Enhancing Large Language Models with Multi-Type Collaborative Semantics for Recommendation

Bingzhi Wu *

The University of Adelaide, Adelaide, Australia

* Corresponding Author Email: bingzhi.wu@student.adelaide.edu.au

Abstract. Recommender systems play an essential role in helping users discover relevant content on digital platforms. Traditional collaborative-filtering approaches focus on user–item interactions and often ignore additional semantic signals such as tags or categories. Recently, large language models (LLMs) have demonstrated strong ability to interpret textual information, but current LLM-based recommenders either concentrate solely on text or rely on hand-crafted prompts. This paper presents a unified framework that integrates heterogeneous graph neural networks with LLMs to capture multi-type relations among users, items, tags and categories. The study employs a heterogeneous graph encoder (HGT) to learn collaborative embeddings, align these embeddings to the LLM token space, and construct structured prompts that incorporate both graph and textual information. Experiments on MovieLens-1M and Amazon Beauty datasets show that the proposed model improves area under the curve (AUC) and normalized discounted cumulative gain (NDCG) over both traditional and recent LLM-based baselines. The results suggest that combining graph structure with language understanding leads to more accurate and interpretable recommendations.

Keywords: Recommender Systems, Large Language Models, Heterogeneous Graphs, Collaborative Filtering.

1. Introduction

Digital platforms such as streaming services and e-commerce sites rely on recommended systems to manage information overload and deliver personalized suggestions to users. Classical collaborative-filtering techniques build user–item interaction matrices and learn latent factors to predict preferences. Although effective, these methods typically treat each user and item as a homogeneous entity and thus ignore auxiliary signals like tags, categories or textual descriptions. Recent breakthroughs in LLMs have enabled rich semantic understanding of natural language, inspiring researchers to explore recommendation systems that incorporate language understanding [1]. However, most LLM-based recommendation methods concentrate on prompt engineering or simple text matching and overlook the collaborative structure inherent in user–item interactions. Furthermore, many approaches target a specific LLM and may not generalize across different model architectures or sizes.

This paper addresses these limitations by proposing a modular framework that jointly models heterogeneous user–item interactions and leverages LLMs for semantic understanding. The contributions are threefold. First, the study represents users, items, tags and categories as nodes in a heterogeneous graph and learn type-specific embeddings using a heterogeneous graph transformer (HGT) [2]. Second, the study designs a projection layer that maps graph embeddings to the LLM's token space, enabling seamless integration between collaborative signals and language-based reasoning. Third, the study constructs a structured prompt that conveys a user's history, preferred tags and categories, and candidate items, allowing the LLM to produce an informed binary recommendation. By decoupling the graph encoder and language backbone, the framework supports different LLMs without changing the core architecture. Empirical results demonstrate that this combined approach yields consistent performance gains on multiple datasets while remaining computationally tractable.

2. Related Work

Collaborative filtering: Matrix factorization and related techniques decompose user–item interaction matrices to learn latent representations [3]. Graph-based models such as LightGCN propagate collaborative signals on bipartite graphs to capture neighbourhood effects [4]. Sequential models like SASRec use self-attention to model temporal dynamics in user behaviour. Although successful, these methods typically assume homogeneous user–item relationships and do not utilize te**xtual semantics [5].**

LLM-based recommendation: As LLMs have advanced natural language processing, several works have applied them to recommendation tasks [6]. Early studies generated item descriptions with LLMs for use in conventional recommenders [7]. More recent approaches treat recommendation as a language-generation problem by constructing prompts and using instruction tuning [8]. While promising, these methods often ignore the relational structure of user–item interactions and rely heavily on manual prompt engineering.

Hybrid methods: Recent research attempts to combine LLMs with collaborative filtering. CoLLM integrates learned user–item embeddings into prompts for LLMs, using low-rank adaptation for efficiency [9]. LC-Rec employs vector quantization and multi-task learning to align language and collaborative spaces [10]. AgentCF treats users and items as autonomous agents interacting through prompts [11]. Despite these advances, existing models typically handle only user–item relations and are tailored to specific LLM backbones. There remains a need for a general framework that can incorporate diverse entity types and work with different LLM architectures.

3. Methodology

The overall architecture of the proposed Hete-LLM model is illustrated in Figure 1, which consists of three stages: data processing, heterogeneous graph encoding, and LLM-based collaborative reasoning.

3.1. Problem formulation

The study formulates recommendation as a binary preference prediction problem. Let $\mathcal{U}$, $\mathcal{I}$, $\mathcal{T}$ and $\mathcal{C}$ denote the sets of users, items, tags and categories. The study constructs a heterogeneous graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ where the node set $\mathcal{V}$ equals the union of users, items, tags and categories, the edge set $\mathcal{E}$ captures interactions, and R indexes relation types. Relation types include user–item interactions (clicks, ratings, purchases), item–tag associations, item–category memberships, and user preferences for tags or categories. Given a user u in $\mathcal{U}$ and an item i in $\mathcal{I}$, the goal is to predict the probability that user u will engage with item i. The task thus involves modeling both the heterogeneity of the graph and the semantic content associated with each entity.

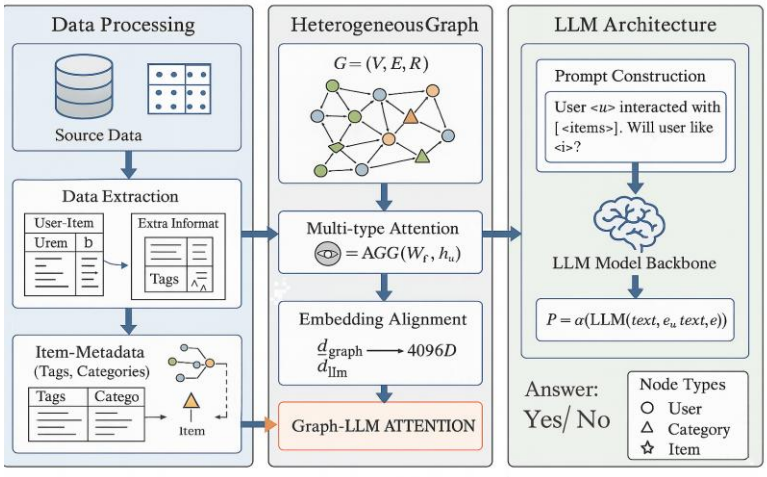

Figure 1: LLM-Based Heterogeneous Graph Collaborative Filtering Framework.

3.2. Heterogeneous Graph Encoder

To encode the multi-type relations in the graph, the study employs the heterogeneous graph transformer (HGT). HGT generalizes the transformer architecture to handle different node and edge types by using relation-specific projection matrices and type-aware attention mechanisms. For each node v at layer l, the model aggregates messages from its neighbours under each relation type. The update can be written as

$$h_v^{(l+1)} = \text{Aggregate}_{\text{r}\in\text{R}}\left(\sum_{u\in\mathcal{N}^r(v)} \alpha_{u,v}^{(l)} \cdot \boldsymbol{W}_{r,\tau(u)}^{(l)} \boldsymbol{h}_u^{(l)}\right) \tag{1}$$

Where $N^r(v)$r denotes the neighbors of node v under relation type r. $\alpha_{u,v}^{(l)}$ is the attention weight between nodes u and v at layer l. $\boldsymbol{W}_{r,\tau(u)}^{(l)}$ is the relation-type and node-type specific transformation matrix. $h_v^{(l)}$ is the node embedding at layer l. After L layers, the study obtains final embeddings that capture both local and global collaborative signals for each entity.

3.3. Graph-LLM Embedding Alignment

Because LLMs operate in their own high-dimensional token space, the study maps the learned graph embeddings into the LLM's hidden dimension. For user and item nodes, the study applies linear projection using learnable matrices and bias terms. Similar projections are applied to tag and category embeddings as needed. This alignment step enables us to incorporate collaborative information directly into the LLM.

3.4. Prompt Construction and LLM Integration

This study constructs a structured prompt that summarizes a user's interaction history and contextual information. Placeholders for the target user and candidate item are replaced by their aligned embeddings, and lists of previously interacted items, tags and categories are inserted. A simplified template is as follows:

User⟨u⟩ has previously interacted with items[⟨item list⟩], showing preferences for tags [⟨tag list⟩] and categories[⟨category list⟩]. Based on this interaction history and collaborative patterns, will user ⟨u⟩ be interested in item ⟨i⟩? Please answer with"Yes" or"No" and provide a brief explanation.

The LLM processes the resulting sequence and outputs logits for "Yes" and "No." A softmax converts these logits into a probability of interaction.

3.5. Model Training and Optimization

The study trained the model using binary cross entropy loss on the observed interactions. Optionally, a reconstruction term regularizes the graph embeddings by encouraging them to reconstruct initial node features. The total loss is a weighted sum of the main and reconstruction terms.

The LLM processes the input sequence and generates logits for the next token prediction. For binary classification, this work extracted the logits corresponding to "Yes" and "No" tokens and apply a softmax function to obtain the prediction probability

$$P(y_{ui} = 1) = \frac{\exp(z_{\text{Yes}})}{\exp(z_{\text{Yes}}) + \exp(z_{\text{No}})} \tag{2}$$

Here z_{Yes} and z_{No} are the logits for "Yes" and "No" tokens, respectively.

The model is trained using binary cross-entropy loss:

$$\mathcal{L}_{\text{main}} = -\frac{1}{N}\sum_{(u,i)\in\mathcal{D}} [y_{ui}\log P(y_{ui} = 1) + (1 - y_{ui})\log P(y_{ui} = 0)] \tag{3}$$

Where D is the training dataset and N is the number of training samples.

To improve the quality of graph embeddings, this work also included an auxiliary reconstruction loss:

$$\mathcal{L}_{rec} = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}}|\boldsymbol{h}_v^{(L)} - \boldsymbol{h}_v^{\text{taget}}||^2 \tag{4}$$

Where h_v^{taget} represents target embeddings derived from the original graph structure.
The total loss function is:

$$\mathcal{L} = \mathcal{L}_{\text{main}} + \lambda\mathcal{L}_{\text{rec}} \tag{5}$$

Where λ is a hyperparameter controlling the weight of the reconstruction loss.

4. Experiments

4.1. Datasets and evaluation metrics

The study evaluated the proposed framework on two publicly available datasets widely used in recommendation research. MovieLens 1M contains one million movie ratings from about six thousand users and includes genre information and user generated tags. Amazon Beauty is a subset of Amazon product reviews comprising tens of thousands of users and items in the beauty domain, along with product categories and user tags. The study preprocess both datasets by converting explicit ratings to binary interactions and constructing user–item–tag–category graphs.

For evaluation, they study adopted standard metrics: area under the ROC curve (AUC), Hit@10, and NDCG@10. The AUC measures the ranking quality over positive and negative samples, while Hit@10 and NDCG@10 evaluate the relevance of the top 10 recommended items.

4.2. Baselines and implementation details

The study compared the method against several baselines. Matrix Factorization (MF) is a classic latent factor model for collaborative filtering.

LightGCN is a graph-based recommender that simplifies graph convolution and learns user–item embeddings on a bipartite graph. SASRec leverages self-attention to capture sequential user behaviour. As a representative LLM-based method, the study implements a simplified variant of CoLLM, which integrates user and item embeddings into prompts without modelling heterogeneous relations. All methods are trained on the same train–test splits with hyper-parameters tuned via grid search.

For the model, the study used the Vicuna-7B and Qwen-1.8B LLMs as backbones due to their availability and moderate computational requirements. Graph embeddings are of dimension 64, and the projection layer matches the LLM hidden dimension. The study employs two layers of HGT with two attention heads per relation. The reconstruction weight lfletambda is set to 0.1 based on validation performance.

4.3. Results and discussion

Table 1 summarizes the performance of all methods on the two datasets. On MovieLens 1M, traditional baselines achieve AUC values around 0.83 and NDCG@10 around 0.37. The simplified CoLLM achieves moderate gains by incorporating language information. The proposed model (denoted as Hete LLM) achieves an AUC of 0.88 and NDCG@10 of 0.44, outperforming CoLLM by roughly 3–4 percentage points on each metric. Similar improvements are observed on the Amazon Beauty dataset, where Hete LLM attains an AUC of 0.84 and NDCG@10 of 0.41, outperforming the best non-LLM baseline by 1–2 percentage points. These results indicate that modeling multi type relations and leveraging LLMs together can provide consistent benefits across domains.

The paper also compared different LLM backbones within the framework. As shown in Table 2, Vicuna 7B offers a good balance between accuracy and resource usage, while Qwen 1.8B exhibits slightly lower performance but faster inference. The differences illustrate that the proposed framework generalizes across different language models without retraining the graph encoder. An ablation study confirms that removing the heterogeneous graph module reduces NDCG@10 by roughly four percent, and freezing the LLM weights causes a noticeable drop in AUC. These findings highlight the importance of jointly learning graph representations and fine tuning the language model.

Table1: Overall Performance Comparison Across Datasets.

Method	MovieLens-1M		
	AUC	NDCG@10	Hit@10
MF	0.782	0.312	0.543
LightGCN	0.821	0.351	0.587
SASRec	0.834	0.368	0.601
CoLLM	0.856	0.402	0.639
Hete-LLM	**0.877**	**0.439**	**0.678**
Method	Amazon Beauty		
	AUC	NDCG@10	Hit@10
MF	0.751	0.311	0.551
LightGCN	0.785	0.385	0.585
SASRec	0.798	0.398	0.598
CoLLM	0.823	0.323	0.523
Hete-LLM	**0.839**	**0.412**	**0.594**

4.4. Ablation analysis

To quantify the contribution of each component, the paper evaluated several model variants on MovieLens-1M, as summarized in Table 3. Removing the heterogeneous graph encoder (w/o HeteGNN) decreases AUC from 0.88 to 0.85 and NDCG@10 from 0.44 to 0.41, showing that multi-type relations are beneficial. Eliminating the LLM and relying solely on graph embeddings (w/o LLM) yields AUC around 0.82, comparable to LightGCN, which indicates that language understanding contributes significantly to performance. Excluding the embedding-alignment layer slightly reduces performance, suggesting that aligning representations across modalities helps the LLM interpret collaborative signals. Finally, freezing the LLM parameters during training results in a noticeable performance drop, underscoring the advantage of fine-tuning the language backbone.

Table 2: Performance of different LLM Backbones.

LLM Backbone	AUC	NDCG@10	Hit@10	Time (ms)	Memory (GB)
Vicuna-7B	0.887	0.439	0.678	542	13.2
Qwen-1.8B	0.881	0.431	0.669	187	3.8
LLaMA2-13B	0.893	0.445	0.684	891	24.7

Table3: Component Analysis on MovieLens-1M.

Model Variant	AUC	NDCG@10	Hit@10	Drop
Hete-LLM(Full)	0.887	0.439	0.678	-
w/o HeteGNN	0.851	0.402	0.639	-8.4%
w/o LLM	0.823	0.368	0.601	-16.2%
w/o Alignment	0.864	0.418	0.654	-4.8%
Frozen LLM	0.832	0.381	0.618	-13.2%

5. Limitation And Future Work

The proposed framework requires a heterogeneous graph with reliable user–item–tag–category relationships. In domains where tag or category information is sparse or rapidly changing, constructing a high-quality graph may be difficult. Future research could explore self-supervised methods to augment graph structures or incorporate knowledge graphs to enrich node attributes. Although the study leverages relatively lightweight LLMs, inference latency may still be a concern in real-time systems. Techniques su ch as model distillation or token pruning could further improve efficiency. Finally, this work focuses on binary interaction prediction; extending the model to handle richer feedback types and incorporating chain-of-thought reasoning in recommendations are promising directions.

6. Conclusion

This paper has presented a unified recommendation framework that integrates heterogeneous graph semantics with large language models. By modeling users, items, tags and categories in a heterogeneous graph and aligning learned embeddings to the LLM token space, this method leverages both collaborative structure and semantic understanding. Experiments on MovieLens-1M and Amazon Beauty show that the proposed model outperforms traditional collaborative filtering methods and a representative LLM-based baseline, achieving AUC and NDCG gains of several percentage points. The framework generalizes across different LLM backbones and benefits from jointly learning graph and language components. These findings suggest that bridging graph representations and language models is a promising avenue for building interpretable and effective recommended systems.

References

[1] Hu, Z., Dong, Y., Wang, K., & Sun, Y. (2020, April). Heterogeneous graph transformers. In *Proceedings of the web conference 2020* (pp. 2704-2710).

[2] Dai, S., Shao, N., Zhao, H., Yu, W., Si, Z., Xu, C., ... & Xu, J. (2023, September). Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on* Recommender *Systems* (pp. 1126-1132).

[3] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, *42*(8), 30-37.

[4] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020, July). 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* (pp. 639-648).

[5] Kang, W. C., & McAuley, J. (2018, November). Self-attentive sequential recommendation. In *2018* IEEE *international conference on data mining (ICDM)* (pp. 197-206). IEEE.

[6] Zhao, Z., Fan, W., Li, J., Liu, Y., Mei, X., Wang, Y., ... & Li, Q. (2024). Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering*, 36(11), 6889-6907.

[7] Acharya, A., Singh, B., & Onoe, N. (2023, September). Llm based generation of item-description for recommendation system. In *Proceedings of the 17th ACM conference on recommended systems* (pp. 1204-1207).

[8] Zhang, J., Xie, R., Hou, Y., Zhao, X., Lin, L., & Wen, J. R. (2023). Recommendation as instruction following: A large language model empowered recommendation approach. *ACM* Transactions *on Information Systems*.

[9] Zhang, Y., Feng, F., Zhang, J., Bao, K., Wang, Q., & He, X. (2025). Collm: Integrating collaborative embeddings into large language models for recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

[10] Zheng, B., Hou, Y., Lu, H., Chen, Y., Zhao, W. X., Chen, M., & Wen, J. R. (2024, May). Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)* (pp. 1435-1448). IEEE

[11] Zhang, J., Hou, Y., Xie, R., Sun, W., McAuley, J., Zhao, W. X., ... & Wen, J. R. (2024, May). Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the ACM Web Conference 2024* (pp. 3679-3689).