



# Training Large Recommendation Models via Graph-Language Token Alignment

Mingdai Yang  
myang72@uic.edu  
University of Illinois at Chicago  
Chicago, USA

Zhiwei Liu  
zhiweiliu@salesforce.com  
Salesforce AI Research  
Palo Alto, USA

Liangwei Yang  
liangwei.yang@salesforce.com  
Salesforce AI Research  
Palo Alto, USA

Xiaolong Liu  
Chen Wang  
xliu262@uic.edu  
cwang266@uic.edu  
University of Illinois at Chicago  
Chicago, USA

Hao Peng\*  
penghao@buaa.edu.cn  
Beihang University, & Hangzhou  
Innovation Institute of BUAA  
Beijing & Hangzhou, China

Philip S. Yu  
psyu@uic.edu  
University of Illinois at Chicago  
Chicago, USA

## Abstract

Recommender systems (RS) have become essential tools for helping users efficiently navigate the overwhelming amount of information on e-commerce and social platforms. However, traditional RS relying on Collaborative Filtering (CF) struggles to integrate the rich semantic information from textual data. Meanwhile, large language models (LLMs) have shown promising results in natural language processing, but directly using LLMs for recommendation introduces challenges, such as ambiguity in generating item predictions and inefficiencies in scalability. In this paper, we propose a novel framework to train Large Recommendation models via Graph-Language Token Alignment. By aligning item and user nodes from the interaction graph with pretrained LLM tokens, **GLTA** effectively leverages the reasoning abilities of LLMs. Furthermore, we introduce Graph-Language Logits Matching (GLLM) to optimize token alignment for end-to-end item prediction, eliminating ambiguity in the free-form text as recommendation results. Extensive experiments on three benchmark datasets demonstrate the effectiveness of GLTA, with ablation studies validating each component.

## CCS Concepts

• Information systems → Retrieval models and ranking.

## Keywords

Recommender System; Large Language Models

### ACM Reference Format:

Mingdai Yang, Zhiwei Liu, Liangwei Yang, Xiaolong Liu, Chen Wang, Hao Peng, and Philip S. Yu. 2025. Training Large Recommendation Models via Graph-Language Token Alignment. In *Companion Proceedings of the ACM WWW Companion '25, Sydney, NSW, Australia*.

\*Corresponding author

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

WWW Companion '25, Sydney, NSW, Australia

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1331-6/25/04

<https://doi.org/10.1145/3701716.3715583>

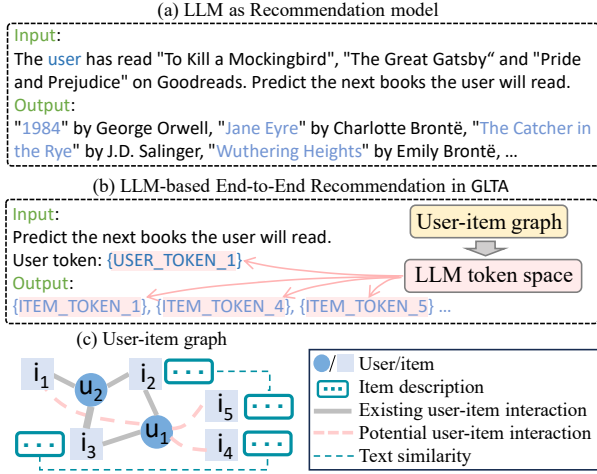
Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3701716.3715583>

## 1 Introduction

With the development of e-commerce and social platforms, information collection and decision-making have become essential yet overwhelming for individual customers. RS plays a key role in simplifying these tasks by offering personalized suggestions, and the recent successes of Graph Neural Networks (GNNs) have been developed to learn RS from graphs [3, 12]. Despite their effectiveness, graph-based RS often struggle to integrate rich semantic information from textual data, limiting their ability to capture nuanced user preferences and item characteristics. Leveraging large language models (LLMs) can bridge this gap by enhancing understanding of textual data and improving recommendation accuracy [7, 14].

A straightforward idea is to train an LLM on recommendation data to serve as a recommender [2]. While previous studies have demonstrated that LLMs can function as recommenders [1], they overlook that established graph-based recommendation models effectively leverage user-item interactions for collaborative filtering [13, 15]. Moreover, when the LLM generates language tokens as item predictions, the output free-form text introduces ambiguity when matching the actual items, compared to traditional recommendation models that output a clear list of item IDs. To mitigate this, additional post-processing is needed to parse text back to specific items, which hinders the actual performance when candidate items are similar. Instead of deploying LLMs directly as RS, some recent works integrate LLM embeddings as supplementary features to enhance existing recommendation models [7, 11]. However, the recommendation process in their approaches is primarily driven by the graph-based model, leaving the reasoning capabilities of LLMs underutilized.

To effectively harness the powerful reasoning capability of LLMs for recommendation, we propose a novel framework, training large recommendation models via Graph-Language Token Alignment, which aligns LLMs with graphs using a carefully designed token alignment paradigm. To be concrete, we first align items with their text descriptions to obtain item tokens, and then align users with pretrained item and text tokens for recommendation. To implement



**Figure 1: A toy example of the input and the output in GLTA, compared with directly deploying LLM for recommendation.**

end-to-end recommendation, we design a GLLM layer to optimize the token alignment by matching predicted item logits with ground-truth items. This GLLM layer eliminates the hallucination issue that non-existing items are generated when directly using LLM outputs as item prediction results. As shown in Figure 1, different from directly adopting LLMs as recommendation models, our GLTA accurately generates existing item tokens instead of plain text outputs. The key contributions of this paper are:

- We propose a novel recommendation framework, GLTA, integrating LLMs and recommendation in an end-to-end manner, where each output of the LLM corresponds precisely to an item in RS, eliminating the hallucination issue and the ambiguity from free-form text as output.
- We adaptively align nodes pretrained on the graph with pretrained tokens of the LLM by token projectors, and introduce a novel GLLM layer for optimizing end-to-end recommendation based on these aligned node tokens. Only projectors and GLLM layers are finetuned for this efficient end-to-end prediction.
- To verify the preeminence of GLTA, we conduct extensive experiments on three publicly available benchmark datasets. The effectiveness of each component in GLTA is verified through ablation studies.

## 2 Preliminary

### 2.1 Recommendation Task

Given two disjoint node sets, including a user set  $\mathcal{U}$  and an item set  $\mathcal{I}$ , and the interactive edges, i.e., user-item edges  $E_{\mathcal{U}, \mathcal{I}}$ , an interaction graph is defined as  $\mathcal{G} = (V, E)$  where  $V = \mathcal{U} \cup \mathcal{I}$ . Besides, each item  $i$  has a text description. An end-to-end recommendation task for a user  $u$  is to predict a ranking list of items  $\{i_1, i_2, \dots, i_m\}$ , with which this user has no interactions in the graph  $\mathcal{G}$ .

### 2.2 Graph Pretraining

Graph-based CF enhances recommendation by using graph structures to model user-item interactions. In this work, we use LightGCN [3] as a graph-based CF method to capture and encode structure information on the user-item graph. In the first stage, user node embeddings and item node embeddings are pretrained as  $\mathbf{E}_u \in \mathbb{R}^{|\mathcal{U}| \times d}$  and  $\mathbf{E}_i \in \mathbb{R}^{|\mathcal{I}| \times d}$  and frozen in the following alignment stages, where  $d$  denotes the dimension of pretrained node embeddings.

## 3 Proposed Framework: GLTA

### 3.1 Item-text Alignment

The proposed GLTA is shown in Figure 2. Following graph pretraining, it is essential to align item node embeddings  $\mathbf{E}_i$  with item descriptions tokens pretrained from the LLM. These two types of embeddings typically reside in different spaces. The item node embeddings capture collaborative signals from the graph, while the text embeddings capture semantic information from the textual descriptions. Inspired by GraphGPT [8], we apply a simple linear layer as an item node projector that maps these item nodes into the same language token space with the descriptions of these items:

$$\mathbf{V}_i = \mathbf{W}_i \mathbf{E}_i + \mathbf{b}_i, \quad (1)$$

where  $\mathbf{V}_i \in \mathbb{R}^{|\mathcal{I}| \times d}$  is the embeddings of item tokens.  $\mathbf{W}_i$  and  $\mathbf{b}_i$  denote the weight and bias of the item node projector. Then, an item-text alignment instruction template, shown in Figure 2 (a), queries the LLM to reorder the item language information and match the token with the predicted logits.

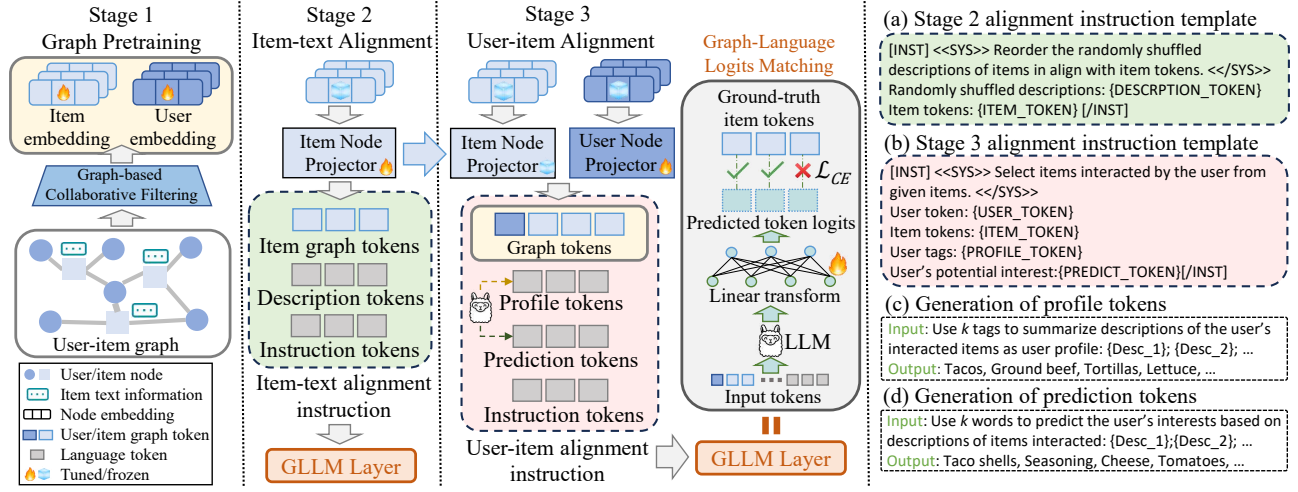
### 3.2 User-item Alignment

Besides the item-text alignment, a user-item alignment is proposed to allow the LLM to process both user and item information in the same context. Similarly, a linear layer is used as a user node projector to map user nodes into the LLM token space.

$$\mathbf{V}_u = \mathbf{W}_u \mathbf{E}_u + \mathbf{b}_u, \quad (2)$$

where  $\mathbf{V}_u \in \mathbb{R}^{|\mathcal{U}| \times d}$  is the embeddings of user tokens.  $\mathbf{W}_u \in \mathbb{R}^{d \times d}$  and  $\mathbf{b}_u \in \mathbb{R}^{1 \times d}$  denote the weight and bias of the user node projector. This projector establishes the correspondence between the user nodes, the language tokens and the item tokens pretrained in previous item-text alignment.

Besides these user tokens and pretrained item tokens, we introduce profile tokens and prediction tokens into the user-item alignment instruction template. These profile and prediction tokens are generated by the LLM based on item descriptions. The instruction templates for generating profile and prediction tokens are shown in Figure 1(c) and Figure 1(d), respectively. In this way, user node embeddings are mapped to the same token space with semantic characteristics reflecting their historical interactions and potential preferences. Then, the user-item alignment instruction template is fed into the LLM to generate the predicted item tokens for each user. During training, the item token prediction is optimized by GLLM for end-to-end recommendation.



**Figure 2: The framework of GLTA consists of three stages: Graph Pretraining, Item-text Alignment, and User-item Alignment. The instruction templates are shown on the right of the figure with the generation process of profile and prediction tokens.**

### 3.3 Graph-Language Logits Matching

In a traditional setup when using LLMs as RS, additional steps are required to parse the text and map it back to specific items [2, 8]. To eliminate the ambiguity that arises when interpreting free-form text outputs of the LLM, a GLLM layer is designed for the end-to-end recommendation in GLTA. After inputting the instruction template to the LLM, a linear layer is applied to transform the last-layer hidden states of the LLM into item token logits  $Z_i^u \in \mathbb{R}^{L \times |I|}$ , where  $L$  denotes the maximum sequence length in the LLM. Then, a cross-entropy loss is applied to match the predicted logits to the ground-truth items interacted by the user:

$$\mathcal{L}_{CE} = -\frac{1}{L} \sum_{t=1}^L \log \left( \frac{\exp(Z_i[t, y_{i,t}^+])}{\sum_{j=1}^{|I|} \exp(Z_i[t, j])} \right), \quad (3)$$

where  $y_{i,t}^+$  is the ground-truth item ID at position  $t$  in the sequence, and  $Z_i[t, y_{i,t}^+]$  represents the logit corresponding to the ground-truth item at position  $t$ . This cross-entropy loss provides direct supervision by comparing the model's predicted probability distribution against the actual ground-truth item IDs. In real-world datasets, the number of items interacted by the user is not necessarily equal to  $L$ , and the order information of the interacted item  $y_{i,t}^+$  can be unavailable. In that case, we only optimize item token logits in the first  $k$  positions, according to  $k$  ground-truth items randomly shuffled from all the items interacted by the user. In this work, we use a quantized version of LLaMA-2-7B<sup>1</sup> as the LLM for training efficiency and adopt Adam [5] as the optimizer.

## 4 Experiment

### 4.1 Experiment Settings

**4.1.1 Datasets.** We conduct experiments on three publicly available datasets: Goodreads [9], Amazon [6] and MovieLens<sup>2</sup>. We use

<sup>1</sup><https://huggingface.co/TheBloke/Llama-2-7B-GPTQ>

<sup>2</sup><https://grouplens.org/datasets/movielens/1m/>

**Table 1: Statistics of the Datasets**

Dataset	Goodreads	Amazon	MovieLens
#Users	10,131	1,032	6,040
#Items	10,725	1,7609	3,706
#U-I interactions	478,334	30,510	1,000,209
Density	0.440%	0.168%	4.468%

history books and groceries as items with their corresponding descriptions in Goodreads and Amazon datasets, respectively. For MovieLens, we regard movies as items and movie genres as item descriptions since no movie descriptions are provided in this dataset. The details of the datasets are shown in Table 1.

**4.1.2 Baselines.** To demonstrate the effectiveness of GLTA, we compare it with three groups of representative baselines. 1.) GNN-based recommendation with only interaction information (LightGCN [3], HCCF [12]) that applies graph or hypergraph neural network on the user-item interaction graph for information propagation. 2.) GNN-based recommendation with text information (LightGCN+, HCCF+) that uses InfoNCE loss to align the encoded node embeddings with description embeddings from a sentence transformer [10]. 3.) LLM-based recommendation [7] that leverages contrastive (RLMRec-Con) or generative (RLMRec-Gen) alignment.

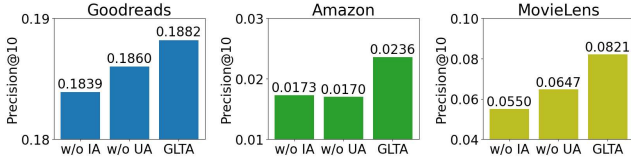
**4.1.3 Evaluation Metrics.** We evaluate the recommendation in an end-to-end manner by ranking the test users with all non-interacted items. Precision (P@5, P@10) and NDCG (N@5, N@10) are adopted as evaluation metrics.

### 4.2 Overall Performance

Performance comparison between GLTA and other baselines are shown in Table 2. We have the following observations. First, GLTA exhibits superior performance on all datasets, especially in MovieLens where the dense user-item interactions lead to over-smoothing

**Table 2: Overall performance comparison. The best and second-best methods are in boldface and underlined.**

Dataset	Goodreads				Amazon				MovieLens			
Metric	P@5	P@10	N@5	N@10	P@5	P@10	N@5	N@10	P@5	P@10	N@5	N@10
LightGCN	0.1999	0.1618	0.2346	0.2352	0.0204	0.0139	0.0301	0.0305	0.0470	0.0426	0.0463	0.0466
HCCF	0.1998	0.1632	0.2371	0.2439	0.0209	0.0143	0.0275	0.0304	0.0467	0.0419	0.0482	0.0486
LightGCN+	0.2004	0.1616	0.2347	0.2358	0.0211	0.0150	0.0268	0.0290	<u>0.0527</u>	0.0451	0.0467	0.0469
HCCF+	0.2034	0.1641	0.2379	0.2483	0.0217	<u>0.0156</u>	0.0310	0.0315	0.0516	0.0436	0.0486	0.0488
RLMRec-Con	0.2062	<u>0.1676</u>	<u>0.2441</u>	<u>0.2542</u>	<u>0.0227</u>	0.0142	<u>0.0348</u>	<u>0.0358</u>	0.0499	<u>0.0460</u>	<u>0.0488</u>	<u>0.0492</u>
RLMRec-Gen	0.2062	0.1653	0.2407	0.2424	0.0223	0.0145	0.0336	0.0339	0.0517	0.0451	0.0475	0.0477
GLTA	<b>0.2465</b>	<b>0.1882</b>	<b>0.2722</b>	<b>0.2905</b>	<b>0.0359</b>	<b>0.0236</b>	<b>0.0499</b>	<b>0.0500</b>	<b>0.0968</b>	<b>0.0821</b>	<b>0.1162</b>	<b>0.1403</b>
Improv.	19.54%	12.29%	11.51%	14.28%	58.14%	51.28%	43.39%	39.66%	83.68%	78.48%	138.11%	185.16%

**Figure 3: Performance of GLTA compared to its variants without item-text alignment and without user-item alignment.**

in GNN-based baselines [4]. Second, compared to graph-based recommendation with only interaction information, introducing text information into graph-based methods improves the recommendation performance in most cases, which verifies the importance of incorporating rich semantic content to enhance user-item matching and better capture the contextual nuances of items. Third, the LLM-based recommendation method, RLMRec-Con, has better overall performance than other baselines. This justifies that aligning the knowledge of LLMs with collaborative relation learning through contrastive learning is able to enhance recommendation performance. However, the recommendation process is still done by the backbone LightGCN model in RLMRec, leaving the reasoning abilities of LLMs untouched. On the contrary, GLTA projects users and items into language space as tokens first, and then completely leverages the LLM for end-to-end recommendation, which is a more holistic approach that fully utilizes the reasoning capabilities and contextual understanding of LLMs.

### 4.3 Item-text and User-item Alignment

In GLTA, item-text alignment and user-item alignment are designed to align CF with the reasoning ability of the LLM. To verify the effectiveness of these two alignment methods, we compare the performance of GLTA to its variant without item-text alignment (w/o IA) and without user-item alignment (w/o UA) on the three datasets in Figure 3. For the variant *w/o IA*, we directly remove the item-text alignment stage and update item and user node projectors simultaneously in the user-item alignment stage. For the variant *w/o UA*, we use user IDs instead of aligned user tokens in the user-item alignment instruction template before feeding it into the GLLM layer. We find that employing item-text alignment or user-item alignment leads to stable enhancement in all datasets, which implies that both alignment methods are crucial for effectively leveraging the LLM’s reasoning capabilities in the recommendation process.

**Table 3: Performance of GLTA compared to its three variants without profile (PF) tokens and/or without prediction (PD) tokens on three datasets.**

Dataset	Goodreads		Amazon		MovieLens	
Metric	P@5	N@5	P@5	N@5	P@5	N@5
w/o both	0.2462	0.2705	0.0316	0.0408	0.0736	0.0658
w/o PF	0.2466	0.2668	0.0318	0.0414	0.0763	0.0713
w/o PD	<b>0.2467</b>	0.2688	0.0341	0.0449	0.0860	0.0761
GLTA	0.2465	<b>0.2722</b>	<b>0.0359</b>	<b>0.0499</b>	<b>0.0968</b>	<b>0.1162</b>

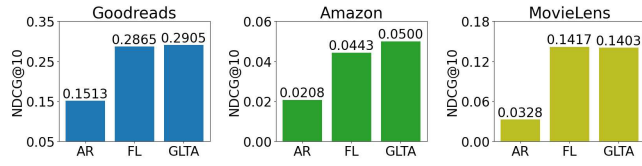
### 4.4 Profile and Prediction Tokens

To quantify the contribution of profile and prediction tokens used in user-item alignment, we conduct an ablation study to investigate the performance of GLTA without these tokens generated by the LLM. The results are shown in Table 3. Both profile and prediction tokens generally improve performance if included in the instruction template. Notably, the advantages of using these LLM-generated tokens become more pronounced when the distinctions between items in the dataset are clearer. For instance, the performance improvement is more evident in MovieLens, where items span a wide range of movie genres, but less pronounced in Goodreads, where items consist only of history books. We speculate that the broader diversity of items in datasets like MovieLens allows the LLM-generated tokens to better capture user features from nuanced differences between items, thereby enhancing recommendation accuracy. In contrast, as the distinctions between items are less varied and therefore less reliant on the LLM’s enhanced representation capabilities, the additional benefits provided by these tokens are also limited.

### 4.5 Item Prediction in GLLM

We further explore two other item prediction patterns in GLLM layers. Autoregressive inference (AR): During inference, we strictly follow the text generation of LLMs in which each item token is generated based on the previous item tokens the model has generated. First-logit optimization (FL): To predict the top  $k$  favorite items for each user, we use the largest  $k$  elements in the first logit from  $Z_i^u \in \mathbb{R}^{L \times |I|}$ , instead of the first  $k$  logits in GLTA. The results are demonstrated in Figure 4. The poor performance of autoregressive inference indicates that, unlike natural language processing tasks





**Figure 4: Performance of GLTA compared to its variants with different optimization methods on three datasets.**

where contextual continuity is vital, the sequential dependency between items is not as strong in recommendation. The inconsistency between training and inference patterns exacerbates this issue, particularly because the model is trained using item descriptions and collaborative signals rather than direct sequential user-item interaction data.

## 5 Conclusion

In this paper, we propose a novel recommendation framework GLTA, which applies token alignment to integrate graph-based CF with the reasoning capabilities of LLMs. In GLTA, we begin by employing a graph encoder to capture user and item node features from the collaborative relationships within the user-item graph. Next, item and user node embeddings are adaptively aligned with other language tokens using node projectors. Finally, the user and item node projectors are optimized for end-to-end recommendation through the GLLM layer. Compared with finetuning LLMs as RS, GLTA overcomes hallucination and is efficient since only projectors and GLLM layers are finetuned. Future works may explore integrating additional modalities with graph-based CF through multimodal large language models.

## 6 Acknowledgments

This work is supported in part by NSF under grants III-2106758, and POSE-2346158. Hao Peng is supported by the National Key R&D Program of China through grant 2022YFB3104703, the NSFC through grants 62322202 and 62441612, Local Science and Technology Development Fund of Hebei Province Guided by the Central Government of China through grant 246Z0102G, the "Pioneer" and "Leading Goose" R&D Program of Zhejiang" through grant 2025C02044, Hebei Natural Science Foundation through grant F2024210008, and the Guangdong Basic and Applied Basic Research Foundation through grant 2023B1515120020.

## References

- [1] Yuwei Cao, Nikhil Mehta, Xinyang Yi, Raghunandan Hulikal Keshavan, Lukasz Heldt, Lichan Hong, Ed Chi, and Maheswaran Sathiamoorthy. 2024. Aligning Large Language Models with Recommendation Knowledge. In *Findings of the Association for Computational Linguistics: NAACL 2024*. Association for Computational Linguistics, Mexico City, Mexico, 1051–1066.
- [2] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *RecSys '22: Sixteenth ACM Conference on Recommender Systems*. ACM, 299–315.
- [3] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. ACM, 639–648.
- [4] Nicolas Keriven. 2022. Not too little, not too much: a theoretical analysis of graph (over)smoothing. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems*.
- [5] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.
- [6] Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 188–197.
- [7] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation Learning with Large Language Models for Recommendation. In *Proceedings of the ACM on Web Conference*. ACM, 3464–3475.
- [8] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2024. GraphGPT: Graph Instruction Tuning for Large Language Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 491–500.
- [9] Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian J. McAuley. 2019. Fine-Grained Spoiler Detection from Large-Scale Review Corpora. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*. Association for Computational Linguistics, 2605–2610.
- [10] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*.
- [11] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. LLMRec: Large Language Models with Graph Augmentation for Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. ACM, 806–815.
- [12] Lianghao Xia, Chao Huang, Yong Xu, Jiahu Zhao, Dawei Yin, and Jimmy X. Huang. 2022. Hypergraph Contrastive Collaborative Filtering. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 70–79.
- [13] Mingdai Yang, Zhiwei Liu, Liangwei Yang, Xiaolong Liu, Chen Wang, Hao Peng, and Philip S. Yu. 2023. Group Identification via Transitional Hypergraph Convolution with Cross-view Self-supervised Learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. ACM, 2969–2979.
- [14] Mingdai Yang, Zhiwei Liu, Liangwei Yang, Xiaolong Liu, Chen Wang, Hao Peng, and Philip S. Yu. 2024. Instruction-based Hypergraph Pretraining. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 501–511.
- [15] Mingdai Yang, Zhiwei Liu, Liangwei Yang, Xiaolong Liu, Chen Wang, Hao Peng, and Philip S. Yu. 2024. Unified Pretraining for Recommendation via Task Hypergraphs. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. ACM, 891–900.