Denoising Alignment with Large Language Model for Recommendation

YINGTAO PENG, School of Information, Renmin University of China, China
CHEN GAO, Beijing National Research Center for Information Science and Technology, Tsinghua University, China
YU ZHANG, Department of Electronic Engineering, Tsinghua University, China
TANGPENG DAN, School of Information, Renmin University of China, China
XIAOYI DU and HENGLIANG LUO, China Meituan, China
YONG LI, Department of Electronic Engineering, Tsinghua University, China
XIAOFENG MENG*, School of Information, Renmin University of China, China

The mainstream approach of GNN-based recommendation aggregates high-order ID information associated with the node in the user-item graph. The aggregation pattern using ID as signal has two disadvantages: lack of textual semantics and the impact of interaction noise. These disadvantages pose a threat to effectively learn user preferences, especially in capturing intricate user-item semantic relationships. Although large language models (LLMs) allow the integration of rich textual information into recommenders and have had groundbreaking applications in recommender systems, current works need to bridge the gap between different representation spaces. This is because LLMs-based methods align the representations of GNN-based models only by using text embedding of LLM, leading to unsatisfactory results. To address this challenge, we propose a **D**enoising **A**lignment framework with **L**LMs for GNN-based **R**ecommenders (DALR), which aims to align structural representation with textual representation and mitigate the effects of noise. Specifically, We propose a modeling framework that integrates the representation of graph structure with textual information from LLMs to capture intricate user-item interactions. We also suggest an alignment paradigm to enhance representation performance by aligning semantic signals from LLMs and structural features from GNN models. Additionally, we introduce a contrastive learning scheme to relieve the impact of noise and improve model performance. Extensive experiments on public datasets demonstrate that our model consistently outperforms the state-of-the-art methods. DALR achieves improvements ranging from 2.82% to 12.20% in Recall@5 and from 1.04% to 3.48% in NDCG@5 compared to the strongest baseline model, using the Steam dataset as an example.

CCS Concepts: • **Information systems** → **Recommender systems**.

Additional Key Words and Phrases: Recommender system, graph neural network, large language models, contrastive learning

[*] Xiaofeng Meng is corresponding author

Authors' addresses: Yingtao Peng, School of Information, Renmin University of China, China, yingtaopeng@ruc.edu.cn; Chen Gao, Beijing National Research Center for Information Science and Technology, Tsinghua University, China, chgao96@gmail.com; Yu Zhang, Department of Electronic Engineering, Tsinghua University, China, zhangyuthu14@gmail.com; Tangpeng Dan, School of Information, Renmin University of China, China, tangpengdan@ruc.edu.cn; Xiaoyi Du, duxiaoyi@meituan.com; Hengliang Luo, luohengliang@meituan.com, China Meituan, China; Yong Li, Department of Electronic Engineering, Tsinghua University, China, liyong07@tsinghua.edu.cn; Xiaofeng Meng, School of Information, Renmin University of China, China, xfmeng@ruc.edu.cn.

ACM 1558-2868/2024/9-ART
https://doi.org/10.1145/3696662

1 INTRODUCTION

In practical situations, recommendation systems (RS) are designed to assist people in finding high-value information across various large-scale data platforms, such as e-commerce, multimedia, and online news [24]. A recommendation system is crucial for users to discover items of interest from severe information overload, bringing significant convenience and attracting widespread attention from academic and industrial communities. The graph data form provides a powerful tool for modeling user preferences [19], which expresses the behavioral relationship between users and items appropriately, and offers significant advantages in understanding intricate relationships. With the advancement of deep learning technologies, neural networks have demonstrated superior capability in extracting deeper data features. Hence, mainstream recommendation technologies have evolved from traditional methods to those based on neural networks [28].

Many studies based on graph neural networks (GNNs) [44] heavily rely on ID information throughout the learning process. Although this reliance can improve the effectiveness, it poses a notable constraint for representation learning due to neglecting the rich textual semantics (e.g., attributes and generated profiles by a large language model (LLM) in Fig. 1 (a)) associated with users and items. Additionally, noise in interaction data, such as misclicks [42] or information bias [5] (as shown in Fig. 1 (b)), can lead to an inaccurate representation of the model. Models rely on sparse interaction data to deduce user preferences and item features and generate vector representations during the training phase. In sparse datasets, the impact of noise is readily captured and magnified by conventional neural models [67, 68]. It is mainly because conventional GNN-based models [20] typically treat all nodes and their neighbors equally, disregarding the potential presence of noise or outliers among nodes. When confronted with these challenges, it is observed that GNN-based models tend to exhibit suboptimal performance. To address the above limitations, some approaches integrate self-supervised data augmentation techniques with graph-based recommenders. These methods [29, 52, 57] enhance the semantic representation of data by leveraging intrinsic information within the graph structure. Specifically, techniques such as subgraph reconstruction enable the model to learn the relationships between nodes and the global structure of the graph, thereby extracting more semantically meaningful feature representations. Moreover, the model learns robust representations to filter and mitigate noise in the graph data by introducing self-supervised tasks, reducing its impact on model performance. Recent models (such as SGL [29], HCCF [57], and DCCF [36]) have demonstrated promising performance by employing contrastive augmentation techniques. For example, HCCF [57] leverages a hypergraph-enhanced cross-view contrastive learning architecture to capture local and global semantic relationships effectively. DCCF [36] utilizes self-supervised augmentation to achieve adaptive intent disentanglement, refining latent factors and reducing noise from augmentation. Despite their success in enhancing recommendation accuracy, these models need to address the challenge of lacking textual semantics. This is because the paradigm of these methods still relies on ID information and lacks external semantics despite generating self-supervised signals through a data augmentation approach.

To better represent the semantic features of nodes in the graph, LLMs can be regarded as an excellent additional semantic extractor. Recently, outstanding LLMs such as GPT-4 [1] have demonstrated remarkable prowess in semantic understanding in natural language processing (NLP), fueling a rising enthusiasm to harness their potential for enhancing recommendation systems [18, 22]. Incorporating LLMs into recommendation systems is advantageous as they can extract high-quality textual information and leverage essential external knowledge encoded from LLMs[54]. The prevailing approach of utilizing LLMs in GNN-based recommendation is to design meticulous prompts, which generate compelling profiles for users/items [49]. By leveraging these textual profiles, recommendation models accurately capture user preference representations to enhance the model's performance [35]. For example, RLMRec [35] meticulously integrates representation learning with LLM to capture intricate semantic facets of user preferences. GraphTranslator [65] connects pre-trained graph models (GMs) and large language models (LLMs) through instruction fine-tuning to address pre-defined tasks within graphs.

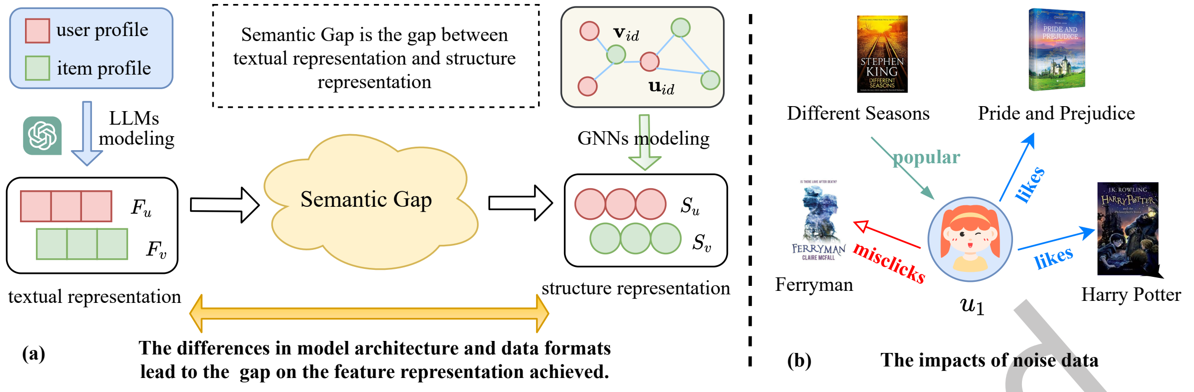


Fig. 1. Figure (a) illustrates an instance of the semantic disparity between GNN-based representations and semantic representations. Figure (b) demonstrates the influence of noise in interactive data.

However, despite the effective recommendation performance demonstrated by the LLMs-based methods above, they still have the following shortcomings: **(1) Representation alignment issue.** We define the semantic gap as the alignment issue between different representation spaces. Specifically, the semantic gap is the gap between textual representation and structure representation in the recommender scenario (as shown in Fig. 1 (a)). existing GNN-based methods [4, 45] mainly adhere to the aggregation paradigm, which captures rich structural properties but lacks textual semantics. In contrast, methods based on LLMs can comprehend the textual semantics of user preferences, yet they lose the rich structural properties of interaction relationships. Therefore, it is necessary to align the semantic space with the representation space of GNN-based models and integrate cross-view features. However, existing approaches (like RLMRec [35]) need more capability to effectively bridge the gap between structural and textual space, as direct contrastive representations from different spaces limit their effectiveness. Additionally, fine-tuning large language models based on instruction prompts incurs significant costs. **(2) Impact of noise issues.** In real-world recommendation scenarios, the observed user-item interactions denoted as Y often exhibit noise, encompassing both false positives (e.g., interactions of u_1 influenced by popular book *Different Seasons* in Fig. 1 (b)) and false negatives (e.g., instances where users do not engage with potentially interesting items). Consequently, the performance of the recommender is susceptible to the influence of such noise data, leading to a detrimental impact on recommendation accuracy. However, the ability of LLMs-based methods to mitigate noise needs to be improved, as it solely relies on contrastive learning from the textual semantic perspective, neglecting important structural property features.

To alleviate the limitations above, we propose a **D**enoising **A**lignment framework with **L**arge language model for GNN-based **R**ecommendation (DALR), aiming to leverage the power of LLMs to align representation and mitigate noise impacts. The core idea of DALR is to align structure representation from GNN-based models and the intricate semantic features from LLM. Our framework contains the two components: **(1) Hybrid feature alignment.** We propose a hybrid alignment scheme to bridge the gap between distinct representational spaces. Firstly, hybrid features alleviate the independence between semantic and structural features in the representation space by integrating the textual semantics generated by LLM with the structural information derived from the user-item graph. Secondly, by aligning the hybrid features with the representation of the backbone model, DALR smoothly integrates information from different spaces, enhancing the representation performance of the model with the generation of hybrid semantic features. **(2) Sementic attribute alignment.** To better mitigate the impact of noise, we devised a denoising contrastive strategy by aligning semantic and structural attributes. The semantic attribute contrastive pattern enhances the model's representation capacity by aligning the semantic representations of LLM and GNN-based features with a cross-view contrastive module. This approach integrates the textual semantics of LLM into the encoded representations of users and items from the GNN-based model, thereby augmenting the recommender's representational capacity. Besides, we design a profile representation module to provide profile information and semantic representations for users and items separately. The profile module relieves the noise of the profile through controllable instruction prompts and enhances the hybrid features by fusing ID and text semantics. Technically, we design a prompting paradigm to generate textual profiles for users and items using ChatGPT. Then, we transform textual profiles into meaningful embedding and utilize them as crucial signals for downstream alignment tasks. Ultimately, we refine both the alignment framework and the model's parameters via joint training to enhance synergy and improve recommendation accuracy.

To validate the effectiveness of our approach, we conducted a series of relevant experiments. The results indicate effective improvements in the Recall and NDCG metrics. For instance, on the SGL backbone network, our method achieved a 12.20% increase in the Recall@5 metric on the steam dataset. Furthermore, the intermediate results of aligning representations, as shown in section 5.4, demonstrate notable enhancements in addressing noise and semantic issues with our method. We summarize the contributions of this work as follows:

- We propose a general alignment framework DALR, to explore the integration of LLMs to GNN-based recommenders by fusing their semantic and structural representations.
- We develop a hybrid attribute alignment module aim at bridging the gap between representations in different spaces, a challenge previously addressed mainly through semantic alignment with limited effectiveness.
- We devise a denoising alignment strategy, which operates from both semantic and structural perspectives. It utilizes mutual information maximization as the optimization objective to enhance representation quality.
- We compare DALR with state-of-the-art LLM-based and Graph-based recommendation models and validate the effectiveness of our approach.

2 RELATED WORK

In this work, we propose a denoising alignment framework with large language models for GNN-based recommendation, which is closely related to GNN-based recommendation, contrastive learning for recommendation, and large language models for recommendation.

2.1 GNN-based recommendation

Numerous endeavors have been devoted to constructing recommenders using various graph neural network techniques [12]. These GNN-based recommenders model high-order collaborative relationships by executing message passing on graph structures [20, 38, 44]. Firstly, graph convolutional networks (GCNs) have emerged as prevalent encoders for modeling user-item interaction graphs, exemplified by LightGCN [20], LR-GCCF [6], and HGCF [39]. LightGCN streamlines collaborative filtering by leveraging the user-item interaction graph without additional user/item side information. Secondly, graph-enhanced attention mechanisms play a crucial role in delineating the impact of embedding propagation among neighboring nodes and serve as pivotal components in numerous recommenders, such as DGCF [45] and ASR [32]. DGCF [26] employs graph-enhanced attention mechanisms to dissect user-item relationships at a granular level of user intents, thereby generating disentangled representations. Furthermore, further innovations extend to hypergraph learning [27, 60], and intention disentanglement [36, 51], aimed at unraveling intricate collaboration patterns. DCCF [36] excels in unraveling complex collaboration patterns and mitigating augmentation-induced noise through techniques like hypergraph learning and intention disentanglement. Recent advancements in denoising implicit feedback for GNN-based recommenders focus on mitigating the negative impacts of noisy data ([13, 41]). BOD ([47]) introduces a bi-level optimization approach that dynamically adjusts weights and eliminates the need for prior knowledge, enhancing efficiency and performance. SocialRec ([34]) addresses noise in the social recommendation by refining social graphs through preference-guided denoising and adaptive strategies. Benefiting from the advantages of GNN, the GNN-based paradigm has been applied in various recommendation scenarios. These encompass social relationship learning, exemplified by DGRec [37] and DANSER [55], as well as multi-behavioral recommendation [60] and knowledge graph-based recommenders like KGAT [43]. KGAT [43] utilizes knowledge graphs to improve recommendation performance by integrating user-item relationships with entities and relations from the KG.

2.2 Contrastive learning for recommendation

Recently, the impressive achievements of self-supervised learning in NLP [11] and CV [7, 8] have sparked significant interest in the field of recommender systems. Self-supervised learning is a machine learning paradigm that learns semantic representations from unlabeled data by generating self-supervised signals from the data itself rather than relying on external labels [52, 58]. Recommender systems enhance the representation performance of models through contrastive learning on different structural views [48, 52, 67]. Current research in this area is divided into two categories: same-scale contrast and cross-scale contrast [25, 63]. In same-scale contrast, the views being compared originate from two entities of equivalent scale [40, 52]. Approaches within this category utilize data augmentation techniques to create different views on graph structures. Specifically, ContraRec [40] introduces a contrastive learning task, called context-context contrast, to encourage sequences after augmentation and sequences with the same target item to have similar representations. SGL [52] employs random dropout operations to augment the interaction graph structure, enhancing recommendation performance. NCL [29] conducts representation alignment between individual users and semantically centered nodes to improve recommendation effectiveness. Conversely, cross-scale contrast involves views from entities at different scales [67, 69]. For example, MCCLK [69] employs contrastive learning across three views at both local and global levels, mining comprehensive graph feature and structure information in a self-supervised manner. Additionally, contrastive learning strategies have enhanced some sequential models to improve recommendation performance, such as CL4SRec [59]. CL4SRec [59] is a multi-task model for a sequential recommendation that integrates traditional next-item prediction with contrastive learning, enabling more meaningful user feature extraction and effective semantic representation.

2.3 Large language models for recommendation

With the significant advancements of large language models [3] in the field of natural language processing, researchers have repeatedly attempted to leverage these models for modeling user preference [15, 53]. Existing studies [30, 50] predominantly fall into two categories: (1)As predictor, which employ LLMs to generate recommendations for users directly; (2) As extractor, which provides profiles for downstream recommendation tasks utilizing rich external knowledge and reasoning capabilities.

In the first approach, LLMs generate recommendations directly by employing techniques such as context learning [30], prompt tuning [2], and instruction tuning [64]. However, such fine-tuning methods typically expend high computational costs. TALLRec [2] is a framework crafted to narrow the divide between large language models and recommendation tasks. It effectively fine-tunes LLMs with recommendation data, mitigating the mismatch between LLM training tasks and recommendation tasks and overcoming the scarcity of recommendation data during pre-training. Similarly, GraphTranslator [65] connects pre-trained graph models (GMs) and large language models through instruction fine-tuning to implement downstream tasks within graphs. In the second approach, the goal is to parse relevant semantic information and generate user/item profiles [31], and other textual data [14] for recommendation tasks, enriching the accuracy and personalization of the recommendations. P5 [14] is a text-to-text paradigm for recommendation that unifies diverse data into natural language sequences, enabling instruction-based recommendation and reducing the need for extensive fine-tuning with adaptive personalized prompts. In GNN-based recommendation, LLMRec [50] employs Large Language Models (LLMs) to enhance

Table 1. Frequently used notations in this article.

Notation	Description
$\mathcal{U}, \mathcal{V}, u, v$	The user set, the item set, user u, and item v
$\mathcal{G}, Y$	The user-item graph and the interaction data
$\mathcal{N}(i), h_i^{(l)}$	Set of neighbors of node i, feature vector of node i at layer l
$a_i^{(l+1)}$	Aggregated feature vector for node i at layer $l+1$
$\hat{y}_{uv}$	Predicted interaction score between user u and item v
e_{id}, e_t, e_a, e_d	e_{id}, e_t, e_a, e_d denote unique number, title, attributes, descriptions
$\mathcal{S}_u^I, \mathcal{S}_v^I$	The system prompt instructions of user and item
$\mathcal{P}_u, \mathcal{P}_u$	The input prompt of user and items
$\mathcal{A}_u, \mathcal{A}_v$	The generation profiles of user and item
$\mathbf{r}_{\mathcal{A}_u}, \mathbf{r}_{\mathcal{A}_v}$	The semantic representation of user's profile and item's profile
$\mathcal{R}_G(\cdot,\cdot), \mathbf{u}_i, \mathbf{v}_j$	a GNN-based model, the representation of user and item
$\mathcal{L}_{Hyb}, \mathcal{L}_{Sem}$	the loss of hybrid feature alignment and the loss of semantic contrastive modeling
s_u, s_v	The user's structure attributes and The item's structure attributes
$\mathbf{h}_{s_u}, \mathbf{h}_{s_v}$	LLM-augmented user/item hybrid features of user and item
$\mathcal{L}, \mathcal{L}_{BPR}$	The total loss and the BPR loss

user-item interaction information and multimodal data, mitigating data sparsity issues. Meanwhile, RLMRec [35] enhances text semantics by aligning the textual representations of large models with those of GNNs to improve representation performance.

However, current LLM-based methods fail to effectively bridge the gap across different spaces by directly aligning text. Besides, there is room for improvement in mitigating the impact of noise, as we explained in the introduction. We propose a denoising alignment framework incorporating hybrid features and semantic contrast to achieve more effective recommendation results.

3 PRELIMINARY

3.1 Problem Statement

First, we present the concept of the recommendation task based on GNN models. Next, we describe the interaction data and introduce the User-Item Graph. Besides, we provide Table 1 to explain the meaning of frequently used notations in this article.

Task Description. In this work, we define our model's task as follows: Utilizing interaction data $\mathbf{Y}$, the User-Item Graph $\mathcal{G}$, and Large Language Models (LLMs), we aim to learn the probability $\hat{y}_{uv} = \sigma(\mathbf{u} \cdot \mathbf{v})$. This probability $\hat{y}_{uv}$ predicts whether user u will engage with a candidate item v. Here, σ represents the activation function, $\mathbf{u}$ is the user's vector representation, and $\mathbf{v}$ is the vector of the candidate item.

Interaction Data. In our study, we gather user-item interactions (such as browsing, clicking, purchasing, and commenting behaviors) as feedback data. We define $\mathcal{U} = \{u_1, u_2, \cdots, u_n\}$ as the set of users and $\mathcal{V} = \{v_1, v_2, \cdots, v_m\}$ as the set of items. Consequently, we establish $\mathrm{Y} = \{y_{uv} | u \in \mathcal{U}, v \in \mathcal{V}\}$ to represent the user-item interaction matrix, where y_{uv} signifies that user u has clicked on or purchased item v.

User-Item Graphs. In our work, we construct an User-Item Graph as follows: Drawing on methods from existing literature [46], we generate an User-Item Graph $\mathcal{G}$ based on historical interaction data. Specifically, $\mathcal{G}$ is constructed as $\{(u, y_{uv}, v \ | u \in \mathcal{U}, v \in \mathcal{V})\}$, representing the interactions between users and items.

3.2 Theory of GNN-based Recommendation

The collaborative filtering method in recommender systems leverages embedding vectors $(\mathbf{u}_i, \mathbf{v}_j)$ for users u_i and items v_j, and calculates the predicted rating $r_{i,j}$ for a target user-item pair through a dot product operation. The embedding vectors are learned by minimizing the discrepancy between observed ratings and model-predicted ratings.

GNN-based recommendation. A GNN-Based model treats users and items as nodes in a graph $\mathcal{G}$. Edges between nodes represent interactions, such as ratings or purchases. The model learns to embed nodes (users and items) into a low-dimensional space where the embedding vectors capture the complex relationships and characteristics of users and items. These embeddings can then be used to predict future interactions or preferences.

$$a_i^{(l+1)} = Aggregate^{(l)}\left(\left\{h_j^{(l)} : j \in \mathcal{N}(i)\right\}\right), \tag{1}$$

Were, $a_i^{(l+1)}$ is the aggregated feature of node j at layer $l+1$, $h_j^{(l)}$ represents the features of neighbor nodes j at layer l, and $\mathcal{N}(i)$ denotes the neighbors of node i. The aggregate function can vary, common choices include mean, sum, or max pooling.

$$h_i^{(l+1)} = \text{Update}^{(l)}\left(h_i^{(l)}, a_i^{(l+1)}\right) \tag{2}$$

In this step, $h_i^{(l+1)}$ is the updated feature of node i at layer $l+1$, which is computed by combining its previous feature $h_i^{(l)}$ with the aggregated feature $a_i^{(l+1)}$. The update function often involves non-linear transformations, such as applying a neural network.

After several iterations of feature aggregation and update, the final embeddings (features) of users and items are used to predict the likelihood of interaction between a user and an item, typically through a dot product or a neural network:

$$\hat{y}_{ui} = f\left(h_u^{(L)}, h_i^{(L)}\right), \tag{3}$$

where, $\hat{y}_{ui}$ is the predicted score of interaction between user u and item i, $h_u^{(L)}$ and $h_i^{(L)}$ are the final embeddings of u and item i at layer L of GNN. f is a function to compute the prediction score.

3.3 Feature Augmentation with Contrastive Learning

To mitigate the impact of irrelevant data on representation, incorporating text semantics is an effective approach [35]. These textual information (e.g., user and item profiles) provide crucial support for learning interaction interests. the semantic vectors of these profiles can effectively capture users' semantic interests by using pre-trained language model encoders. Notably, both the semantic encoder and the neural graph model capture shared information from interaction behaviors (i.e., both contain useful information). Therefore, our goal is to learn the optimal value of this shared information by maximizing the conditional probability.

Mutual Information Maximization (MIM) [21] is a critical technique in the DALR model. Mutual Information (MI) quantifies the degree of information shared between two random variables, indicating how much observing one variable reduces the uncertainty of the other. In our model, MIM is employed to align semantic features $\mathbf{r_s}$ from large language models (LLMs) with structural features $\mathbf{r_g}$ from graph data, thereby enhancing the performance of the recommendation system. Formally,

$$I\left(\mathbf{r}_g, \mathbf{r}_s\right) = \mathbb{E}_{(\mathbf{r}_g, \mathbf{r}_s)}\left[\log \frac{p\left(\mathbf{r}_g, \mathbf{r}_s\right)}{p\left(\mathbf{r}_g\right) p\left(\mathbf{r}_s\right)}\right], \tag{4}$$

here, $p\left(\mathbf{r}_g, \mathbf{r}_s\right)$ is the joint probability distribution of $\mathbf{r}_g$ and $\mathbf{r}_s$. $p\left(\mathbf{r}_g\right)$ and $p\left(\mathbf{r}_g\right)$ are the marginal probability distributions of $\mathbf{r}_g$ and $\mathbf{r}_s$, respectively. To optimize the above objective function, we can approximate the maximization of mutual information through InfoNCE (Contrastive Estimation of Neural Entropy) [17, 67]. The InfoNCE loss is formulated as follows:

$$\mathcal{L}_{\text{InfoNCE}} = \mathbb{E}_{p(\mathbf{r}_g,\mathbf{r}_s)}\left[f_{sim}\left(\mathbf{r}_g,\mathbf{r}_s\right) - \log\sum_{s'\in S}\exp f_{sim}\left(\mathbf{r}_g,\mathbf{r}_{s'}\right)\right], \tag{5}$$

where, We introduce a discriminator function $f_{sim}(\cdot,\cdot)$ to measure the similarity between different views. The goal is to ensure that the similarity of positive sample pairs is higher than that of negative sample pairs. During training, the model minimizes the InfoNCE loss using gradient descent methods, updating the parameters of both the GNNs and the LLMs.

4 METHODOLOGY

In this section, we initially present our DALR framework. We then delve into a detailed examination of each module within this framework.

4.1 An Overview of DALR

We propose a general framework named DALR, which aims to leverage the capabilities of LLMs to align the representations of GNN-based models and mitigate the noise impacts. The structure is shown in Fig. 2. DALR predicts the probability of a user interacting with a candidate item by utilizing interaction data and LLMs' textual information as input. We improve recommender's performance and addresses noise issues through two key components: hybrid feature alignment and semantic attribute alignment. **(i) Hybrid feature alignment.** The objective of hybrid feature alignment is to co-align hybrid features with the feature representations from the backbone network, thereby bridging the gap between different spaces and enhancing the model's representation. Hybrid features are defined as a fusion of semantic information from LLMs and structural information from graph data. Additionally, our sampling strategy can effectively control and capture the similarity of neighbourhood graph nodes based on transition factors, thereby reducing interference from irrelevant nodes. **(ii) Semantic attribute alignment.** We enhance model representation by maximizing the mutual information paradigm. By contrasting the similarities and differences between samples from different views, this method helps our model to alleviate noise issues. We treat the representations from the backbone network and the semantic information from the LLM as two distinct perspectives. Besides, we design a profile representation module to provide profile information and semantic representations for users and items separately using ChatGPT. We aim to provide valuable profiles and semantic representation for user preference learning by a carefully designed prompt template. Finally, our model yields high-quality representations of items and users with joint training.

4.2 Profile Representation Module

This section discusses the significance and design patterns of user and item profiles. To effectively understand the semantic preferences underlying user-item interactions, valuable profiles play a crucial role in the model. Technically, we generate user and item profiles from interaction data and user/item information using ChatGPT API from OpenAI[1]. We have established precise prompt instructions to control the quality of profiles and avoid the influence of text noise. User profiles should effectively summarize the types and features of items representing the user's historical preferences, reflecting the user's personalized interests. Item profiles should clearly describe the characteristics, features, and audience types, facilitating better alignment with user preferences. Furthermore, we incorporate unique user identifiers into the input prompts to mitigate the gap between diverse spatial representations.

[1] https://platform.openai.com/

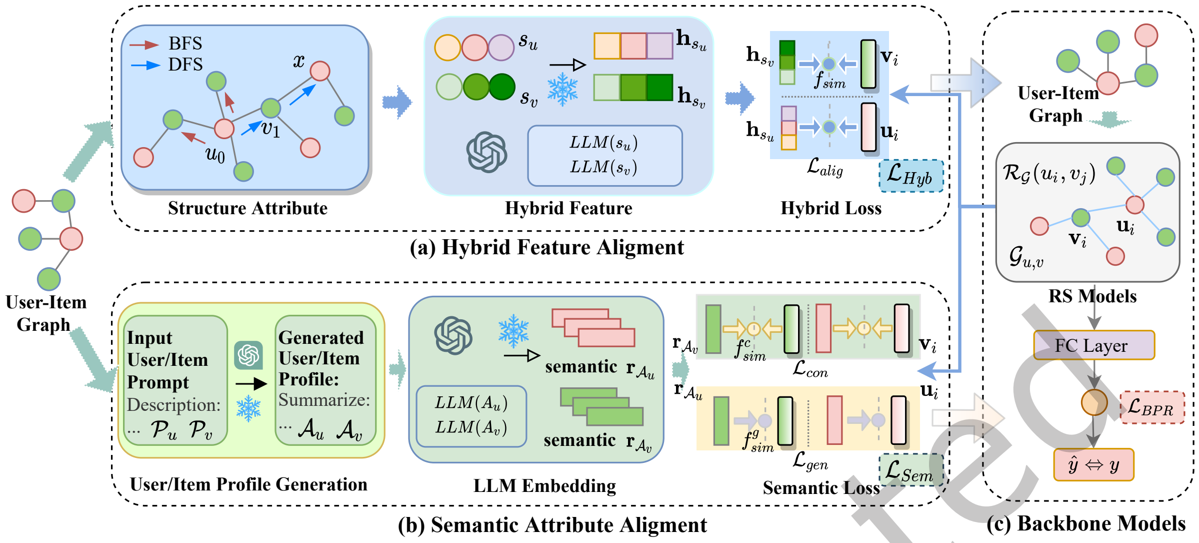


Fig. 2. An intuitive illustration of our DALR framework. DALR is a universal recommendation framework comprising three modules: (a) Hybrid Feature Alignment, (b) Semantic Attribute Alignment, and (c) Backbone Models. In the Hybrid Feature Alignment stage, cross-representation space contrastive learning is utilized to align knowledge from two modalities. During the Semantic Attribute Alignment stage, mutual information learning is employed to contrast self-supervised signals from different views. In the third stage, a joint collaborative model fine-tunes the backbone model and evaluates its performance on the test dataset.

4.2.1 Item Prompt Construction. Fig. 8 illustrates the design of input prompts and system instructions for generating item profiles. Inspired by prior research [9, 14], we recognize that effective prompts must encompass key semantic details like the item's unique number, title, category, and description. Accordingly, we crafted the following input prompt template for items:

$$\mathcal{P}_v = f_v\left([e_{id}, e_t, e_a, e_d]\right), \tag{6}$$

where, the function $f_v(\cdot)$ merges different elements into a unified string. e_{id}, e_t, e_a, e_d represent unique number, title, attributes, and descriptions, respectively. If the element e is absent, we assign it as NULL.

4.2.2 User Prompt Construction. Similar to constructing prompts for items, we leverage historical interaction data and item details to create user profiles, illustrated in Fig. 9. We select a random subset of items the user has interacted with and consolidate them into a coherent string. To capture user preferences accurately, comments made by the user on interactive items are incorporated as a crucial component. The user's input prompt $\mathcal{P}_u$ is established as follows:

$$\mathcal{P}_u = f_u\left(\{\mathbf{r}_v \mid v \in \mathcal{V}_u\}\right), \tag{7}$$

where, the function $f_u(\cdot)$ merges various elements into a single string, similarly to how $f_v(\cdot)$ operates. $\mathcal{V}_u$ is is the collection of items that the user interacted. We form an interaction record as $\mathbf{r}_v = \left[\mathcal{P}_{u,v}, c_u^v\right]$, where c_u^v denotes the comments given by user u. This format of user prompts captures authentic user interests and feedback in text format, which are subsequently translated into semantic features for utilization in downstream tasks.

4.2.3 Profile Generation and Semantic Representation. To alleviate the hallucination issues [61] in inference generation of large language models and enhance the quality of generated data, we designed the system prompt instructions $\mathcal{S}^I_{v/u}$ as a crucial input provided to the LLMs. Our objective is to accurately set the boundaries for generating user and item profiles by detailing the input-output content and format. Following existing handling approaches [35], our profile generation approach is outlined as follows:

$$\mathcal{A}_v = LLM_{ChatGPT}\left(\mathcal{S}_v^I, \mathcal{P}_v\right), \quad \mathcal{A}_u = LLM_{ChatGPT}\left(\mathcal{S}_u^I, \mathcal{P}_u\right), \tag{8}$$

where, $\mathcal{P}_u$ and $\mathcal{P}_v$ denote the input prompts of users and items. $\mathcal{S}_u^I$ and $\mathcal{S}_v^I$ are the system prompts for users and items, respectively. $\mathcal{A}_u$ and $\mathcal{A}_v$ denote the generated profiles for users and items. $LLM_{ChatGPT}$ is a large language model developed by OpenAI.

The enhanced information on users and items is transformed into feature representations and used as input in the recommendation system. The LLM plays a crucial role as an encoder, analyzing what users are interested in, reducing noise in item features, and tackling issues of sparse interactions with its accurate and advanced understanding of semantics. Formally, the process for generating LLM-based semantic representations is outlined as follows:

$$\mathbf{r}_{\mathcal{A}_v} = LLM_{emb}\left(\mathcal{A}_v\right), \quad \mathbf{r}_{\mathcal{A}_u} = LLM_{emb}\left(\mathcal{A}_u\right), \tag{9}$$

where, $\mathbf{r}_{\mathcal{A}_v}, \mathbf{r}_{\mathcal{A}_u}, \in \mathcal{R}^{d_{LLM}}$ are LLM-augmented representation of users and items. LLM_{emb} denotes text-embedding-ada-002, which is a new and improved text embedding model from OpenAI.

4.3 Hybrid Feature Alignment

As previously discussed, existing efforts enhance recommendation systems by leveraging the external knowledge and semantics of large language models. However, current LLMs-based methods struggle to effectively bridge the gap between textual semantic expression and the structured spatial representation offered by graph neural networks. This challenge stems from the direct alignment of textual semantics with aggregated ID representations being unsmooth. Such alignment fails to maximize mutual information across different spaces. To address this problem, we propose a hybrid feature alignment module designed to improve the representational capabilities of recommendation systems. Specifically, We define hybrid features as the fusion of semantic information from LLMs and structural information from graph data. These hybrid features are used for collaborative alignment with the feature representations from the backbone network to alleviate the representation gap and enhance performance effectively.

4.3.1 Hybrid Feature Representation. The concept of homogeneity in a network means that embeddings of closely linked nodes are alike. Meanwhile, homology in a network indicates that nodes with similar structures yield similar embeddings. Both homogeneity and homology play vital roles in the representation of graph-based recommendation systems. Drawing inspiration from node2vec [16], our approach integrates both the homogeneity and structural attributes into the node representations. The above function is implemented by utilizing the transition probabilities between nodes in the graph.

Sampling strategy. The sampling strategy aims to obtain structure sequences centred on the current node n_i from the interaction graph to generate hybrid features. This strategy is a core component of the hybrid feature alignment module, designed to bridge the gap between different representation spaces and enhance the model's representation capability. In the sampling module, we control the transition probability of sampling through transition factors q and p. By controlling transition factors of the sampling strategy, we can effectively manage the homogeneity and isomorphism of the sampled nodes centred on node n_i.

Following the design of node2vec [16], we employ a neighbor sampling strategy. This approach allows the model to seamlessly transition between breadth-first search (BFS) and depth-first search (DFS) sampling methods. During the sampling process, two primary parameters are involved: the return parameter p and the in-out parameter q. Given a current node v_1 and a previously visited node u_0, the transition probability $\pi_{v_1,x}$ to a next node x is calculated as follows: $\pi_{v_1,x} = \alpha_{p,q}\left(u_0, x\right) \cdot w_{v_1,x}$. Here, $w_{v_1,x}$ is the edge weight between vertices v_1 and

x, it is set to 1 by default. $\alpha_{p,q}\left(u_0,x\right)$ is a correction factor, defined as follows:

$$\alpha_{p,q}\left(u_0,x\right)=\begin{cases}\frac{1}{p}, & \text{if} \quad d_{u_0,x}=0\\ 1, & \text{if} \quad d_{u_0,x}=1\\ \frac{1}{q}, & \text{if} \quad d_{u_0,x}=2\end{cases}. \tag{10}$$

where, $d_{u_0,x}=0$ indicates that the distance from node u_0 to node x is 0 (also referring to u_0 itself), then the transition probability is $\frac{1}{p}$. If there is a path with a distance of 1 from node u_0 to x, then the probability of walking to x in the next step is 1. If x is a direct neighbor of u_0, but u_0 and the current node v_1 are not direct neighbors, then the transition probability is $\frac{1}{q}$.

By adjusting p and q, we can flexibly balance between preserving local neighborhood information (homophily) and exploring distant nodes (structural exploration). For each node e ($e \in \{u_i, v_i\}$), the sequence s_e obtained by sampling strategy can be expressed as $s_e=(e_1,\cdots,e_l)$, Here, e_i is the node from $\mathcal{G}_{u,v}$ and l is the length of the sequence.

Hybrid Feature. Through the sampling strategy, we obtain a sequence of structural attributes centered around the current node. To represent the enhanced structural sequence, we utilize LLM_{emb} as an encoder to provide efficient and state-of-the-art representation understanding for aligning structural attributes. The LLM_{emb} encoder refers to ”text-embedding-ada-002,” an advanced language model embedding technique developed by OpenAI. This model can transform text into high-dimensional vector representations, thereby capturing the semantic information of the text.

Given the structured sequence of the current node $s_{id}=\{e_1,\cdots,e_l\}$, which is obtained through a sampling strategy. We aim to generate hybrid features by integrating the LLM-generated profiles with the structured sequences. This integration produces a hybrid feature sequence containing both structural and textual features: $s_h=f\left(\left[e_i,\mathcal{A}_{e_i}|e_i\in s_{id}\right]\right)$. Subsequently, this hybrid feature sequence is encoded by the LLM_{emb} encoder, transforming the hybrid sequence into a vector form $\mathbf{h}_s$. Formally, the LLM-based hybrid feature augmentation is as:

$$\begin{cases} user: & \mathbf{h}_{s_u}=LLM_{emb}\left(s_u\right)\\ item: & \mathbf{h}_{s_v}=LLM_{emb}\left(s_v\right)\end{cases}, \tag{11}$$

where, $\mathbf{h}_{s_u},\mathbf{h}_{s_v},\in\mathcal{R}^{d_{LLM}}$ are LLM-augmented user/item hybrid features. $LLM_{emb}\left(\cdot\right)$ is an advanced language model encoder (text-embedding-ada-002) by OpenAI. Here, we obtain the hybrid features by fusing structural sequences and textual semantics from LLMs.

4.3.2 *alignment modeling module.*

To align backbone model representations with hybrid structural features, we consider the user/item representation by the GNNs module and the LLM module, respectively, as two different views. Our model boosts the learning of representations by optimizing for maximum mutual information between positive and negative examples. In detail, each user and item is assigned initial embeddings u and v in GNN-based recommenders. The goal is to learn user and item representations $\mathbf{u}$, $\mathbf{v}$ through a GNN-based model, formally:

$$\mathbf{u}_i,\mathbf{v}_j=\mathcal{R}_{\mathcal{G}}\left(u_i,v_j\right), \tag{12}$$

where, $\mathcal{R}_{\mathcal{G}}$ is a GNN-based backbone model. $\mathbf{u}_i$ and $\mathbf{v}_j$ are the feature representation by the backbone model. u_i and v_j mean the initialize feature of the user and item. The user's hybrid feature $\mathbf{h}_{s_u}$ and the item's hybrid feature $\mathbf{h}_{s_v}$ are obtained from the section 4.3.1. We calculate the alignment loss function for two different spatial features. From a technological standpoint, we adopt a modeling strategy centered on similarity contrast, denoted as $f_{sim}\left(\cdot,\cdot\right)$, inspired by existing techniques [35]. The function $f_{sim}\left(\cdot,\cdot\right)$ serves as a measure of similarity, assessing the resemblance between $\mathbf{u}_i/\mathbf{v}_j$ and $\mathbf{h}_{s_u}/\mathbf{h}_{s_v}$. The optimization formula for the representation of users and items is specified as follows:

$$\mathcal{L}^{u}_{alig} = f_{sim}\left(\mathbf{h}_{s_u}, \mathbf{u}\right) - \log \sum_{u' \in \mathcal{U}} \exp\left(f_{sim}\left(\mathbf{h}_{s_{u'}}, \mathbf{u}\right)\right), \tag{13}$$

$$\mathcal{L}^{v}_{alig} = f_{sim}\left(\mathbf{h}_{s_v}, \mathbf{v}\right) - \log \sum_{v' \in \mathcal{V}} \exp\left(f_{sim}\left(\mathbf{h}_{s_{v'}}, \mathbf{v}\right)\right), \tag{14}$$

where u' and v' are the negative sample. $f_{sim}(\cdot, \cdot)$ is implemented with cosine similarity function. Finally, we denote the alignment loss of the hybrid feature module as $\mathcal{L}_{Hyb} = \mathcal{L}^{u}_{alig} + \mathcal{L}^{v}_{alig}$.

4.4 Semantic Attribute Alignment

Existing GNN-based recommendation methods predict user preferences based on the similarity of user and item representations. With their exceptional ability to harness external knowledge and semantics, large language models can significantly enhance recommendation systems' semantic representations [66]. Inspired by RLMRec, we have explored a semantic alignment attribute module to mitigate the effects of noise better. Denoising contrast improves model performance by the paradigm of mutual information maximization, which regards the representations (aggregated information) of backbone networks and the LLM's semantics (interaction and profiles) as two different views. It aims to bolster the model's representative capacity by aligning semantic representations of features based on LLM and GNN with a cross-view contrastive module. To ensure alignment consistency, we also utilize two modeling methodologies: comparative and generative. This strategy improves the alignment between representations from GNN-based models and those from LLM, reducing the effects of noise during the learning of representations.

4.4.1 Contrastive modeling module. Contrastive modeling has been widely validated on recommender systems to align different views efficiently [23]. As shown in Fig. 2 (b), we consider the representation of users/items from the GNNs and LLM modules to be two different views. Effective contrastive learning can be achieved by maximizing the similarity between similar samples and minimizing the similarity between dissimilar samples. The optimization function of the user's representation is as follows:

$$\mathcal{L}^{u}_{con} = -\mathbb{E}\log\left[\frac{f^{c}_{sim}\left(\mathbf{r}_{\mathcal{A}_u}, \mathbf{u}\right)}{\sum_{u' \in \mathcal{U}} f^{c}_{sim}\left(\mathbf{r}_{\mathcal{A}_{u'}}, \mathbf{u}\right)}\right], \tag{15}$$

where, $f^{c}_{sim}\left(\mathbf{r}_{\mathcal{A}_u}, \mathbf{u}\right) = exp(sim(f^{-}_{mlp}(\mathbf{r}_{\mathcal{A}_u}, \mathbf{u})))$ is the cosine similarity function. The goal of the function f^{-}_{mlp} is to map the semantic representation $\mathbf{r}_{\mathcal{A}_u}$ into the GNNs representation $\mathbf{u}$.

Similar to the handling of users, the optimization function of item's representation can be denoted:

$$\mathcal{L}^{v}_{con} = -\mathbb{E}\log\left[\frac{f^{c}_{sim}\left(\mathbf{r}_{\mathcal{A}_v}, \mathbf{v}\right)}{\sum_{v' \in \mathcal{V}} f^{c}_{sim}\left(\mathbf{r}_{\mathcal{A}_{v'}}, \mathbf{v}\right)}\right], \tag{16}$$

where $f^{c}_{sim}\left(\mathbf{r}_{\mathcal{A}_v}, \mathbf{v}\right) = exp(sim(f^{-}_{mlp}(\mathbf{r}_{\mathcal{A}_v}), \mathbf{v}))$. In the end, we denote the loss of contrastive modeling module as $\mathcal{L}_{con}$, $\mathcal{L}_{con} = \mathcal{L}^{u}_{con} + \mathcal{L}^{v}_{con}$.

4.4.2 Generative modeling module. Inspired by generative self-supervised learning [56], we utilize a masked autoencoder (MAE) as a generative modeling component following RMLRec [35]. The generative process aims to reconstruct the semantic representations for the masked samples by a single-direction reconstruction approach. Technically, we explore the reconstruction capability within the semantic feature space by masking and reconstructing random subsets of users/items. For the user's representation, the calculation function of cosine similarity between the GNNs representation and the LLMs representation can be denoted as follows:

$$f^{g}_{sim}\left(\mathbf{r}_{\mathcal{A}_u}, \mathbf{u}\right) = exp(sim(\mathbf{r}_{\mathcal{A}_u}, f^{+}_{mlp}(\tilde{\mathbf{u}}))), \tag{17}$$

where, $\tilde{\mathbf{u}} = \mathcal{R}\left(\{\mathbf{u} \setminus \mathbf{u}_i\}\right)$ denotes the initial embedding of the i-th sample with a masking technique. The function f^{+}_{mlp} maps the GNNs representation to the semantic representation via a multi-layer perception layer. The generative process adopts a single-direction reconstruction method, with a focus on reconstructing the semantic representations solely for the masked samples. This permits us to explore the generative power in semantic space.

After obtaining generative self-supervised signals from users and items, we optimize the generative alignment using a contrastive learning approach. Formally:

$$\mathcal{L}_{gen} = \mathcal{L}^{u}_{gen} + \mathcal{L}^{v}_{gen} = -\mathbb{E}\log\left[\frac{f^{g}_{sim}\left(\mathbf{r}_{\mathcal{A}_u}, \mathbf{u}\right)}{\sum_{u' \in \mathcal{U}} f^{g}_{sim}\left(\mathbf{r}_{\mathcal{A}_{u'}}, \mathbf{u}\right)}\right] - \mathbb{E}\log\left[\frac{f^{g}_{sim}\left(\mathbf{r}_{\mathcal{A}_v}, \mathbf{v}\right)}{\sum_{v' \in \mathcal{V}} f^{g}_{sim}\left(\mathbf{r}_{\mathcal{A}_{v'}}, \mathbf{v}\right)}\right] \tag{18}$$

where, $u' \neq u, v' \neq v$, $\mathcal{L}^{u}_{gen}$ and $\mathcal{L}^{v}_{gen}$ denote the user's optimization loss and item's optimization, respectively. The loss $\mathcal{L}_{gen}$ of the generative modeling module is obtained by summing the user generative loss $\mathcal{L}^{u}_{gen}$ and the item generative loss $\mathcal{L}^{v}_{gen}$.

4.5 Prediction and Model Optimization

Prediction Module. In the prediction layer, we calculate the relevance probability by a user's feature $\mathbf{u}$ and vectors $\mathbf{v}$ of candidate items as follows:

$$\hat{y}_{uv} = \sigma\left(\mathbf{u} \cdot \mathbf{v}\right), \tag{19}$$

where σ is a sigmoid activation function, $\mathbf{u}$ and $\mathbf{v}$ indicate the embedding representation of user and items, respectively.

Model Optimization. We minimize the following objective function to learn the model's parameter by combining the independence loss and BPR loss [44]:

$$\mathcal{L} = \mathcal{L}_{BPR} + \lambda_1 \mathcal{L}_{Hyb} + \lambda_2 \mathcal{L}_{Sem} + \lambda_3 \left\|\Theta\right\|_2^2, \tag{20}$$

where Θ is the set of parameters; λ_1, λ_2 and λ_3 are the hyper-parameters to control the independence loss and L_2 regularization term, respectively. $\mathcal{L}_{Sem}$ contains two different strategies: contrastive loss $\mathcal{L}_{con}$ and generative loss $\mathcal{L}_{gen}$. In particular, we assume that observed interactions indicate more user preferences and should be given higher predictive values than unobserved interactions:

$$\mathcal{L}_{BPR} = \sum_{(u,v^{+},v^{-}) \in O} - \ln \sigma\left(\hat{y}_{u,v^{+}} - \hat{y}_{u,v^{-}}\right), \tag{21}$$

where $O = \{(u, v_{+}, v_{-}) \mid (u, v_{+}) \in \mathcal{R}^{+}, (u, v_{-}) \in \mathcal{R}^{-}\}$ is the pairwise training data, $\mathcal{R}^{+}$ means the positive samples, $\mathcal{R}^{-}$ denotes the negative samples, v_{+} is user clicked item, v_{-} is no click item. $\hat{y}_{u,v^{+}}$ represents the calculated score.

To optimize the loss $\mathcal{L}$, we employ the mini-batch Adam algorithm, which adaptively adjusts both the gradient and the learning rate. For training data, we select samples from both observed and unobserved interactions, representing positive and negative instances, respectively. In practice, we randomly select a batch consisting of user-item pairs (u, v_{+}, v_{-}) to capture the representation of users and items. The training procedure of our model is illustrated in Algorithm 1.

The learning process of our model is detailed in Algorithm 1. Lines 1-2 initiate the process by setting up model parameters and the adjacency matrix based on the user-item graph. The algorithm samples interactions and generates collaborative representations from lines 3-6. Lines 7-8 focus on deriving semantic representations from LLMs and calculating the corresponding loss. In lines 9-11, the model infers hybrid features and computes the overall loss by integrating semantic and hybrid loss components with specific weights. Finally, line 12 updates the model's parameters, concluding with the optimized parameters being returned in line 13. Finally, DALR enhances the performance of the recommender through a mix of collaborative signal and alignment representation. We provide the source code repository of our DALR framework at the following link: https://github.com/pengyingtao/DALR.

Algorithm 1 Our model Learning algorithm.

```
Require:
    Interaction matrix $\mathbf{Y}$, user-item graph $\mathcal{G}$, LLMs, the backbone model $\mathcal{R}_G$, loss weights and hyper-parameters $\lambda_1$, $\lambda_2$, $\lambda_3$, learning rate $\eta$.
Ensure:
    Model parameters $\mathbf{\Theta}$
Randomly initialize neural parameters $\mathbf{\Theta}$
Constructe adjacency matrix of entities $\mathbf{A}_e$ from $\mathcal{G}$
while An epoch is not end do
    Sample minibatch of positive and negative interactions from Interaction matrix $\mathbf{Y}$;
    Inference collaborative-side representation $\mathbf{e}_{u/v}$ with $\mathcal{R}_G$;
    Compute model optimization objective $\mathcal{L}_{BPR}(\mathbf{\Theta})$ $(Eq.(21))$
    Inference semantic representation $\mathbf{r}_{\mathcal{A}_u}$ and $\mathbf{r}_{\mathcal{A}_v}$ with LLMs
    Compute the loss $\mathcal{L}_{Sem}(\mathbf{\Theta})$ $(Sec.4.3)$
    Inference hybrid feature $\mathbf{h}_{s_u}$ and $\mathbf{h}_{s_v}$
    Compute the loss $\mathcal{L}_{Hyb}(\mathbf{\Theta})$ $(Sec.4.4.2)$
    $\mathcal{L}(\mathbf{\Theta}) = \mathcal{L}_{BPR} + \lambda_1\mathcal{L}_{Hyb}(\mathbf{\Theta}) + \lambda_2\mathcal{L}_{Sem}(\mathbf{\Theta})$
    Update neural parameters
end while
return $\mathbf{\Theta}$
```

5 EXPERIMENTS

In this section, we assess the performance of our framework. Initially, we outline the datasets used for evaluation. Subsequently, we detail the baseline models and the settings of our experiments. Lastly, we present the results of these experiments and analyze the impact of various hyper-parameters.

5.1 Experimental Setings

We aim to answer the following three research questions (RQ) in experiments.

- RQ1: How does our proposed DALR framework perform compared with state-of-the-art GNN-based recommendation methods? Specifically, has it effectively bridged the representation gap that was a notable issue in prior methodologies?
- RQ2: Whether the modules of our model can work well, including the hybrid feature alignment and the semantic attribute alignment?
- RQ3: Can DALR perform well in the presence of noise and data sparsity issues?
- RQ4. How do different hyper-parameter settings (e.g., λ_i, q) affect the recommendation performance?

5.1.1 Datasets Description. The Amazon-Book[2] dataset is a collection of data on book products from the Amazon website, which includes user reviews, ratings, and other relevant information about the books. Specifically, the user information includes unique user IDs and profile information. The book information includes unique identifiers, titles, authors, and so on. Other relevant information also includes user purchase records, click behaviors, preference tags, and more. Similarly, the Steam dataset[3], sourced from the Steam platform, encompasses user reviews and game attribute data, featuring a vast user base, numerous game titles, and millions of reviews. In addition to textual reviews, it provides temporal data on user gameplay activities. The Yelp2018 dataset, sourced

[2] http://jmcauley.ucsd.edu/data/amazon/

[3] https://www.kaggle.com/tamber/steam-video-games/data

Table 2. Statistics of the experimental datasets.

Dataset	#Users	#Items	#Interactions	#Density
Amazon-Book	11,000	9,331	200,860	1.96E-03
Steam	23,310	5,236	525,923	4.31E-03
Yelp	11,091	11,010	166,620	1.41E-03
Movie-1m	6,040	3,952	1,000,209	4.19E-02

from the Yelp website[4], includes user and business information, ratings, reviews, and millions of interaction records, making it ideal for evaluating large-scale recommender systems. The MovieLens-1M (Movie1M) dataset[5], sourced from GroupLens Research, comprises 1 million ratings on almost 3,900 movies by over 6,000 users.

For preprocessing, we followed the established protocols outlined in previous studies [35, 56]. Specifically, we filtered out interactions with ratings below 3 in the Amazon-Book and Yelp dataset, while no filtering was applied to the Steam and Movie1m dataset. Each dataset was then partitioned into training, validation, and test sets using a 6:2:2 ratio. For detailed dataset information, please refer to Table 2.

5.1.2 Baselines. To evaluate the performance improvement of our framework, we conduct comparisons with the state-of-the-art GNN-based recommender enhanced with LLMs. This approach allows for an independent assessment of our framework's effectiveness compared to baseline models. These baselines fall into three categories: aggregation-based approaches (such as LightGCN [20]), denoising models (like DenoisingRec [41], SGDL [13] and BOD [47]), self-supervised models (including SGL [52], SimGC [62]L, DCCF [36], and AutoCF [56]), and LLMs-enhanced approaches (like LLMRec [50] and RLMRec [35]).

- LightGCN [20], a simplified graph convolutional neural network, models user-item interactions in recommenders.
- DenoisingRec [41] improves recommender systems by adaptively identifying and pruning noisy feedback during training, using Truncated Loss and Reweighted Loss strategies to enhance recommendation quality.
- SGDL [13] leverages early memorized interactions to guide training and automatically adapts learning phases, improving robustness across various recommendation models and loss functions.
- BOD [47] models denoising recommendation as a bi-level optimization problem, addressing implicit feedback noise by adapting weights dynamically based on previous iterations and avoiding prior knowledge.
- SGL [52] is a recommendation system approach based on graph self-supervised learning, which leverages the structural information of user-item interaction graphs to learn embedding representations without requiring additional annotations, thereby improving recommendation performance.
- SimGCL [62] aims to enhance recommendation performance by conducting contrastive learning on simplified graph representations of user-item interaction graphs to learn more discriminative user and item representations.
- DCCF [36] is a collaborative filtering method that disentangles user-item interactions into distinct aspects and applies contrastive learning to these aspects separately. DCCF improves recommendation quality by effectively capturing diverse user preferences and item characteristics.
- AutoCF [56] provides an automated graph-enhancement scheme by combining subgraph semantic relevance based on information maximization with self-supervised learning signals. This approach automatically extracts rich self-supervised information in an unsupervised manner.
- LLMRec [50] improves data reliability and recommendation performance through three LLM-based graph augmentation strategies and a denoising data robustification mechanism.

[4] https://www.yelp.com

[5] https://grouplens.org/datasets/movielens/1m/

- RLMRec [35] proposes a model-agnostic framework to enhance existing recommendation systems through representation learning empowered by LLMs. It combines representation learning with large language models to capture complex semantic aspects of user behavior and preferences.

5.1.3 Evaluation Metrics. When evaluating GNN-based recommender systems, it is essential to assess model performance from two perspectives: (1) the average accuracy of the recommended items and (2) the accuracy of item ranking within the recommended set. To take both aspects into account, we use commonly adopted metrics [20, 33]: Recall@K and Normalized Discounted Cumulative Gain (NDCG@K). We set the default values of K to 5, 10, and 20 and report the average metrics for all users in the test dataset.

Recall: It is the proportion of correctly predicted positive samples by the model out of the total true positive samples.

$$\text{Recall} = \frac{\sum\limits_{u \in U} |R(u) \cap T(u)|}{\sum\limits_{u \in U} |T(u)|}, \tag{22}$$

here, $R(u)$ represents the recommended list given based on user behavior in the training set, and $T(u)$ represents the recommended list given based on user behavior in the test set.

Discounted Cumulative Gain (DCG): This metric indicates that placing the items liked by the user at the top of the recommendation list can significantly enhance the user experience compared to placing them further down.

$$DCG(b, L) = \sum_{i=1}^{b} r_i + \sum_{i=b+1}^{L} \frac{r_i}{\log_b i}, \tag{23}$$

where r_i denotes whether the item ranked in position i is liked by the user. If $r_i = 1$, it indicates that the user likes the item; if $r_i = 0$, it means the user does not like the item. b is a free parameter, and L represents the length of the recommendation list. Then, since DCG values are not directly comparable between users, they need to be normalized. Specifically, all items in the test set are first sorted in the ideal order, and the top K items are selected to calculate their DCG. Then, the original DCG is divided by the DCG under the ideal order, resulting in the Normalized Discounted Cumulative Gain (NDCG):

$$NDCG@K = \frac{DCG}{iDCG} \tag{24}$$

5.1.4 Parameter Settings. The experiments were performed using the PyTorch framework on an Nvidia GPU equipped with 48 GB of memory. User and item profiles were generated using the ChatGPT model (gpt-3.5-turbo), while semantic representations were obtained using the text embedding model (text-embedding-ada-002) from OpenAI. Baseline models were evaluated using their respective released implementation source codes to ensure fairness in comparison. The dimension of representations was standardized to 32 across all base models. The training process employed the Adam optimizer with a fixed batch size of 4096 and a learning rate of 0.001. Additional insights on parameter analysis can be found in Section 5.5.

5.2 Performance Comparison (RQ1)

To validate the effectiveness of our approach in improving recommendation performance, we compared it with five state-of-the-art graph-based recommenders. We conducted three random initializations in the experiments and averaged the results, as shown in Table 3, 4, 5, and 6. We report the performance comparison in the result tables, where the performance of the strongest baseline is presented in an underlined format, and the best performance is in bold font. From these experimental results, we can draw the following conclusions:

(1) Firstly, we observed that our denoising alignment framework for recommendation outperforms LLMRec and RLMRec in enhancing the performance of the backbone recommender. The results in Table 1 provide compelling

Table 3. Overall performance comparisons with baseline models on Amazon-Book.

		Recall@5	Recall@10	Recall@20	NDCG@5	NDCG@10	NDCG@20
DenoisingRec	Denoising	0.0330	0.0543	0.0838	0.0348	0.0419	0.0517
NGCF		0.0486	0.0841	0.1312	0.0512	0.0613	0.0789
LightGCN	Base	0.0579	0.0916	0.1434	0.0584	0.0700	0.0869
	LLMRec	0.0608	0.0972	0.1480	0.0625	0.0743	0.0911
	RLMRec-Con	0.0613	0.0976	0.1490	0.0616	0.0742	0.0910
	RLMRec-Gen	0.0615	0.0969	0.1492	0.0628	0.0747	0.0916
	Our model	**0.0642**	**0.1023**	**0.1527**	**0.0634**	**0.0764**	**0.0927**
	Imp.%	4.41%	4.87%	2.36%	0.89%	2.32%	1.24%
SGDL	Denoising	0.0664	0.1067	0.1500	0.0677	0.0797	0.0954
SGL	Base	0.0644	0.1023	0.1480	0.0643	0.0774	0.0925
	RLMRec-Con	0.0660	0.1026	0.1527	0.0661	0.0786	0.0950
	RLMRec-Gen	0.0644	0.1010	0.1529	0.0653	0.0779	0.0948
	Our model	**0.0676**	**0.1039**	**0.1560**	**0.0664**	**0.0793**	**0.0960**
	Imp.%	2.45%	1.21%	2.02%	0.49%	0.97%	1.00%
BOD	Denoising	0.0713	0.1126	0.1659	0.0733	0.0875	0.1048
SimGCL	Base	0.0610	0.0991	0.1513	0.0612	0.0747	0.0915
	RLMRec-Con	0.0651	0.1022	0.1559	0.0653	0.0781	0.0955
	RLMRec-Gen	0.0619	0.0984	0.1518	0.0631	0.0755	0.0928
	Our model	**0.0664**	**0.1058**	**0.1616**	**0.0663**	**0.0799**	**0.0981**
	Imp.%	1.92%	3.50%	3.64%	1.47%	2.27%	2.68%
DCCF	Base	0.0670	0.1034	0.1517	0.0662	0.0791	0.0945
	RLMRec-Con	0.0665	0.1039	0.1555	0.0661	0.0791	0.0961
	RLMRec-Gen	0.0680	0.1027	0.1542	0.0672	0.0789	0.0959
	Our model	**0.0690**	**0.1065**	**0.1583**	**0.0678**	**0.0802**	**0.0972**
	Imp.%	1.47%	2.52%	1.78%	0.85%	1.31%	1.11%
AutoCF	Base	0.0688	0.1058	0.1559	0.0714	0.0837	0.0998
	RLMRec-Con	0.0696	0.1064	0.1578	0.0703	0.0827	0.0995
	RLMRec-Gen	0.0691	0.1088	0.1608	0.0711	0.0845	0.1014
	Our model	**0.0731**	**0.1118**	**0.1644**	**0.0735**	**0.0862**	**0.1028**
	Imp.%	4.96%	2.71%	2.28%	3.39%	1.98%	1.43%

evidence of the effectiveness of our model. Specifically, taking the SimGCL backbone model as an example, on the Movie1m dataset, our model improves Recall@5, Recall@10, Recall@20, NDCG@5, NDCG@10, and NDCG@20 by 3.55%, 2.47%, 5.17%, 3.54%, 3.36%, and 2.91%, respectively, compared to the strongest baseline RLMRec. On the Steam dataset, our model achieves improvements of 11.09%, 7.67%, and 5.64% in Recall@5, Recall@10, and Recall@20, and 1.04%, 0.97%, and 1.14% in NDCG@5, NDCG@10, and NDCG@20, respectively, compared to the strongest baseline RLMRec. These enhancements are attributed to three main factors: (1) Hybrid feature alignment further mitigates the gap between LLM semantic space and GNN-based structural space. (2) By contrasting semantic information from different views, DALR effectively alleviates the impact of irrelevant noise on the recommendation features. (3) Our carefully designed prompt representation module via LLM effectively controls the quality of text semantic generation, providing semantic support for downstream tasks.

(2) The hybrid feature alignment supported by LLM enhances the recommender's coverage of user interests. The results show that the improvement in recall metrics surpasses that of NDCG metrics in most datasets. Specifically,

Table 4. Overall performance comparisons with baseline models on Steam.

		Recall@5	Recall@10	Recall@20	NDCG@5	NDCG@10	NDCG@20
DenoisingRec	Denoising	0.0415	0.0671	0.1083	0.0476	0.0554	0.0689
NGCF		0.0588	0.0819	0.1292	0.0544	0.0631	0.0788
LightGCN	Base	0.0519	0.0863	0.1351	0.0571	0.0688	0.0852
	LLMRec	0.0551	0.0908	0.1435	0.0601	0.0725	0.0905
	RLMRec-Con	0.0553	0.0900	0.1417	0.0604	0.0721	0.0897
	RLMRec-Gen	0.0553	0.0910	0.1445	0.0604	0.0726	0.0907
	Our model	**0.0574**	**0.0931**	**0.1465**	**0.0625**	**0.0745**	**0.0928**
	Imp.%	3.82%	2.26%	1.41%	3.48%	2.64%	2.33%
SGDL	Denoising	0.0559	0.0925	0.1431	0.0618	0.0740	0.0911
SGL	Base	0.0559	0.0911	0.1425	0.0614	0.0734	0.0909
	RLMRec-Con	0.0592	0.0945	0.1481	0.0647	0.0766	0.0947
	RLMRec-Gen	0.0575	0.0930	0.1463	0.0625	0.0745	0.0927
	Our model	**0.0664**	**0.1016**	**0.1530**	**0.0668**	**0.0789**	**0.0961**
	Imp.%	12.20%	7.48%	3.37%	3.21%	2.99%	1.46%
BOD	Denoising	0.0568	0.0929	0.1442	0.0635	0.0759	0.0937
SimGCL	Base	0.0567	0.0930	0.1450	0.0621	0.0742	0.0920
	RLMRec-Con	0.0581	0.0948	0.1475	0.0639	0.0761	0.0941
	RLMRec-Gen	0.0578	0.0934	0.1462	0.0632	0.0753	0.0932
	Our model	**0.0645**	**0.1021**	**0.1558**	**0.0645**	**0.0768**	**0.0951**
	Imp.%	11.09%	7.67%	5.64%	1.04%	0.97%	1.14%
DCCF	Base	0.0562	0.0921	0.1453	0.0622	0.0743	0.0923
	RLMRec-Con	0.0568	0.0934	0.1466	0.0627	0.0749	0.0929
	RLMRec-Gen	0.0566	0.0920	0.1437	0.0626	0.0745	0.0921
	Our model	**0.0584**	**0.0946**	**0.1478**	**0.0640**	**0.0759**	**0.0940**
	Imp.%	2.82%	1.30%	0.85%	2.16%	1.34%	1.16%
AutoCF	Base	0.0519	0.0853	0.1357	0.0574	0.0687	0.0858
	RLMRec-Con	0.0524	0.0873	0.1383	0.0577	0.0695	0.0868
	RLMRec-Gen	0.0542	0.0894	0.1408	0.0592	0.0711	0.0886
	Our model	**0.0562**	**0.0923**	**0.1448**	**0.0608**	**0.0728**	**0.0906**
	Imp.%	3.73%	3.32%	2.86%	2.75%	2.45%	2.29%

considering the Amazon-book dataset, on the LightGCN backbone model, our model outperforms the strongest baseline model by 4.41%, 4.87%, and 2.36% in Recall@5, Recall@10, and Recall@20, respectively, while achieving improvements of 0.89%, 2.32%, and 1.24% in NDCG@5, NDCG@10, and NDCG@20. For Yelp, compared to the strongest baseline model on the LightGCN backbone model, our model achieves enhancements of 5.57%, 4.19%, and 2.33% in Recall@5, Recall@10, and Recall@20, respectively, while achieving improvements of 5.55%, 2.19%, and 1.92% in NDCG@5, NDCG@10, and NDCG@20. These results indicate that our framework better captures user interests than the baseline models. This improvement is attributed to the meticulously designed prompts via LLM and the alleviation of the gap between different spaces through hybrid feature alignment.

(3) Both the hybrid alignment and semantic alignment methods generally lead to performance improvements. From the table, it can be observed that the RLMRec and LLMRec methods outperform all baseline methods. Specifically, on the Steam dataset, the RLMRec method achieves Recall@5, Recall@10, and Recall@20 scores of 2.51%, 1.93%, and 1.74% higher than the SimGCL method, respectively. Similarly, the RLMRec method achieves

Table 5. Overall performance comparisons with baseline models on Movie1M.

		Recall@5	Recall@10	Recall@20	NDCG@5	NDCG@10	NDCG@20
DenoisingRec	Denoising	0.0589	0.1038	0.1784	0.2668	0.2589	0.2633
NGCF		0.0579	0.1092	0.1799	0.2651	0.2524	0.2547
LightGCN	Base	0.0575	0.1071	0.1793	0.2646	0.2516	0.2538
	LLMRec	0.0663	0.1132	0.1849	0.2732	0.2617	0.2616
	RLMRec-Con	0.0675	0.1139	0.1861	0.2751	0.2623	0.2618
	RLMRec-Gen	0.0647	0.1093	0.1794	0.2702	0.2563	0.2554
	Our model	**0.0716**	**0.1196**	**0.1920**	**0.2893**	**0.2741**	**0.2696**
	Imp.%	6.12%	4.98%	3.17%	5.13%	4.50%	2.99%
SGDL	Denoising	0.0685	0.1256	0.1930	0.2791	0.2689	0.2713
SGL	Base	0.0672	0.1148	0.1877	0.2689	0.2589	0.2601
	RLMRec-Con	0.0739	0.1256	0.2020	0.2847	0.2746	0.2763
	RLMRec-Gen	0.0723	0.1238	0.1998	0.2848	0.2719	0.2737
	Our model	**0.0777**	**0.1294**	**0.2073**	**0.2900**	**0.2791**	**0.2828**
	Imp.%	5.10%	2.96%	2.66%	1.85%	1.64%	2.38%
BOD	Denoising	0.0775	0.1309	0.2135	0.2893	0.2801	0.2825
SimGCL	Base	0.0680	0.1161	0.1903	0.2691	0.2574	0.2592
	RLMRec-Con	0.0758	0.1285	0.2077	0.2798	0.2696	0.2728
	RLMRec-Gen	0.0758	0.1287	0.2079	0.2818	0.2692	0.2715
	Our model	**0.0785**	**0.1316**	**0.2184**	**0.2897**	**0.2787**	**0.2807**
	Imp.%	3.55%	2.47%	5.17%	3.54%	3.36%	2.91%
DCCF	Base	0.0757	0.1287	0.2058	0.2868	0.2781	0.2789
	RLMRec-Con	0.0767	0.1296	0.2092	0.2917	0.2801	0.2822
	RLMRec-Gen	0.0746	0.1262	0.2054	0.2884	0.2762	0.2788
	Our model	**0.0795**	**0.1342**	**0.2185**	**0.2980**	**0.2892**	**0.2870**
	Imp.%	3.53%	3.55%	4.46%	2.17%	3.27%	1.69%
AutoCF	Base	0.0270	0.0501	0.0853	0.1908	0.1835	0.1680
	RLMRec-Con	0.0288	0.0506	0.0856	0.1953	0.1828	0.1729
	RLMRec-Gen	0.0299	0.0537	0.0927	0.2092	0.1925	0.1798
	Our model	**0.0314**	**0.0556**	**0.0951**	**0.2114**	**0.1985**	**0.1869**
	Imp.%	4.95%	3.43%	2.61%	1.06%	3.13%	3.96%

NDCG@5, NDCG@10, and NDCG@20 scores of 2.90%, 2.51%, and 2.24% higher than the SimGCL method, respectively. However, it is worth noting that the DALR method exhibits superior performance when combined with the backbones of SGL and SimGCL. Specifically, on the Steam dataset, the DALR method achieves Recall@5, Recall@10, and Recall@20 scores of 13.88%, 9.75%, and 7.47% higher than the SimGCL method. Likewise, the DALR method achieves NDCG@5, NDCG@10, and NDCG@20 scores of 3.98%, 3.51%, and 3.40% higher than the SimGCL method. Thus, this further validates the efficacy of our framework over the RLMRec method.

(4) We achieved a significant improvement compared to the denoising baseline. From the results presented in the tables, the following conclusions can be drawn: a) Our model consistently outperforms both the baseline model (Base) and the denoising model (Denoising) across all datasets and evaluation metrics. This indicates that the proposed hybrid alignment and denoising alignment methods significantly mitigate the impact of noise and enhance the model's performance. b) The BOD model demonstrates a notable improvement over SGDL,

Table 6. Overall performance comparisons with baseline models on Yelp.

		Recall@5	Recall@10	Recall@20	NDCG@5	NDCG@10	NDCG@20
DenoisingRec	Denoising	0.0260	0.0451	0.0766	0.0316	0.0373	0.0481
NGCF		0.0367	0.0621	0.1064	0.0443	0.0528	0.0619
LightGCN	Base	0.0416	0.0706	0.1150	0.0485	0.0577	0.0728
	LLMRec	0.4105	0.1716	0.1210	0.0492	0.0600	0.0745
	RLMRec-Con	0.0417	0.0717	0.1134	0.0492	0.0606	0.0739
	RLMRec-Gen	0.0416	0.0719	0.1212	0.0494	0.0596	0.0748
	Our model	**0.0440**	**0.0749**	**0.1240**	**0.0521**	**0.0620**	**0.0762**
	Imp.%	5.57%	4.19%	2.33%	5.55%	2.19%	1.92%
SGDL	Denoising	0.0428	0.0735	0.1241	0.0488	0.0598	0.0756
SGL	Base	0.0429	0.0731	0.1207	0.0502	0.0597	0.0758
	RLMRec-Con	0.0453	0.0768	0.1227	0.0528	0.0625	0.0782
	RLMRec-Gen	0.0459	0.0761	0.1219	0.0526	0.0622	0.0781
	Our model	**0.0484**	**0.0785**	**0.1273**	**0.0556**	**0.0643**	**0.0807**
	Imp.%	5.52%	2.31%	3.77%	5.39%	2.91%	3.15%
BOD	Denoising	0.0480	0.0795	0.1309	0.0561	0.0657	0.0835
SimGCL	Base	0.0460	0.0758	0.1255	0.0536	0.0626	0.0793
	RLMRec-Con	0.0464	0.0782	0.1304	0.0542	0.0642	0.0816
	RLMRec-Gen	0.0462	0.0759	0.1241	0.0537	0.0634	0.0803
	Our model	**0.0468**	**0.0795**	**0.1316**	**0.0556**	**0.0653**	**0.0835**
	Imp.%	1.02%	1.72%	0.92%	2.69%	1.75%	2.31%
DCCF	Base	0.0463	0.0770	0.1249	0.0548	0.0639	0.0803
	RLMRec-Con	0.0480	0.0805	0.1326	0.0553	0.0661	0.0829
	RLMRec-Gen	0.0476	0.0789	0.1268	0.0549	0.0646	0.0815
	Our model	**0.0499**	**0.0831**	**0.1340**	**0.0572**	**0.0673**	**0.0847**
	Imp.%	3.98%	3.22%	1.07%	3.30%	1.80%	2.17%
AutoCF	Base	0.0470	0.0793	0.1287	0.0541	0.0642	0.0808
	RLMRec-Con	0.0485	0.0802	0.1311	0.0560	0.0656	0.0829
	RLMRec-Gen	0.0493	0.0820	0.1330	0.0571	0.0673	0.0840
	Our model	**0.0509**	**0.0853**	**0.1365**	**0.0586**	**0.0693**	**0.0866**
	Imp.%	3.22%	3.91%	2.60%	2.67%	2.92%	3.00%

DenoisingRec, and the Base model. This suggests that the BOD model exhibits substantial advancements in recommendation accuracy and relevance.

5.3 Ablation Study (RQ2)

To evaluate the effectiveness of our proposed denoising alignment based on large language models (LLMs), we conducted two ablation experiments: (1) Effectiveness of different alignment components and (2) Effectiveness of representation components using different LLMs.

5.3.1 Effectiveness of different alignment components. In this section, we explore the effects of various alignment components on performance through a series of ablation experiments. These experiments evaluate the alignment performance individually based on hybrid features and textual semantics. The outcomes of these experiments are summarized in Table 7, leading to the derivation of two pivotal observations.

Table 7. The results of ablation study.

	Amazon				Steam			
	Recall@5	Recall@10	NDCG@5	NDCG@10	Recall@5	Recall@10	NDCG@5	NDCG@10
NGCF	0.0486	0.0841	0.0512	0.0613	0.0588	0.0819	0.0544	0.0631
LightGCN Base	0.0579	0.0916	0.0584	0.0700	0.0519	0.0863	0.0571	0.0688
RLMRec[con]	0.0613	0.0976	0.0616	0.0742	0.0553	0.0900	0.0604	0.0721
RLMRec[gen]	0.0615	0.0969	0.0628	0.0747	0.0553	0.0910	0.0604	0.0726
w/o -RLMRec[con]	0.0638	0.0986	0.0629	0.0759	0.0566	0.0923	0.0609	0.0732
w/o -RLMRec[gen]	0.0625	0.0982	0.0628	0.0752	0.0570	0.0924	0.0614	0.0736
our Model	**0.0642**	**0.1023**	**0.0634**	**0.0764**	**0.0574**	**0.0931**	**0.0625**	**0.0745**
SGL Base	0.0644	0.1023	0.0643	0.0774	0.0559	0.0911	0.0614	0.0734
RLMRec-Con	0.0660	0.1026	0.0661	0.0786	0.0592	0.0945	0.0647	0.0766
RLMRec-Gen	0.0644	0.1010	0.0653	0.0779	0.0575	0.0930	0.0625	0.0745
w/o -RLMRec[con]	0.0487	0.0786	0.0495	0.0597	0.0543	0.0892	0.0596	0.0714
w/o -RLMRec[gen]	0.0638	0.1024	0.0645	0.0778	0.0600	0.0954	0.0655	0.0773
Our model	**0.0676**	**0.1039**	**0.0664**	**0.0793**	**0.0664**	**0.1016**	**0.0668**	**0.0789**
SimGCL Base	0.0610	0.0991	0.0612	0.0747	0.0567	0.0930	0.0621	0.0742
RLMRec-Con	0.0651	0.1022	0.0653	0.0781	0.0581	0.0948	0.0639	0.0761
RLMRec-Gen	0.0619	0.0984	0.0631	0.0755	0.0578	0.0934	0.0632	0.0753
w/o -RLMRec[con]	0.0641	0.1025	0.0638	0.0771	0.0565	0.0897	0.0608	0.0720
w/o -RLMRec[gen]	0.0557	0.0912	0.0561	0.0684	0.0588	0.0939	0.0637	0.0751
our Model	**0.0664**	**0.1058**	**0.0663**	**0.0799**	**0.0645**	**0.1021**	**0.0645**	**0.0768**
AutoCF Base	0.0688	0.1058	0.0714	0.0837	0.0519	0.0853	0.0574	0.0687
RLMRec-Con	0.0696	0.1064	0.0703	0.0827	0.0524	0.0873	0.0577	0.0695
RLMRec-Gen	0.0691	0.1088	0.0711	0.0845	0.0542	0.0894	0.0592	0.0711
w/o -RLMRec[con]	0.0727	0.1109	0.0731	0.0859	0.0518	0.0855	0.0568	0.0681
w/o -RLMRec[gen]	0.0717	0.1097	0.0725	0.0852	0.0477	0.0803	0.0512	0.0627
Our model	**0.0731**	**0.1118**	**0.0735**	**0.0862**	**0.0562**	**0.0923**	**0.0608**	**0.0728**

(1) Our proposed alignment based on hybrid features is effective, indicating the importance of emphasizing hybrid feature alignment in DALR. Specifically, the results of DALR outperform all ablation models across all evaluation metrics on the Amazon-book and Steam datasets, as observed in Table 7.

(2) The performance improvement brought by single-level alignment components is suboptimal. Specifically, on models based on SGL as the backbone, the hybrid feature alignment component enhances Recall and NDCG performance on the Steam dataset. On the Amazon dataset, the hybrid feature alignment component instead leads to a slight decrease in performance compared to the textual semantics alignment component. Similarly, on networks based on autocf as the backbone, the effectiveness of the hybrid feature alignment component is lower than that of the textual semantics component on the Steam dataset. We attribute these observations to two main reasons: (i) while the process of learning mixed alignment representations can help obtain better user and item embeddings, we cannot ignore the impact of fine-grained text semantics; (ii) we attribute the performance differences to the differences between the Amazon and Steam datasets when aligning features in a single dimension.

5.3.2 Effectiveness of representation components using different LLMs. In this section, we investigate the impact of semantic representations from different large language models on performance. For this purpose, we analyze

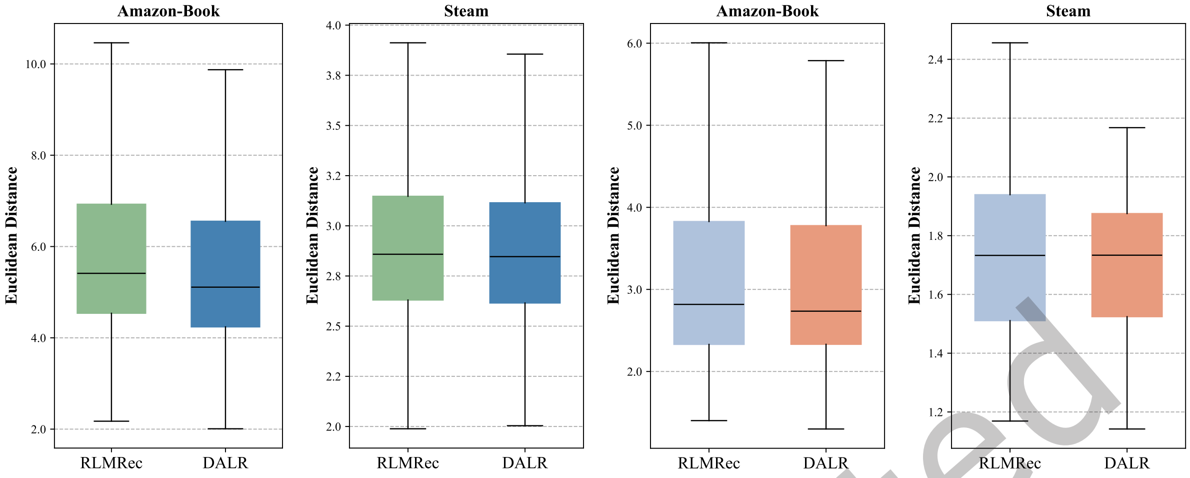
Table 8. Comparison with LLMs-enhanced Approaches.

SimGCL		Based	RLMRec-con (BERT)	RLMRec-gen (BERT)	RLMRec-con (GPT2)	RLMRec-gen (GPT2)	Our Model (BERT)	Our Model (GPT2)	**Our Model**
Amazon	R@5	0.0610	0.0551	0.0611	0.0412	0.0615	0.0616	0.0593	**0.0664**
	R@10	0.0991	0.0901	0.0973	0.0881	0.0977	0.0988	0.0962	**0.1058**
	R@20	0.1513	0.1410	0.1505	0.1369	0.1491	0.1525	0.1485	**0.1616**
	N@5	0.0612	0.0607	0.0621	0.0511	0.0618	0.0613	0.0607	**0.0663**
	N@10	0.0747	0.0725	0.0744	0.0664	0.0739	0.0745	0.0736	**0.0799**
	N@20	0.0915	0.0909	0.0918	0.0719	0.0907	0.0920	0.0908	**0.0981**
Steam	R@5	0.0567	0.0572	0.0571	0.0438	0.0566	0.0589	0.0585	**0.0645**
	R@10	0.0930	0.0942	0.0927	0.0867	0.0925	0.0953	0.0956	**0.1021**
	R@20	0.1450	0.1481	0.1453	0.1344	0.1451	0.1488	0.1512	**0.1558**
	N@5	0.0621	0.0633	0.0626	0.0543	0.0622	0.0643	0.0643	**0.0644**
	N@10	0.0742	0.0759	0.0745	0.0635	0.0744	0.0765	0.0767	**0.0768**
	N@20	0.0920	0.0942	0.0924	0.0887	0.0922	0.0947	0.0957	**0.0947**

AutoCF		Based	RLMRec-con (BERT)	RLMRec-gen (BERT)	RLMRec-con (GPT2)	RLMRec-gen (GPT2)	Our Model (BERT)	Our Model (GPT2)	**Our Model**
Amazon	R@5	0.0688	0.0687	0.0687	0.6048	0.0689	0.0710	0.0695	**0.0731**
	R@10	0.1058	0.1054	0.1075	0.0897	0.1066	0.1073	0.1076	**0.1118**
	R@20	0.1559	0.1554	0.1580	0.1499	0.1567	0.1599	0.1610	**0.1644**
	N@5	0.0714	0.0699	0.0711	0.0589	0.0700	0.0708	0.0709	**0.0735**
	N@10	0.0837	0.0821	0.0841	0.0668	0.0828	0.0834	0.0838	**0.0862**
	N@20	0.0998	0.0985	0.1005	0.0746	0.0991	0.1005	0.1011	**0.1028**
Steam	R@5	0.0519	0.0453	0.0526	0.0487	0.0526	0.0506	0.0524	**0.0562**
	R@10	0.0853	0.0784	0.0857	0.0749	0.0866	0.0841	0.0860	**0.0923**
	R@20	0.1357	0.1261	0.1372	0.1210	0.1376	0.1365	0.1380	**0.1448**
	N@5	0.0574	0.0509	0.0581	0.0488	0.0578	0.0554	0.0570	**0.0608**
	N@10	0.0687	0.0623	0.0691	0.0598	0.0692	0.0667	0.0684	**0.0728**
	N@20	0.0858	0.0785	0.0864	0.0795	0.0865	0.0845	0.0860	**0.0906**

the effects of three different pre-trained large language models on performance: ChatGPT3.5, GPT2, and BERT [10]. Among them, DALR defaults to using the semantic encoding model text-embedding-ada-002 based on ChatGPT3.5. We evaluate our approach using two backbone methods (i.e., SGL and AutoCF). The results are summarized in Table 8, yielding two key observations:

(1) The advancement of ChatGPT3.5 improves the recommender's performance. As shown in Table 8, when DALR framework utilizes ChatGPT3.5's text embedding model (default setting of text-embedding-ada-002), it significantly enhances the baseline performance, outperforming the backbone model and RLMRec. This result indicates that DALR effectively leverages the text encoder to transform text semantics into preference representations, thereby enhancing the performance of the recommendation backbone. Additionally, compared to pre-trained models based on GPT2 and BERT, the advanced pre-trained model ChatGPT3.5 can capture semantic information with higher precision, leading to more significant improvements. (2) Integrating BERT and GPT2 into RLMRec did not significantly outperform the baseline models and, in some cases, even performed worse than the base models. This result indicates that for specific datasets and backbone models, the richness of semantics in BERT and GPT2 may not be as effective as that in GPT3.5. Moreover, simply injecting semantic information from this paper into the backbone network does not significantly bridge the gap between semantic representation and



(a) LightGCN as the backbone model (b) LightGCN as the backbone model (c) SimGCL as the backbone model (d) SimGCL as the backbone model

Fig. 3. Visualization of the representations before and after the alignment.

spatial structural representation. Therefore, our hybrid structural alignment component can effectively mitigate this issue.

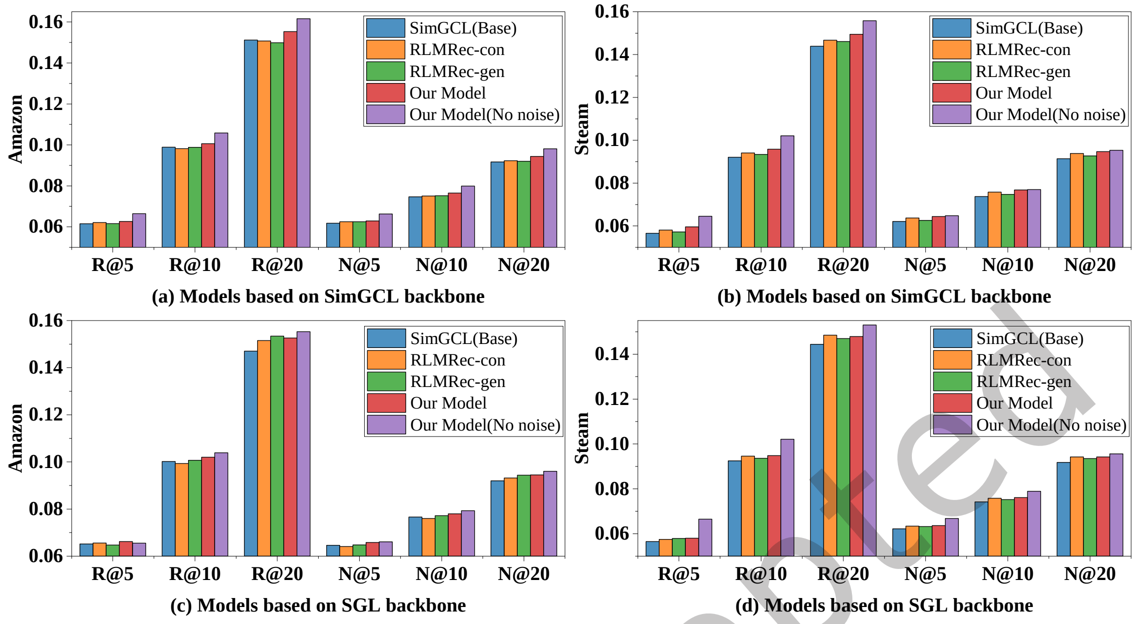
5.4 In-depth Analysis of DALR (RQ3)

5.4.1 Visualization of bridging the gap. To investigate the impact of hybrid feature alignment in bridging the gap between different representation spaces, we capture users' representations on the backbone, RLMRec, and our model, respectively. Then, we compute the distances between the user's embedding vector on the backbone and the vectors obtained from RLMRec and DALR, respectively. To ensure the reliability of our experiments, we select two backbone networks, LightGCN and SimGCL, and include two datasets: Amazon-Book and Steam. The visualization format we utilize is the box plot.

DALR effectively reduces the gap between different representation spaces and significantly enhances the representational power of the recommendation system. In the subfigure of Fig. 3, the left side illustrates distances from RLMRec to the base model, while the right side illustrates distances from DALR to the base model. Initially, the vectors for GNN and RLMRec exhibit a large Euclidean distance on the left side. However, the distance between vectors is markedly reduced after applying hybrid feature alignment. These findings affirm DALR's effectiveness in aligning the spaces of structural and textual modalities.

5.4.2 Performance of noise data. We evaluate the robustness of DALR to data noise by augmenting the original training data with virtual interactions. The noise level is set at 5% relative to the size of the training set. We compare our approach with backbone networks SimGCL, SGL, and RLMRec-Con/Gen on the Amazon and Steam datasets.

The main findings of Fig. 4 are as follows: (i) Our model consistently outperforms the backbone models of SimGCL and SGL, as well as RLMRec-Con/Gen. This result underscores the advantage of aligning semantic information and structural features, wherein the framework leverages contrastive learning to filter out Irrelevant data influences, thereby enhancing recommendation performance and robustness against noise. (ii) Compared to the base model, all alignment methods demonstrate better resistance to data noise. The results suggest that the introduction of alignment methods, whether based on semantic information or hybrid features, can alleviate noise effects in feature representation, thus exhibiting superior performance under the same noise ratio.



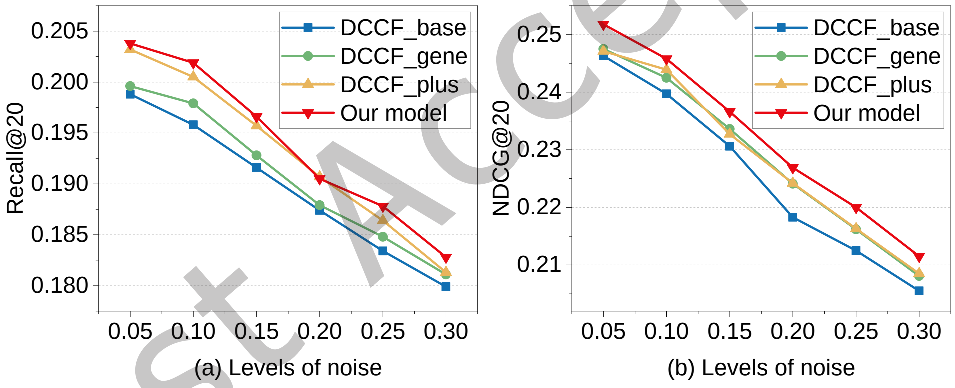
(a) Models based on SimGCL backbone

(b) Models based on SimGCL backbone

(c) Models based on SGL backbone

(d) Models based on SGL backbone

Fig. 4. The Impact of noise data.



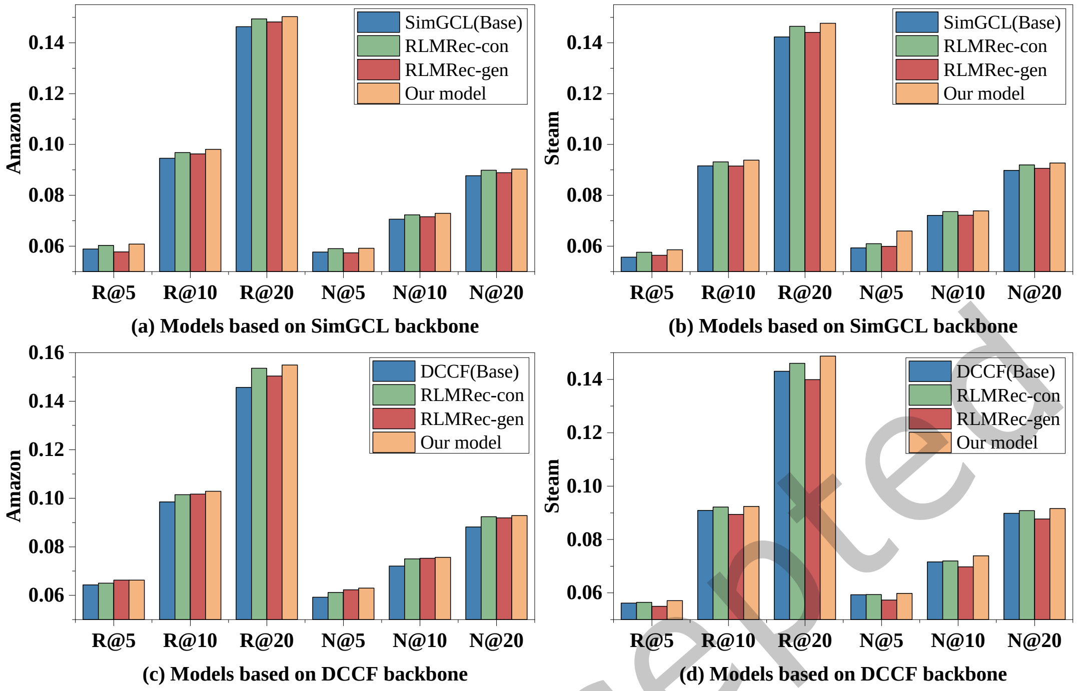
(a) Levels of noise

(b) Levels of noise

Fig. 5. Comparing performance on different levels of noise in the Movie1m dataset with DCCF backbone models.

Moreover, We evaluated six noise levels (5%, 10%, 15%, 20%, 25%, and 30%) to investigate the impact of different levels of interaction noise on the model. We generated new noisy training data by adding varying levels of random noise to the interaction data. We compared the performance of the DCCF backbone network and its variant models on Movie1m dataset. The results in Fig. 5 indicate that our model outperforms both the DCCF backbone and its RLMRec variant across all noise levels. The above results demonstrate the advantage of combining hybrid feature alignment and leveraging denoising alignment to mitigate noisy data, thereby enhancing noise robustness and improving recommendation performance.

5.4.3 Performance on Sparse Interaction Data. To evaluate the performance of our model in handling real-world sparse datasets, we conducted sparse interaction testing experiments. Specifically, we attempted to retain only data with interactions below the average as training data across the Amazon-Book and Steam datasets. Additionally,



(a) Models based on SimGCL backbone

(b) Models based on SimGCL backbone

(c) Models based on DCCF backbone

(d) Models based on DCCF backbone

Fig. 6. The impact of sparse data.

we assessed performance under sparse interaction conditions across the SimGCL and DCCF backbone networks and the RLMRec and DALR models.

The experimental results are presented in Fig. 6. From these results, we draw the following conclusions: (1) DALR consistently outperforms the backbone models of SimGCL and DCCF, as well as RLMRec-Con/Gen. This result underscores the critical role of large-scale language models in semantic information and the alignment of hybrid features. Our framework compensates for the deficiencies of sparsity from a data augmentation perspective by utilizing ChatGPT to generate user and item profile information. (2) Relative to RLMRec, our framework achieves significant improvements. The above result indicates that introducing hybrid features enables learning of relationships between nodes in the network, resulting in superior performance under the same sparse interaction conditions. (3) Compared to RLMRec-Gen, RLMRec-Con demonstrates better resistance to data sparsity. This may be attributed to the generative approach increasing the likelihood of introducing noise through node masking. In contrast, the contrastive method exhibits better performance.

5.4.4 Analysis of Efficiency. We analyzed the time complexity of our model from the perspectives of model complexity and running time. We can find that compared with those competitive baselines, our proposed models' efficiency is similar to baseline models, while our models significantly outperform baselines on recommendation performance.

Analysis of the computational complexity. For the hybrid feature alignment module, the theoretical time complexity of the sampling strategy is $O(|V|)$, and the complexity of the alignment loss computation is $O\left(N^2 \times d\right)$. For the semantic attribute alignment module, the time complexity of the contrastive modelling module is $O\left(N^2 \times d\right)$, and the generative modelling module has a time complexity of $O(M \times N \times d)$. For BPR (Bayesian Personalized Ranking), the time complexity is $O(U \times n_u \times d)$.

Table 9. Timing cost analysis of our model and its variants.

DCCF backbone	Amazon-Book	Steam	Yelp	Movie1m
Base	2.02s	11.54s	4.05s	18.21s
RLMRec-Con	2.55s	12.02s	4.53s	20.35s
RLMRec-Gen	2.10s	12.03s	4.12s	20.16s
Our model	3.01s	14.05s	5.02s	22.45s

Table 10. The impact of hyperparameter λ_1.

λ_1	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
Recall@5	0.0661	0.0660	0.0664	0.0676	0.0660	0.0653	0.0660	0.0653	0.0652
Recall@10	0.1004	0.1014	0.1033	0.1049	0.1026	0.1028	0.1030	0.1028	0.1034
Recall@20	0.1521	0.1552	0.1560	0.1561	0.1563	0.1547	0.1549	0.1544	0.1551
NDCG@5	0.0644	0.0646	0.0649	0.0663	0.0648	0.0647	0.0649	0.0645	0.0646
NDCG@10	0.0764	0.0768	0.0778	0.0792	0.0775	0.0776	0.0778	0.0777	0.0779
NDCG@20	0.0933	0.0944	0.0950	0.0981	0.0946	0.0948	0.0947	0.0948	0.0950

Table 11. The impact of hyperparameter λ_2.

λ_2	0.01	0.05	0.1	0.15	0.2	0.25	0.3	0.4	0.5	0.6
Recall@5	0.0400	0.0573	0.0620	0.0647	0.0650	0.0676	0.0659	0.0465	0.0447	0.0408
Recall@10	0.0645	0.0897	0.0982	0.1001	0.1011	0.1039	0.1026	0.0710	0.0677	0.0644
Recall@20	0.1020	0.1408	0.1498	0.1511	0.1519	0.1560	0.1522	0.1040	0.0997	0.0964
NDCG@5	0.0412	0.0571	0.0620	0.0649	0.0649	0.0664	0.0655	0.0469	0.0453	0.0414
NDCG@10	0.0491	0.0680	0.0746	0.0769	0.0773	0.0793	0.0781	0.0552	0.0531	0.0494
NDCG@20	0.0612	0.0847	0.0914	0.0935	0.0939	0.0960	0.0943	0.0660	0.0637	0.0598

Runtime performance. We measured the epoch time during training on a server equipped with an NVIDIA A40 GPU to further evaluate the time cost. Specifically, we used DCCF as the backbone network and tested the runtime on four datasets (as shown in Table 9). The additional time cost of our model is only about 10% to 20% higher than the original RLMRec. However, our model achieves significant performance improvements.

5.5 Hyper-parameter Sensitivity (RQ4)

To evaluate the effect of the hyperparameter λ_i and sample strategy parameter q, we conduct experiments on two datasets by varying their values.

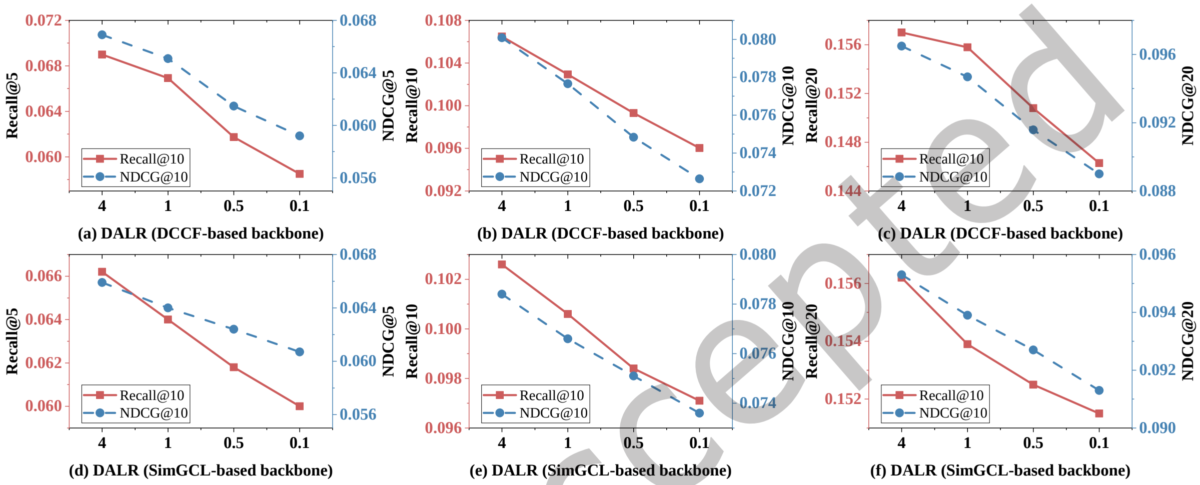
5.5.1 The hyperparameter λ_i. Using a controlled variable design, we explored the impact of different hyperparameter weights on experimental results while keeping other parameters constant. We used the Amazon-Book dataset as an example and conducted a detailed analysis on the SGL-based backbone network. The experimental results are shown in Table 10, 11, and 12.

The hyperparameters λ_1: Within the specified threshold range, as λ_1 increases, the performance of DALR initially improves and then declines. Specifically, DALR achieves optimal results at $\lambda_1 = 0.4$ (such as Recall@5=0.0676 and NDCG@5=0.0663) and $\lambda_1 = 0.5$ (Recall@20=0.1563). These findings suggest that the hybrid feature alignment enhances the representation capability and effectiveness of the recommender system. However, if $\lambda_1 = 0$, the model lacks the alignment signal of hybrid features. Conversely, if λ_1 is 1, the model suffers from overgeneralization, leading to a decrease in recommendation accuracy.

The hyperparameters λ_2: As λ_2 increases, the performance of DALR exhibits a rising and then falling curve. DALR achieves the best results at $\lambda_2 = 0.25$ (such as Recall@10=0.1039 and NDCG@10=0.0793). These results

Table 12. The impact of hyperparameter λ_3.

λ_3	1.0E-06	5.0E-06	1.0E-05	5.0E-05	1.0E-04	1.0E-03	1.0E-02	1.0E-01
Recall@5	0.0634	0.0641	0.0654	0.0676	0.0624	0.0472	0.0376	0.0288
Recall@10	0.0991	0.0997	0.1013	0.1039	0.0981	0.0740	0.0594	0.0462
Recall@20	0.1485	0.1508	0.1535	0.1560	0.1501	0.1119	0.0883	0.0690
NDCG@5	0.0630	0.0629	0.0637	0.0664	0.0621	0.0485	0.0389	0.0289
NDCG@10	0.0754	0.0755	0.0761	0.0793	0.0744	0.0575	0.0460	0.0349
NDCG@20	0.0915	0.0921	0.0933	0.0960	0.0913	0.0697	0.0555	0.0422



(a) DALR (DCCF-based backbone)

(b) DALR (DCCF-based backbone)

(c) DALR (DCCF-based backbone)

(d) DALR (SimGCL-based backbone)

(e) DALR (SimGCL-based backbone)

(f) DALR (SimGCL-based backbone)

Fig. 7. The impact of sample strategy parameter q on Amazon-Book.

indicate that denoising alignment enhances the representation capability of our model. If $\lambda_2 = 0.01$, the impact weight of the denoising alignment signal is too small, contributing insufficiently to the model. Conversely, if $\lambda_2 > 0.4$, the model overgeneralizes during the training process, resulting in decreased recommendation accuracy.

The hyperparameters λ_3: The choice of the regularization parameter λ_3 is a balancing act that must prevent both overfitting and underfitting, ultimately aiming to enhance the model's generalization capability and stability. The experimental results show that the model achieves stable performance at λ_3 = 5.0E-05 (such as Recall@5=0.0676, Recall@10=0.1039 and Recall@20=0.1560). If the regularization parameter λ_3 is small, the regularization effect is insignificant, and the model may overfit the training data, leading to poor performance on the test data. Conversely, if λ_3 is too large, the regularization effect is too strong, excessively limiting the model's freedom and preventing it from adequately learning the data patterns, resulting in underfitting.

The analysis reveals that optimal performance is achieved with specific hyperparameter values: $\lambda_1 = 0.4$ and 0.5 for alignment, $\lambda_2 = 0.25$ for denoising, and λ_3 = 5.0E-05 for regularization. At these values, the model demonstrates enhanced representation capability and stability. Deviations from these optimal settings result in insufficient alignment, overgeneralization, or inadequate regularization, affecting the model's accuracy and performance.

5.5.2 *Samle strategy parameter q.*

In this section, we conducted hyperparameter experiments to validate the impact of the sampling strategy parameter q on performance. As indicated in Section 4.4.1, the results of structural features are predominantly influenced by the sampling strategy parameters p and q. A smaller p value causes the sampling to hover around the root node, exhibiting BFS characteristics, while a smaller q value leads to the sampling gradually moving away from the root node, demonstrating DFS characteristics. This approach enables us to employ different sampling strategies for various graphs, obtaining higher-quality node structural properties.

Specifically, we fixed the p value ($p = 1$) and varied the q value to examine its impact on different structural attributes and model performance. We tested q values of 4, 1, 0.5, 0.1 on two backbone networks based on SimGCL and DCCF, respectively, using the Amazon-Book dataset. From the experimental results in Fig. 7, we observed the following conclusions: Under the condition of fixed p values, a decreasing trend is observed in Recall and NDCG@5, 10, 20 as the q values decrease 4, 1, 0.5, 0.1. This trend indicates that nodes sampled farther away from the root node exhibit significant dissimilarities with the root node, which aligns with our theoretical expectations.

6 CONCLUSION AND FUTURE WORK

In summary, we have proposed DALR, a denoise alignment framework that bridges the representation gap between the semantics of LLMs and GNN-based recommenders. Our framework addresses the challenge of aligning structural with textual representations and mitigating noise effects to enhance recommendation accuracy. In detail, DALR consists of two key components: a hybrid alignment paradigm and a semantic contrast component. Firstly, we propose an alignment paradigm to enhance representation performance by aligning hybrid and structural features from backbone models. Secondly, we introduce a semantic contrast component to relieve the impact of noise and improve model performance. Lastly, a lightweight collaborative framework fine-tunes the backbone model and evaluates its performance on the test dataset. We have demonstrated that DALR consistently outperforms state-of-the-art methods through extensive experiments on public datasets. Furthermore, our future research will focus on enhancing model effectiveness while reducing the costs associated with large language model deployment to deploy these methods in online recommenders.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1007–1014.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[4] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. *arXiv preprint arXiv:2302.08191* (2023).

[5] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[6] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 27–34.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[8] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. 2020. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems* 33 (2020), 22243–22255.

[9] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084* (2022).

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[11] Hao Fu, Shaojun Zhou, Qihong Yang, Junjie Tang, Guiquan Liu, Kaikui Liu, and Xiaolong Li. 2021. LRC-BERT: latent-representation contrastive knowledge distillation for natural language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 12830–12838.

[12] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. 2023. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender*

Systems 1, 1 (2023), 1–51.

[13] Yunjun Gao, Yuntao Du, Yujia Hu, Lu Chen, Xinjun Zhu, Ziquan Fang, and Baihua Zheng. 2022. Self-guided learning to denoise for robust recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1412–1422.

[14] 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 *Proceedings of the 16th ACM Conference on Recommender Systems*. 299–315.

[15] Jiahui Gong, Jingtao Ding, Fanjin Meng, Guilong Chen, Hong Chen, Shen Zhao, Haisheng Lu, and Yong Li. 2024. A Population-to-individual Tuning Framework for Adapting Pretrained LM to On-device User Intent Prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 896–907.

[16] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[17] Michael U Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of machine learning research* 13, 2 (2012).

[18] Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging large language models for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1096–1102.

[19] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.

[20] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong 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*. 639–648.

[21] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670* (2018).

[22] Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2023. Genrec: Large language model for generative recommendation. *arXiv e-prints* (2023), arXiv–2307.

[23] Yangqin Jiang, Chao Huang, and Lianghao Huang. 2023. Adaptive graph contrastive learning for recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4252–4261.

[24] Dongho Lee, Byungkook Oh, Seungmin Seo, and Kyong-Ho Lee. 2020. News recommendation with topic-enriched knowledge graphs. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 695–704.

[25] Ming Li, Lin Zhang, Lixin Cui, Lu Bai, Zhao Li, and Xindong Wu. 2023. BLoG: Bootstrapped graph representation learning with local and global regularization for recommendation. *Pattern Recognition* 144 (2023), 109874.

[26] Xiaohan Li, Mengqi Zhang, Shu Wu, Zheng Liu, Liang Wang, and S Yu Philip. 2020. Dynamic graph collaborative filtering. In *2020 IEEE international conference on data mining (ICDM)*. IEEE, 322–331.

[27] Yinfeng Li, Chen Gao, Hengliang Luo, Depeng Jin, and Yong Li. 2022. Enhancing hypergraph neural networks with intent disentanglement for session-based recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1997–2002.

[28] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*. 689–698.

[29] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM Web Conference 2022*. 2320–2329.

[30] Peng Liu, Lemei Zhang, and Jon Atle Gulla. 2023. Pre-train, Prompt, and Recommendation: A Comprehensive Survey of Language Modeling Paradigm Adaptations in Recommender Systems. *Transactions of the Association for Computational Linguistics* 11 (2023), 1553–1571.

[31] Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2023. A first look at llm-powered generative news recommendation. *arXiv preprint arXiv:2305.06566* (2023).

[32] Dongsheng Luo, Yuchen Bian, Xiang Zhang, and Jun Huan. 2020. Attentive social recommendation: Towards user and item diversities. *arXiv preprint arXiv:2011.04797* (2020).

[33] Yingrong Qin, Chen Gao, Shuangqing Wei, Yue Wang, Depeng Jin, Jian Yuan, Lin Zhang, Dong Li, Jianye Hao, and Yong Li. 2023. Learning from hierarchical structure of knowledge graph for recommendation. *ACM Transactions on Information Systems* 42, 1 (2023), 1–24.

[34] Yuhan Quan, Jingtao Ding, Chen Gao, Lingling Yi, Depeng Jin, and Yong Li. 2023. Robust preference-guided denoising for graph based social recommendation. In *Proceedings of the ACM Web Conference 2023*. 1097–1108.

[35] 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 2024*. 3464–3475.

[36] Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. 2023. Disentangled Contrastive Collaborative Filtering. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*. 1137–1146.

[37] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the Twelfth ACM international conference on web search and data mining*. 555–563.

[38] Yuzhi Song, Hailiang Ye, Ming Li, and Feilong Cao. 2022. Deep multi-graph neural networks with attention fusion for recommendation. *Expert Systems with Applications* 191 (2022), 116240.

[39] Jianing Sun, Zhaoyue Cheng, Saba Zuberi, Felipe Pérez, and Maksims Volkovs. 2021. Hgcf: Hyperbolic graph convolution networks for collaborative filtering. In *Proceedings of the Web Conference 2021*. 593–601.

[40] Chenyang Wang, Weizhi Ma, Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. Sequential recommendation with multiple contrast signals. *ACM Transactions on Information Systems* 41, 1 (2023), 1–27.

[41] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on WSDM*. 373–381.

[42] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded recommendation for alleviating bias amplification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1717–1725.

[43] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 950–958.

[44] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[45] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1001–1010.

[46] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, and Tat-Seng Chua. 2020. Reinforced negative sampling over knowledge graph for recommendation. In *Proceedings of the web conference 2020*. 99–109.

[47] Zongwei Wang, Min Gao, Wentao Li, Junliang Yu, Linxin Guo, and Hongzhi Yin. 2023. Efficient bi-level optimization for recommendation denoising. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2502–2511.

[48] Wei Wei, Chao Huang, Lianghao Xia, Yong Xu, Jiashu Zhao, and Dawei Yin. 2022. Contrastive meta learning with behavior multiplicity for recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 1120–1128.

[49] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. Llmrec: Large language models with graph augmentation for recommendation. *arXiv preprint arXiv:2311.00423* (2023).

[50] 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*. 806–815.

[51] Jiahao Wu, Wenqi Fan, Jingfan Chen, Shengcai Liu, Qing Li, and Ke Tang. 2022. Disentangled contrastive learning for social recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4570–4574.

[52] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.

[53] Likang Wu, Zhaopeng Qiu, Zhi Zheng, Hengshu Zhu, and Enhong Chen. 2023. Exploring large language model for graph data understanding in online job recommendations. *arXiv preprint arXiv:2307.05722* (2023).

[54] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2023. A Survey on Large Language Models for Recommendation. *arXiv preprint arXiv:2305.19860* (2023).

[55] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The world wide web conference*. 2091–2102.

[56] Lianghao Xia, Chao Huang, Chunzhen Huang, Kangyi Lin, Tao Yu, and Ben Kao. 2023. Automated Self-Supervised Learning for Recommendation. In *Proceedings of the ACM Web Conference 2023*. 992–1002.

[57] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*. 70–79.

[58] Lianghao Xia, Chao Huang, and Chuxu Zhang. 2022. Self-supervised hypergraph transformer for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2100–2109.

[59] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.

[60] Yuhao Yang, Chao Huang, Lianghao Xia, Yuxuan Liang, Yanwei Yu, and Chenliang Li. 2022. Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 2263–2274.

[61] Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, and Li Yuan. 2023. Llm lies: Hallucinations are not bugs, but features as adversarial examples. *arXiv preprint arXiv:2310.01469* (2023).

[62] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1294–1303.

[63] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. 2023. Self-supervised learning for recommender systems: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[64] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Recommendation as instruction following: A large language model empowered recommendation approach. *arXiv preprint arXiv:2305.07001* (2023).

[65] Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng Yang, and Chuan Shi. 2024. GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks. *arXiv preprint arXiv:2402.07197* (2024).

[66] Wenxuan Zhang, Hongzhi Liu, Yingpeng Du, Chen Zhu, Yang Song, Hengshu Zhu, and Zhonghai Wu. 2023. Bridging the Information Gap Between Domain-Specific Model and General LLM for Personalized Recommendation. *arXiv preprint arXiv:2311.03778* (2023).

[67] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th CIKM*. 1893–1902.

[68] Yaochen Zhu and Zhenzhong Chen. 2022. Mutually-regularized dual collaborative variational auto-encoder for recommendation systems. In *Proceedings of The ACM Web Conference 2022*. 2379–2387.

[69] Ding Zou, Wei Wei, Xian-Ling Mao, Ziyang Wang, Minghui Qiu, Feida Zhu, and Xin Cao. 2022. Multi-level cross-view contrastive learning for knowledge-aware recommender system. In *Proceedings of the 45th international ACM SIGIR*. 1358–1368.

A SUPPLEMENTARY MATERIAL

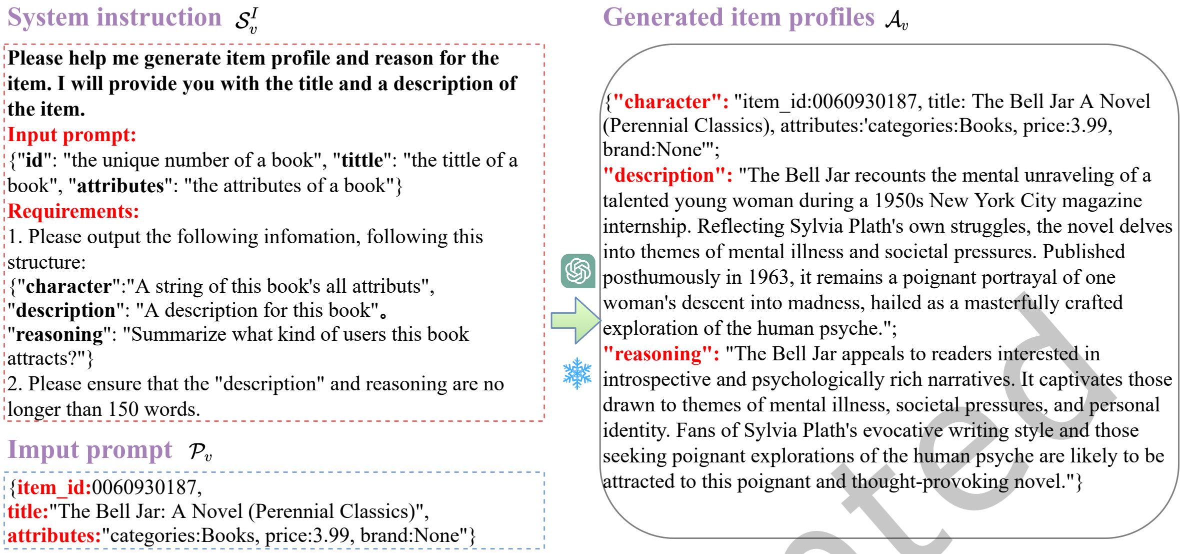
In the supplementary material, we provide an in-depth analysis of prompt design. Through examples, we demonstrate the visualization transition from user-generated samples to the profile generation process.

A.1 Profile Generation

This section elaborates on the generation of user and project profiles. We illustrate this process using real examples from the Amazon-book dataset, as shown in Fig. 8 and Fig. 9. A standard interaction paradigm with large language models (LLMs) is employed, where system prompts serve as directives for profile generation. Although the focus here is on the Amazon-book dataset, the generation approach for the Steam dataset is similar, with only minor differences in the commands used to represent data information in Fig. 10 and Fig. 11. Furthermore, we illustrate the generation of user and item profiles for both the Yelp and Movie1m datasets through examples presented in Fig. 12, 13, 14, and 15.

An illustration of the generated item profile. We aim to provide explicit instruction prompts for the project, guiding LLMs in generating book profiles to offer valuable information for recommendations in Fig. 8. Input information consists of unique identifiers, book titles, and attribute descriptions. To ensure profile quality, we mandate that the language model output adheres to specified requirements and prevents the occurrence of illusions. Output information encompasses basic book features, descriptions, and inferences. Results indicate that LLMs accurately capture significant book features and potential reader types they may attract.

An illustration of the generated user profile. We generate user profiles from project descriptions and user interaction data. This design enables the large language models to identify user preferences with precision by merging both sources of information. Furthermore, we employ system commands to control the quality of outputs. For example, by analyzing descriptions and reviews of books, the model precisely extracts user preference features in Fig. 9.



System instruction $\mathcal{S}_v^I$

Please help me generate item profile and reason for the item. I will provide you with the title and a description of the item.
Input prompt:
{"**id**": "the unique number of a book", "**tittle**": "the tittle of a book", "**attributes**": "the attributes of a book"}
Requirements:
1. Please output the following infomation, following this structure:
{"**character**":"A string of this book's all attributs",
"**description**": "A description for this book"。
"**reasoning**": "Summarize what kind of users this book attracts?"}
2. Please ensure that the "description" and reasoning are no longer than 150 words.

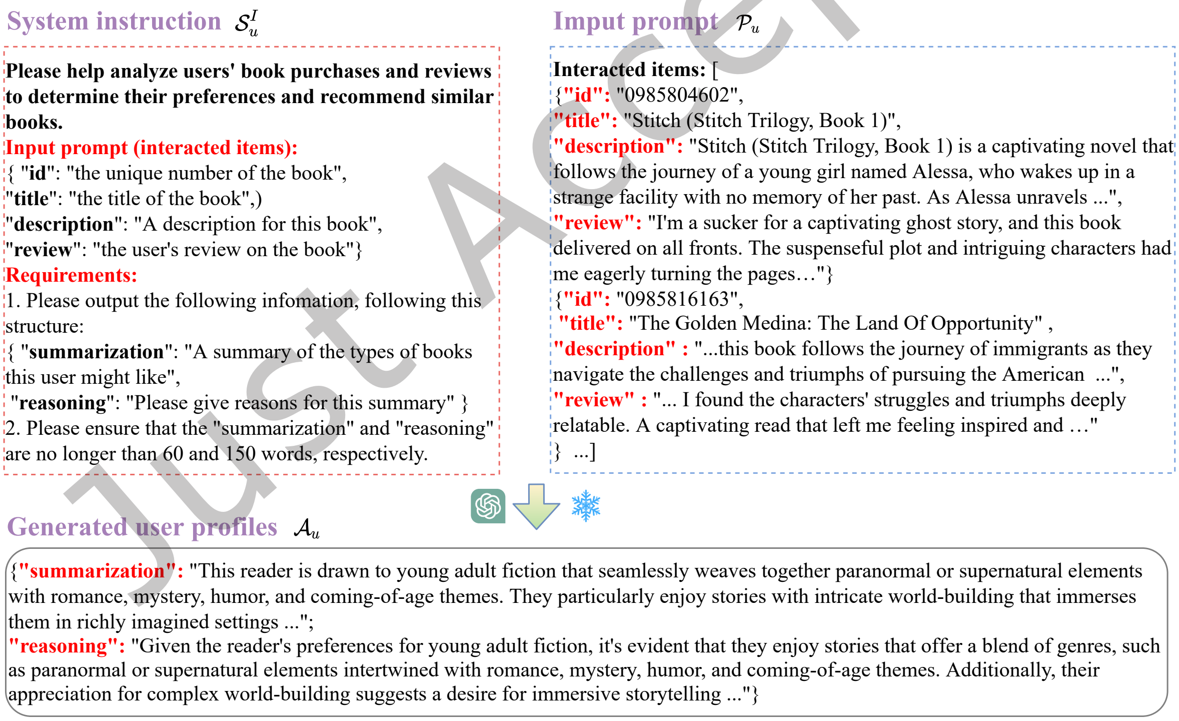
Imput prompt $\mathcal{P}_v$

{**item_id:**0060930187,
title:"The Bell Jar: A Novel (Perennial Classics)",
attributes:"categories:Books, price:3.99, brand:None"}

Generated item profiles $\mathcal{A}_v$

{**"character":** "item_id:0060930187, title: The Bell Jar A Novel (Perennial Classics), attributes:'categories:Books, price:3.99, brand:None'";
"description": "The Bell Jar recounts the mental unraveling of a talented young woman during a 1950s New York City magazine internship. Reflecting Sylvia Plath's own struggles, the novel delves into themes of mental illness and societal pressures. Published posthumously in 1963, it remains a poignant portrayal of one woman's descent into madness, hailed as a masterfully crafted exploration of the human psyche.";
"reasoning": "The Bell Jar appeals to readers interested in introspective and psychologically rich narratives. It captivates those drawn to themes of mental illness, societal pressures, and personal identity. Fans of Sylvia Plath's evocative writing style and those seeking poignant explorations of the human psyche are likely to be attracted to this poignant and thought-provoking novel."}

Fig. 8. An illustration of item profile generation in Amazon-book data.



System instruction $\mathcal{S}_u^I$

Please help analyze users' book purchases and reviews to determine their preferences and recommend similar books.
Input prompt (interacted items):
{ "**id**": "the unique number of the book",
"**title**": "the title of the book",)
"**description**": "A description for this book",
"**review**": "the user's review on the book"}
Requirements:
1. Please output the following infomation, following this structure:
{ "**summarization**": "A summary of the types of books this user might like",
"**reasoning**": "Please give reasons for this summary" }
2. Please ensure that the "summarization" and "reasoning" are no longer than 60 and 150 words, respectively.

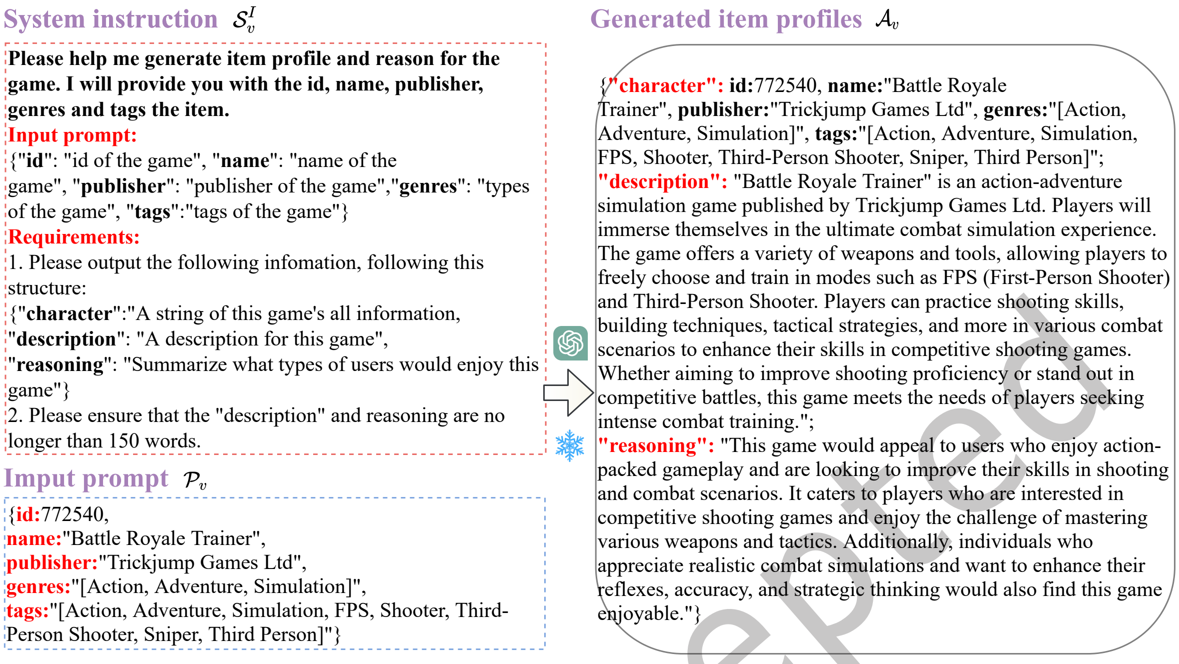
Imput prompt $\mathcal{P}_u$

Interacted items: [
{**"id":** "0985804602",
"title": "Stitch (Stitch Trilogy, Book 1)",
"description": "Stitch (Stitch Trilogy, Book 1) is a captivating novel that follows the journey of a young girl named Alessa, who wakes up in a strange facility with no memory of her past. As Alessa unravels ...",
"review": "I'm a sucker for a captivating ghost story, and this book delivered on all fronts. The suspenseful plot and intriguing characters had me eagerly turning the pages…"}
{**"id":** "0985816163",
"title": "The Golden Medina: The Land Of Opportunity" ,
"description" : "...this book follows the journey of immigrants as they navigate the challenges and triumphs of pursuing the American ...",
"review" : "... I found the characters' struggles and triumphs deeply relatable. A captivating read that left me feeling inspired and …"
} ...]

Generated user profiles $\mathcal{A}_u$

{**"summarization":** "This reader is drawn to young adult fiction that seamlessly weaves together paranormal or supernatural elements with romance, mystery, humor, and coming-of-age themes. They particularly enjoy stories with intricate world-building that immerses them in richly imagined settings ...";
"reasoning": "Given the reader's preferences for young adult fiction, it's evident that they enjoy stories that offer a blend of genres, such as paranormal or supernatural elements intertwined with romance, mystery, humor, and coming-of-age themes. Additionally, their appreciation for complex world-building suggests a desire for immersive storytelling ..."}

Fig. 9. An illustration of user profile generation in Amazon-book data.



System instruction $\mathcal{S}_v^I$

Please help me generate item profile and reason for the game. I will provide you with the id, name, publisher, genres and tags the item.

Input prompt:

{"**id**": "id of the game", "**name**": "name of the game", "**publisher**": "publisher of the game","**genres**": "types of the game", "**tags**":"tags of the game"}

Requirements:

1. Please output the following infomation, following this structure:
{"**character**":"A string of this game's all information,
"**description**": "A description for this game",
"**reasoning**": "Summarize what types of users would enjoy this game"}
2. Please ensure that the "description" and reasoning are no longer than 150 words.

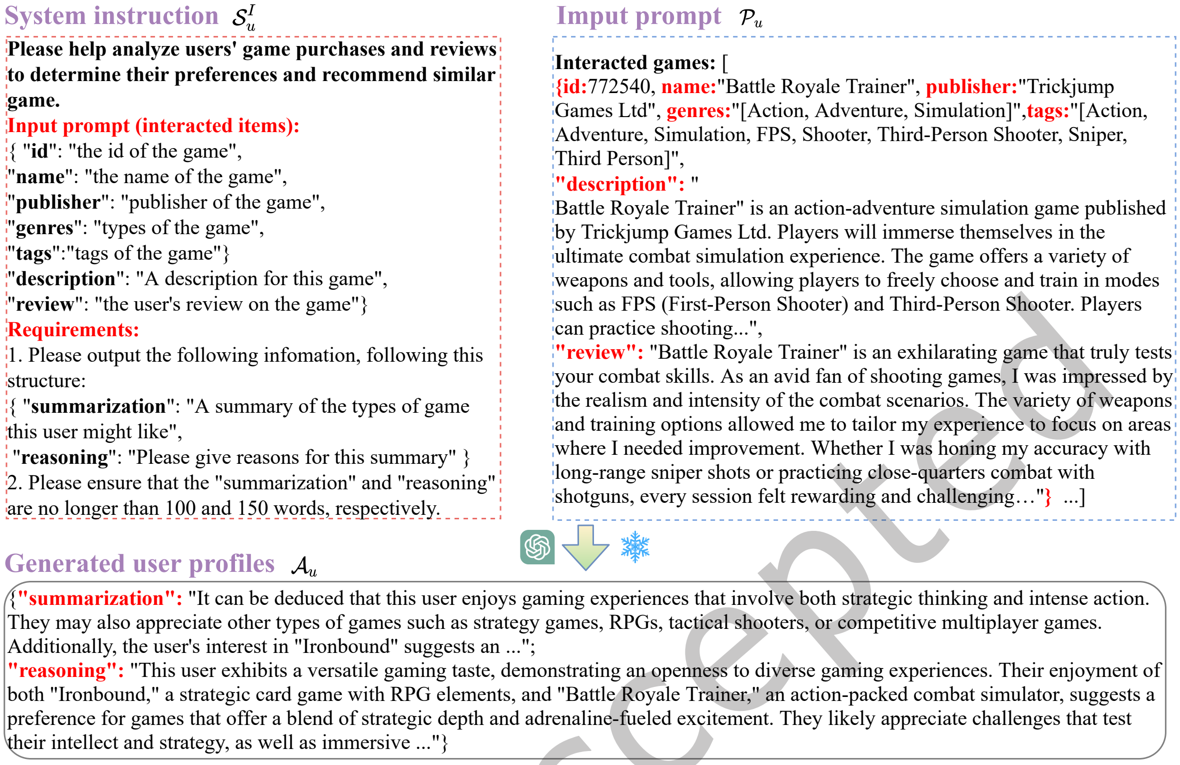
Imput prompt $\mathcal{P}_v$

{**id:**772540,
name:"Battle Royale Trainer",
publisher:"Trickjump Games Ltd",
genres:"[Action, Adventure, Simulation]",
tags:"[Action, Adventure, Simulation, FPS, Shooter, Third-Person Shooter, Sniper, Third Person]"}

Generated item profiles $\mathcal{A}_v$

{**"character":** **id:**772540, **name:**"Battle Royale Trainer", **publisher:**"Trickjump Games Ltd", **genres:**"[Action, Adventure, Simulation]", **tags:**"[Action, Adventure, Simulation, FPS, Shooter, Third-Person Shooter, Sniper, Third Person]";
"description": "Battle Royale Trainer" is an action-adventure simulation game published by Trickjump Games Ltd. Players will immerse themselves in the ultimate combat simulation experience. The game offers a variety of weapons and tools, allowing players to freely choose and train in modes such as FPS (First-Person Shooter) and Third-Person Shooter. Players can practice shooting skills, building techniques, tactical strategies, and more in various combat scenarios to enhance their skills in competitive shooting games. Whether aiming to improve shooting proficiency or stand out in competitive battles, this game meets the needs of players seeking intense combat training.";
"reasoning": "This game would appeal to users who enjoy action-packed gameplay and are looking to improve their skills in shooting and combat scenarios. It caters to players who are interested in competitive shooting games and enjoy the challenge of mastering various weapons and tactics. Additionally, individuals who appreciate realistic combat simulations and want to enhance their reflexes, accuracy, and strategic thinking would also find this game enjoyable."}

Fig. 10. An illustration of item profile generation in Steam data.



System instruction $\mathcal{S}_u^I$

Please help analyze users' game purchases and reviews to determine their preferences and recommend similar game.

Input prompt (interacted items):

{ "**id**": "the id of the game",
"**name**": "the name of the game",
"**publisher**": "publisher of the game",
"**genres**": "types of the game",
"**tags**":"tags of the game"}
"**description**": "A description for this game",
"**review**": "the user's review on the game"}

Requirements:

1. Please output the following infomation, following this structure:
{ "**summarization**": "A summary of the types of game this user might like",
"**reasoning**": "Please give reasons for this summary" }
2. Please ensure that the "summarization" and "reasoning" are no longer than 100 and 150 words, respectively.

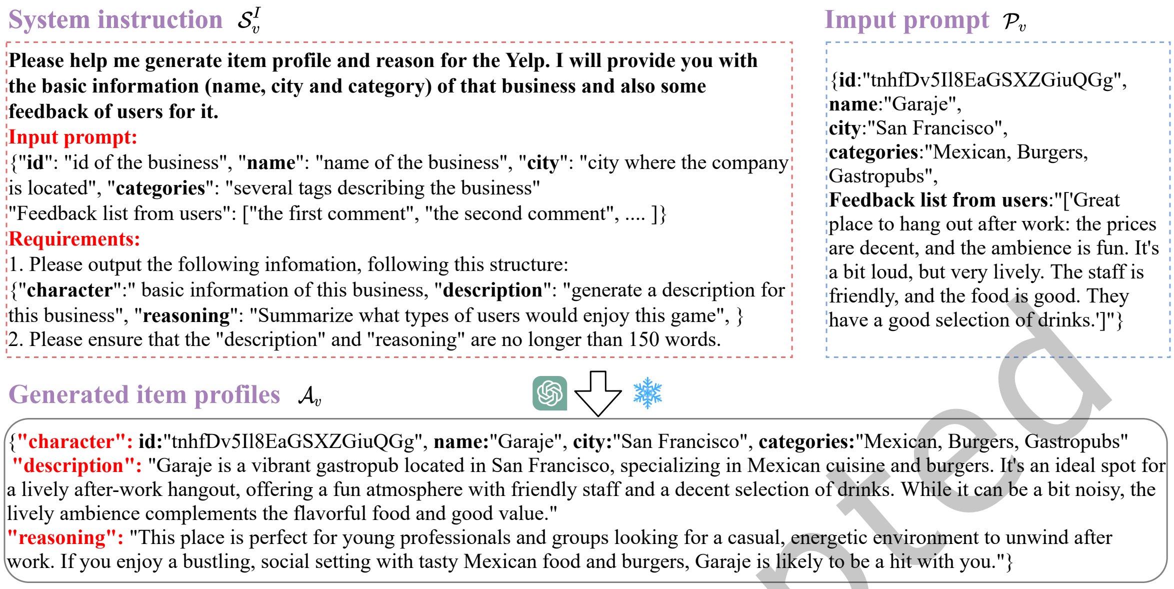
Imput prompt $\mathcal{P}_u$

Interacted games: [
{id:772540, **name:**"Battle Royale Trainer", **publisher:**"Trickjump Games Ltd", **genres:**"[Action, Adventure, Simulation]",**tags:**"[Action, Adventure, Simulation, FPS, Shooter, Third-Person Shooter, Sniper, Third Person]",
"description": "
Battle Royale Trainer" is an action-adventure simulation game published by Trickjump Games Ltd. Players will immerse themselves in the ultimate combat simulation experience. The game offers a variety of weapons and tools, allowing players to freely choose and train in modes such as FPS (First-Person Shooter) and Third-Person Shooter. Players can practice shooting...",
"review": "Battle Royale Trainer" is an exhilarating game that truly tests your combat skills. As an avid fan of shooting games, I was impressed by the realism and intensity of the combat scenarios. The variety of weapons and training options allowed me to tailor my experience to focus on areas where I needed improvement. Whether I was honing my accuracy with long-range sniper shots or practicing close-quarters combat with shotguns, every session felt rewarding and challenging…"**}** ...]

Generated user profiles $\mathcal{A}_u$

{**"summarization":** "It can be deduced that this user enjoys gaming experiences that involve both strategic thinking and intense action. They may also appreciate other types of games such as strategy games, RPGs, tactical shooters, or competitive multiplayer games. Additionally, the user's interest in "Ironbound" suggests an ...";
"reasoning": "This user exhibits a versatile gaming taste, demonstrating an openness to diverse gaming experiences. Their enjoyment of both "Ironbound," a strategic card game with RPG elements, and "Battle Royale Trainer," an action-packed combat simulator, suggests a preference for games that offer a blend of strategic depth and adrenaline-fueled excitement. They likely appreciate challenges that test their intellect and strategy, as well as immersive ..."}

Fig. 11. An illustration of user profile generation in Steam data.



System instruction $\mathcal{S}_v^I$

Please help me generate item profile and reason for the Yelp. I will provide you with the basic information (name, city and category) of that business and also some feedback of users for it.

Input prompt:

{"**id**": "id of the business", "**name**": "name of the business", "**city**": "city where the company is located", "**categories**": "several tags describing the business"
"Feedback list from users": ["the first comment", "the second comment", ...]}

Requirements:

1. Please output the following infomation, following this structure:
{"**character**":" basic information of this business, "**description**": "generate a description for this business", "**reasoning**": "Summarize what types of users would enjoy this game", }
2. Please ensure that the "description" and "reasoning" are no longer than 150 words.

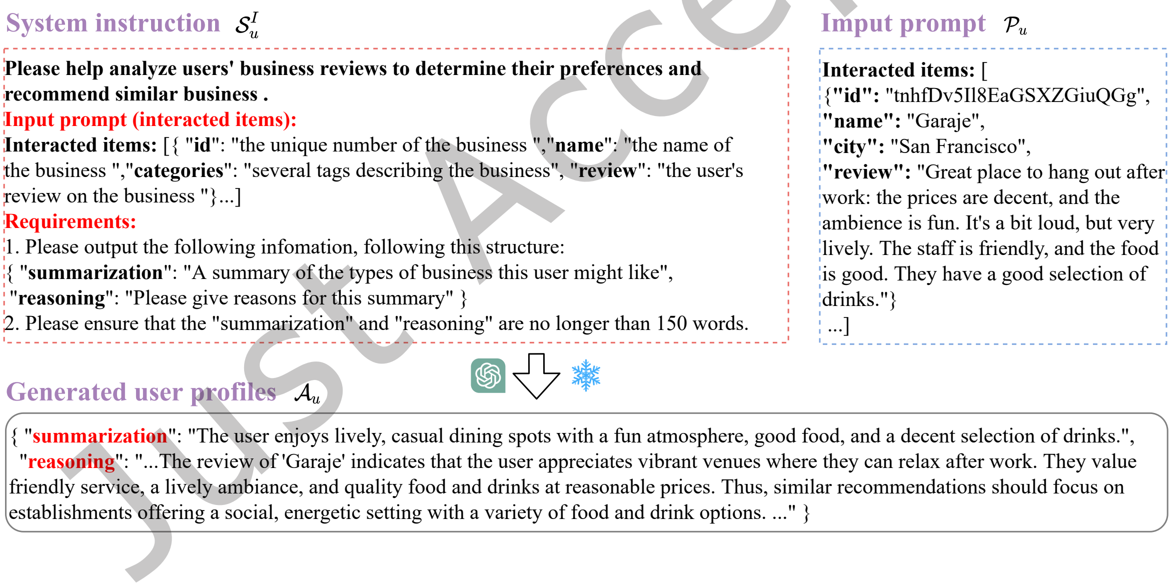
Imput prompt $\mathcal{P}_v$

{**id**:"tnhfDv5Il8EaGSXZGiuQGg",
name:"Garaje",
city:"San Francisco",
categories:"Mexican, Burgers, Gastropubs",
Feedback list from users:"['Great place to hang out after work: the prices are decent, and the ambience is fun. It's a bit loud, but very lively. The staff is friendly, and the food is good. They have a good selection of drinks.']"}

Generated item profiles $\mathcal{A}_v$

{**"character":** **id:**"tnhfDv5Il8EaGSXZGiuQGg", **name:**"Garaje", **city:**"San Francisco", **categories:**"Mexican, Burgers, Gastropubs"
"description": "Garaje is a vibrant gastropub located in San Francisco, specializing in Mexican cuisine and burgers. It's an ideal spot for a lively after-work hangout, offering a fun atmosphere with friendly staff and a decent selection of drinks. While it can be a bit noisy, the lively ambience complements the flavorful food and good value."
"reasoning": "This place is perfect for young professionals and groups looking for a casual, energetic environment to unwind after work. If you enjoy a bustling, social setting with tasty Mexican food and burgers, Garaje is likely to be a hit with you."}

Fig. 12. An illustration of item profile generation in Yelp data.



System instruction $\mathcal{S}_u^I$

Please help analyze users' business reviews to determine their preferences and recommend similar business .

Input prompt (interacted items):

Interacted items: [{ "**id**": "the unique number of the business ","**name**": "the name of the business ","**categories**": "several tags describing the business", "**review**": "the user's review on the business "}...]

Requirements:

1. Please output the following infomation, following this structure:
{ "**summarization**": "A summary of the types of business this user might like",
"**reasoning**": "Please give reasons for this summary" }
2. Please ensure that the "summarization" and "reasoning" are no longer than 150 words.

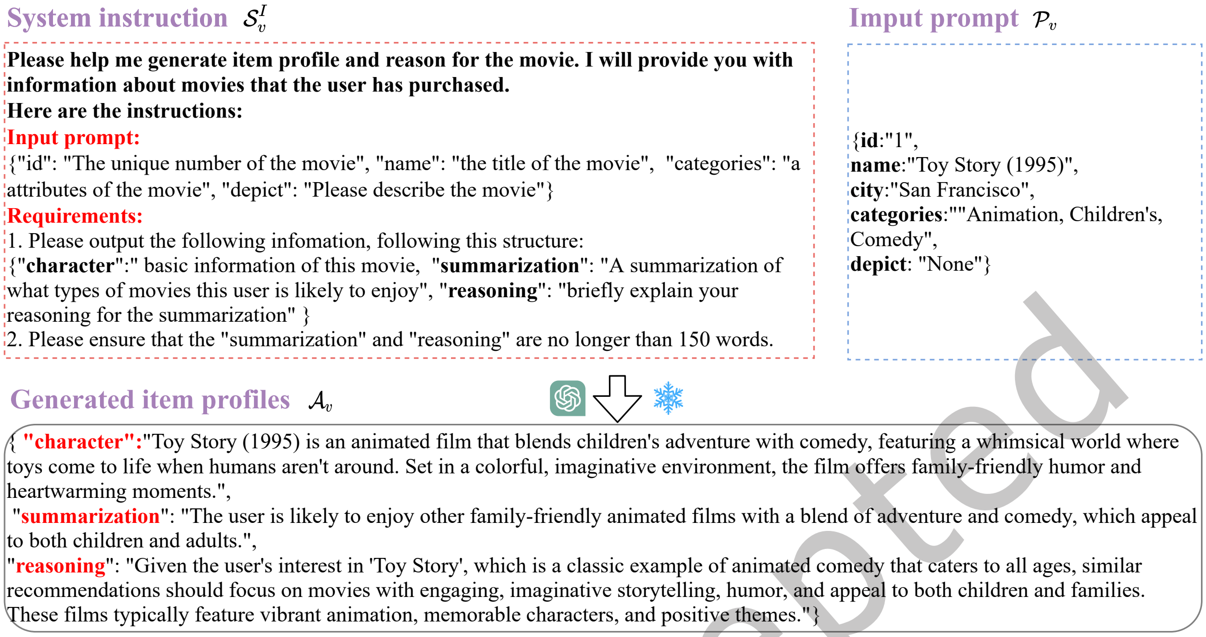
Imput prompt $\mathcal{P}_u$

Interacted items: [
{**"id":** "tnhfDv5Il8EaGSXZGiuQGg",
"name": "Garaje",
"city": "San Francisco",
"review": "Great place to hang out after work: the prices are decent, and the ambience is fun. It's a bit loud, but very lively. The staff is friendly, and the food is good. They have a good selection of drinks."}
...]

Generated user profiles $\mathcal{A}_u$

{ "**summarization**": "The user enjoys lively, casual dining spots with a fun atmosphere, good food, and a decent selection of drinks.",
"**reasoning**": "...The review of 'Garaje' indicates that the user appreciates vibrant venues where they can relax after work. They value friendly service, a lively ambiance, and quality food and drinks at reasonable prices. Thus, similar recommendations should focus on establishments offering a social, energetic setting with a variety of food and drink options. ..." }

Fig. 13. An illustration of user profile generation in Yelp data.



System instruction $\mathcal{S}_v^I$

Please help me generate item profile and reason for the movie. I will provide you with information about movies that the user has purchased.
Here are the instructions:
Input prompt:
{"id": "The unique number of the movie", "name": "the title of the movie", "categories": "a attributes of the movie", "depict": "Please describe the movie"}
Requirements:
1. Please output the following infomation, following this structure:
{"**character**":" basic information of this movie, "**summarization**": "A summarization of what types of movies this user is likely to enjoy", "**reasoning**": "briefly explain your reasoning for the summarization" }
2. Please ensure that the "summarization" and "reasoning" are no longer than 150 words.

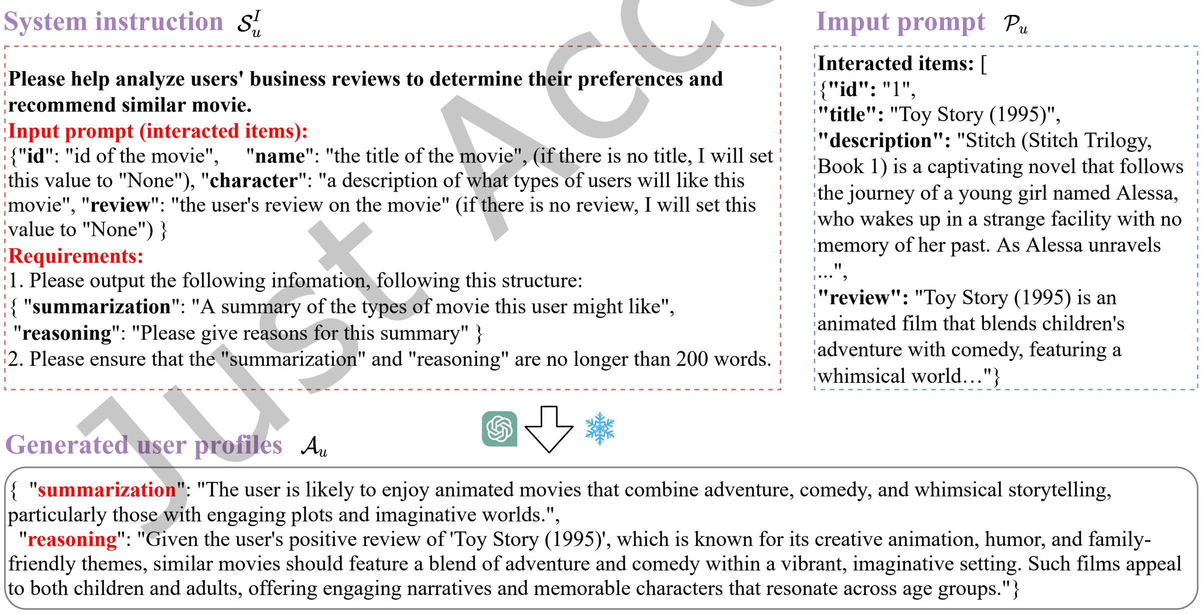
Imput prompt $\mathcal{P}_v$

{**id**:"1",
name:"Toy Story (1995)",
city:"San Francisco",
categories:""Animation, Children's, Comedy",
depict: "None"}

Generated item profiles $\mathcal{A}_v$

{ **"character":**"Toy Story (1995) is an animated film that blends children's adventure with comedy, featuring a whimsical world where toys come to life when humans aren't around. Set in a colorful, imaginative environment, the film offers family-friendly humor and heartwarming moments.",
"**summarization**": "The user is likely to enjoy other family-friendly animated films with a blend of adventure and comedy, which appeal to both children and adults.",
"**reasoning**": "Given the user's interest in 'Toy Story', which is a classic example of animated comedy that caters to all ages, similar recommendations should focus on movies with engaging, imaginative storytelling, humor, and appeal to both children and families. These films typically feature vibrant animation, memorable characters, and positive themes."}

Fig. 14. An illustration of item profile generation in Movie1m data.



System instruction $\mathcal{S}_u^I$

Please help analyze users' business reviews to determine their preferences and recommend similar movie.
Input prompt (interacted items):
{"**id**": "id of the movie", "**name**": "the title of the movie", (if there is no title, I will set this value to "None"), "**character**": "a description of what types of users will like this movie", "**review**": "the user's review on the movie" (if there is no review, I will set this value to "None") }
Requirements:
1. Please output the following infomation, following this structure:
{ "**summarization**": "A summary of the types of movie this user might like",
"**reasoning**": "Please give reasons for this summary" }
2. Please ensure that the "summarization" and "reasoning" are no longer than 200 words.

Imput prompt $\mathcal{P}_u$

Interacted items: [
{**"id":** "1",
"title": "Toy Story (1995)",
"description": "Stitch (Stitch Trilogy, Book 1) is a captivating novel that follows the journey of a young girl named Alessa, who wakes up in a strange facility with no memory of her past. As Alessa unravels …",
"review": "Toy Story (1995) is an animated film that blends children's adventure with comedy, featuring a whimsical world…"}

Generated user profiles $\mathcal{A}_u$

{ "**summarization**": "The user is likely to enjoy animated movies that combine adventure, comedy, and whimsical storytelling, particularly those with engaging plots and imaginative worlds.",
"**reasoning**": "Given the user's positive review of 'Toy Story (1995)', which is known for its creative animation, humor, and family-friendly themes, similar movies should feature a blend of adventure and comedy within a vibrant, imaginative setting. Such films appeal to both children and adults, offering engaging narratives and memorable characters that resonate across age groups."}

Fig. 15. An illustration of user profile generation in Movie1m data.