

Intent Representation Learning with Large Language Model for Recommendation

Yu Wang
Anhui University
Hefei, China
wangyuahu@stu.ahu.edu.cn

Yi Zhang
Anhui University
Hefei, China
zhangyi.ahu@gmail.com

Lei Sang*
Anhui University
Hefei, China
sanglei@ahu.edu.cn

Yiwen Zhang
Anhui University
Hefei, China
zhangyiwen@ahu.edu.cn

Abstract

Intent-based recommender systems have garnered significant attention for uncovering latent fine-grained preferences. Intents, as underlying factors of interactions, are crucial for improving recommendation interpretability. Most methods define intents as learnable parameters updated alongside interactions. However, **existing frameworks often overlook textual information (e.g., user reviews, item descriptions), which is crucial for alleviating the sparsity of interaction intents**. Exploring these multimodal intents, especially the inherent differences in representation spaces, poses two key challenges: **i) How to align multimodal intents and effectively mitigate noise issues; ii) How to extract and match latent key intents across modalities**. To tackle these challenges, we propose a model-agnostic framework, *Intent Representation Learning with Large Language Model* (IRLLRec), which leverages large language models (LLMs) to construct multimodal intents and enhance recommendations. Specifically, IRLLRec employs a dual-tower architecture to learn multimodal intent representations. Next, we propose pairwise and translation alignment to eliminate inter-modal differences and enhance robustness against noisy input features. Finally, to better match textual and interaction-based intents, we employ momentum distillation to perform teacher-student learning on fused intent representations. Empirical evaluations on three datasets show that our IRLLRec framework outperforms baselines¹.

CCS Concepts

• **Information systems** → **Recommender systems**.

Keywords

Recommendation, Collaborative Filtering, Large Language Models, Intent Modeling, Alignment

*Lei Sang is the corresponding author.

¹Code available at <https://github.com/wangyu0627/IRLLRec>

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

SIGIR '25, July 13–18, 2025, Padua, Italy.

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

ACM ISBN 979-8-4007-1592-1/25/07

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

ACM Reference Format:

Yu Wang, Lei Sang, Yi Zhang, and Yiwen Zhang. 2025. Intent Representation Learning with Large Language Model for Recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25)*, July 13–18, 2025, Padua, Italy. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/XXXXXX.XXXXXX>

1 INTRODUCTION

Recommender systems [32] have become indispensable tools in modern life, enabling personalized content delivery across various domains such as short videos [35], news [45], and E-commerce [52]. These systems predominantly rely on collaborative filtering (CF) [14], which infers user preferences from historical interaction data. Inspired by the advantages of graph neural networks (GNNs) [5, 13, 40]) in aggregating higher-order collaborative signals, CF models leverage GNNs and graph contrastive learning (GCL [3, 46, 49]) methods to model high-quality representations for users and items. Although effective, **they largely ignore the latent fine-grained intents between users and items**. In real-world scenarios, user-item interactions are often influenced by multiple intent factors [42, 43]. For instance, users may select skincare products based on specific skin issues like dryness or sensitivity.

Some studies [7, 24, 42] have investigated how intents positively influence recommendation. These works focus on disentangling latent intents from interactions and mapping them to unique feature spaces for modeling. As shown in Figure 1 (a) and (b), we introduce multiple intents between users and items to explain how interactions are driven by intent. For instance, the interaction between user u_2 and item i_1 may be captured by intent c_1 , which subsequently recommends more businesses offering food diversity. KGIN [41] proposes shared intents and leveraged item-side knowledge graphs to capture users' path intents. In contrast, DCCF [31] enhances self-supervised signals by learning disentangled representations with global context. Although disentangled intents have proven effective in recommendation tasks, the rich semantics conveyed by users in unstructured data are often ignored. In E-commerce, for instance, implicit preferences in user reviews or search logs are challenging to capture solely through explicit interaction data [21].

The emergence of large language models (LLMs) like GPT-4 [1] has recently driven major advancements in user profiling and text summarization for recommendation. Some studies [2, 27, 47] use the extensive world knowledge of LLMs to deeply describe

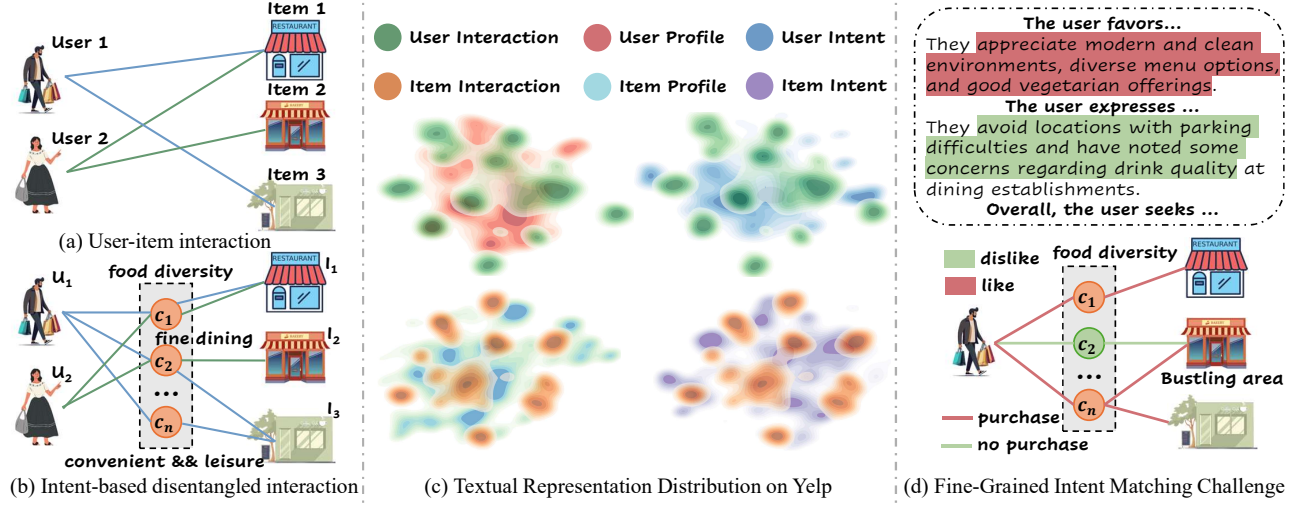


Figure 1: (a) User-item interaction bipartite graph; (b) Disentangled interactions incorporating user intents: u_1 - l_3 is influenced by intents c_2 and c_n , reflecting a preference for businesses offering fine dining, convenience, and leisure; (c) Gaussian kernel density estimation (KDE [38]) visualizes three embedding types: interaction from the pre-trained LightGCN [13], profile from RLMRec [29] extracted attribute summaries, and intent from our chain-of-thought reasoning summaries (Figure 8); (d) The text represents user intents, with red for likes and green for dislikes, and lines indicating interaction or non-interaction.

side information, improving the understanding of complex semantic relationships. For instance, RLMRec [29] focuses on aligning semantic and interaction spaces, while AlphaRec [36] enhances recommendation performance by replacing ID-based embeddings with language embeddings. Moreover, using LLMs to model user and item intents not only captures implicit fine-grained preferences but also provides stronger interpretability. However, **aligning text-based explicit intents with interaction-based implicit intents remains some significant challenges**, as the follow:

- **Cross-modal semantic alignment.** Semantic alignment can bridge the gap between unstructured textual data and structured user-item interactions, effectively capturing fine-grained intents in both modalities to enhance contextual representations of users and items. However, the inherent differences between textual and interaction data in the representation space require complex mapping mechanisms [36] to ensure consistency and semantic fidelity. As illustrated in Figure 1 (c), profile and intent representations overlap with interaction-based representations but differ significantly in distribution. This suggests that intents capture behavioral drivers often missed by fixed profiles in interactions. Moreover, during the alignment process, misaligned representations influenced by noise or sparse interaction data [29] can lead to suboptimal intent fusion.
- **Fine-Grained Intent Matching.** In multimodal data, key information is often embedded in specific text or interactions. Precise matching mechanisms [18] extract relevant needs and behaviors from fragmented signals, capturing fine-grained intents. For example, as shown in Figure 1 (d), a user may express likes (e.g., "diverse menu options") alongside dislikes (e.g., "parking difficulties"). These likes align with interaction intents like c_1 's food

diversity, while dislikes correspond to c_2 's bustling area. Additionally, redundant textual information (e.g., general environment descriptions [33]) and noisy interaction signals (e.g., misclicks or popularity bias [48]) reduce matching accuracy.

To tackle the challenges mentioned above, we propose an Intent Representation Learning with Large Language Model (IRLLRec) framework for recommendation. First, for **Challenge 1**, a dual-tower-based **Intent Alignment (IA)** module is proposed, aiming to bring intents in two distinct representation spaces (text and interaction) closer and to capture fine-grained semantics advantageous for recommendations. IA employs specific encoders to map multimodal intents into a shared space for fusion. It includes two alignment strategies: pairwise alignment enhances consistency by maximizing multimodal mutual information, to mitigate spatial differences. Translation alignment perturbs multimodal representations of individual users and items, improving robustness to input noise features. For **Challenge 2**, we design a **Interaction-text Matching (ITM)** module to extract and match latent key intents for precise preference recommendations. ITM employs momentum distillation [18] to perform teacher-student learning on the fusing intent representations. It also maps the dual-tower encoders into two momentum unimodal encoders to generate pseudo-labels for the teacher model, aiming to identify the optimal matching positions for multimodal intents. Overall, IRLLRec is an LLM-based model-agnostic recommendation framework that considers the potential of multimodal intents to enhance traditional recommendation models and effectively mitigating noise and matching issues in intent alignment. The contributions are summarized as follows:

- We investigate multimodal intent extraction methods and introduce IRLLRec, a dual-tower alignment paradigm that effectively

captures fine-grained semantic information for recommendations. This framework utilizes mutual information maximization and noise perturbation to generate high-quality representations.

- We further propose an Interaction-Text Matching module, which employs momentum distillation for teacher-student learning on fusing intent representations. This approach accurately extracts latent key intents and enables cross-modal matching.
- We conduct extensive experiments on three public datasets, integrating state-of-the-art recommendation methods. IRLRec outperformed existing baselines and significantly enhanced the base models. Furthermore, we analyzed how intent alignment and matching contribute to optimizing recommendation outcomes.

2 PRELIMINARIES

Collaborative Filtering. In a general recommendation model, there exists a set of M users and a set of N items, represented as $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$ and $\mathcal{I} = \{i_1, i_2, \dots, i_N\}$, respectively. According to previous research [13, 49], historical interactions between these users and items are stored in a user-item interaction matrix $\mathbf{R} \in \mathbb{R}^{M \times N}$. The edges, denoted as $\mathcal{E}_{ui} = \mathcal{E}_{iu} = R_{ui}$, correspond to the entries in $\mathcal{G} = \langle \mathcal{V} = \mathcal{U}, \mathcal{I}, \mathcal{E} \rangle$. This formulation frames the recommendation problem as a bilateral graph generation task. The probability of the existence of edges in the user-item interaction graph \mathcal{G} is calculated using the following formula:

$$\mathbb{P}(\hat{R}_{ui} = 1 | \mathbf{e}_u, \mathbf{e}_i) = \sigma(\mathbf{e}_u^\top \mathbf{e}_i), \quad (1)$$

where σ is the sigmoid function. \mathbf{e}_u and \mathbf{e}_i are the embeddings of user u and item i via encoding, respectively.

Text-enhanced Recommendation. Recent works [29, 36] draw inspiration from the powerful summarization capabilities of LLM, leveraging rich textual information such as user reviews and item descriptions in interactions to enhance collaborative filtering. Existing study [44] indicate that system prompts can effectively alleviate LLM hallucinations and improve the quality of its responses. Therefore, the common approach is to design specific prompts $S_{u/v}$ for each user u and item i to obtain summarization responses or profiles from the LLM. This process is represented as follows:

$$\mathcal{P}_u = \text{LLMs}(S_u, Q_u), \quad \mathcal{P}_i = \text{LLMs}(S_i, Q_i), \quad (2)$$

where $Q_{u/i}$ represents user/item profile generation prompts, and $\mathcal{P}_{u/i}$ serves as input for the subsequent stage. Formally, the semantic information about users and items is transformed into features $\mathbf{s}_{u/v} = \text{LLMs}_{emb}(\mathcal{P}_{u/v})$ by the LLM's transformer [37] and integrated into the recommender system.

InfoNCE-based Alignment. In recent years, contrastive learning (CL [6]) has demonstrated significant advantages in cross-modal alignment by bringing the positive samples of two modalities closer in embedding space while pushing the negative samples apart. The core component of CL, the InfoNCE loss [6], has been proven to be a critical means of Mutual Information Maximization (MIM). To mitigate the interference of irrelevant information (noise) on recommendations, some studies [27, 29] typically use it to align two modalities. Formally,

$$\mathcal{L}_{\text{InfoNCE}} = \mathbb{E}_{p(\mathbf{e}, \mathbf{s})} \left[f_{\text{sim}}(\mathbf{e}, \mathbf{s}) - \log \sum_{\mathbf{s}' \in \mathcal{B}} \exp f_{\text{sim}}(\mathbf{e}, \mathbf{s}') \right]. \quad (3)$$

where f_{sim} represents the similarity function, usually the inner product or cosine similarity, and \mathbf{e} , \mathbf{s} , \mathbf{s}' are the embeddings of users, positive samples, and negative samples in a batch \mathcal{B} . The loss optimization guides the similarity of positive sample pairs to be higher than that of negative sample pairs.

3 METHODOLOGY

In this section, we propose an intent-enhanced framework, IRLRec, designed to assist recommendation models capture the fine-grained preferences of users and items hidden in interactions. As shown in the figure 2, we will introduce the model's inputs and the three key modules of IRLRec step by step.

3.1 LLM-based Intent Construction

Textual intent. As illustrated in the Figure 2, constructing textual intent follows a structured multi-step process designed to extract and summarize user and item preferences, encoding them into fine-grained intent representations to improve recommendation.

The first step involves summarizing the intents, where Chain of Thought (CoT [44]) prompts are crafted to incrementally direct the LLM to produce well-defined preferences, ensuring a diverse range of intents. As shown in Figures 7 and 8, we summarize user and item intents as $I_{u/i} = \text{Aggregate}(\{\mathcal{A}_{\text{pos}}, \mathcal{A}_{\text{neg}}\})$ where \mathcal{A}_{pos} and \mathcal{A}_{neg} represent the user's or item's likes and dislikes. The LLM extracts features of each item to obtain I_i , while I_u analyzes the features of items in each interaction to deeply explore the user's intents. The specific forms are as follows:

$$I_u = \text{Aggregate} \left(\bigcup_{i \in I_u} \{\mathcal{A}_{\text{pos}}(i), \mathcal{A}_{\text{neg}}(i)\} \right). \quad (4)$$

where I_u represents the set of items interacted with by user u . "Aggregate" refers to the third step in the CoT prompt, summarizing the extracted information to obtain the final I_u and I_i . Subsequently, following [29], the textual descriptions of these intents are fed into a decoder-only LLM to generate the intent features $\mathbf{x}_u = \text{LLMs}_{emb}(I_u)$ and $\mathbf{x}_i = \text{LLMs}_{emb}(I_i)$ for users and items.

Interaction intent. Finally, we model the interaction-based intents \mathbf{c}_u and \mathbf{c}_i for users and items. User intent refers to the user's needs in a specific context (e.g., preferred brands or movie genres), while item intent refers to the item's context, such as products with a "health" label being suitable for fitness enthusiasts. In this study, we assume that the latent intents \mathbf{c}_u and \mathbf{c}_i follow the distributions $P(\mathbf{c}_u | u)$ and $P(\mathbf{c}_i | i)$, respectively. According to statistical derivation, when the interaction probability $P(y | \mathbf{c}_u, \mathbf{c}_i)$ between users and items is modeled as the joint distribution of latent intent factors, this probability can be expressed as the following:

$$P(y | u, i) = \mathbb{E}_{P(\mathbf{c}_u | u)P(\mathbf{c}_i | i)} [P(y | \mathbf{c}_u, \mathbf{c}_i)], \quad (5)$$

where $P(\mathbf{c}_u | u)$ and $P(\mathbf{c}_i | i)$ describe the intent distributions of users and items, respectively. Furthermore, to simplify the computation, the mathematical expectation of the prediction function $f(\cdot)$ can be approximated as follows:

$$\mathbb{E}_{P(\mathbf{c}_u | u)P(\mathbf{c}_i | i)} [f(\mathbf{c}_u, \mathbf{c}_i)] \approx f \left(\mathbb{E}_{P(\mathbf{c}_u | u)} [\mathbf{c}_u], \mathbb{E}_{P(\mathbf{c}_i | i)} [\mathbf{c}_i] \right). \quad (6)$$

where $\mathbb{E}_{P(\mathbf{c}_u | u)} [\mathbf{c}_u]$ and $\mathbb{E}_{P(\mathbf{c}_i | i)} [\mathbf{c}_i]$ represent the expectations of \mathbf{c}_u and \mathbf{c}_i , respectively. The above formulas provide a theoretical

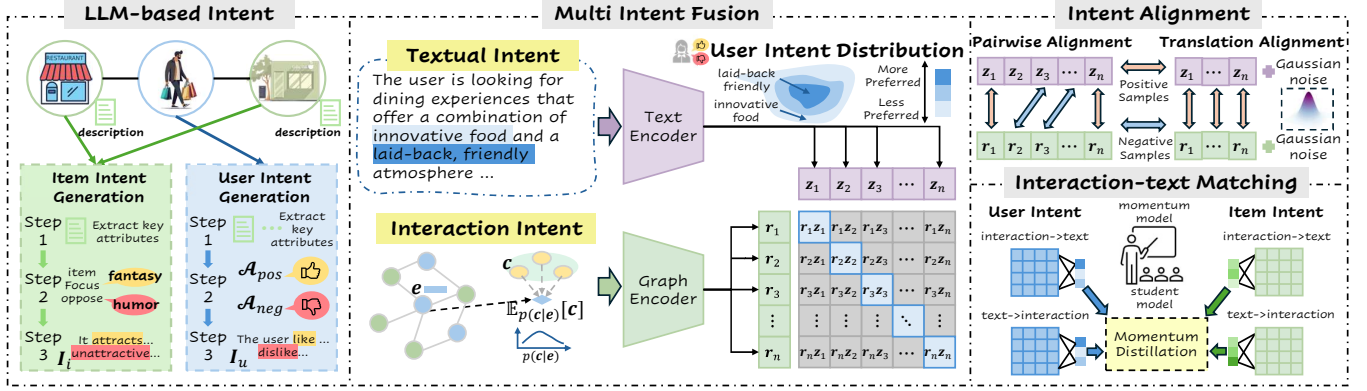


Figure 2: Illustration of IRLRec. Multi Intent Fusion (MIF): MIF takes textual and interaction-based intents as inputs, learning intent embeddings z and r through a dual-tower model and fusing them. **Intent Alignment (IA):** IA bridges spatial discrepancies by aligning two distinct representation spaces. **Interaction-text Matching (ITM):** ITM employs momentum distillation for teacher-student learning, enabling optimal matching of multimodal intents for users and items.

basis for modeling the interaction probability between users and items in recommender systems. Specific mathematical derivations can be found in the references [31, 39].

3.2 Multi Intent Fusion

In this section, we configure two encoders: a text encoder and a graph encoder. Inspired by [36] and [27], we design the text encoder as a linear mapping, which effectively controls computational cost and aligns the textual space with the interaction space. The specific formula is as follows:

$$z = W_2(\sigma(W_1 x + b_1)) + b_2, \quad (7)$$

where σ is the LeakyReLU function, and W and b are the weight matrix and bias vector, corresponding to the linear transformation and bias term, respectively. This mapping has been proven effective in capturing fine-grained preferences in the textual space, laying the foundation for multi-intent alignment. Next, we empirically adopt LightGCN [13] as the graph encoder because the message-passing mechanism of graphs is widely recognized for its advantages in capturing collaborative signals. The representations E_u and E_i for users and items are shared parameters in most recommenders, and our update mechanism is as follows:

$$e_u^{(l)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} e_i^{(l-1)}, e_i^{(l)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}} e_u^{(l-1)}, \quad (8)$$

where l is the encoder layer, and $e_u^{(0)}$ and $e_i^{(0)}$ denote one of the nodes E_u and E_i . \mathcal{N}_u and \mathcal{N}_i respectively represent the first-order receptive fields of users and items.

However, the intents of users and projects are often diverse, making it challenging to fully capture potential preferences with a global representation alone. To address this, we further introduce the concept of interaction intent modeling. By employing the mechanism defined in Eq. 6 and incorporating K intent prototypes, the global representations e_u and e_i are mapped to intent-aware

embeddings r_u and r_i , as shown below:

$$\begin{aligned} r_u^{(l)} &= \mathbb{E}_{P(c_u | e_u^{(l)})} [c_u] = \sum_{k=1}^K c_u^k P(c_u^k | e_u^{(l)}), \\ r_i^{(l)} &= \mathbb{E}_{P(c_i | e_i^{(l)})} [c_i] = \sum_{k=1}^K c_i^k P(c_i^k | e_i^{(l)}), \end{aligned} \quad (9)$$

where $r_u^{(l)}$ represents the intent-aware representation of user u at layer l . The c_u^k and c_i^k are the intent prototypes of the user and item, respectively. To obtain these probability distributions, we need to further clarify the relevance between each user or item and the intent prototypes. The \mathbb{E} of c is defined as follows:

$$\begin{aligned} P(c_u^k | e_u^{(l)}) &= \exp(e_u^{(l)} c_u^k) / \sum_{k'=1}^K \exp(e_u^{(l)} c_u^{k'}), \\ P(c_i^k | e_i^{(l)}) &= \exp(e_i^{(l)} c_i^k) / \sum_{k'=1}^K \exp(e_i^{(l)} c_i^{k'}), \end{aligned} \quad (10)$$

Considering that intents are susceptible to noise in interactions, we employ a graph structure learning (GSL [22, 34]) approach to reconstruct a clean interaction graph and filter intent representations. GSL has been widely applied to various graph tasks and proven effective in alleviating noise issues, summarized as follows: $\mathcal{G}^{(l)} = M^{(l)} \odot \mathcal{G}$, where the mask matrix $M_{ij}^{(l)}$ assigns a weight to each edge in \mathcal{G} . The closer the value is to 0, the less significant the interaction, and vice versa. The formula is as follows:

$$M_{ij}^{(l)} = D^{-1} \cdot \alpha_{ij}, \quad \alpha_{ij} = (s(r_u^{(l)}, r_i^{(l)}) + 1) / 2 \quad (11)$$

where $s(\cdot)$ is the cosine similarity, and normalization is used to obtain the weight α_{ij} within the range $[0, 1]$. D is the degree matrix of the graph $\mathcal{G}^{(l)}$. This method emphasizes the importance of low-degree nodes in the graph (through inverse degree weighting), thereby preventing high-degree nodes (e.g., hub nodes) from excessively dominating the edge weight distribution.

$$R_u^{(l)} = \mathcal{G}^{(l)} \cdot R_u^{(l)}, \quad R_i^{(l)} = \mathcal{G}^{(l)} \cdot R_i^{(l)}. \quad (12)$$

Finally, we perform an averaging operation on the intent representations to obtain $R_u = 1/L \sum_{l=1}^L R_u^{(l)}$ and $R_i = 1/L \sum_{l=1}^L R_i^{(l)}$.

3.3 Intent Alignment

This section is divided into two alignment strategies: pairwise alignment and translation alignment. As illustrated in Figure 2 under “Intent Alignment”, pairwise alignment aims to align the two modalities using interaction-based CF information, while translation achieves the same goal through simple sample shifting.

Pairwise alignment. In our IRLRec, textual intents and interaction intents are treated as two distinct views. Contrastive Learning (CL) achieves this alignment strategy by maximizing representation consistency between positive samples in both views while pushing negative samples apart in the embedding space. For optimizing the representations of the two types of intent embeddings, the proposed objective function is as follows:

$$\mathcal{L}_{\text{pair}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} -\log \frac{\exp(s(\mathbf{r}_i, \mathbf{z}_i) / \tau)}{\sum_{j \in \mathcal{B}} \exp(s(\mathbf{r}_i, \mathbf{z}_j) / \tau)}, \quad (13)$$

In a batch \mathcal{B} , the base model includes embeddings for three types of samples: user, positive, and negative. Therefore, we contrast them separately to obtain the losses $\mathcal{L}_{\text{pair}}^{\text{user}}$, $\mathcal{L}_{\text{pair}}^{\text{pos}}$ and $\mathcal{L}_{\text{pair}}^{\text{neg}}$.

Translation alignment. Given the difficulty in obtaining positive and negative samples in interactions, we employ a positive pair alignment strategy to mitigate this limitation (e.g., only the intents of the same user or item are regarded as positive pairs). To better adapt to the inherent perturbations in input features, we add Gaussian noise [15] to both embeddings to enhance robustness:

$$\begin{aligned} \mathbf{r}' &= \mathbf{r} + \epsilon_r \odot \mathbf{r}, & \epsilon_r &\sim \mathcal{N}(0, \mathbf{I}), \\ \mathbf{z}' &= \mathbf{z} + \epsilon_z \odot \mathbf{z}, & \epsilon_z &\sim \mathcal{N}(0, \mathbf{I}), \end{aligned} \quad (14)$$

where ϵ_r and ϵ_z are auxiliary noise variables sampled from $\mathcal{N}(0, \mathbf{I})$, and each element in $\mathcal{N}(0, \mathbf{I})$ follows a standard normal distribution. As shown in the Figure 2, we construct the diagonal elements as pairs of positive samples in set \mathbb{R} , while the other elements constitute negative samples in set \mathbb{N} . Therefore, in the text view, the contrastive loss is defined as follows:

$$\mathcal{L}_{\text{tran}}^{\text{text}} = -\log \frac{\sum_{i \in \mathbb{R}} \exp(\eta \cdot \mathbf{r}'_i \cdot \mathbf{z}'_i^\top)}{\sum_{j \in \mathbb{N}} \exp(\eta \cdot \mathbf{r}'_i \cdot \mathbf{z}'_j^\top)} \quad (15)$$

the η represents the scaling factor, used to adjust the similarity intensity during the alignment process. $\mathcal{L}_{\text{tran}}^{\text{text}}$ represents the text-interaction loss. Similar to the text view, the interaction view also has a loss $\mathcal{L}_{\text{tran}}^{\text{inter}}$, derived from the transposed similarity matrix. Finally, the overall loss is as follows:

$$\mathcal{L}_{\text{IA}} = \lambda_1 (\mathcal{L}_{\text{pair}}^{\text{user}} + \mathcal{L}_{\text{pair}}^{\text{pos}} + \mathcal{L}_{\text{pair}}^{\text{neg}}) + \lambda_2 (\mathcal{L}_{\text{tran}}^{\text{text}} + \mathcal{L}_{\text{tran}}^{\text{inter}}). \quad (16)$$

where λ_1 and λ_2 represent the adjustable weights for two alignment strategies, respectively.

3.4 Interaction-text Matching

The information about interaction intents is often contained in certain parts of the text, so they tend to be noisy. The alignment process is typically weakly correlated, as the text may contain words unrelated to the true intents driving the interactions or intents not reflected in the interactions. To address this issue, we propose Interaction-text Matching (ITM), which is used to determine whether a pair of intents is matched or unmatched. ITM is a momentum model, serving as an evolving teacher model.

Inspired by [12], we maintain two momentum unimodal encoder g'_t (g_t is the text encoder) and g'_i (g_i is the graph encoder) for obtaining pseudo-labels for the teacher model. Specifically, we use g'_t and g'_i to calculate the representation similarities $s'(T, I) = g'_t(\mathbf{x})^\top g'_i(\mathbf{e})$ and $s'(I, T) = s'(T, I)^\top$, where T and I denote the textual and interaction perspectives, respectively. Next, we represent the student model's results $s(T, I) = \mathbf{z}^\top \mathbf{r}$ and $s(I, T) = s(T, I)^\top$ as \mathbf{p}^{t2i} and \mathbf{p}^{i2t} , separately. Similarly, $s'(T, I)$ and $s'(I, T)$ denote \mathbf{q}^{t2i} and \mathbf{q}^{i2t} . Formally, the ITM loss is defined as:

$$\begin{aligned} \mathcal{L}_{\text{ITM}} &= (1 - \alpha) \mathcal{L}_{\text{tran}} + \alpha \mathbb{E}_{(I, T) \sim \mathcal{D}} \left[\text{KL}(\mathbf{q}^{i2t}(I) \parallel \mathbf{p}^{i2t}(I)) \right. \\ &\quad \left. + \text{KL}(\mathbf{q}^{t2i}(T) \parallel \mathbf{p}^{t2i}(T)) \right], \end{aligned} \quad (17)$$

where α is the balancing coefficient between the teacher and student models, and $i2t$ represents the interaction-to-text matching task, while $t2i$ is similar. To simplify, we test and set the weight is set to 0.4. $\text{KL}(\parallel)$ is employed to quantify the difference between two distributions. After each training iteration, the momentum model's parameters are updated using Exponential Moving Average (EMA) [18], smoothly transferring the student model's parameters to the teacher model to enhance the teacher model's stability:

$$\theta_{\text{teacher}} = \beta \cdot \theta_{\text{teacher}} + (1 - \beta) \cdot \theta_{\text{student}}. \quad (18)$$

where β is the momentum coefficient (e.g., 0.999). The momentum model is typically used to generate pseudo-labels, filtering out high-frequency noise in the student model and allowing the momentum model to capture more stable features.

3.5 Model Analysis

Multi-task Training. Finally, we employ multi-task joint learning to complete the framework's optimization process, and the optimization objective of IRLRec is as follows:

$$\mathcal{L}_{\text{IRLLRec}} = \mathcal{L}_{\text{IA}} + \lambda_3 \mathcal{L}_{\text{ITM}} + \|\Theta\|_2^2. \quad (19)$$

where $\|\Theta\|_2^2$ are trainable model parameters and L_2 regularization.

Time Complexity Analysis. We analyze the batch time complexity for this model-agnostic framework, excluding graph convolution, which is common to all base models. Let d_t denote the dimension of the text representation and d the dimension of user and item embeddings. The time complexity of the text encoder MLP encoding is $O((M + N)d_t d)$, where M and N represent the number of users and items, respectively. Next, interaction intent encoder has a complexity of $O((M + N)LKd)$, where K is the number of latent intents and L is the number of encoder layers. Additionally, the intent interaction reconstruction complexity is $O(|\mathcal{G}|Ld)$, which is used to remove noise from the intent. Finally, the complexity of the loss functions for IA and ITM are $O(2\mathcal{B}d + (M + N)^2 d)$ and $O(M^2 + N^2)$, respectively, where \mathcal{B} is a sampled batch. Momentum distillation doubles the complexity of the encoding process.

Table 1: Statistics of the experimental datasets.

Dataset	#Users	#Items	#Interactions	Density
Amazon-book	11,000	9,332	200,860	2.0×10^{-3}
Yelp	11,091	11,010	277,535	2.3×10^{-3}
Amazon-movie	16,994	9,370	168,243	1.1×10^{-3}

Table 2: Recommendation performance Imprvment of all backbone methods on different datasets in terms of Recall and NDCG. The superscript * indicates the Imprvment is statistically significant where the p-value is less than 0.05.

Dataset	Amazon-book						Yelp						Amazon-movie					
Metric	Recall			NDCG			Recall			NDCG			Recall			NDCG		
Model	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20	@5	@10	@20
Semantic Only	0.0081	0.0125	0.0199	0.0072	0.0088	0.0112	0.0013	0.0022	0.0047	0.0014	0.0018	0.0026	0.0029	0.0084	0.0117	0.0022	0.0069	0.0098
AlphaRec	0.0598	0.0941	0.1412	0.0605	0.0721	0.0873	0.0431	0.0726	0.1212	0.0493	0.0586	0.0752	0.0909	0.1213	0.1661	0.0706	0.0827	0.0958
LightGCN	0.0570	0.0915	0.1411	0.0574	0.0694	0.0856	0.0421	0.0706	0.1157	0.0491	0.0580	0.0733	0.0796	0.1193	0.1661	0.0623	0.0761	0.0891
KAR	0.0596	0.0934	0.1416	0.0590	0.0705	0.0860	0.0437	0.0740	0.1194	0.0506	0.0602	0.0756	0.0824	0.1195	0.1653	0.0645	0.0775	0.0905
RLMRec-Con	0.0608	0.0969	0.1483	0.0606	0.0734	0.0903	0.0445	0.0754	0.1230	0.0518	0.0614	0.0776	0.0834	0.1193	0.1697	0.0652	0.0778	0.0918
RLMRec-Gen	0.0596	0.0948	0.1446	0.0605	0.0724	0.0887	0.0435	0.0734	0.1209	0.0505	0.0600	0.0761	0.0823	0.1185	0.1705	0.0643	0.0770	0.0921
IRLLRec	0.0643*	0.1009*	0.1538*	0.0638*	0.0765*	0.0938*	0.0460*	0.0781*	0.1278*	0.0542*	0.0642*	0.0810*	0.0950*	0.1341*	0.1855*	0.0752*	0.0888*	0.1032*
Improv.	5.72%	4.13%	3.71%	5.21%	4.28%	3.85%	3.39%	3.63%	3.90%	4.57%	4.55%	4.39%	13.96%	12.40%	8.82%	15.37%	14.11%	12.03%
SGL	0.0637	0.0994	0.1473	0.0632	0.0756	0.0913	0.0432	0.0722	0.1197	0.0501	0.0592	0.0753	0.0917	0.1239	0.1658	0.0722	0.0835	0.0953
KAR	0.0595	0.0941	0.1435	0.0596	0.0713	0.0875	0.0441	0.0735	0.1200	0.0510	0.0599	0.0757	0.0889	0.1230	0.1647	0.0714	0.0838	0.0958
RLMRec-Con	0.0655	0.1017	0.1528	0.0652	0.0778	0.0945	0.0452	0.0763	0.1248	0.0530	0.0626	0.0790	0.0916	0.1275	0.1747	0.0722	0.0847	0.0978
RLMRec-Gen	0.0644	0.1015	0.1537	0.0648	0.0777	0.0947	0.0467	0.0771	0.1263	0.0537	0.0631	0.0798	0.0892	0.1292	0.1760	0.0701	0.0841	0.0972
IRLLRec	0.0671*	0.1031*	0.1539*	0.0680*	0.0803*	0.0964*	<u>0.0465</u>	0.0784*	0.1275*	0.0541	0.0640*	0.0805	0.0963*	0.1349*	0.1857*	0.0757*	0.0892*	0.1035*
Improv.	2.50%	1.40%	0.14%	4.33%	3.16%	1.75%	-0.43%	1.66%	0.95%	0.74%	1.36%	0.88%	5.15%	4.46%	5.52%	4.86%	5.42%	5.82%
SimGCL	0.0618	0.0992	0.1512	0.0619	0.0749	0.0919	0.0467	0.0772	0.1254	0.0546	0.0638	0.0801	0.0925	0.1276	0.1700	0.0738	0.0861	0.0978
KAR	0.0623	0.1003	0.1520	0.0633	0.0756	0.0924	0.0466	0.0768	0.1265	0.0538	0.0636	0.0804	0.0930	0.1285	0.1685	0.0754	0.0876	0.1005
RLMRec-Con	0.0633	0.1011	0.1552	0.0633	0.0765	0.0942	0.0470	0.0784	0.1292	0.0546	0.0642	0.0814	0.0979	0.1323	0.1763	0.0775	0.0896	0.1020
RLMRec-Gen	0.0617	0.0991	0.1525	0.0622	0.0752	0.0925	0.0464	0.0767	0.1267	0.0541	0.0634	0.0803	0.0969	0.1324	0.1781	0.0780	0.0902	0.1034
IRLLRec	0.0669*	0.1057*	0.1575*	0.0666*	0.0802*	0.0970*	0.0493*	0.0818*	0.1328*	0.0571*	0.0671*	0.0844*	0.1011*	0.1374*	0.1863*	0.0817*	0.0946*	0.1082*
Improv.	5.66%	4.51%	1.46%	5.21%	4.84%	2.95%	4.81%	4.31%	2.77%	4.54%	4.52%	3.71%	3.21%	3.81%	4.63%	4.72%	4.83%	4.62%
DCCF	0.0662	0.1019	0.1517	0.0658	0.0780	0.0943	0.0468	0.0778	0.1249	0.0543	0.0640	0.0800	0.0965	0.1317	0.1723	0.0763	0.0905	0.1014
KAR	0.0665	0.1022	0.1521	0.0657	0.0785	0.0949	0.0465	0.0784	0.1257	0.0539	0.0642	0.0808	0.0977	0.1334	0.1750	0.0778	0.0914	0.1029
RLMRec-Con	0.0665	0.1040	0.1563	0.0668	0.0798	0.0968	<u>0.0486</u>	<u>0.0813</u>	<u>0.1321</u>	<u>0.0561</u>	<u>0.0663</u>	<u>0.0836</u>	<u>0.0987</u>	<u>0.1349</u>	<u>0.1790</u>	<u>0.0796</u>	<u>0.0921</u>	<u>0.1048</u>
RLMRec-Gen	0.0666	0.1046	0.1559	0.0670	0.0801	0.0969	0.0475	0.0785	0.1281	0.0549	0.0646	0.0815	0.0980	0.1349	0.1778	0.0792	0.0918	0.1041
IRLLRec	0.0702*	0.1076*	0.1590*	0.0690*	0.0822*	0.0992*	0.0506*	0.0843*	0.1360*	0.0581*	0.0689*	0.0864*	0.1012*	0.1374*	0.1826*	0.0812*	0.0939*	0.1066*
Improv.	5.41%	2.87%	1.73%	2.96%	2.60%	2.41%	4.04%	3.69%	2.96%	3.56%	3.91%	3.35%	2.53%	1.82%	2.00%	2.06%	1.98%	1.72%
BIGCF	0.0662	0.1028	0.1552	0.0658	0.0784	0.0955	0.0458	0.0758	0.1237	0.0536	0.0627	0.0789	0.0947	0.1330	0.1793	0.0759	0.0893	0.1022
KAR	0.0670	0.1034	0.1559	0.0664	0.0792	0.0961	0.0446	0.0763	0.1245	0.0533	0.0630	0.0799	0.0952	0.1342	0.1808	0.0766	0.0904	0.1030
RLMRec-Con	0.0685	0.1052	0.1574	0.0680	0.0807	0.0977	0.0470	0.0784	0.1284	0.0548	0.0646	0.0815	0.0987	0.1373	0.1845	0.0779	0.0914	0.1046
RLMRec-Gen	0.0676	0.1044	0.1555	0.0654	0.0796	0.0965	0.0467	0.0764	0.1259	0.0545	0.0635	0.0802	0.0954	0.1339	0.1814	0.0765	0.0899	0.1031
IRLLRec	0.0693*	0.1082*	0.1631*	0.0694*	0.0831*	0.1010*	0.0495*	0.0823*	0.1342*	0.0576*	0.0678*	0.0854*	0.1048*	0.1431*	0.1916*	0.0845*	0.0979*	0.1115*
Improv.	1.23%	2.83%	3.63%	2.09%	2.90%	3.32%	5.32%	4.97%	4.52%	5.11%	4.95%	4.79%	6.16%	4.20%	3.87%	8.44%	7.11%	6.62%

4 EXPERIMENTS

In this section, we conduct extensive experiments and answer the following research questions:

- **RQ1:** How does IRLRec compare to the current state-of-the-art (SOTA) framework in terms of performance?
- **RQ2:** Are the key components in our IRLRec delivering the expected performance gains?
- **RQ3:** What are the reasons for IRLRec’s superior performance?
- **RQ4:** How do different hyperparameters affect IRLRec?

4.1 Experiment Setup

4.1.1 Datasets. The performance of IRLRec has been validated on three real-world datasets. A statistical overview of all the datasets is presented in Table 1. **Amazon-book**[29] contains users’ book purchase, rating, and review records from Amazon. **Yelp**² covers users’ reviews and rating information for local businesses. **Amazon-movie**³ provides users’ movie-watching records, ratings, and review data from Amazon. The data processing follows the previous work [14, 29], where interactions with ratings below 2 in Amazon-book, below 3 in Yelp, and below 3 in Amazon-movie are filtered out. Next, we perform k-core filtering. Amazon-book and Yelp datasets are split into training, validation, and test sets in a 3:1:1 ratio, while Amazon-movie is split in a ratio of 8:1:1.

²<https://business.yelp.com/data/resources/open-dataset/>

³<https://amazon-reviews-2023.github.io/>

4.1.2 Baselines and Base Models. Our IRLRec employs an intent alignment component to enhance representation-based recommenders. Therefore, we selected representative models as base models. To evaluate the effectiveness of the method, we selected two state-of-the-art methods as baselines.

Baselines.

- **KAR** [47] introduces external knowledge of user preferences and factual knowledge about items, which are compressed into vectors to improve recommendations.
- **RLMRec** [29] proposes a framework that leverages the representation learning empowered by LLMs and designs contrastive and generative alignment methods to enhance recommendations.
- **AlphaRec** [36] replaces ID-based embeddings with language embeddings and combines GCN and CL to achieve a simple yet effective recommendation paradigm.

Base Models.

- **LightGCN** [13] removes the feature transformation and nonlinear activations of GCN to achieve light recommendation.
- **SGL** [46] generates contrast views by edge dropout to aid contrastive learning to enhance recommendation.
- **SimGCL** [49] considers the relationship between neighbor nodes to enhance collaborative filtering.
- **DCCF** [31] enhances self-supervised signals by learning disentangled representations with global context.
- **BIGCF** [51] explores the individuality and collectivity of intents behind interactions for collaborative filtering.

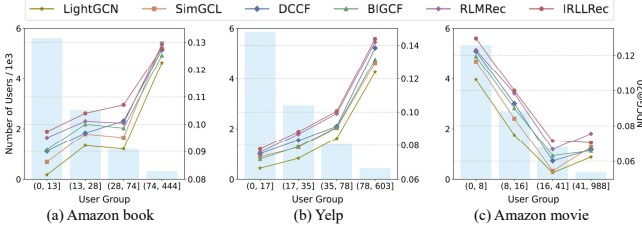


Figure 3: Performance comparison of different sparsity levels. The bar graph shows users' number per group on the left y-axis, and the line graph shows the performance of each method w.r.t. NDCG@20 on the right y-axis.

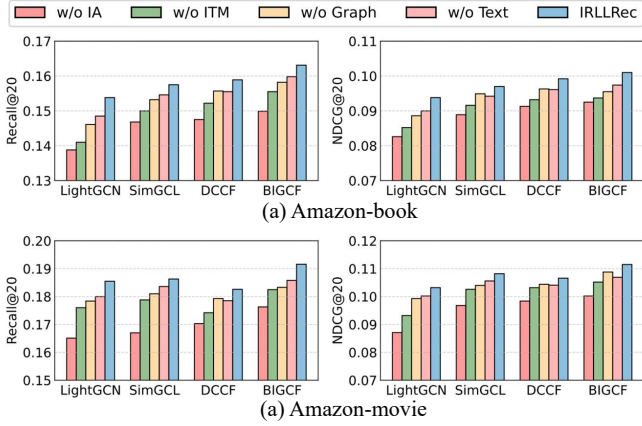


Figure 4: Ablation studies of model variants on the Amazon book and movie datasets w.r.t. Recall@20 and NDCG@20.

4.1.3 Implementations. To ensure fair comparison, we adopted the sampling method and dataset format from the open-source framework SSLRec [30]. In the experimental setup, the embedding dimension of the models and the transformed LLM embeddings were uniformly set to 32. All models, including baselines and base models, were trained using the Adam optimizer with Xavier [11] initialization for embeddings, a fixed batch size of 4096, and a learning rate of $1e-3$. Additionally, early stopping was applied based on the model's performance on the validation set to prevent overfitting. For textual intent generation, we used OpenAI's GPT-4o-mini for intent generation and text-embeddings-3-large [26] for semantic representation to ensure accuracy in intent expression and consistency in semantic representation.

4.2 Performance Comparisons(RQ1)

4.2.1 Comparisons w.r.t. Overall Performance. Table 2 presents the improvement in recommendation performance of the intent-based, model-agnostic framework across three public datasets. We will discuss notable phenomena and provide potential explanations. The experimental results are averaged over 5 runs.

- Compared to all baselines, IRLRec shows significant improvements across all metrics (5.72% and 15.37% for Amazon book and movie, respectively, and 5.32% for Yelp). The only exception is the

recall@5 metric on Yelp, which is 0.43% lower than RLMRec-Gen. We believe this is due to the random perturbation of the graph structure in SGL [46], which disrupts intent propagation and shifts the alignment direction towards noise, yet still highlights the framework's superiority and rationale.

- Both KAR [47] and RLMRec [29] are LLM-enhanced methods that improve base models by generating textual user/item descriptions. However, KAR is less stable than RLMRec due to its lack of integration between textual knowledge and user behavior, making it more susceptible to noise. Our IRLRec addresses this limitation effectively through intent alignment and enhance recommendations by aligning fine-grained multimodal intents.
- The AlphaRec's poor performance may stem from our adherence to fairness, where we use RLMRec's profile representation as the item embedding instead of the proposed item title.

4.2.2 Comparisons w.r.t. Data Sparsity. A common challenge in existing models is data sparsity, where users with few interactions dominate. As shown in Figure 3, we divide users into four groups based on interaction count across three datasets, ensuring roughly equal total interactions per group. For instance, in the Amazon book dataset, the first group includes 5612 users with no more than 13 interactions, meaning over 50% of users in the test set have limited interaction records. We observe that as interaction scale increases, the model generally makes more accurate recommendations (i.e., higher NDCG value), suggesting that 'cold-start' (not strictly defined) users are a key limitation in recommendations. Compared to RLMRec, IRLRec improves performance in the sparsest user group by 2.31%, 2.51%, and 5.20%, respectively, demonstrating that multimodal intent better captures the preferences of these 'cold-start' users, leading to more accurate recommendations.

4.3 Ablation Study (RQ2)

4.3.1 Impact of key components. In this section, we assess the impact of key components in IRLRec and provide potential explanations for the results. The following model variants are evaluated:

- IRLRec w/o IA: Removes the intent alignment module;
- IRLRec w/o ITM: Removes the interaction-text matching;
- IRLRec w/o Graph: Replaces GCN [13] operation on interaction intent embeddings with user and item embeddings;
- IRLRec w/o Text: Replaces MLP mapping of text intent embeddings with direct alignment.

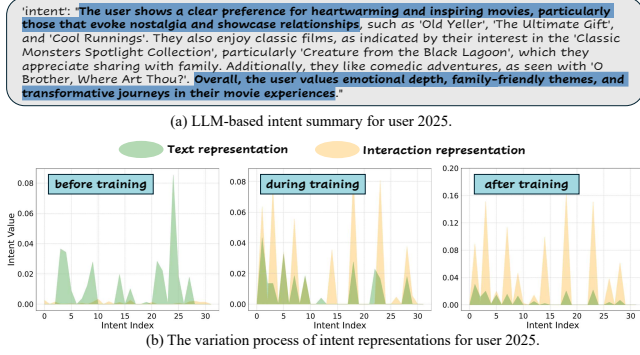
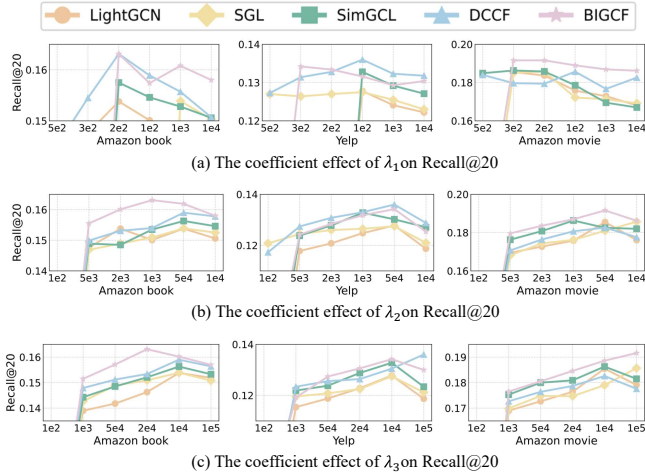
Figure 4 presents the experimental results on the Amazon book and movie datasets, leading to the following insights:

Experimental results demonstrate that IA contributes the most to recommendation performance, with its absence often leading to worse outcomes than the base model. This highlights that the a single modality's intent cannot capture fine-grained preferences; instead, the two intents must be brought closer together in their spatial distance while filtering noise [27]. In contrast, the ITM module ranks second in importance, consistently outperforming the base model due to its momentum distillation mechanism, which effectively aligns the two types of intents.

Subsequently, we further decompose the framework into a text encoder and a graph encoder to evaluate the necessity of the dual-tower architecture. Notably, removing the graph encoder typically

Table 3: Ablation studies of different semantic representations on Amazon-movie datasets w.r.t. Recall@20 and NDCG@20.

Dataset	Variants	LightGCN		SGL		SimGCL		DCCF		BIGCF	
		R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20
Amazon Movie	Llama3-8B-Instruct [8]	0.1816	0.1005	0.1804	0.1003	0.1806	0.1036	0.1781	0.1025	0.1859	0.1074
	text-embedding-ada-002 [26]	0.1837	0.1026	0.1835	0.1029	0.1844	0.1070	0.1805	0.1059	0.1898	0.1102
	text-embeddings-3-large [26]	0.1855	0.1032	0.1857	0.1035	0.1863	0.1082	0.1826	0.1066	0.1916	0.1115
	SFR-Embedding-Mistral [25]	0.1864	0.1040	0.1866	0.1043	0.1874	0.1084	0.1832	0.1076	0.1935	0.1123

**Figure 5: Case study on Amazon movie.****Figure 6: Performance comparison w.r.t. loss coefficients applied to the base model by IRLRec.**

results in poorer performance. RLMRec seeks to align user representations with profile representations, highlighting that text-based intent representation goes beyond a simple user profile, capturing fine-grained preferences. Overall, substituting either encoder leads to a loss of IA's information, emphasizing the advantage of the dual-tower architecture.

4.3.2 Impact of different semantic representations. To assess the impact of text embedding models, we tested additional transformation models (e.g., Llama3 and GPT) to evaluate the effectiveness of LLMs for intent summarization. As shown in Table 3, despite

Llama3's larger embedding dimension (4096), its performance is the worst. This is due to the 8B model's insufficient parameter count, limiting its ability to capture fine-grained user preferences. In contrast, SFR-Embedding-Mistral, the latest model, achieved the best recommendation performance, demonstrating significant progress in LLM-based semantic transformation. Overall, both locally deployable and closed-source LMs significantly enhance the base model, highlighting the importance of intent summarization.

4.4 Case Study (RQ3)

We investigate the impact of interaction-text matching on intent alignment. Figure 5 (b) shows a case study where we split the 32-dimensional multimodal intent representation of user 2025 into individual values by index. Before training, the matching score (vector dot product) between the two representations was -0.0005, during training it increased to 0.0270, and after training, it reached 0.0399. Initially, the interaction-based intent representation was indistinguishable, but over time, it developed clearer preferences, and the bias in the multimodal intent representation decreased. Notably, the trained multimodal intent focuses on the head and tail, as shown in the blue sections of Figure 5 (a). This demonstrates that IRLRec effectively aligns interaction-driven intents with LLM-summarized intents, validating Challenge 2.

4.5 Hyperparameter Sensitivity (RQ4)

In this study, we keep the parameters of RLMRec fixed while analyzing the loss coefficients of the IA and ITM components, corresponding to λ_1 and λ_2 in Eq. 16, and λ_3 in Eq. 19.

- As shown in Figure 6 (a), pairwise alignment in IA plays a key role, with optimal performance typically achieved between 0.01 and 0.03. Truncated cases in the figure are due to excessive non-primary losses that hinder model convergence and are not displayed. This suggests that alignment is the primary source of performance gains, with interaction-based alignment effectively integrating user behavior and semantic information.
- Translation alignment and interaction-text matching in IA exhibit clear trends in Figure 6 (b) and (c). For instance, recall typically peaks as λ_2 decreases, then gradually declines, with a similar pattern for λ_3 . This indicates that these two components must maintain a delicate balance with pairwise alignment, where combining alignment and matching effectively addresses the noise issue in multimodal intents.

5 RELATED WORK

5.1 LLMs for Recommendation

Large language models (LLMs) have gained significant attention in recommender systems for their advanced language understanding and reasoning abilities. Research in this area primarily follows three paradigms: LLMs as recommenders, enhancers, and encoders. LLMs as recommenders [21, 50]: These methods use user interaction histories as prompts to guide LLMs in selecting recommendation targets from a candidate set. TallRec [2] fine-tunes Llama on constructed recommendation datasets, enhancing LLM decision-making in recommendations. RosePO [20] employs smoothing personalized preference optimization to fine-tune LLMs, improving performance while ensuring the recommender remains "helpful and harmless". LLMs as enhancers [16, 27]: RLMRec [29] introduces a framework leveraging LLM-driven representation learning, with contrastive and generative alignment methods to improve recommendations. AlphaRec [36] replaces ID-based embeddings with language embeddings and combines GCN and CL for a simple yet effective recommendation approach. LLMs as encoders [10, 17]: EasyRec [28] leverages collaborative information and textual data from users and items to retrain language models for recommendation, achieving impressive performance in zero-shot scenarios. Despite their impact in respective fields, they overlook the potential of LLM-based intents to enhance interpretability.

5.2 Disentanglement-based Recommendation

Disentanglement-based methods generally focus on modeling user-item interactions by projecting them into distinct feature spaces [9, 19]. For instance, MacridVAE [24] leverages variational autoencoders to encode various user intents [23]. DGCF [42] employs graph neural networks to learn disentangled user representations. DisenHAN [43] utilizes meta-relation decomposition along with disentangled propagation layers to capture semantic meanings. In the case of CDR [4], a dynamic routing mechanism is developed to characterize the correlations among user intents for embedding denoising. KGIN [41] introduces the concept of shared intents and uses an item-side knowledge graph to capture user's path-based intents. Some innovative approaches have started integrating contrastive learning into intent modeling, such as ICLRec [7], DCCF [31], and BIGCF [51]. DCCF [31] enhances self-supervised signals by learning disentangled representations with a global context, while BIGCF investigates the individuality and collectivity of intents behind interactions for collaborative filtering. While effective, multimodal intents present a promising avenue for exploration.

6 CONCLUSION

The paper leveraged LLMs to infer the semantic intents of users and items, and employed joint distribution and graph convolution to capture interaction intents. We proposed a model-agnostic framework, IRLRec, incorporated a dual-tower model where textual and interaction intents were processed by separate encoders. We applied pairwise alignment to capture common features between the two intent types, reduced spatial differences, while translation alignment disrupted multimodal representations to enhance robustness against input noise. Additionally, we introduced an interaction-text

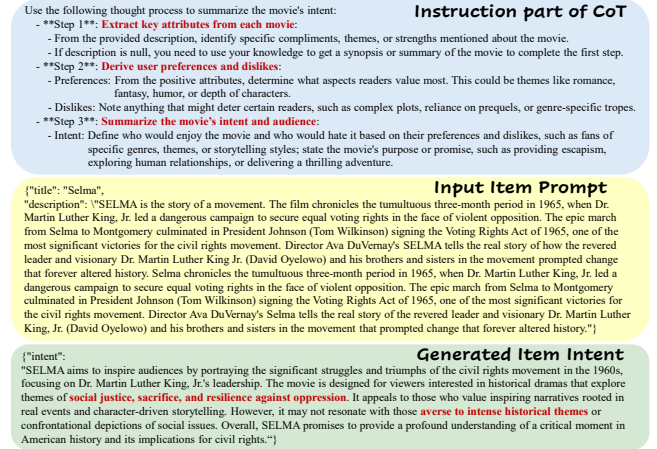


Figure 7: Case study on LLM-based item intent generation in Amazon-movie dataset.

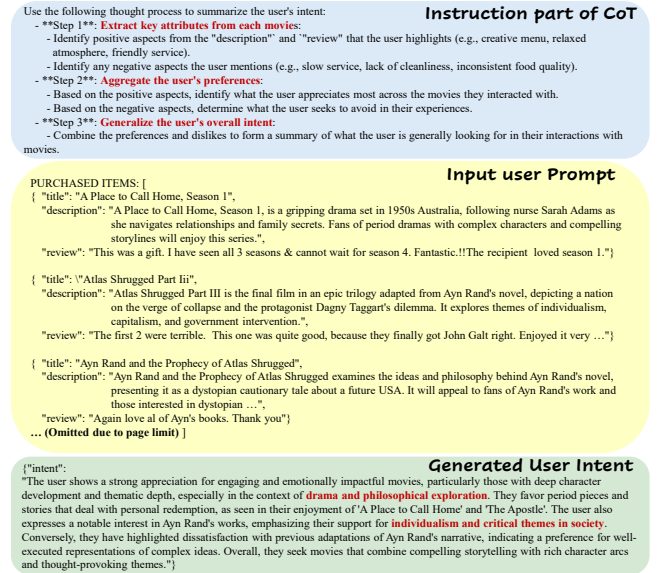


Figure 8: Case study on LLM-based user intent generation in Amazon-movie dataset.

matching method with momentum distillation for teacher-student learning of fused intent representations. Experiments on three public datasets validated the superiority of IRLRec.

Acknowledgments

This work is supported by the National Science Foundation of China (No. 62206002, No. 62272001 and No. 62206004), Natural Science Foundation of Anhui Province (No. 2208085QF195) and Xunfei Zhiyuan Digital Transformation Innovation Research Special for Universities (No. 2023ZY001).

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Floren-
cia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, 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 LLarge
Language Model with Recommendation. In *RecSys*. 1007–1014.
- [3] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple
Yet Effective Graph Contrastive Learning for Recommendation. In *ICLR*.
- [4] Hong Chen, Yudong Chen, Xin Wang, Ruobing Xie, Rui Wang, Feng Xia, and
Wenwu Zhu. 2021. Curriculum Disentangled Recommendation with Noisy Multi-
feedback. *NeurIPS* 34 (2021), 26924–26936.
- [5] 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 *AAAI*, Vol. 34. 27–34.
- [6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A
Simple Framework for Contrastive Learning of Visual Representations. In *ICML*.
PMLR, 1597–1607.
- [7] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent
Contrastive Learning for Sequential Recommendation. In *WWW*. 2172–2182.
- [8] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad
Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan,
et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783* (2024).
- [9] Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and
Yongliang Li. 2019. Metapath-guided Heterogeneous Graph Neural Network for
Intent Recommendation. In *KDD*. 2478–2486.
- [10] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei
Zhang. 2023. Chat-Rec: Towards Interactive and Explainable LLMs-augmented
Recommender System. *arXiv preprint arXiv:2303.14524* (2023).
- [11] Xavier Glorot and Yoshua Bengio. 2010. Understanding the Difficulty of Training
Deep Feedforward Neural Networks. In *Proceedings of the thirteenth International
Conference on Artificial Intelligence and Statistics (ICAIS)*. JMLR Workshop and
Conference Proceedings, 249–256.
- [12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Mo-
mentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*.
9729–9738.
- [13] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng
Wang. 2020. Lightgcn: Simplifying and Powering Graph Convolution Network
for Recommendation. In *SIGIR*. 639–648.
- [14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng
Chua. 2017. Neural Collaborative Filtering. In *WWW*. 173–182.
- [15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic
Models. In *NeurIPS*, Vol. 33. 6840–6851.
- [16] Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. 2023. Learning
Vector-quantized Item Representation for Transferable Sequential Recommenders.
In *WWW*. 1162–1171.
- [17] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiushi Chen, and Julian McAuley.
2024. Bridging language and items for retrieval and recommendation. *arXiv
preprint arXiv:2403.03952* (2024).
- [18] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty,
Caiming Xiong, and Steven Hoi. 2021. Align before Fuse: Vision and Language
Representation Learning with Momentum Distillation. In *NeurIPS*.
- [19] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018.
Variational Autoencoders for Collaborative Filtering. In *WWW*. 689–698.
- [20] Jiayi Liao, Xiangnan He, Ruobing Xie, Jiancan Wu, Yancheng Yuan, Xingwu Sun,
Zhanhui Kang, and Xiang Wang. 2024. RosePO: Aligning LLM-based Recom-
menders with Human Values. *arXiv preprint arXiv:2410.12519* (2024).
- [21] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang,
and Xiangnan He. 2024. Llara: Large Language-Recommendation Assistant. In
SIGIR. 1785–1795.
- [22] Dongsheng Luo, Wei Cheng, Wenchao Yu, Bo Zong, Jingchao Ni, Haifeng Chen,
and Xiang Zhang. 2021. Learning to Drop: Robust Graph Neural Network via
Topological Denoising. In *WSDM*. 779–787.
- [23] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled
Graph Convolutional Networks. In *ICML*. PMLR, 4212–4221.
- [24] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learn-
ing Disentangled Representations for Recommendation. *NeurIPS* 32 (2019).
- [25] Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih
Yavuz. 2024. SFR-Embedding-Mistral: Enhance Text Retrieval with Transfer
Learning. *Salesforce AI Research Blog* 3 (2024).
- [26] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry
Torek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al.
2022. Text and Code Embeddings by Contrastive Pre-training. *arXiv preprint
arXiv:2201.10005* (2022).
- [27] Yingtao Peng, Chen Gao, Yu Zhang, Tangpeng Dan, Xiaoyi Du, Hengliang Luo,
Yong Li, and Xiaofeng Meng. 2024. Denoising Alignment with Large Language
Model for Recommendation. *ACM Transactions on Information Systems (TOIS)*
(2024).
- [28] Xubin Ren and Chao Huang. 2024. EasyRec: Simple yet Effective Language
Models for Recommendation. *arXiv preprint arXiv:2408.08821* (2024).
- [29] 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 *WWW*. 3464–3475.
- [30] Xubin Ren, Lianghao Xia, Yuhao Yang, Wei Wei, Tianle Wang, Xuheng Cai,
and Chao Huang. 2024. SSLRec: A Self-Supervised Learning Framework for
Recommendation. In *WSDM*. 567–575.
- [31] Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. 2023. Dis-
entangled Contrastive Collaborative Filtering. In *SIGIR*. 1137–1146.
- [32] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2010. Introduction to Rec-
ommender Systems Handbook. In *Recommender Systems Handbook*. Springer,
1–35.
- [33] Lei Sang, Yu Wang, Yiwen Zhang, and Xindong Wu. 2024. Denoising Heteroge-
neous Graph Pre-training Framework for Recommendation. *ACM Transactions
on Information Systems (TOIS)* (2024).
- [34] Lei Sang, Yu Wang, Yi Zhang, Yiwen Zhang, and Xindong Wu. 2025. Intent-
guided Heterogeneous Graph Contrastive Learning for Recommendation. *IEEE
Transactions on Knowledge and Data Engineering (TKDE)* 37, 4 (2025), 1915–1929.
- [35] Lei Sang, Min Xu, Shengsheng Qian, Matt Martin, Peter Li, and Xindong Wu. 2020.
Context-dependent Propagating-based Video Recommendation in Multimodal
Heterogeneous Information Networks. *IEEE Transactions on Multimedia (TMM)*
23 (2020), 2019–2032.
- [36] Leheng Sheng, An Zhang, Yi Zhang, Yuxin Chen, Xiang Wang, and Tat-Seng Chua.
2025. Language Representations Can be What Recommenders Need: Findings
and Potentials. *ICLR* (2025).
- [37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones,
Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All
you Need. In *NeurIPS*, Vol. 30.
- [38] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation
learning through alignment and uniformity on the hypersphere. In *ICML*. PMLR,
9929–9939.
- [39] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021.
Deconfounded Recommendation for Alleviating Bias Amplification. In *KDD*.
1717–1725.
- [40] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019.
Neural Graph Collaborative Filtering. In *SIGIR*. 165–174.
- [41] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhengguang Liu,
Xiangnan He, and Tat-Seng Chua. 2021. Learning Intents behind Interactions
with Knowledge Graph for Recommendation. In *WWW*. 878–887.
- [42] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua.
2020. Disentangled Graph Collaborative Filtering. In *SIGIR*. 1001–1010.
- [43] Yifan Wang, Suyao Tang, Yuntong Lei, Weiping Song, Sheng Wang, and Ming
Zhang. 2020. DisenHAN: Disentangled Heterogeneous Graph Attention Nnet-
work for Recommendation. In *CIKM*. 1605–1614.
- [44] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi,
Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought Prompting Elicits Reason-
ing in Large Language Models. *NeurIPS* 35 (2022), 24824–24837.
- [45] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2023. Personal-
ized News Recommendation: Methods and Challenges. *ACM Transactions on
Information Systems (TOIS)* 41, 1 (2023), 1–50.
- [46] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian,
and Xing Xie. 2021. Self-supervised Graph Learning for Recommendation. In
SIGIR. 726–735.
- [47] Yunjia Xi, Weiwei Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo
Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards Open-world
Recommendation with Knowledge Augmentation from Large Language Models.
In *RecSys*. 12–22.
- [48] Yonghui Yang, Le Wu, Zihan Wang, Zhuangzhuang He, Richang Hong, and Meng
Wang. 2024. Graph Bottlenecked Social Recommendation. In *KDD*. 3853–3862.
- [49] 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 *SIGIR*. 1294–1303.
- [50] Zhenrui Yue, Sara Rabhi, Gabriel de Souza Pereira Moreira, Dong Wang, and Even
Oldridge. 2023. LlamaRec: Two-Stage Recommendation using Large Language
Models for Ranking. *CIKM* (2023).
- [51] Yi Zhang, Lei Sang, and Yiwen Zhang. 2024. Exploring the Individuality and
Collectivity of Intents behind Interactions for Graph Collaborative Filtering. In
SIGIR. 1253–1262.
- [52] Yi Zhang, Yiwen Zhang, Lei Sang, and Victor S Sheng. 2024. Simplify to the
Limit! Embedding-less Graph Collaborative Filtering for Recommender Systems.
ACM Transactions on Information Systems (TOIS) 43, 1 (2024), 1–30.