Proceedings of CONF-MLA 2025 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithm
DOI: 10.54254/2755-2721/165/2025.LD23499

GLEM-Rec—A Research on Cross-modal Recommendation Framework Based on Semantic-Graph Structure

Yunziyu Zhang

School of Information Management & Engineering, Shanghai University of Finance and Economics, Shanghai, China
sweato2022@163.com

Abstract: This paper proposes GLEM-Rec, a cross-modal recommendation framework integrating large language models with graph neural networks, effectively addressing three major challenges in traditional recommendation systems: semantic-graph structure feature alignment, long-tail item recommendation, and explainability. The framework consists of five core modules: semantic feature extractor, heterogeneous data processor, heterogeneous GNN integrator, adaptive trainer, and explainable recommendation generator, achieving complementary advantages between LLM's deep semantic understanding and GNN's high-order relationship modeling. Through multi-objective optimization strategies, GLEM-Rec achieves a balance between prediction accuracy, recommendation diversity, personalization, and long-tail coverage. Experiments based on the Movies Dataset demonstrate that this framework significantly outperforms existing methods, achieving an RMSE of 0.9122, coverage rate of 0.7723, and long-tail item recommendation performance of 0.9851, comprehensively surpassing traditional baseline models. System ablation experiments confirm the necessity and effectiveness of each functional module, validating the critical contribution of semantic and graph structure feature collaboration to recommendation system performance. This research not only provides new theoretical support for cross-modal recommendation systems but also offers effective technical solutions for key challenges in recommendation system practice.

Keywords: Recommendation Systems, GNN, LLM, Explainability, Contrastive Learning

1. Introduction

With the intelligent development of the internet, recommendation systems have become core technologies across various domains, with conversion rates of algorithms on leading platforms exceeding 20%, and deep learning recommendations increasing video platform user viewing time by 20%. Recommendation methods have continuously evolved from collaborative filtering to hybrid recommendations, yet still face significant challenges.

Cross-modal recommendation systems integrate diverse data to enhance personalization effects. The MM-Rec framework combines textual and visual information [1], while the DEAMER model alleviates data sparsity through multi-modal generation [2]. GNNs demonstrate outstanding performance in processing interaction graphs, with GraphRec modeling social relationships through heterogeneous message passing [3], and dual-graph attention networks implementing dynamic context awareness [4]. Large language models show potential in semantic understanding and

DOI: 10.54254/2755-2721/165/2025.LD23499

generative recommendations, with research optimizing the consistency between recommendation reasons and user preferences through prompt engineering [5]. Explainability research generates credible explanations through attention weights [6], while multi-objective optimization employs contrastive learning to balance various metrics [7].

Current systems still face key challenges: rapidly changing user preferences require improved timeliness and long-tail coverage [8]; privacy restrictions lead to data scarcity; cross-modal alignment between LLM and GNN is difficult; insufficient cold start solutions and explainability affect user trust [9].

This research proposes a framework integrating LLM and GNN, achieving alignment between text semantics and graph structure. Theoretically, it breaks through GNN's dependence on structured data, achieves feature alignment through contrastive learning, and establishes bidirectional mapping. Practically, it constructs a dual-modal system, leverages textual data to alleviate data scarcity, and enhances user trust through natural language explanations. The research demonstrates model parameter and feature dimension selection, providing a new paradigm for cross-modal recommendations while simultaneously addressing explainability and cold start problems.

2. Framework

2.1. Framework architecture diagram

As shown in Figure 1, the architecture diagram identifies each module in the GLEM-Rec framework, as well as the interaction relationships between modules, while reflecting the complete data flow from metadata and interaction data to the final recommendation results.

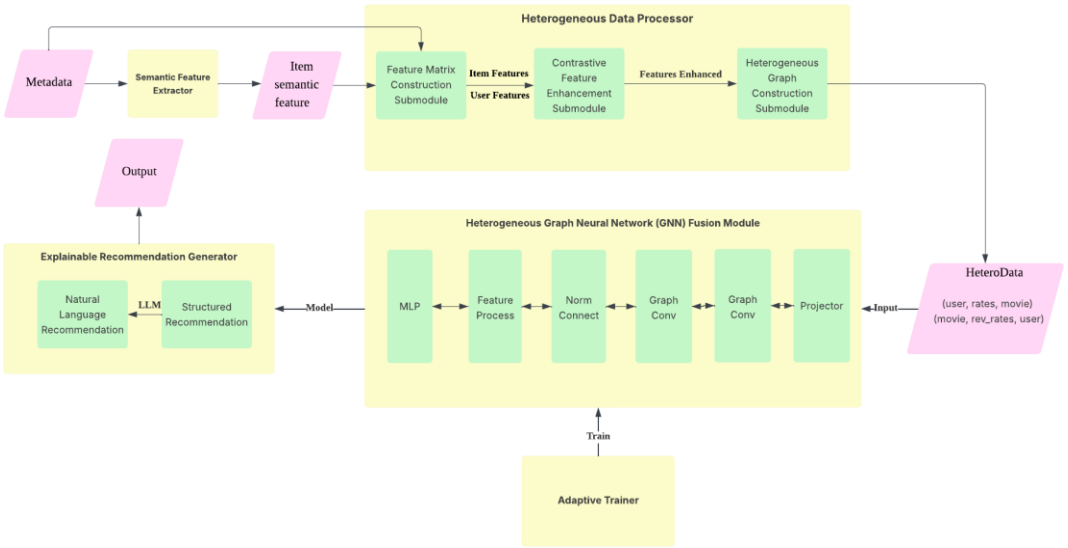

Figure 1: GLEM-Rec framework architecture diagram

2.2. Semantic feature extractor

The semantic feature extractor transforms natural language into learnable feature embeddings for GNN processing, establishing cross-modal alignment between textual and graph structural information. This module manages the comprehensive pipeline of language data collection, preprocessing, tokenization, and feature conversion.

To optimize information utilization, this component aggregates all available textual data as extraction input. It employs enhanced preprocessing techniques to extract critical information from

DOI: 10.54254/2755-2721/165/2025.LD23499

complex text formats, substantially improving model robustness. Addressing the high-dimensionality of textual data, the module implements a Max-Pooling strategy (1) rather than conventional CLS token or Mean-Pooling approaches, which more effectively captures distinctive elements in item descriptions—an approach particularly advantageous for recommendation systems prioritizing item differentiation.

For feature extraction, the module employs BERT-base, RoBERTa, and ALBERT models, selected for their balance of performance and efficiency. These models offer significant advantages in computational resource management, implementation maturity, bidirectional encoding capabilities, and fine-tuning flexibility compared to larger language models such as GPT-3 and ModernBERT, making them appropriate for recommendation systems requiring responsive performance with predominantly short text inputs [10].

2.3. Heterogeneous data processor

The heterogeneous data processor, as a key module for data preparation and quality improvement, includes three functional sub-modules that transform raw data and encoded text features into heterogeneous graph data for direct learning by subsequent models.

2.3.1. Feature matrix construction sub-module

This sub-module maximizes the use of existing data resources, effectively addresses user data missing situations, and supports real-time personalized recommendations.

For item data, the sub-module automatically processes various features, including date conversion, missing data filling, and category encoding. The processed features are concatenated with text features output by the semantic feature extractor to form a complete item feature matrix.

For user data, considering data missing under privacy protection and cold start problems, the module builds user profiles based only on item data and rating data. Specifically, it constructs user profiles through weighted average aggregation of features from items with which users have historically interacted, with weights determined by rating values and time decay factors, enhancing the influence of recent high-rating behaviors. The module also calculates supplementary features including users' historical rating mean, standard deviation, and quantity, ultimately forming the user feature matrix.

$$x_i^{\sim user} = \frac{\sum_{v_j \in V_j} \omega_{ij} \cdot x_j^{item}}{\sum_{v_j} \omega_{ij} + \varepsilon} \tag{1}$$

2.3.2. Contrastive feature enhancement sub-module

This sub-module is dedicated to feature enhancement and semantic space alignment, improving GNN message passing effects and system explainability.

The module maps original user and item features to intermediate representations of the same dimension, projects them to a low-dimensional latent space through two independent projection networks, and applies L2 normalization to ensure calculation stability.

In the contrastive learning stage, it builds similarity matrices using predefined positive sample pairs, making positive sample pairs closer while distinguishing other samples through cross-entropy loss function with temperature scaling. This process is conducted bidirectionally, effectively aligning feature representations of different modalities.

DOI: 10.54254/2755-2721/165/2025.LD23499

$$S_{ij} = \frac{\exp\left(z_i^{user} \cdot \frac{z_j^{item}}{\tau}\right)}{\sum_{k=1}^{N} \exp\left(z_i^{user} \cdot \frac{z_k^{item}}{\tau}\right)} \tag{2}$$

In the feature fusion stage, original features are concatenated with features obtained from contrastive learning, then integrated and reconstructed through a non-linear fusion network, eliminating redundant information and refining joint representations, ultimately outputting highly discriminative features.

2.3.3. Heterogeneous graph construction sub-module

This sub-module integrates user-item interaction data and features to construct a heterogeneous graph structure. The module uses enhanced user and item features as initial embeddings for corresponding nodes, with ratings as edge labels. Besides basic ['user', 'rates', 'item'] edges, the module also constructs reverse edges ['item', 'rev_rates', 'user'], enabling GNN to achieve bidirectional information transmission and more effectively capture potential influence relationships between interacting parties.

2.4. Heterogeneous GNN integrator

The heterogeneous GNN integrator significantly improves recommendation system prediction accuracy by modeling user-item interaction graphs and comprehensively utilizing graph topology structure and node embedding information. This module consists of four key components:

Feature Embedding Layer: Transforms user features and item features into low-dimensional embedding vectors through non-linear mapping, serving as initial representations.

Where $u \in R^{d_u}$ is the original user feature, $m \in R^{d_m}$ is the original item feature, and the outputs are the initial embedding representations of users and items respectively.

Heterogeneous Graph Convolution Layer: Implements message passing on heterogeneous graphs using HeteroConv, including user-to-item rating edges and corresponding reverse edges. Considering the over-smoothing problem that may be caused by data sparsity, each edge type uses GraphSAGE convolution operations to update node features through neighborhood aggregation. The aggregation function can capture high-order interaction relationships between nodes, enhancing model expressiveness [11].

For the ℓ -th layer, message passing can be represented as:

$$x_i^{(\ell+1)} = \text{Norm}\left\{x_i^{(\ell)} + \text{Agg}_{j \in N(i)}\left(x_i^{(l)}, x_j^{(\ell)}\right)\right\} \tag{3}$$

Residual Connection and Layer Normalization: To alleviate the over-smoothing problem in deep graph neural networks, the model adds residual connections and layer normalization operations after each graph convolution layer.

$$x_i^{(\ell+1)} = \text{LayerNorm}\left(x_i^{(\ell+1)} + x_i^{(\ell)}\right) \tag{4}$$

The module also introduces weak residual connections of initial embeddings at the final output layer. This multi-residual design effectively prevents feature degradation during information transmission.

This module effectively captures high-order relationships in user-item interactions through heterogeneous graph neural networks, successfully solving typical problems in graph neural

DOI: 10.54254/2755-2721/165/2025.LD23499

networks. It not only accurately predicts user ratings but also learns semantic features of users and items, providing technical support for personalized recommendations. The model combines L2 regularization to control complexity, has strong generalization capabilities, and can effectively address cold start challenges in recommendation systems, with modeling capabilities for unseen user and item nodes.

2.5. Adaptive trainer

2.5.1. Multi-objective optimization framework

The adaptive trainer proposed in this research adopts a multi-objective optimization strategy, aiming to overcome the limitations of traditional single rating difference loss, comprehensively considering rating prediction accuracy, recommendation diversity, personalized recommendation, and attention to long-tail items. This framework guides model training through a weighted combination of four different loss functions, with the overall loss function represented as:

$$L_{total} = L_{rating} + \lambda_{diversity} \cdot L_{diversity} + \lambda_{personalization} \cdot L_{personalization} + \lambda_{coverage} \cdot L_{coverage} \quad (5)$$

where $\lambda_{diversity}$, $\lambda_{personalization}$, $\lambda_{coverage}$ are hyperparameters controlling the weights of diversity, personalization, and coverage losses. These hyperparameters can be adjusted according to specific application scenarios to balance different optimization objectives.

2.5.2. Core loss function design

This research presents a comprehensive loss function framework addressing multiple recommen dation challenges simultaneously. The rating prediction loss incorporates a weighted mechanism based on item popularity to tackle long-tail distribution problems, assigning higher weights to long-tail items during training:

$$L_{rating} = \frac{1}{|E|} \sum_{(u,i)\in E} \omega_i \cdot (r_{ui} - \hat{r}_{ui})^2$$

where ω_i is calculated as

$$\omega_i = \frac{\max_{j\in I} count_j}{count_i}$$

with $count_i$ representing item i's received ratings.

To enhance recommendation diversity, the diversity loss minimizes average cosine similarity between item embeddings:

$$L_{diversity} = \frac{1}{|I|(|I|-1)} \sum_{i\in I} \sum_{j\in I, j\neq i} \cos\left(e_i, e_j\right)$$

The personalization loss leverages contrastive learning to simultaneously maximize similarity between users and their preferred items while minimizing similarity with disliked items:

$$L_{personalization} = \left(1 - \frac{1}{|H|} \sum_{(u,i)\in H} \cos\left(u_u, e_i\right)\right) + \max\left(0, \frac{1}{|L|} \sum_{(u,j)\in L} \cos\left(u_u, e_j\right)\right)$$

DOI: 10.54254/2755-2721/165/2025.LD23499

Finally, the coverage loss minimizes the standard deviation of item embedding norms, preventing popular items from dominating the embedding space:

$$L_{coverage} = \sigma\left(\left\{ \left\| e_i \right\|_2 \middle| i \in I \right\}\right)$$

2.5.3. Hyperparameter sensitivity analysis

Hyperparameter sensitivity analysis shows that the optimal values for the three key weight parameters ($\lambda_{diversity}$, $\lambda_{personalization}$, $\lambda_{coverage}$) are 200, 7, and 200 respectively. When $\lambda_{diversity}$ and $\lambda_{coverage}$ are too small, the model tends to recommend popular items; when they are too large, prediction accuracy decreases. Parameter $\lambda_{personalization}$ needs moderate values to balance personalized recommendations and exploratory recommendations.

2.6. Explainable recommendation generator

The explainable recommendation generator, as the terminal module of the GLEM-Rec framework, integrates the recommendation capabilities of graph neural networks with the semantic generation capabilities of large language models, solving the black-box problem of traditional recommendation systems. This module achieves cross-modal mapping from low-dimensional graph embeddings to high-dimensional semantic space, constructing a system that provides both personalized recommendations and natural language explanations.

The generator builds a multi-dimensional explanation system, first constructing representation vectors by filtering users' historically preferred items through rating thresholds, while using heterogeneous graph edge prediction mechanisms to simulate user-item interaction probabilities. Second, it builds semantic association networks by combining cosine similarity calculations of item feature embeddings, and establishes genre preference models based on analysis of genre distribution features from user interaction history.

The system generates structured explanations in three core dimensions for top-k items in the recommendation list: genre-based explanations (when recommended items intersect with user-preferred genres), similar item-based explanations (when content similarity exceeds preset thresholds), and semantic matching-based explanations (when item descriptions achieve semantic proximity with user preference features). This structured explanation adopts user-friendly forms, clearly showing recommendation bases, ensuring transparency, credibility, and personalization, while providing high-quality input for subsequent LLM integration.

In terms of converting structured explanations into natural language, this research compared the performance of three models: GPT-4 achieved the highest naturalness score (4.87) but with longer inference latency (~850ms); Mistral 7B had a lower naturalness score (4.23) but significantly reduced latency (~120ms); ChatGLM-6B performed well in Chinese contexts (4.41).

3. Experimental results and analysis

3.1. Dataset description

This research uses Kaggle's The Movies Dataset as the experimental data source, which integra tes movie information and user rating data from TMDB and GroupLens. The dataset contains multidimensional metadata for 45,000 movies and 26 million user ratings, with a sparsity of 99. 46%.

The dataset consists of multiple related files: movies_metadata.csv records basic movie meta data; keywords.csv stores plot keywords; credits.csv presents actors and production team infor

DOI: 10.54254/2755-2721/165/2025.LD23499

mation; links.csv provides cross-platform ID mapping; ratings.csv includes 100,000 rating records from 700 users for 9,000 movies. These files are interconnected through movie IDs, forming a multidimensional analysis system.

This dataset contains rich movie features and interaction information but lacks user information, which aligns with current data distribution scenarios under privacy protection, representing one of the key problems this research aims to solve.

3.2. Evaluation methods

3.2.1. Evaluation framework overview and process

This research employs a comprehensive evaluation framework encompassing multiple metric dimensions. For prediction accuracy, RMSE and MAE quantify rating prediction error. To assess ranking quality, the system incorporates Normalized Discounted Cumulative Gain (NDCG@k), which evaluates the quality of ranked recommendations while considering item relevance and position; Precision@k and Recall@k, which measure the proportion of relevant items in the recommendation list and the proportion of all relevant items that are recommended, respectively; Mean Average Precision (MAP@k), which calculates the average precision across different recall levels; and Mean Reciprocal Rank (MRR), which evaluates the system's ability to rank the first relevant item highly.

For diversity and novelty assessment, this research introduces Coverage, which measures the proportion of all items that appear in recommendation lists across all users:

$$\text{Coverage} = \frac{\left|\bigcup_{u \in U} \text{Rec}_u^k\right|}{|I|}$$

and Novelty, which quantifies the average unexpectedness of recommended items:

$$\text{Novelty} = \frac{1}{|U|} \sum_{u \in U} \frac{1}{\left|\text{Rec}_u^k\right|} \sum_{i \in \text{Rec}_u^k} -\log_2 \frac{|U_i|}{|U|}$$

represents the number of users who have interacted with item i.

Additionally, to analyze popularity bias, items are categorized into popular (Head), medium (Torso), and long-tail (Tail) groups based on interaction frequency, with performance metrics calculated separately for each group to evaluate the model's effectiveness across different popularity segments.

3.3. Baseline model selection

This research selects various models from the recommendation system field for the experimental section, incorporating classical algorithms and diverse model structures that demonstrate excellent performance across different recommendation scenarios, including: Popularity_Baseline; User_CF and Item_CF, which represent collaborative filtering approaches from user and item perspectives respectively; Matrix Factorization, which captures latent factors in user-item interactions; Bayesian Personalized Ranking (BPR); and Neural Collaborative Filtering, which employs neural networks to replace the inner product operation in matrix factorization, thereby modeling non-linear interactions between users and items.

DOI: 10.54254/2755-2721/165/2025.LD23499

3.4. Experimental results and analysis

3.4.1. Overall performance comparison

Table 1: Experimental results-rating prediction aspects

Model/Metric	RMSE	MAE	MRR
GLEM-Rec	0.9122	0.7103	0.9600
Popularity_Based	0.9842	0.7683	0.9658
User-CF	0.9793	0.7619	0.9642
Item-CF	0.9304	0.7097	0.9570
Matrix_Factorization	0.9751	0.7576	0.9666
BPR	1.7783	1.4461	0.9555
NCF	0.9292	0.7173	0.9721

Table 2: Experimental results-top-10 recommendation aspects

Model/Metric	Precision@10	Recall@10	NDCG@10	MAP@10
GLEM-Rec	0.8875	0.7007	0.8732	0.6556
Popularity_Based	0.9170	0.6227	0.8822	0.5909
User-CF	0.9175	0.6233	0.8853	0.5946
Item-CF	0.9150	0.6221	0.8768	0.5862
Matrix_Factorization	0.9080	0.6178	0.8679	0.5852
BPR	0.8875	0.6069	0.8380	0.5622
NCF	0.913	0.6221	0.8841	0.5906

Table 3: Experimental results-diversity aspects

Model/Metric	coverage	Performance on niche tail movies
GLEM-Rec	0.7723	0.9851
Popularity_Based	0.6987	0.9036
User-CF	0.7075	0.9108
Item-CF	0.7143	0.9085
Matrix_Factorization	0.7473	0.9113
BPR	0.5384	0.9002
NCF	0.7230	0.9356

The GLEM-Rec framework demonstrates strong performance across key evaluation metrics. For rating prediction (Table 1), it achieves an RMSE of 0.9122 and MAE of 0.7103, significantly outperforming traditional collaborative filtering models like User-CF (RMSE: 0.9793) and Item-CF (RMSE: 0.9304), indicating substantial reduction in prediction errors.

In Top-10 recommendation quality assessment (Table 2), GLEM-Rec records a Precision@10 of 0.8875 and NDCG@10 of 0.8732. While some benchmark models like User-CF show marginally higher precision (0.9175), GLEM-Rec maintains a superior balance between precision and recall, with a Recall@10 of 0.7007 compared to User-CF's 0.6233.

GLEM-Rec excels particularly in addressing the long-tail recommendation challenge (Table 3), achieving the highest coverage rate (0.7723) among all comparison methods and significantly surpassing popularity-based (0.6987) and traditional collaborative filtering approaches. For long-tail movie recommendations, it achieves an exceptional performance measure of 0.9851, substantially

DOI: 10.54254/2755-2721/165/2025.LD23499

outperforming other methods. This demonstrates how integrating language models' semantic understanding with graph neural networks' structural knowledge effectively overcomes traditional recommendation systems' limitations when handling items with limited interaction history.

3.4.2. Ablation experiments

Table 4: Ablation experiment results

Model/Metric	GLEM-Rec	Without Semantic Extractor	Without Heterogeneous Data Processor	Without Heterogeneous GNN Integrator
RMSE	0.8800	1.0248	0.9334	0.9427
MAE	0.6902	0.8034	0.7198	0.8253
MRR	0.9731	0.9533	0.9688	0.9512
Precision@10	0.9086	0.8837	0.8705	0.8727
Recall@10	0.6994	0.6855	0.7121	0.6881
coverage	0.7327	0.7259	0.7400	0.7193
Performance on tail movies	0.9607	0.9703	0.9486	0.9382

To verify the contribution of each component in the GLEM-Rec framework, ablation experiments were conducted, as shown in Table 4. Removing the semantic extractor led to significant performance declines across all metrics, with RMSE increasing from 0.8800 to 1.0248 and MAE increasing from 0.6902 to 0.8034. This confirms the critical role of the LLM component in extracting meaningful semantic features from item descriptions and user profiles.

Similarly, the heterogeneous data processor proved essential, with its removal leading to increased error metrics (RMSE: 0.9334, MAE: 0.7198) and decreased performance in standard and long-tail recommendations. Interestingly, while standard GNN performed reasonably well in some aspects, it could not match GLEM-Rec's performance in accuracy (RMSE: 0.9427 vs. 0.8800) and long-tail item recommendation (0.9382 vs. 0.9607).

The results of the ablation experiments indicate that each module in the proposed framework makes a significant contribution to overall performance, and their integration creates synergies that enable GLEM-Rec to surpass traditional methods and individual modules.

4. Conclusion

This research effectively integrates graph neural networks (GNN) with large language models (LLM) through the GLEM-Rec framework, establishing a cross-modal fusion recommendation system solution. Results demonstrate GLEM-Rec's exceptional performance across multiple metrics: competitive prediction accuracy (RMSE and MAE), balanced precision-recall in Top-N recommendations, enhanced item coverage, superior long-tail item recommendations, and explainable recommendation capabilities.

Experimental findings confirm the research hypothesis that cross-modal fusion of graph structure and semantic information creates a more comprehensive recommendation framework, effectively addressing limitations in long-tail item representation, recommendation diversity, and explainability.

In practical application, the system recommended "The Cotton Club" to user 670 with a predicted rating of 4.52, based on past preferences for films like "My Best Friend's Wedding" and "Bad Boys II." The recommendation included detailed rationale highlighting the film's crime drama and romantic elements, Harlem jazz venue setting, and underworld-romance narrative. Additional

DOI: 10.54254/2755-2721/165/2025.LD23499

recommendations of "Beyond the Sea," "The Age of Innocence," and "Ragtime" further demonstrated the system's understanding of user preferences and capacity for recommendation diversity.

Future research directions include: utilizing timestamp information to create dynamic temporal graph structures capturing behavioral evolution patterns; developing time-window-based graph sampling strategies; upgrading from BERT to more advanced language models (GPT-4 or Claude series) for enhanced semantic understanding; and implementing attention mechanisms to dynamically adjust weights between semantic and structural features across different scenarios.

References

[1] *Wu C., Wu F., Qi T., et al. Mm-rec: multimodal news recommendation [J]. arXiv preprint arXiv:2104.07407, 2021.*

[2] *Hong Y., Li H., Wang X., Lin C. (2020). DEAMER: A Deep Exposure-Aware Multimodal Content-Based Recommendation System. Database Systems for Advanced Applications. DASFAA 2020. Lecture Notes in Computer Science vol 12114. Springer, Cham. https://doi.org/10.1007/978-3-030-59419-0_38*

[3] *Fan W., Ma Y., Li Q., et al. Graph neural networks for social recommendation [C]// The world wide web conference. 2019: 417-426.*

[4] *Wu Q., Zhang H., Gao X., et al. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems [C]// The world wide web conference. 2019: 2091-2102.*

[5] *Liu Q., Zhu J., Yang Y., et al. Multimodal pretraining, adaptation, and generation for recommendation: A survey [C]// Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024: 6566-6576.*

[6] *Xu C., Guan Z., Zhao W., et al. Recommendation by users' multimodal preferences for smart city applications [J]. IEEE Transactions on Industrial Informatics, 2020, 17(6): 4197-4205.*

[7] *Liu Q., Hu J., Xiao Y., et al. Multimodal recommender systems: A survey [J]. ACM Computing Surveys, 2024, 57(2): 1-17.*

[8] *Jiang J., Zhang M. Overspinning a rotating black hole in semiclassical gravity with type-A trace anomaly [J]. The European Physical Journal C, 2023, 83(8): 687.*

[9] *Hongzhi Wen, Jiayuan Ding, Wei Jin, Yiqi Wang, Yuying Xie, and Jiliang Tang. (2022). Graph Neural Networks for Multimodal Single-Cell Data Integration. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22). Association for Computing Machinery, New York, NY, USA, 4153–4163. https://doi.org/10.1145/3534678.3539213*

[10] *Reimers N., Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks[J]. arXiv preprint arXiv:1908.10084, 2019.*

[11] *Hamilton W., Ying Z., Leskovec J. Inductive representation learning on large graphs [J]. Advances in neural information processing systems, 2017, 30.*