	0.0385	0.0481	0.0629	0.0941	0.0341	0.0420
DA-MRS	0.0626	0.0966	0.0339	0.0426	0.0708	0.1078	0.0379	0.0475	0.0633	0.0924	0.0342	0.0415
AM2HRec	0.0628	0.0980	0.0339	0.0431	0.0754	0.1113	0.0410	0.0508	0.0645	0.0958	0.0354	0.0434
MIG-GT	0.0665	0.1021	0.0361	0.0452	0.0753	0.1130	0.0414	0.0511	0.0636	0.0934	0.0347	0.0422
DDGEM	**0.0675**	**0.1041**	**0.0370**	**0.0463**	**0.0762**	**0.1136**	**0.0421**	**0.0516**	**0.0662**	**0.0971**	**0.0366**	**0.0446**
p-value	1.51×10^{-2}	1.05×10^{-2}	1.11×10^{-2}	4.49×10^{-2}	2.70×10^{-2}	3.32×10^{-2}	2.17×10^{-2}	3.13×10^{-2}	3.00×10^{-3}	2.20×10^{-3}	1.10×10^{-3}	1.90×10^{-3}
Improv.	1.50%	1.95%	2.49%	2.43%	1.20%	0.53%	1.69%	0.98%	4.09%	3.96%	5.48%	5.69%

Note: R@10, Recall@10; R@20, Recall@20; N@10, NDCG@10; N@20, NDCG@20; Improv., improvements.

4.3 Ablation Study (RQ2)

To examine the contribution of DDGEM's main modules and internal sub-designs, we conduct module-level and sub-design-level ablation studies. The module-level variants remove NDD, RDGR, CGIE, BMMF, or both diffusion modules. The sub-design variants further remove the node diffusion loss, disable relation diffusion refinement, remove the CVAE KL regularization, remove the popularity-aware gate, or replace behavior-aware fusion with average fusion.

As shown in Table 3, all variants underperform DDGEM. Removing NDD and RDGR together causes the largest decline, indicating that collaborative denoising and relation reconstruction are complementary. The drops caused by w/o CGIE and w/o CVAE KL regularization show that content-conditioned generation benefits from latent regularization. In addition, w/o BMMF, w/o popularity-aware gate, and Average fusion reduce performance, confirming the need for adaptive generation balance and user–item-specific multimodal fusion.

Table 3: Ablation results of DDGEM on the Baby, Sports, and Clothing datasets (**best** in bold).

Variants	Baby		Sports		Clothing	
	R@20	N@20	R@20	N@20	R@20	N@20
w/o NDD	0.1019	0.0450	0.1123	0.0512	0.0950	0.0430
w/o RDGR	0.1030	0.0454	0.1126	0.0508	0.0956	0.0438
w/o CGIE	0.1023	0.0455	0.1124	0.0509	0.0951	0.0436
w/o BMMF	0.1035	0.0458	0.1130	0.0513	0.0959	0.0441
w/o NDD+RDGR	0.1018	0.0446	0.1120	0.0504	0.0941	0.0428
w/o node diffusion loss	0.1016	0.0451	0.1122	0.0499	0.0964	0.0440
w/o relation diffusion refinement	0.1022	0.0449	0.1120	0.0505	0.0949	0.0432
w/o CVAE KL regularization	0.1029	0.0454	0.1123	0.0503	0.0956	0.0439
w/o popularity-aware gate	0.1021	0.0452	0.1117	0.0500	0.0952	0.0436
Average fusion	0.1025	0.0456	0.1126	0.0510	0.0966	0.0442
DDGEM	**0.1041**	**0.0463**	**0.1136**	**0.0516**	**0.0971**	**0.0446**

Note: R@20, Recall@20; N@20, NDCG@20; w/o, without; NDD, node-wise diffusion denoiser; RDGR, relational diffusion graph reconstructor; CGIE, conditional generative item enhancement; BMMF, behavior-aware multimodal fusion; CVAE, conditional variational autoencoder; KL, Kullback–Leibler.

To further compare diffusion-based denoising with CVAE-based generation, we conduct an incremental analysis. Here, *base* removes both designs from DDGEM; *base-cvae* and *base-diff* add only CVAE-based generation and diffusion-based denoising, respectively. As shown in Fig. 6, the single-component variants do not consistently outperform base, suggesting that either generation or denoising alone is insufficient and may require coordination with the other component. In contrast, the full DDGEM achieves the best results on all datasets, indicating that diffusion-based denoising and CVAE-based generation are complementary rather than redundant.

4.4 Efficiency Analysis (RQ3)

Table 4 reports the training time, inference time, GPU memory usage, and parameter size under the same hardware setting. DDGEM introduces extra memory and inference cost because it combines diffusion-based denoising with CVAE-based generation. However, its parameter size remains close to GoRec and MIG-GT and is much smaller than DiffMM. Its training time is also lower than DiffMM on all datasets.

Overall, the main overhead of DDGEM comes from memory usage and inference, while its parameter scale and training cost remain within an acceptable range.

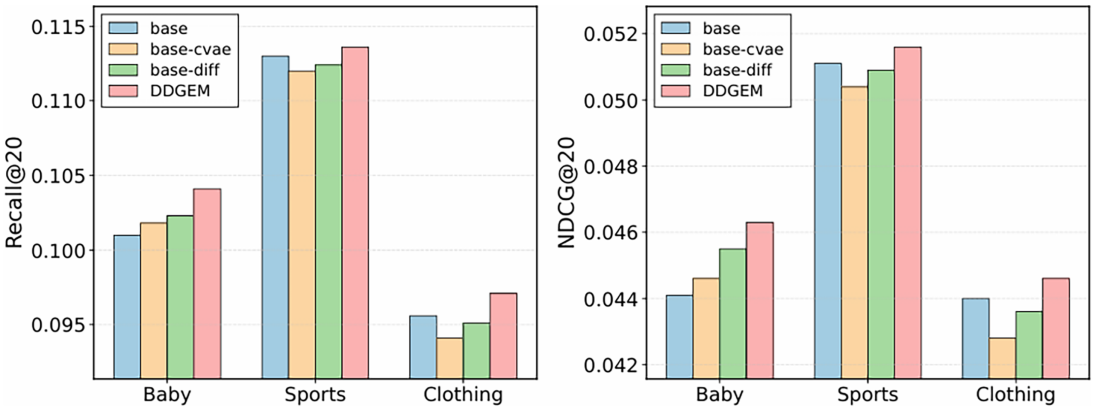


Figure 6: Incremental contribution analysis of diffusion-based denoising and CVAE-based generation. base-cvae, CVAE-based generation; base-diff, diffusion-based denoising.

Table 4: Efficiency comparison on the Baby, Sports, and Clothing datasets. Param., Train, Infer., and Mem. denote parameter size (M), training time (s/epoch), inference time (s), and GPU memory usage (GB), respectively.

Model	Baby				Sports				Clothing			
	Param. (M)	Train (s/epoch)	Infer. (s)	Mem. (GB)	Param. (M)	Train (s/epoch)	Infer. (s)	Mem. (GB)	Param. (M)	Train (s/epoch)	Infer. (s)	Mem. (GB)
DiffMM	30.26	7.09	3.20	1.33	77.23	18.97	9.64	2.91	96.48	21.99	13.98	3.46
GoRec	2.87	1.28	1.03	1.45	4.63	1.72	1.19	2.01	5.17	2.78	2.11	3.25
MIG-GT	2.63	1.52	6.32	1.84	4.39	2.42	14.91	2.70	7.70	4.31	16.89	5.11
DDGEM	2.99	2.67	6.78	3.40	4.75	6.70	16.99	6.53	8.06	7.16	17.39	10.24

4.5 Graph Noise Robustness Analysis (RQ4)

To evaluate robustness to noisy collaborative graphs, we randomly remove user–item interactions from the training graph with noise ratios of 5%, 10%, 15%, and 20%. As shown in Fig. 7, both models degrade as the noise ratio increases. DDGEM consistently obtains higher Recall@20 and lower drop rates than MIG-GT on all datasets. This indicates that node-wise denoising and relation-level reconstruction help DDGEM reduce the effect of unreliable collaborative signals.

4.6 Strict Cold-Start Analysis (RQ5)

We further evaluate DDGEM under a strict item cold-start setting. In this setting, cold-start items are defined as test items with no user–item interactions in the training graph; only textual and visual features are available for these items. Fig. 8 reports Recall@20 and NDCG@20 on the three datasets. DDGEM consistently outperforms MIG-GT, with the largest margin on Sports, suggesting that it can provide more useful item representations when collaborative signals are unavailable. To further illustrate this effect, Fig. 9 visualizes warm-item and cold-item embeddings on Baby using t-distributed stochastic neighbor embedding (t-SNE). Compared with MIG-GT, DDGEM shows closer alignment between cold and warm items, indicating that the generated cold-start representations are better integrated into the latent space.

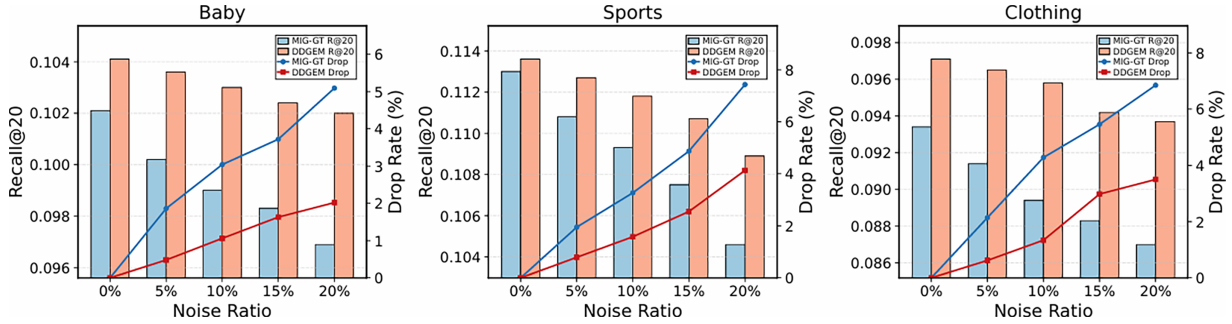


Figure 7: Model performance with respect to graph noise ratio. Bars denote Recall@20, and lines denote the percentage of performance degradation. R@20, Recall@20.

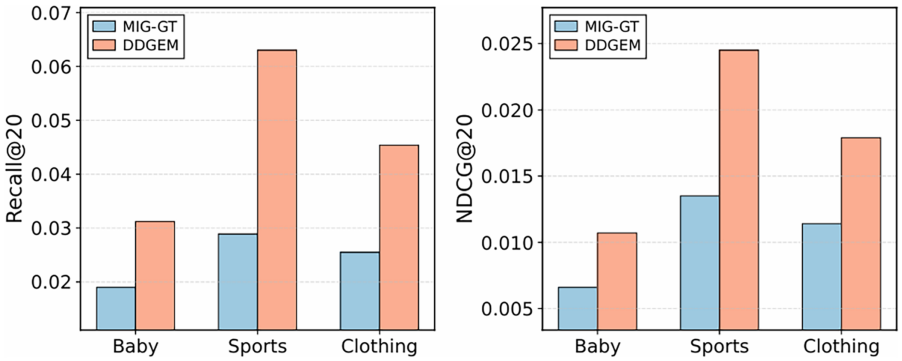


Figure 8: Performance under the strict item cold-start setting.

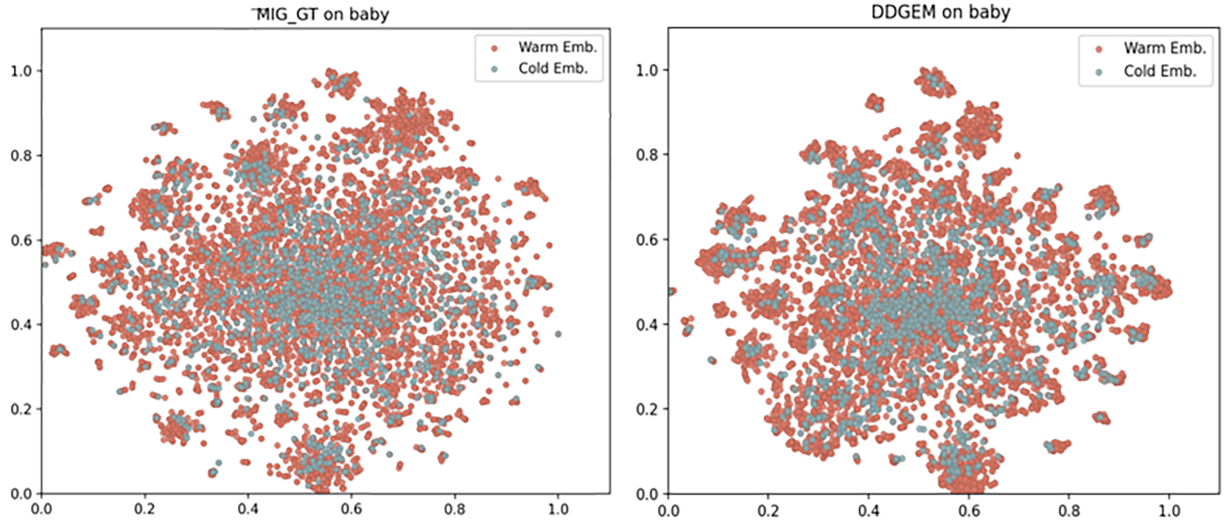


Figure 9: t-SNE visualization of warm-item and cold-item embeddings on Baby. Emb., embedding.

4.7 Modality Conflict Analysis (RQ6)

We further evaluate DDGEM under modality conflict settings. Here, *Full* denotes the original setting with matched textual and visual features; *Text Conflict* mismatches textual features across items while keeping visual features unchanged; and *Image Conflict* mismatches visual features across items while keeping textual features unchanged. As shown in Fig. 10, both models degrade under modality conflict, but DDGEM keeps

higher Recall@20 and NDCG@20 than MIG-GT on all datasets. The drop is larger under text conflict, suggesting that textual inconsistency has a stronger effect on recommendation performance.

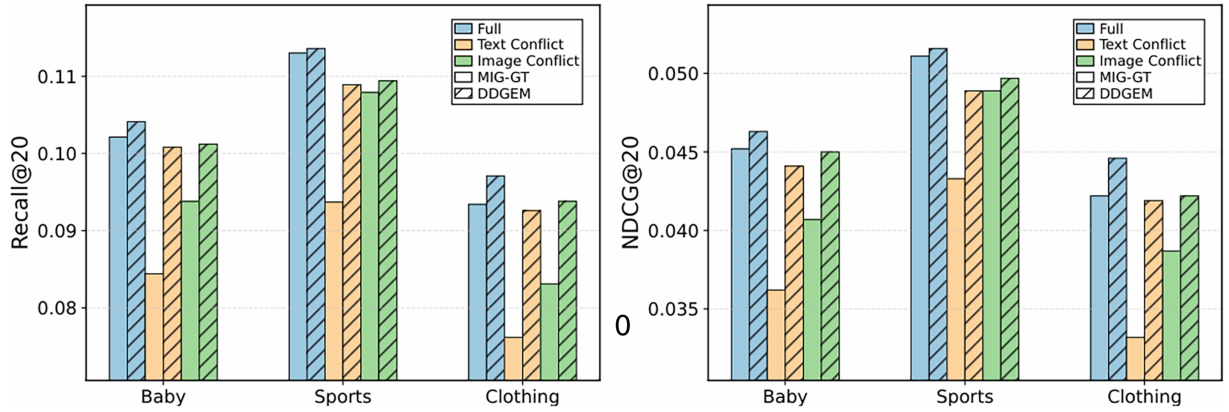
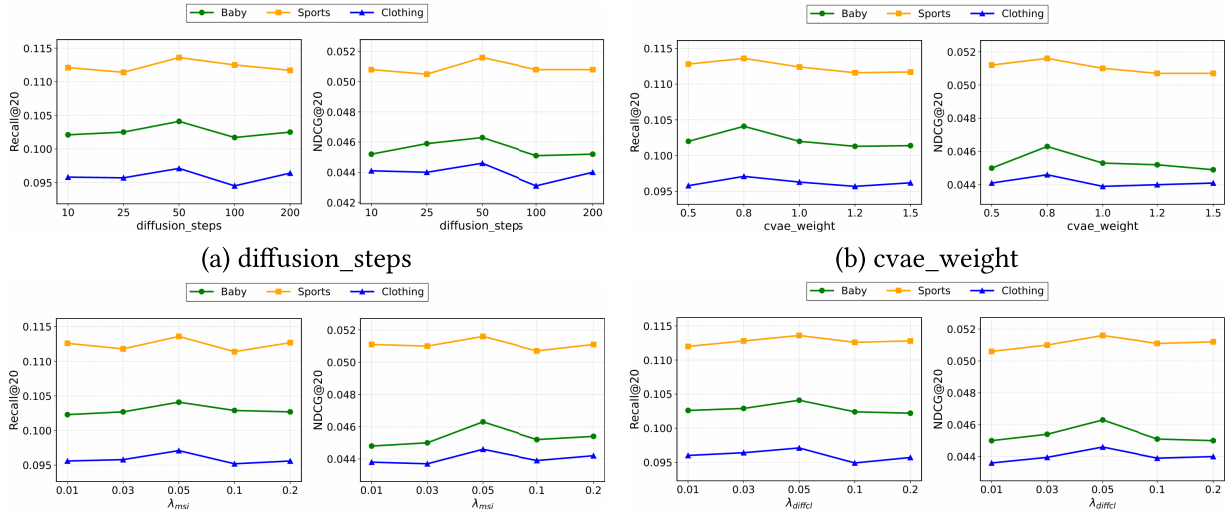


Figure 10: Performance under modality conflict settings.

4.8 Hyperparameter Sensitivity (RQ7)

We analyze the sensitivity of DDGEM to four key hyperparameters: diffusion_steps, cvae_weight, λ_{msi}, and λ_{diffcl}. As shown in Fig. 11, DDGEM remains relatively stable within moderate ranges. The best results are generally obtained when diffusion_steps is around 50, cvae_weight is around 0.8, and both λ_{msi} and λ_{diffcl} are around 0.05. Very small values may provide insufficient denoising or regularization, whereas overly large values may weaken useful collaborative signals.



(a) diffusion_steps (b) cvae_weight

(c) λ_{msi} (d) λ_{diffcl}

Figure 11: Hyperparameter sensitivity of DDGEM on the Baby, Sports, and Clothing datasets. (**a**) Performance with different numbers of diffusion steps; (**b**) performance with different CVAE weights; (**c**) performance with different values of λ_{msi}, the weight of the cosine-based representation consistency loss; (**d**) performance with different values of λ_{diffcl}, the weight of the diffusion-related contrastive loss. CVAE, conditional variational autoencoder.

5 Conclusion

We proposed DDGEM, a diffusion denoising and generative enhancement framework for multimodal recommendation. DDGEM refines collaborative representations through node-wise diffusion, reconstructs adaptive item–item relations with relational diffusion, enhances sparse item representations through CVAE-based generation, and integrates multimodal signals with behavior-aware fusion. Experiments on three Amazon datasets show that DDGEM improves recommendation accuracy over representative baselines, with statistically reliable gains over MIG-GT. Further ablation, efficiency, graph-noise robustness, strict cold-start, modality-conflict, and hyperparameter analyses verify the contribution and practical behavior of the main designs.

Although DDGEM improves robustness and cold-start recommendation, it introduces additional memory and inference cost due to the diffusion and generative modules. Future work will focus on lighter denoising architectures, more efficient inference, evaluation on industrial-scale recommendation scenarios, and temporal preference modeling for dynamic recommendation.

Acknowledgement: The authors appreciate the support of the Youth Project of Humanities and Social Sciences of the Ministry of Education of China.

Funding Statement: This research was supported by the Youth Project of Humanities and Social Sciences of the Ministry of Education of China under Grant No. 25YJC860021.

Author Contributions: Conceptualization and methodology, Li Zhao and Weiwei Li; software, validation, data curation and visualization, Li Zhao, Chengshan Li and Wenjie Geng; writing—original draft preparation, Li Zhao; writing—review and editing, Li Zhao and Weiwei Li; supervision and project administration, Weiwei Li. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are publicly available from the Amazon Review datasets cited in [43]. The data that support the findings of this study are available from the Corresponding Author upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Goldberg D, Nichols D, Oki BM, Terry D. Using collaborative filtering to weave an information tapestry. Commun ACM. 1992;35(12):61–70. doi:10.1145/138859.138867.
2. Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer. 2009;42(8):30–7. doi:10.1109/MC.2009.263.
3. Liu Q, Hu J, Xiao Y, Zhao X, Gao J, Wang W, et al. Multimodal recommender systems: a survey. ACM Comput Surv. 2024;57(2):1–17. doi:10.1145/3695461.
4. Guo F, Wang Z, Wang X, Lu Q, Ji S. Dual-view multi-modal contrastive learning for graph-based recommender systems. Comput Electr Eng. 2024;116(1):109213. doi:10.1016/j.compeleceng.2024.109213.
5. Mnih A, Salakhutdinov R. Probabilistic matrix factorization. Adv Neural Inf Process Syst. 2007;20:1257–64.
6. Cheng HT, Koc L, Harmsen J, Shaked T, Chandra T, Aradhye H, et al. Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems; 2016 Sep 15; Boston, MA, USA. p. 7–10. doi:10.1145/2988450.2988454.
7. Wang X, He X, Wang M, Feng F, Chua TS. Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2019 Jul 21–25; Paris, France. p. 165–74. doi:10.1145/3331184.3331267.

8. Yu J, Xia X, Chen T, Cui L, Nguyen QVH, Yin H. XSimGCL: towards extremely simple graph contrastive learning for recommendation. IEEE Trans Knowl Data Eng. 2024;36(2):913–26. doi:10.1109/TKDE.2023.3288135.
9. He R, McAuley J. VBPR: visual Bayesian personalized ranking from implicit feedback. Proc AAAI Conf Artif Intell. 2016;30(1):144–50. doi:10.1609/aaai.v30i1.9973.
10. Wei Y, Wang X, Nie L, He X, Hong R, Chua TS. MMGCN: multi-modal graph convolution network for personalized recommendation of micro-video. In: Proceedings of the 27th ACM International Conference on Multimedia; 2019 Oct 21–25; Nice, France. p. 1437–45. doi:10.1145/3343031.3351034.
11. Zhang J, Zhu Y, Liu Q, Wu S, Wang S, Wang L. Mining latent structures for multimedia recommendation. In: Proceedings of the 29th ACM International Conference on Multimedia; 2021 Oct 20–24; Chengdu, China. p. 3872–80. doi:10.1145/3474085.3475259.
12. Tao Z, Wei Y, Wang X, He X, Huang X, Chua TS. MGAT: multimodal graph attention network for recommendation. Inf Process Manag. 2020;57(5):102277. doi:10.1016/j.ipm.2020.102277.
13. Wei Y, Wang X, Li Q, Nie L, Li Y, Li X, et al. Contrastive learning for cold-start recommendation. In: Proceedings of the 29th ACM International Conference on Multimedia; 2021 Oct 20–24; Chengdu, China. p. 5382–90. doi:10.1145/3474085.3475665.
14. Zhou X, Zhou H, Liu Y, Zeng Z, Miao C, Wang P, et al. Bootstrap latent representations for multi-modal recommendation. arXiv:2207.05969. 2023. doi:10.1145/3543507.3583251.
15. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L. BPR: Bayesian personalized ranking from implicit feedback. arXiv:1205.2618. 2012.
16. Aljunid MF, Manjaiah DH, Hooshmand MK, Ali WA, Shetty AM, Alzoubah SQ. A collaborative filtering recommender systems: survey. Neurocomputing. 2025;617(12):128718. doi:10.1016/j.neucom.2024.128718.
17. Wu G, Zha Z, Tu L, Tao H, Song F. Research advances in graph neural network recommendation. CAAI Trans Intell Syst. 2020;15(1):14–24. (In Chinese). doi:10.11992/tis.201908034.
18. He X, Deng K, Wang X, Li Y, Zhang Y, Wang M. 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; 2020 Jul 25–30; Xi'an, China. p. 639–48. doi:10.1145/3397271.3401063.
19. Chen J, Zhou J, Ma L. GNNCL: a graph neural network recommendation model based on contrastive learning. Neural Process Lett. 2024;56(2):45. doi:10.1007/s11063-024-11545-9.
20. Zhang Y, Yu J, Liu Z, Wang G, Nguyen M, Sheng QZ, et al. Improving graph collaborative filtering with network motifs. Neural Comput Appl. 2025;37(16):9413–32. doi:10.1007/s00521-025-11079-8.
21. Liu J, Wang P, Sun G. Multi-modal recommendation algorithm based on graph structure learning. IAENG Int J Comput Sci. 2025;52(10):3862–9.
22. Dang Y, Pan Z, Zhang X, Chen W, Cai F, Chen H. Discrepancy learning guided hierarchical fusion network for multi-modal recommendation. Knowl Based Syst. 2025;317(1):113496. doi:10.1016/j.knosys.2025.113496.
23. Zhou X, Shen Z. A tale of two graphs: freezing and denoising graph structures for multimodal recommendation. In: Proceedings of the 31st ACM International Conference on Multimedia; 2023 Jan 8–10; Nara, Japan. p. 935–43. doi:10.1145/3581783.3611943.
24. Xv G, Li X, Xie R, Lin C, Liu C, Xia F, et al. Improving multi-modal recommender systems by denoising and aligning multi-modal content and user feedback. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2024 Aug 25–29; Barcelona, Spain. p. 3645–56. doi:10.1145/3637528.3671703.
25. Yu Y, Zhang C, Cong S, Yue X, Shen Y, Zhao J. AM2HRec: dual-representation adaptive noise reduction for multi-modal recommendation. Eng Lett. 2025;33(2):382–93.
26. Volkovs M, Yu G, Poutanen T. DropoutNet: addressing cold start in recommender systems. Adv Neural Inf Process Syst. 2017;30:4957–66.
27. Liang S, Pan Z, Liu W, Yin J, de Rijke M. A survey on variational autoencoders in recommender systems. ACM Comput Surv. 2024;56(10):1–40. doi:10.1145/3663364.

28. Li X, She J. Collaborative variational autoencoder for recommender systems. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mini; 2017 Aug 13–17; Halifax, NS, Canada. p. 305–14. doi:10.1145/3097983.3098077.
29. Walker J, Zhang F, Zhong T, Zhou F, Baagyere EY. Variational cold-start resistant recommendation. Inf Sci. 2022;605:267–85. doi:10.1016/j.ins.2022.05.025.
30. Ren J, Zhang R. When alignment makes a difference: a content-based variational model for cold-start CTR prediction. Adv Data Min Appl Lect Notes Comput Sci. 2023;14176(4):724–39. doi:10.1007/978-3-031-46661-8_48.
31. Xu X, Yang C, Yu Q, Fang Z, Wang J, Fan C, et al. Alleviating cold-start problem in CTR prediction with a variational embedding learning framework. In: Proceedings of the ACM Web Conference 2022; 2022 Apr 25–29; Virtual. p. 27–35. doi:10.1145/3485447.3512048.
32. Bai H, Hou M, Wu L, Yang Y, Zhang K, Hong R, et al. GoRec: a generative cold-start recommendation framework. In: Proceedings of the 31st ACM International Conference on Multimedia; 2023 Oct 29–Nov 3; Ottawa, ON, Canada. p. 1004–12. doi:10.1145/3581783.3612238.
33. Jiang Y, Yang Y, Xia L, Huang C. DiffKG: knowledge graph diffusion model for recommendation. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining; 2024 Mar 4–8; Yucatán, Mexico. p. 313–21. doi:10.1145/3616855.3635850.
34. Wang W, Xu Y, Feng F, Lin X, He X, Chua TS. Diffusion recommender model. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2023 Jul 23–27; Taipei, Taiwan. p. 832–41. doi:10.1145/3539618.3591663.
35. Du H, Yuan H, Huang Z, Zhao P, Zhou X. Sequential recommendation with diffusion models. arXiv:2304.04541. 2023.
36. Ma H, Xie R, Meng L, Chen X, Zhang X, Lin L, et al. Plug-in diffusion model for sequential recommendation. Proc AAAI Conf Artif Intell. 2024;38(8):8886–94. doi:10.1609/aaai.v38i8.28736.
37. Ma H, Xie R, Meng L, Yang Y, Sun X, Kang Z. SeeDRec: sememe-based diffusion for sequential recommendation. In: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence; 2024 Aug 3–9; Jeju, Republic of Korea. p. 2270–8. doi:10.24963/ijcai.2024/251.
38. Lee G, Zhu Y, Yu H, Zhou Y, Li J. Collaborative diffusion model for recommender system. In: Companion Proceedings of the ACM on Web Conference 2025; 2025 Apr 28–May 2; Sydney, Australia. p. 1091–5. doi:10.1145/3701716.3715516.
39. Jiang Y, Xia L, Wei W, Luo D, Lin K, Huang C. DiffMM: multi-modal diffusion model for recommendation. In: Proceedings of the 32nd ACM International Conference on Multimedia; 2024 Oct 28–Nov 1; Melbourne, Australia. p. 7591–9. doi:10.1145/3664647.3681498.
40. Islam K, Zaheer MZ, Mahmood A, Nandakumar K, Akhtar N. GenMix: effective data augmentation with generative diffusion model image editing. Expert Syst Appl. 2026;322(2):132273. doi:10.1016/j.eswa.2026.132273.
41. Islam K, Zaheer MZ, Mahmood A, Nandakumar K. DiffuseMix: label-preserving data augmentation with diffusion models. In: Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024 Jun 16–22; Seattle, WA, USA. p. 27611–20. doi:10.1109/CVPR52733.2024.02608.
42. Kong Y, Chen X, Jiang Y, Sun D, Zhang J. Novel discretized zeroing neural network models for time-varying optimization aided with predictor-corrector methods. IEEE Trans Neural Netw Learn Syst. 2025;36(8):14037–48. doi:10.1109/TNNLS.2024.3512505.
43. He R, McAuley J. Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In: Proceedings of the 25th International Conference on World Wide Web; 2016 Apr 11–15; Montreal, QC, Canada. p. 507–17. doi:10.1145/2872427.2883037.
44. Hu J, Hooi B, He B, Wei Y. Modality-independent graph neural networks with global transformers for multimodal recommendation. Proc AAAI Conf Artif Intell. 2025;39(11):11790–8. doi:10.1609/aaai.v39i11.33283.
45. Wang C, Yu Y, Ma W, Zhang M, Chen C, Liu Y, et al. Towards representation alignment and uniformity in collaborative filtering. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2022 Aug 14–18; Washington, DC, USA. p. 1816–25. doi:10.1145/3534678.3539253.

46. Chen H, Wang Z, Huang F, Huang X, Xu Y, Lin Y, et al. Generative adversarial framework for cold-start item recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2022 Jul 11–15; Madrid, Spain. p. 2565–71. doi:10.1145/3477495.3531897.
47. Zhou X, Miao C. Disentangled graph variational auto-encoder for multimodal recommendation with interpretability. IEEE Trans Multimed. 2024;26:7543–54. doi:10.1109/TMM.2024.3369875.