Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Interaction-Grounded Semantic Graph Refinement for LLM-based Recommendation

WOOSEOK JEONG, YOUNG-JIN KIM, HAE-YOON KOO, JIMYEUNG SEO, JINHO CHOI, BYUNGKOOK OH
Department of Computer Science and Engineering, Konkuk University, Gwangjin-gu, Seoul 05029, Republic of Korea

Corresponding author: Byungkook Oh (e-mail: bkoh@konkuk.ac.kr).

Following are results of a study on the "Convergence and Open Sharing System" Project, supported by the Ministry of Education and National Research Foundation of Korea. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2024-00457435). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the metaverse support program to nurture the best talents (IITP-2025-RS-2023-00256615) grant funded by the Korea government(MSIT).

ABSTRACT Large language models (LLMs) have recently demonstrated remarkable potential for recommendation by reframing it as a text-generation task. Recent LLM-based approaches apply GNNs to capture higher-order collaborative patterns from interaction graphs but handle textual data separately, failing to extract higher-order semantic patterns from text. One potential solution is to construct semantic graphs to better capture such semantic relationships. However, naively selecting top-k connections by profile similarity introduces significant challenges: 1) preference gap between textual similarity and actual user behavior, and 2) semantic distortion where identical attributes carry different meanings for users versus items. To overcome these issues, we propose an **I**nteraction-**G**rounded **S**emantic **Rec**ommender (**IGSRec**), which constructs an interaction-grounded semantic graph by aligning profile-based connections with observed interactions. IGSRec employs an LLM-based profile generator, constructs a top-k semantic graph, then refines it using a learnable scoring function that identifies relevant semantic neighborhoods conditioned on user-item interactions. Through dual-graph propagation over both the refined semantic and interaction graphs, IGSRec captures higher-order semantic and collaborative patterns. Experiments on Amazon Review benchmarks demonstrate state-of-the-art performance in both direct and sequential recommendation tasks. Our code and data are publicly available at https://github.com/oseoko/IGSRec/.

INDEX TERMS Large language models, recommendation systems, graph refinement, graph neural networks

I. INTRODUCTION

Traditional collaborative filtering (CF) relies solely on user-item interactions [1], [2], suffering from data sparsity and cold-start issues. Large language models (LLMs) address these limitations by reframing recommendation as a text-generation task [3]–[5], enabling human-like reasoning [6], [7] to exploit rich semantic information [8] from reviews, item attributes, and external knowledge. LLM-based methods have explored various ways to leverage textual data. Some methods treat users and items as numeric IDs [9]–[15], which lack semantic meaning and show limited generalization. Other methods generate semantic IDs from text metadata such as titles, categories, and descriptions [3], [16], [17], enabling natural language understanding of user preferences. Many of these methods further employ explanation generation objectives within multi-task learning to exploit textual semantics and user preferences in diverse recommendation scenarios.

Recent advances integrate LLMs with graph learning [5], [18], [19]. The most effective approaches employ graph neural networks (GNNs) [20], [21] to capture higher-order collaborative patterns (e.g., user→item→user) by propagating embeddings over user-item interaction graphs [22], [23]. For instance, ELMRec [14] extends POD [10] by integrating LightGCN [24] propagation with ID embeddings. However, these GNN-enhanced methods apply propagation exclusively to interaction graphs while handling textual descriptions separately, failing to extract higher-order semantic patterns from text. Moreover, interaction graphs rely on historical behavior and suffer from inherent sparsity, which leads to cold-start issues and prevents inference of higher-order collaborative patterns such as indirect user-user or item-item relationships. This motivates the need for complementary structures that leverage rich textual information to establish semantic connections beyond sparse behavioral interactions.

A straightforward solution is to construct a semantic graph based on profile similarity, where nodes sharing common textual characteristics can be connected through embedding similarity. This complements the sparse structure of interaction graphs, enhancing the diversity and expressiveness of user and item representations. However, as shown in Figure 1, such naive attempts introduce two fundamental challenges:

- **[C1] Preference Gap:** Textual similarity may not align with actual user preferences. A "romance" attribute in a user profile does not guarantee genuine interest in romance content. In other words, a gap occurs between explicit textual signals and implicit behavioral patterns.
- **[C2] Semantic Distortion:** Identical attributes in user and item profiles may carry different semantic meanings depending on context, leading to misleading connections when relying solely on textual matching.

These challenges cause semantic graphs to form suboptimal dense structures containing noisy connections. During embedding propagation, this noise accumulates and amplifies, distorting learned representations and degrading recommendation performance. Thus, bridging the gap between textual similarity and behavioral evidence remains a critical challenge in semantic graph construction.

Motivated by these issues, we propose an **I**nteraction-**G**rounded **S**emantic **Rec**ommender **(IGSRec)**, which constructs an interaction-grounded semantic graph by aligning profile-based connections with observed interactions. IGSRec captures higher-order collaborative and semantic patterns while mitigating preference gap and semantic distortion.

Specifically, we generate user and item profiles through GPT-based review summarization to capture explicit preferences, latent interests, and contextual attributes. A top-k semantic graph constructor then computes embedding similarity across node pairs (i.e., user-user, user-item, item-item) and selects top-k similar neighbors to form an initial top-k semantic graph. To refine these connections beyond simple profile similarity, we introduce a semantic graph refinement module. Inspired by recent advances in subgraph retrieval for knowledge graph reasoning [25], it employs a lightweight MLP-based scoring function that learns to identify semantically relevant neighborhoods conditioned on actual user-item interactions. Unlike prior work that retrieves subgraphs from external knowledge graphs, we refine connections within the constructed graph by treating shortest paths between observed interactions as weak-supervision signals. This allows the refinement module to align semantic relationships with behavioral patterns. Although the initial graph exhibits dense user-user and item-item connections, it has sparse user-item links. The refinement process creates meaningful bridges across all node types and discovers potential user–item preference relationships beyond profile similarity alone.

Through dual-graph propagation over the refined semantic and interaction graphs, IGSRec learns enriched user and item representations that capture higher-order semantic and collaborative patterns. An embedding fusion mechanism then integrates these representations with ID embeddings. This

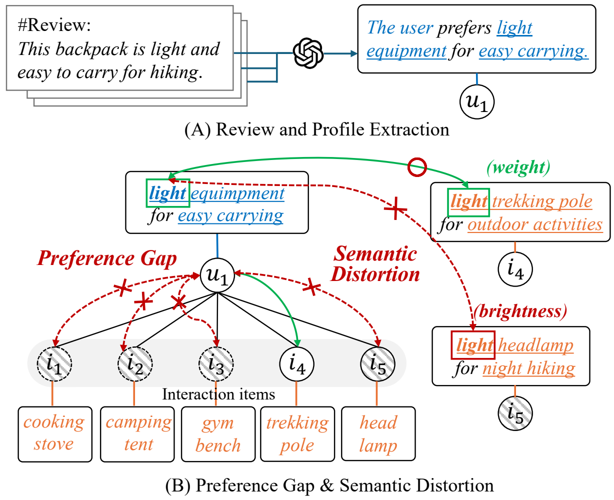


FIGURE 1. Illustration of challenges in semantic graph construction. LLM-based profile generation (A) followed by naive similarity-based connections (B) leads to two issues: [C1] Preference Gap and [C2] Semantic Distortion.

enables the LLM-based recommender to better interpret node semantics and relationships within recommendation contexts, enhancing recommendation performance. The main contributions of this paper can be summarized as follows:

- We introduce interaction-grounded semantic graphs that explicitly model user-user, item-item, and user-item relationships, complementing sparse interaction structures and capturing higher-order semantic patterns.
- We propose a semantic graph refinement module that refines top-k semantic connections by aligning profile-based similarity with actual interaction patterns, effectively addressing preference gap and semantic distortion.
- We achieve state-of-the-art performance on Amazon Review benchmarks with significantly improved robustness in sparse and cold-start scenarios, validating the effectiveness of our approach.

II. RELATED WORK

A. LLM-BASED RECOMMENDATION

LLMs have demonstrated exceptional performance in recommendation tasks [3]–[5], [17], including explainable recommendation [26] and addressing cold-start scenarios by leveraging their intrinsic knowledge [27]. LLM-based recommendation research can be broadly categorized into three paradigms based on how user and item IDs are handled.

ID direct usage paradigm treats raw numerical IDs as textual tokens for LLMs to process. Pioneering works like P5 [9] and its successors [11], [12] adopted this approach in which numerical IDs (e.g., user_12, item_115) are directly fed as whole-word embeddings to LLMs. However, numerical IDs lack semantic meaning and exhibit limited generalization capability. To address this limitation, ID Translation paradigm converts numerical IDs into natural language text such as titles, categories, and descriptions [3]. This allows LLMs to leverage their pre-trained knowledge for text-based reasoning [17]. However, this approach faces constraints when dealing with long user histories due to input length limitations. ID Representation paradigm addresses these issues by encoding IDs into continuous prompt vectors, as seen in Ra-Rec [28], or by employing hybrid methods that combine discrete and continuous prompts [29]. This offers greater flexibility across various recommendation tasks.

Beyond ID handling paradigms, recent efforts have focused on enhancing the reasoning capabilities of LLMs. Chain-of-Thought (CoT) prompting extracts intermediate reasoning steps [30], while rationale distillation compresses knowledge into smaller, more efficient models [31]. For instance, POD [10] significantly improved P5's performance through prompt distillation. Building on these advances, RDRec [15] distills interaction rationales from user reviews to construct sophisticated user and item profiles, simultaneously ensuring both recommendation performance and explainability.

While these methods leverage textual semantics via various ID representations and reasoning techniques, they encode users and items independently without modeling the semantic relationships between them. Our approach complements these representations by explicitly capturing user-user, item-item, and user-item connections through a semantic graph.

B. GNN-BASED RECOMMENDATION

GNNs have become a core strategy in recommendation systems [23], [32], prized for their ability to model user-item interactions and capture higher-order collaborative patterns. Foundational models like NGCF [33] and its lightweight successor, LightGCN [24], exemplify this by propagating embeddings over user-item interaction graphs to learn representations through multi-hop neighborhood aggregation.

Recently, integrating GNNs' structural modeling capabilities with LLMs' semantic understanding has attracted significant attention. Various approaches have emerged: using LLMs to augment graph structures via large-scale text prompts [5], enhancing textual attributes of nodes and their neighbors [18], and aligning graph-based interaction patterns with text-based reasoning [19]. Among these, ELMRec [14] represents a notable advancement by internalizing higher-order collaborative patterns directly within an LLM. Building upon the POD framework [10], ELMRec integrates LightGCN's propagation mechanism into the LLM's embedding layer. As a result, the model can capture indirect collaborative patterns that purely text-based models would miss.

However, these methods apply graph propagation exclusively to interaction graphs while handling textual data separately, failing to capture higher-order semantic patterns from rich textual descriptions. Moreover, they do not explicitly model user-user or item-item semantic relationships.

C. GRAPH RETRIEVAL FOR LLMS

Retrieval-augmented generation (RAG) has excelled with textual corpora and further leverages knowledge graphs (KG) as structured knowledge sources to enhance reasoning expressiveness [34]–[36]. However, KG-based RAG frameworks face limitations in retrieval efficiency and reasoning suitability. To address this, recent works have introduced pruning approaches that remove unnecessary structural information not related to the query and retain only core evidence [37].

Some research has shifted towards subgraph retrieval [38], [39], which aims to extract compact, query-relevant subgraphs containing multi-hop relational paths [40], rather than isolated entities and relations. Various strategies have emerged, including query decomposition [41], multi-hop expansion from seed entities [40], and linearizing graph triples for dense retrieval [42]. Other approaches focus on connecting key topic entities and exploring their local neighborhoods to construct retrieval contexts [43]. Among them, SubgraphRAG [25] achieves both efficiency and accuracy through a lightweight MLP-based retriever that leverages structural context without relying on computationally expensive GNNs or complex heuristics.

While these methods extract relevant subgraphs from external knowledge graphs for question answering, we take a fundamentally different approach. Rather than retrieving from external sources, our semantic graph refinement module refines connections within a constructed semantic graph. It employs a learnable scoring function that leverages observed interactions as weak supervision to address preference gap and semantic distortion, thereby constructing an interaction-grounded semantic graph.

III. PRELIMINARY

A. PROBLEM DEFINITION

Following prior research [10], IGSRec transforms the recommendation task into a text generation problem through task-specific prompt distillation. We focus on two core recommendation tasks: **(i) direct recommendation**, which predicts the most relevant item from a candidate list, and **(ii) sequential recommendation**, which forecasts the next item based on a user's interaction history. Additionally, we incorporate (iii) explanation generation as an auxiliary task to produce textual rationales for user-item interactions [10]. The multi-task learning framework enables the model to leverage the reasoning capabilities of LLMs while jointly learning from both behavioral prediction and semantic understanding.

For each task, the input and output sequences are denoted as $X = [x_1, \ldots, x_{|X|}]$ and $Y = [y_1, \ldots, y_{|Y|}]$. We concatenate the input tokens with distilled prompt vectors $P = [p_1, \ldots, p_{|P|}]$, yielding the extended input sequence $X_p = [x_1, \ldots, x_{|X|}, p_1, \ldots, p_{|P|}]$. To ensure distinguishable and coherent ID-level representations, we further adopt whole-word embeddings [9], [10] $X_\omega = [\omega_1, \ldots, \omega_{|X|+|P|}]$. Each ω_i corresponds to the embedding of a user or item ID (when present) or a general token embedding otherwise. In IGSRec, whole-word embeddings for user and item IDs are constructed by fusing collaborative and semantic signals obtained from dual-graph propagation, enabling the LLM to leverage both behavioral patterns and semantic relationships. The final representation is defined as $\hat{X} = X_p + X_\omega$, which is fed into the LLM to model the conditional probability distribution

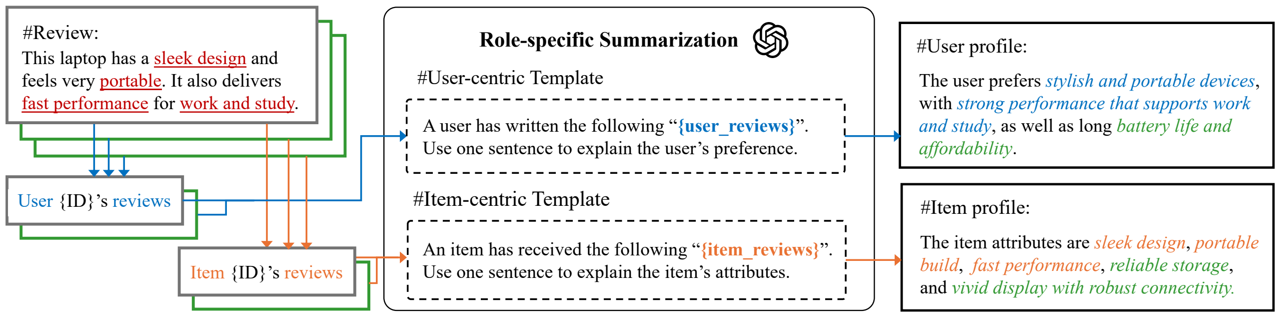


FIGURE 2. Illustration of user and item profile generation with role-specific summarization. Blue texts denote user-centric expressions and orange texts denote item-centric expressions from aggregated reviews, while green texts indicate attributes inferred from other reviews to enrich the profile.

$p(y_{step} \mid Y_{<step}, X)$ at each generation step, where $Y_{<step}$ denotes the previously generated tokens.

The model parameters Θ are optimized by minimizing the negative log-likelihood loss function:

$$\mathcal{L}(\Theta) = -\frac{1}{|D|} \sum_{(X,Y)\in D} \frac{1}{|Y|} \sum_{step=1}^{|Y|} \log p(y_{step} \mid Y_{<step}, X), \quad (1)$$

where D denotes the entire training set consisting of all input–output pairs, $|D|$ is the number of training samples, and $|Y|$ is the average number of tokens in each output sequence.

B. GRAPH REFINEMENT FORMULATION

Following prior work on subgraph-based retrieval [25], we adopt a weak supervision strategy to train our graph refinement module. Since the ground-truth refined graph for a given query $q = \langle n_u, n_i \rangle$ - an interaction pair of user and item anchor nodes - cannot be directly observed, fully supervision training is impossible. Instead, the surrogate graph using shortest path between the anchors serves as heuristic evidence of meaningful connections. During training, nodes lying on the surrogate shortest paths are labeled as positive, while all other nodes are labeled as negative. This weak supervision provides an effective inductive bias for learning to refine semantic graphs without requiring annotated ground truth.

IV. THE PROPOSED MODEL

We present the IGSRec framework, composed of three main stages: (i) top-k semantic graph construction, (ii) semantic graph refinement, and (iii) dual graph propagation and fusion as illustrated in Figure 3.

A. TOP-K SEMANTIC GRAPH CONSTRUCTION

1) User and Item Profile Generation

To generate user and item profiles, we design tailored prompts that guide an LLM to summarize aggregated reviews from two complementary perspectives (i.e., user-centric and item-centric) rather than directly using raw review text. For each user, we collect all reviews they have written and instruct the LLM to extract their overall semantic preferences, forming a user-centric profile. Similarly, for each item, all reviews about it are aggregated and summarized to construct an item-centric profile that encodes its intrinsic attributes and characteristics. This role-specific summarization enforces a clear separation between behavioral preferences (user-centric) and objective characteristics (item-centric), thus producing refined profiles that mitigate noise in raw reviews.

Figure 2 illustrates the role-specific summarization process and prompt templates, where blue and orange texts indicate user-centric and item-centric signals derived from aggregated reviews, while green texts represent complementary attributes incorporated from additional reviews to enrich the profile. Specifically, for a user u and an item v, their profiles are generated as $p_u = \mathrm{LLM}(\{r_{u,\cdot}\}, \mathcal{P}_u)$ and $p_v = \mathrm{LLM}(\{r_{\cdot,v}\}, \mathcal{P}_v)$, where $\{r_{u,\cdot}\}$ denotes all reviews written by user u and $\{r_{\cdot,v}\}$ denotes all reviews about item v, and $\mathcal{P}_u$ and $\mathcal{P}_v$ are prompt templates that guide the LLM to summarize the reviews from a user-centric or item-centric perspective, respectively. The resulting profiles p_u and p_v serve as the foundation for semantic graph construction, enabling to capture of higher-order semantic patterns through graph propagation.

2) Top-k Semantic Graph Initialization

Building on the refined profiles, we construct a semantic graph that encodes latent associations among users and items. Each profile is embedded into a shared representation space by a pre-trained encoder f_θ: $z_u = f_\theta(p_u)$ and $z_v = f_\theta(p_v)$. These embeddings provide semantically consistent representations of user preferences and item attributes.

To construct the top-k semantic graph, we compute pairwise cosine similarities and connect each node only to its top-k neighbors whose similarity exceeds threshold τ. This preserves reliable relations while avoiding unnecessary connectivity. We define the edge weight w_{uv} between nodes u and v as their cosine similarity if both the top-k and threshold conditions are met, and zero otherwise:

$$w_{uv} = \begin{cases} \frac{\langle z_u, z_v \rangle}{\|z_u\|\|z_v\|}, & \text{if } v \in \text{top-}k(u) \text{ and } \frac{\langle z_u, z_v \rangle}{\|z_u\|\|z_v\|} \geq \tau, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The resulting graph contains user–user edges that capture aligned preference patterns, item–item edges that highlight shared characteristics, and user–item edges that indicate semantic matches between interests and attributes. This process

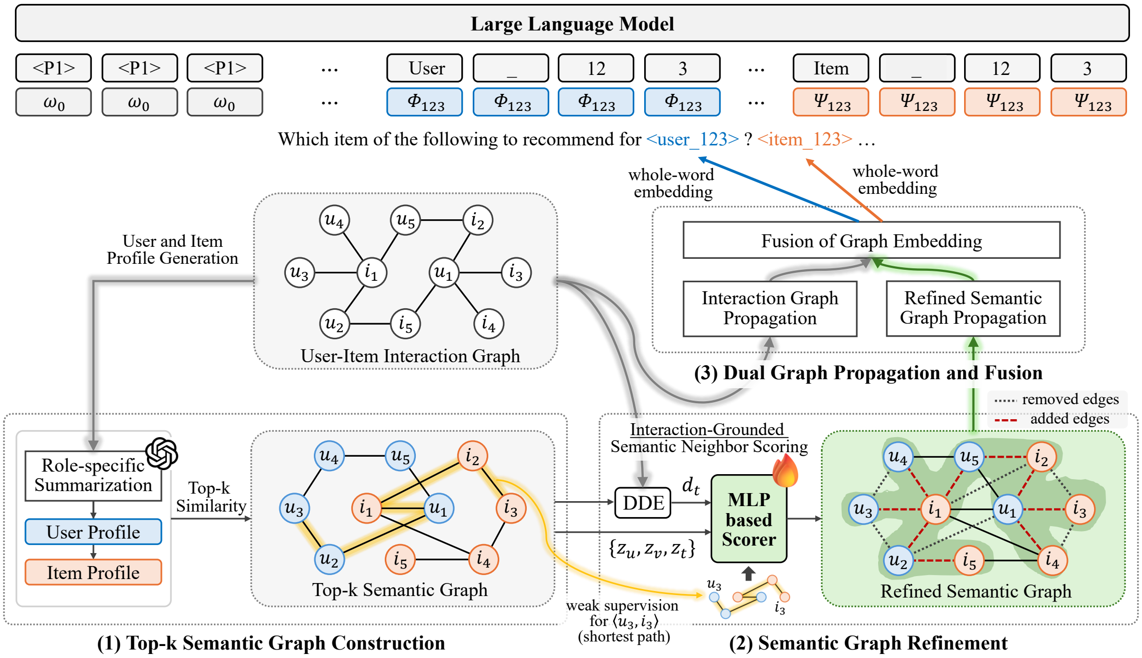


FIGURE 3. Overview of the IGSRec framework. (1) role-specific profiles from a profile generator yield a top-k semantic graph, (2) refined via DDE and MLP-based scoring with shortest paths from observed user-item interactions as weak supervision, and (3) fused with the interaction graph through dual propagation to enrich representations for the LLM-based recommender.

yields the top-k semantic graph G_s, which extends beyond historical interactions and provides a structured foundation for subsequent refinement.

B. SEMANTIC GRAPH REFINEMENT

1) Interaction-Grounded Semantic Neighbor Scoring

We employ a lightweight MLP-based scoring function MLP_θ that leverages both semantic embeddings and structural signals to score semantic neighbors conditioned on observed interactions. Each query q is defined as an interaction pair $q = \langle n_u, n_v\rangle$, where n_u and n_v are the user and item anchor nodes observed in the interaction graph. During training, we construct weak supervision signals using the shortest paths between n_u and n_v in the top-k semantic graph G_s. Specifically, any target node n_t lying on these shortest paths is assigned a positive label ($y_t = 1$), while all other nodes outside the paths are assigned negative labels ($y_t = 0$).

To capture complementary perspectives, we construct dual input representations for each candidate target node n_t based on which anchor serves as the primary reference point:

$$\begin{aligned}\mathbf{x}_{u\to t} &= [z_u \parallel z_v \parallel z_t \parallel d_{u\to t}(G_s, q)],\\ \mathbf{x}_{v\to t} &= [z_v \parallel z_u \parallel z_t \parallel d_{v\to t}(G_s, q)],\end{aligned} \tag{3}$$

where $\mathbf{x}_{u\to t}$ and $\mathbf{x}_{v\to t}$ denote the user-centric and item-centric views, respectively. Here, z_u, z_v, z_t are the profile embeddings of the user anchor, item anchor, and target node. $d_{u\to t}(G_s, q)$ and $d_{v\to t}(G_s, q)$ are the directional distance encodings (DDE) that capture the structural position of n_t relative to the user anchor and item anchor, respectively (detailed in the next section). The MLP then outputs relevance scores from both perspectives:

$$\begin{aligned}\hat{y}_{u\to t} &= \sigma(\mathrm{MLP}_\theta(\mathbf{x}_{u\to t})),\\ \hat{y}_{v\to t} &= \sigma(\mathrm{MLP}_\theta(\mathbf{x}_{v\to t})),\end{aligned} \tag{4}$$

where $\sigma(\cdot)$ denotes the sigmoid activation function and $\hat{y}_{u\to t}, \hat{y}_{v\to t} \in [0, 1]$ represent the probabilities that n_t is semantically relevant from each perspective. The model is trained as a binary classifier by minimizing the binary cross-entropy loss:

$$\mathcal{L}_{\text{scoring}} = \frac{1}{|Q|}\sum_{q\in Q}\sum_{n_t\in N}\sum_{a\in\{u,v\}} \mathrm{BCE}(y_t, \hat{y}_{a\to t}), \tag{5}$$

where $\mathrm{BCE}(y, \hat{y}) = -[y\log\hat{y} + (1-y)\log(1-\hat{y})]$ is the binary cross-entropy function, Q denotes the set of training queries corresponding to observed user-item interactions, N is the set of candidate nodes in G_s, and y_t is the ground-truth label derived from shortest path supervision.

During inference, we restrict the candidate sets to the 2-hop neighborhoods of each anchor in G_s for computational efficiency. For a given query $q = \langle n_u, n_v\rangle$, we define T_q^u and T_q^v as the 2-hop neighborhoods of user anchor n_u and item anchor n_v, respectively. Starting from the item anchor n_v, the trained MLP evaluates each candidate $n_t \in T_q^v$ using $\mathbf{x}_{v\to t}$ to produce scores $\hat{y}_{v\to t}$, then constructs the item-centric refined graph $\hat{G}_v$ by connecting n_v to the top-k highest-scoring nodes, creating item-item and item-user edges. Similarly, starting from the user anchor n_u, it evaluates candidates $n_t \in T_q^u$ using $\mathbf{x}_{u\to t}$ to produce scores $\hat{y}_{u\to t}$, then constructs the user-centric refined graph $\hat{G}_u$ by connecting n_u to the top-k highest-scoring nodes, creating user-user and user-item edges. Finally, these two refined graphs are merged to form $\hat{G}_s = \hat{G}_u \cup \hat{G}_v$.

2) Directional Distance Encoding for Structural Position

To model the structural relationships between a candidate target node n_t and the anchor nodes in query interaction $\langle n_u, n_v\rangle$ within the semantic graph G_s, we introduce directional distance encoding (DDE). As shown in Figure 4, the key idea is that nodes closer to the anchors naturally accumulate stronger signals through graph propagation, while distant nodes receive weaker signals, effectively encoding their relative structural positions.

Given a query $q = \langle n_u, n_v\rangle$, we first initialize the representation of each node $e \in G_s$ as:

$$s_e^{(0)} = \begin{cases} 1, & e \in \{n_u, n_v\}, \\ 0, & \text{otherwise}. \end{cases} \tag{6}$$

That is, one-hot values are assigned only to the anchors while all other nodes are initialized to zero. We then propagate these signals bidirectionally through G_s to incrementally encode directional distances. In the forward propagation, each node e aggregates signals from its incoming neighbors e' located l hops away from the anchors:

$$s_e^{(l+1)} = \mathrm{MEAN}\{\, s_{e'}^{(l)} \mid (e', e) \in G_s \,\}, \tag{7}$$

where $(e', e) \in G_s$ denotes a directed edge from e' to e. At the first hop, the anchor's signal is split among its neighbors (each receiving a value of $1/|\mathcal{N}(e')|$). We also perform reverse propagation, which sends signals backward along the edges:

$$s_e^{(r,l+1)} = \mathrm{MEAN}\{\, s_{e'}^{(r,l)} \mid (e, e') \in G_s, \},\ s_e^{(r,0)} = s_e^{(0)}, \tag{8}$$

where (e, e') denotes an outgoing edge from e to e'. After L propagation steps, each node e accumulates these signals into a fixed-dimensional vector:

$$s_e = \left[s_e^{(0)} \parallel s_e^{(1)} \parallel \cdots \parallel s_e^{(L)} \parallel s_e^{(r,1)} \parallel \cdots \parallel s_e^{(r,L)} \right] \in \mathbb{R}^{2L+1}. \tag{9}$$

This final vector s_e is formed by concatenating the initial signal and all signals from the L forward and reverse propagation steps. The DDE for a candidate target node n_t is then defined based on which anchor serves as the starting point. When starting from the user anchor n_u, we define the user-anchored DDE as $d_{u\to t}(G_s, q) = s_t$, while when starting from the item anchor n_v, we define the item-anchored DDE as $d_{v\to t}(G_s, q) = s_t$. Although both use the same propagated signal s_t, they serve different roles in the input representations shown in Eq. 3, allowing the model to learn distinct scoring patterns for user-user, user-item, item-item, and item-user edge construction during semantic graph refinement.

3) Discussion

The core function of Interaction-Grounded Semantic Neighbor Scoring is to distill the noisy semantic graph G_s into a refined graph $\hat{G}_s$ specific to an interaction pair $q = \langle n_u, n_v\rangle$.

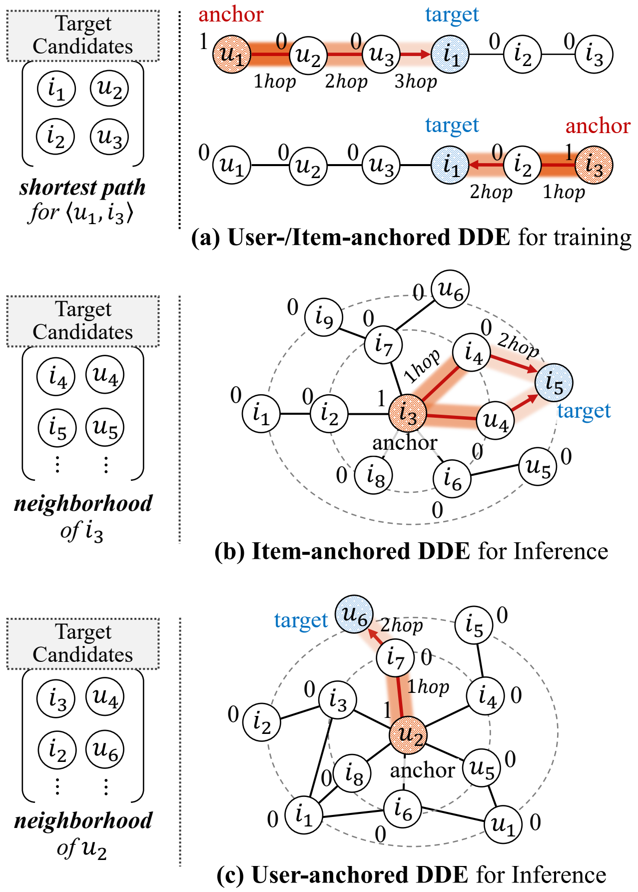


FIGURE 4. Illustration of directional distance encoding (DDE) (a) one-hot signals are propagated from anchors, with shortest-path nodes as weak supervision. (b) 2-hop neighbors of i_3 are scored to construct *item-item* and *item-user* edges. (c) 2-hop neighbors of u_1 are scored to construct *user-user* and *user-item* edges.

This semantic graph refinement is achieved through heuristic-based weak supervision. It is a strategy to train a model using surrogate evidence constructed from heuristics as the ground-truth, instead of finding computationally difficult optimal evidence. In the training process, we use the shortest paths between anchor nodes (n_u, n_v) as surrogate evidence to train the MLP_θ. This binary classification setup is the key to effective noise suppression, as it explicitly trains the MLP_θ to distinguish core interaction signals from abundant, irrelevant noise. During inference, the trained MLP_θ acts as a flexible and precise selection mechanism within the 2-hop neighborhoods, identifying the the most relevant nodes to construct the final, cleanly refined $\hat{G}_s$.

C. DUAL GRAPH PROPAGATION AND FUSION

1) Interaction Graph Propagation

We consider the interaction graph G_{ui}, which serves as the backbone for capturing collaborative filtering signals. Since nodes in G_{ui} carry only ID information without inherent semantics, we initialize user and item embeddings from a Gaussian distribution with zero mean and variance σ^2. This provides unbiased initialization without external priors:

$$\phi_{G_{ui}}(u) \sim \mathcal{N}(0, \sigma^2), \quad \psi_{G_{ui}}(v) \sim \mathcal{N}(0, \sigma^2), \tag{10}$$

where $\phi_{G_{ui}}(u)$ and $\psi_{G_{ui}}(v)$ denote the embeddings of user u and item v in G_{ui}, respectively. With this initialization, subsequent learning is guided solely by collaborative signals without introducing external semantic priors.

To encode collaborative signals, embeddings are updated layer by layer via linear neighborhood aggregation following LightGCN [24]. In this bipartite graph structure, user embeddings aggregate information from neighboring items, while item embeddings aggregate information from neighboring users:

$$\begin{aligned} \psi^{(l+1)}_{G_{ui}}(v) &= \sum_{u \in \mathcal{N}_v} \frac{1}{\sqrt{|\mathcal{N}_v|}\sqrt{|\mathcal{N}_u|}} \phi^{(l)}_{G_{ui}}(u), \\ \phi^{(l+1)}_{G_{ui}}(u) &= \sum_{v \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_v|}} \psi^{(l)}_{G_{ui}}(v), \end{aligned} \tag{11}$$

where $\mathcal{N}_u$ and $\mathcal{N}_v$ denote the neighbors of user u and item v, respectively. Repeating this propagation for L layers allows each node to integrate not only direct interactions but also higher-order information from multi-hop neighbors. After two-hop propagation, the influence of a second-order neighbor u_j on a target user u_i can be written as follows:

$$c_{u_j \to u_i} = \frac{1}{\sqrt{|\mathcal{N}_{u_i}|}\sqrt{|\mathcal{N}_{u_j}|}} \sum_{k \in \mathcal{N}_{u_i} \cap \mathcal{N}_{u_j}} \frac{1}{|\mathcal{N}_k|}. \tag{12}$$

This coefficient reflects three intuitive factors: (i) more co-interactions strengthen the relation; (ii) less popular shared items provide stronger collaborative signals; and (iii) neighbors with fewer connections contribute stronger personalized signals. Finally, embeddings from all layers are combined to form the structural representations as follows:

$$\hat{\psi}_{G_{ui}}(v) = \frac{1}{L+1} \sum_{l=0}^{L} \psi^{(l)}_{G_{ui}}(v), \quad \hat{\phi}_{G_{ui}}(u) = \frac{1}{L+1} \sum_{l=0}^{L} \phi^{(l)}_{G_{ui}}(u). \tag{13}$$

This aggregation balances information from shallow and deep layers, which preserves both direct connections and higher-order collaborative patterns from the interaction graph.

2) Refined Semantic Graph Propagation

We consider the refined semantic graph $\hat{G}_s$, which encodes semantic relations among users and items derived from textual profiles. Unlike G_{ui}, where nodes carry only ID information, nodes in $\hat{G}_s$ are associated with rich semantic profiles generated by the profile generator. Therefore, we initialize user and item embeddings directly with their profile embeddings, providing meaningful semantic priors:

$$\phi_{\hat{G}_s}(u) = z_u, \quad \psi_{\hat{G}_s}(v) = z_v, \tag{14}$$

where $\phi_{\hat{G}_s}(u)$ and $\psi_{\hat{G}_s}(v)$ denote the semantic embeddings of user u and item v, respectively.

To capture semantic dependencies, embeddings are updated layer by layer via weighted neighborhood aggregation over diverse semantic connections. Unlike the bipartite structure of G_{ui}, $\hat{G}_s$ contains user-user, item-item, and user-item edge types, enabling propagation across all node types:

$$\begin{aligned} \psi^{(l+1)}_{\hat{G}_s}(v) &= \sum_{u \in \mathcal{N}_v} \frac{w_{uv}}{\sqrt{d_v}\sqrt{d_u}} \phi^{(l)}_{\hat{G}_s}(u), \\ \phi^{(l+1)}_{\hat{G}_s}(u) &= \sum_{v \in \mathcal{N}_u} \frac{w_{uv}}{\sqrt{d_u}\sqrt{d_v}} \psi^{(l)}_{\hat{G}_s}(v), \end{aligned} \tag{15}$$

where $\mathcal{N}_u$ and $\mathcal{N}_v$ denote the semantic neighbors of user u and item v, respectively. The edge weight between two nodes is defined as w_{uv} and the weighted degree $d_u = \sum_{v \in \mathcal{N}_u} w_{uv}$ (or $d_v = \sum_{u \in \mathcal{N}_v} w_{uv}$) serves as a normalization factor. Repeating this propagation for L layers enables each node to capture not only direct semantic neighbors but also indirect relationships through multi-hop paths.

Finally, embeddings from all layers are averaged to form the semantic embedding of users and items:

$$\hat{\psi}_{\hat{G}_s}(v) = \frac{1}{L+1} \sum_{l=0}^{L} \psi^{(l)}_{\hat{G}_s}(v), \quad \hat{\phi}_{\hat{G}_s}(u) = \frac{1}{L+1} \sum_{l=0}^{L} \phi^{(l)}_{\hat{G}_s}(u). \tag{16}$$

This aggregation balances information from shallow and deep semantic contexts, preserving both direct affinities and higher-order semantic relations within the refined semantic graph.

3) Fusion of Graph Embeddings

We obtain two complementary sets of representations from the dual graph propagation processes. From the interaction graph propagation, we derive $\hat{\phi}_{G_{ui}}(u)$ and $\hat{\psi}_{G_{ui}}(v)$ that capture collaborative patterns, while from the refined semantic graph propagation, we obtain $\hat{\phi}_{\hat{G}_s}(u)$ and $\hat{\psi}_{\hat{G}_s}(v)$ that capture semantic patterns. For user and item tokens, we construct the final whole-word embeddings by linearly combining these propagated embeddings:

$$\omega_i = \begin{cases} \alpha\, \hat{\phi}_{G_{ui}}(u_i) + \beta\, \hat{\phi}_{\hat{G}_s}(u_i), & \text{for user } u_i, \\ \alpha\, \hat{\psi}_{G_{ui}}(v_i) + \beta\, \hat{\psi}_{\hat{G}_s}(v_i), & \text{for item } v_i, \\ \omega_0, & \text{otherwise,} \end{cases} \tag{17}$$

where ω_i denotes the whole-word embedding for token i that will be fed into the LLM, ω_0 represents the pre-trained T5 token embedding for general (non-ID) tokens, and α, β are hyperparameters that control the relative contributions of collaborative and semantic signals. A larger α emphasizes collaborative signals from G_{ui}, while a larger β strengthens semantic signals from $\hat{G}_s$, enabling the model to better capture profile-derived preferences and contextual meanings. These fused embeddings are then integrated into the LLM-based recommender to improve recommendation performance.

V. EXPERIMENTAL SETUP

A. DATASETS

We evaluate IGSRec on three public benchmarks from Amazon product reviews: Sports & Outdoors, Beauty, and Toys &

Dataset	#User	#Item	#Review	Avg.	Density (%)
Sports	48,993	34,298	296,337	8.3	0.0453
Beauty	22,363	12,101	198,502	8.9	0.0734
Toys	19,804	22,086	167,597	8.6	0.0724

TABLE 1. Dataset statistics for Sports & Outdoors, Beauty, and Toys & Games. #User, #Item, #Review, and Avg. denote the number of users, items, reviews, and average reviews per user, respectively

Games[1]. These datasets span different domains and exhibit varying sparsity levels, enabling comprehensive assessment of our approach's generalizability. For direct and sequential recommendation tasks, we adopt a leave-one-out partition strategy following ELMRec [14]: each user's most recent interaction serves as the test instance, the second most recent as validation, and all earlier interactions as training data. To prevent future information leakage, the same partitioning strategy is consistently applied across all recommendation settings. For explanation generation, we use an 8:1:1 train-validation-test split, ensuring that every user and item appears in the training set to support meaningful text generation. The overall dataset statistics are summarized in Table 1.

B. EVALUATION METRICS

We evaluate both direct and sequential recommendation performance using two widely adopted ranking metrics: hit rate@k (H@k) and normalized discounted cumulative gain@k (N@k) with $k \in \{5, 10\}$. H@k indicates whether the ground-truth item is successfully retrieved within the top-k recommendations. As a binary recall metric, it measures the model's ability to identify relevant items regardless of their exact position in the ranking. N@k assesses ranking quality by assigning higher weights to relevant items placed earlier in the recommendation list. This position-sensitive metric captures the practical importance of ranking order.

C. BASELINE MODELS

The effectiveness of IGSRec was evaluated by comparing with traditional, Transformer-based, GNN-based, and LLM-based baselines on direct and sequential recommendation:

Traditional approaches: These methods employ conventional collaborative filtering techniques using matrix factorization, RNNs, or CNNs to model user-item interactions.

- SimpleX [44] unifies collaborative filtering with cosine similarity to improve recommendation.
- GRU4Rec [45] introduces a GRU for recommendation, regarding the item sequence as the user session and optimizing pairwise ranking losses for prediction.
- Caser [46] captures sequential patterns by treating user interaction sequences as images and applying convolutional sequence embedding for recommendation.
- SASRec [47] combines Markov Chains with undirectional self-attention to model sequential recommendation and capture long-term user preferences.
- HGN [48] employs a hierarchical gating network to capture both long- and short-term user interests.

Transformer-based approaches: These methods leverage self-attention mechanisms to model long-range dependencies in sequential recommendation tasks.

- BERT4Rec [49] leverages bidirectional Transformer with a cloze task for sequential recommendation.
- FDSA [50] introduces feature-level self-attention to model both item and feature-level sequential patterns.
- S^3-Rec [51] adopts self-supervised learning paradigm that learns user's latent behavioral features.

GNN-based approaches: These models exploit user–item interaction graphs to leverage collaborative signals through message passing, primarily for direct recommendation.

- LightGCN [24] simplifies graph convolutional for collaborative filtering by removing nonlinear activation.
- NCL [23] enhances contrastive learning by constructing positive pairs from structural and semantic neighbors.
- XSimGCL [32] improves robustness of user and item representations by injecting uniform noise during contrastive learning to mitigate oversmoothing.
- ELMRec [14] enhances LLM-based recommendation by integrating whole-word embeddings to better interpret graph-constructed interaction data.

LLM-based approaches: These models leverage LLMs' text understanding and generation capabilities to reformulate direct and sequential recommendation as a text generation task.

- P5 [9] reformulates recommendation task into a text-to-text paradigm, enabling LLMs to handle diverse tasks with personalized prompts under a unified framework.
- RSL [52] incorporates LLM reasoning ability through novel training and inference strategies.
- POD [10] improves efficiency by distilling prompts within P5 framework for recommendation.
- RDRec [15] distills rationale from reviews into a compact model for Top-N and sequential recommendation.

D. IMPLEMENTATION DETAILS

For a fair comparison, we follow the same settings as ELMRec and adopt T5-small [53] as our backbone. The number of negative items for direct recommendation is set to 99 for both training and evaluation. We prepend three prompt tokens to the input for all tasks, and the batch size is set to 64 across all datasets. The embedding dimensions of both word tokens and whole-word tokens are set to 512. For sequential recommendation, we iteratively and randomly sample a segment from each user's historical sequence for training. We also keep several hyperparameters the same as in ELMRec, including the initialization standard deviation σ and the number of LightGCN layers L. For model-specific hyperparameters, we tune the graph fusion weights α and β by searching over the set $\{5, 7, 9, 11, 13, 15\}$. The optimal values are determined

[1] https://www.amazon.com/

Models	Sports				Beauty				Toys			
	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
Traditional Approach												
SimpleX	0.2362	0.1505	0.3290	0.1800	0.2247	0.1441	0.3090	0.1711	0.1958	0.1244	0.2662	0.1469
GNN-based Approach												
LightGCN	0.4150	0.3002	0.5436	0.3418	0.4205	0.3067	0.5383	0.3451	0.3879	0.2874	0.5106	0.3272
NCL	0.4292	0.3131	0.5592	0.3551	0.4378	0.3228	0.5542	0.3607	0.3975	0.2925	0.5120	0.3325
XSimGCL	0.3547	0.2689	0.4486	0.2992	0.3530	0.2734	0.4392	0.3012	0.3351	0.2614	0.4186	0.2885
LLM-based Approach												
P5	0.1955	0.1355	0.2802	0.1627	0.1564	0.1096	0.2300	0.1332	0.1322	0.0889	0.2023	0.1114
RSL	0.2092	0.1502	0.3001	0.1703	0.1564	0.1096	0.2300	0.1332	0.1423	0.0825	0.1926	0.1028
POD	0.2105	0.1539	0.2889	0.1782	0.1931	0.1404	0.2677	0.1639	0.1461	0.1029	0.2119	0.1244
RDRec	0.2747	0.2033	0.3683	0.2326	0.2572	0.1902	0.3380	0.2160	0.1655	0.1171	0.2375	0.1398
ELMRec	0.5782	0.4792	0.6479	0.4852	0.6052	0.4852	0.6794	0.4973	0.5178	0.4051	0.6045	0.4141
IGSRec	**0.6332**	**0.5320**	**0.7155**	**0.5420**	**0.6504**	**0.5133**	**0.7362**	**0.5248**	**0.5573**	**0.4449**	**0.6457**	**0.4573**
Impv.	+9.5%	+11.0%	+10.4%	+11.7%	+7.5%	+5.8%	+8.4%	+5.5%	+7.6%	+9.8%	+6.8%	+10.4%

TABLE 2. Performance comparison on direct recommendation. Bold and underline indicate the best and second-best results, respectively.

Models	Sports				Beauty				Toys			
	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
Traditional Approach												
Caser	0.0116	0.0072	0.0194	0.0097	0.0205	0.0131	0.0347	0.0176	0.0166	0.0107	0.0270	0.0141
GRU4Rec	0.0129	0.0086	0.0204	0.0110	0.0164	0.0099	0.0283	0.0137	0.0097	0.0059	0.0176	0.0084
HGN	0.0189	0.0120	0.0313	0.0159	0.0325	0.0206	0.0512	0.0266	0.0321	0.0221	0.0497	0.0277
SASRec	0.0233	0.0154	0.0350	0.0192	0.0387	0.0249	0.0605	0.0318	0.0463	0.0306	0.0675	0.0374
Transformer-based Approach												
BERT4Rec	0.0115	0.0075	0.0191	0.0099	0.0203	0.0124	0.0347	0.0170	0.0116	0.0071	0.0203	0.0099
FDSA	0.0182	0.0122	0.0288	0.0156	0.0267	0.0163	0.0407	0.0208	0.0228	0.0140	0.0381	0.0189
S^3-Rec	0.0251	0.0161	0.0385	0.0204	0.0387	0.0244	0.0647	0.0327	0.0443	0.0294	0.0700	0.0376
LLM-based Approach												
P5	0.0387	0.0312	0.0460	0.0336	0.0508	0.0379	0.0664	0.0429	0.0648	0.0567	0.0709	0.0587
RSL	0.0392	0.0330	0.0512	0.0375	0.0508	0.0381	0.0667	0.0446	0.0676	0.0583	0.0712	0.0596
POD	0.0497	0.0399	0.0585	0.0422	0.0559	0.0420	0.0696	0.0471	0.0692	0.0589	0.0744	0.0601
RDRec	0.0505	0.0408	0.0596	0.0433	0.0601	0.0461	0.0743	0.0504	0.0723	0.0593	0.0802	0.0605
ELMRec	0.0538	0.0453	0.0616	0.0471	0.0609	0.0486	0.0750	0.0529	0.0713	0.0608	0.0764	0.0618
IGSRec	**0.0541**	**0.0445**	**0.0618**	**0.0467**	**0.0641**	**0.0505**	**0.0763**	**0.0544**	**0.0732**	**0.0636**	**0.0777**	**0.0644**
Impv.	+0.6%	-1.8%	+0.3%	-0.9%	+5.3%	+3.9%	+1.7%	+2.8%	+1.2%	+4.6%	-3.1%	+4.2%

TABLE 3. Performance comparison on sequential recommendation. Bold and underline indicate the best and second-best results, respectively.

based on validation performance: Toys ($\alpha = 11$, $\beta = 11$), Sports ($\alpha = 11$, $\beta = 11$), and Beauty ($\alpha = 13$, $\beta = 13$). To construct semantic profiles, we employ GPT-4o-mini[2] for summarization and text-embedding-3-large[3] for encoding. We train the retriever with BCEWithLogitsLoss for two epochs. During inference, it selects the top $K = 10$ entities (users or items) for each anchor node to refine semantic graph.

Model training alternates between direct, sequential, and explanation tasks rather than mixing them within a batch to avoid inefficiencies from large variations in input length. We optimize our model with AdamW (learning rate 0.01) for all datasets and apply early stopping with a patience of five epochs based on total validation loss. During inference, beam search explores multiple decoding candidates, and the sequence with the highest likelihood is selected as output. For direct and sequential recommendation, we fix the beam size at 20, while for explanation generation we employ group beam search with 21 beams organized into 3 groups. All experiments are conducted on an NVIDIA RTX 6000 Ada.

[2]https://platform.openai.com/docs/models/gpt-4o-mini
[3]https://platform.openai.com/docs/models/text-embedding-3-large

VI. EXPERIMENTAL RESULTS AND DISCUSSION

Variants	Sports		Beauty		Toys	
	H@10	N@10	H@10	N@10	H@10	N@10
w/o Semantic	0.6479	0.4852	0.6794	0.4973	0.6045	0.4141
w/o Interaction	0.3082	0.1877	0.2827	0.1744	0.1918	0.1053
w/o Refine	0.5999	0.4263	0.5631	0.3859	0.4530	0.3070
IGSRec	**0.7155**	**0.5420**	**0.7362**	**0.5248**	**0.6457**	**0.4573**

TABLE 4. Ablation results of IGSRec. "w/o Semantic" removes the refined semantic graph, leaving only the user–item interaction graph. "w/o Interaction" removes the user–item interaction graph, keeping only the refined semantic graph. "w/o Refine" replaces the refined semantic graph with a naive top-k semantic graph without refinement.

Models	Sports		Beauty		Toys	
	H@10	N@10	H@10	N@10	H@10	N@10
w/o Profile	0.6075	0.4416	0.6022	0.4250	0.4803	0.3206
IGSRec	**0.7155**	**0.5420**	**0.7362**	**0.5248**	**0.6457**	**0.4573**

TABLE 5. Effect of profile embedding initialization on the refined semantic graph. "w/o Profile" initializes user/item profiles randomly instead of LLM-based embeddings.

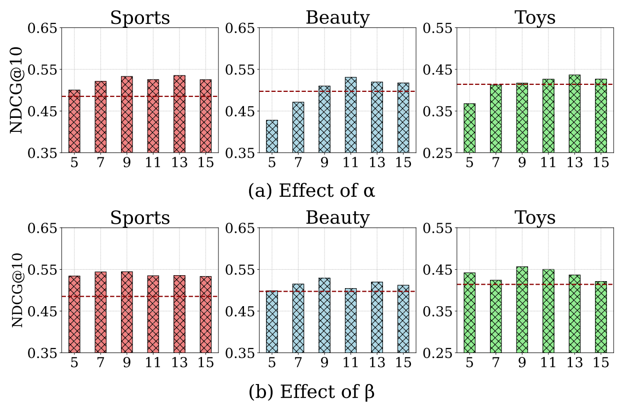


FIGURE 5. Effect of graph fusion weights α (interaction) and β (semantic) on direct recommendation performance across three datasets. The red dashed line denotes the ELMRec baseline for reference, highlighting the robustness of IGSRec across different weight settings.

A. OVERALL PERFORMANCE

Tables 2 and 3 summarize the results on direct and sequential recommendation tasks, respectively. Overall, IGSRec achieves the best performance across all datasets and tasks. In direct recommendation, IGSRec consistently outperforms the strongest baseline with relative improvements ranging from +5.5% to +11.7% in Hit@10 and NDCG@10. For sequential recommendation, IGSRec attains comparable or superior results, with improvements up to +5.3%. These findings highlight that our framework not only strengthens direct matching of user–item preferences but also remains competitive in capturing sequential behaviors. We highlight the following observations:

(1) GNN-based methods such as LightGCN, NCL, and XSimGCL serve as strong baselines by effectively capturing higher-order structural patterns. However, they overlook semantic information that can complement structural modeling. Beyond these methods, IGSRec incorporates graph awareness with semantic refinement, yielding consistent improvements over competitive GNN baselines.

(2) LLM-based approaches such as P5, RSL, POD, and RDRec lag behind GNN-based methods in direct recommendation, highlighting the necessity of modeling higher-order interaction signals. ELMRec addresses this gap by integrating graph awareness into LLMs, achieving notable improvements over prior baselines. Extending this line of work, IGSRec introduces a refined semantic graph that adjusts semantic connections based on observed interactions, thereby enabling a more robust unification of semantic and structural signals.

(3) In contrast to traditional or Transformer-based baselines, which struggle to model user behavior in sequential recommendation, LLM-based methods demonstrate clear advantages. Leveraging pre-trained knowledge, models such as P5, RSL, POD, and ELMRec — as well as IGSRec — exploit diverse textual prompts to capture user preferences and item properties. This enables them to more effectively represent sequential behavioral patterns.

B. ABLATION STUDY

We further analyze how each component contributes to the overall performance of IGSRec through an ablation study (Tables 4–5). The analysis yields three main findings:

1) Complementarity of structural and semantic signals

Eliminating either the user–item interaction graph or the refined semantic graph leads to a noticeable drop in accuracy. As shown in Table 4, the interaction-only variant fails to leverage semantic alignment beyond observed behaviors, limiting its ability to generalize user preferences. Conversely, the semantic-only variant neglects collaborative filtering signals that encode higher-order connectivity, resulting in unstable and incomplete recommendations. When combined, the two graphs consistently achieve the best results, confirming that structural and semantic signals provide complementary strengths and must be jointly exploited.

2) Effectiveness of interaction-grounded refinement

Table 4 also compares our refinement module with a naive cosine-similarity graph. Without refinement, semantic connections are often dominated by spurious similarities or noisy links, which accumulate during propagation and distort learned representations. In contrast, the refinement process selectively strengthens semantic edges that are consistent with observed interactions, filtering out irrelevant neighbors and reinforcing meaningful ones. This refinement module produces reliable semantic bridges between users and items, which not only enhance accuracy but also stabilize training by reducing noise in high-order propagation. The results highlight the importance of refinement of semantic relations in behavioral evidence rather than relying solely on raw similarity.

Hop Size	Sports		Beauty		Toys	
	H@10	N@10	H@10	N@10	H@10	N@10
2	**0.7155**	**0.5420**	**0.7362**	**0.5248**	**0.6457**	**0.4573**
3	0.7083	0.5368	0.6984	0.5241	0.6419	0.4483
10	0.7111	0.5369	0.7398	0.5176	0.5800	0.4208

TABLE 6. Effect of hop size on recommendation performance across three datasets. Specifically, hop size controls the neighborhood depth considered when selecting candidate edges in the refined semantic graph.

3) Sensitivity to profile embedding quality

Initializing the semantic graph with random profile embeddings causes a sharp degradation in performance, as clearly shown in Table 5. Although embeddings can still propagate, the absence of coherent semantics leads to weak and fragmented relational signals, preventing the graph from capturing meaningful user–item associations. By contrast, LLM-based profiles provide semantically consistent and context-aware representations that reflect user preferences and item attributes in a richer way. When these profiles are used as initialization, the refined semantic graph is able to amplify informative correlations while suppressing noise, ultimately enabling stronger semantic bridges and more robust recommendation accuracy. These results demonstrate that the refined semantic graph is highly sensitive to the quality of its profile initialization, and semantically grounded embeddings are essential for unlocking its full potential.

C. HYPERPARAMETER SENSITIVITY ANALYSIS

We further analyze the sensitivity of IGSRec to three key hyperparameters: the fusion weights α and β, and the hop size used for candidate retrieval in refined semantic graph construction. For the fusion weights, we fixed one parameter to a reference value selected from validation results while varying the other to observe its individual effect.

1) Effect of Graph Fusion Weights α and β

As shown in Figure 5, the parameters α and β control the balance between higher-order collaborative patterns from the interaction graph and higher-order semantic patterns from the refined semantic graph, respectively. To analyze the sensitivity of these fusion weights, we conduct two sets of experiments: (1) varying $\alpha \in \{5, 7, 9, 11, 13, 15\}$ while fixing $\beta = 13$, and (2) varying $\beta \in \{5, 7, 9, 11, 13, 15\}$ while fixing $\alpha = 13$.

Increasing α initially improves direct recommendation performance by leveraging structural propagation to capture higher-order collaborative signals. However, excessive reliance on interaction graphs causes performance to saturate and eventually decline, as collaborative patterns alone cannot adequately address semantic nuances in user preferences. This limitation stems from the inherent sparsity of interaction data, which fails to capture the rich semantic relationships encoded in textual profiles. Similarly, moderate levels of β consistently enhance recommendation quality by complementing collaborative patterns with semantic information. Nevertheless, excessively low or high values lead to performance degradation. When β is too low, semantic information becomes underrepresented, failing to mitigate the preference gap between textual similarity and actual user behavior. Conversely, when β is excessively high, semantic distortion is amplified, as profile-based connections may introduce noise that misaligns with real interaction patterns.

Importantly, IGSRec demonstrates robust performance across a broad range of α and β values, validating the effectiveness of our refined semantic graph. By refining semantic connections using observed interactions, IGSRec automatically aligns textual similarity with behavioral patterns, alleviating the trade-off between collaborative and semantic signals. This design enables our framework to synergistically integrate both information sources, achieving superior performance without requiring excessive hyperparameter tuning.

2) Effect of Hop Size

To investigate how the neighborhood depth influences the construction of the refined semantic graph, we varied the hop size, which specifies the maximum hop distance from the anchor nodes (i.e., user node and item node) to candidate target nodes. As shown in Table 6, setting the hop size to 2 consistently achieves the best performance in terms of both Hit@10 and NDCG@10 across all datasets. These results highlight a key property of the refinement process: while the refined semantic graph effectively leverages local structural context to enrich semantic connections, extending the search beyond two hops tends to introduce noise or less relevant neighbors. By restricting candidate selection to closer target nodes, the model captures more reliable and informative structural signals, thereby improving recommendation accuracy.

D. FURTHER ANALYSIS

To better understand how IGSRec addresses the limitations of both sparse interaction graphs and naive semantic graphs, we conduct in-depth structural and empirical analyses. We first examine how the refined semantic graph complements the user–item interaction graph by providing densified connections grounded in semantic similarity and aligned with observed behaviors. We then analyze how the refined semantic graph differs from top-k graphs constructed purely from profile similarity, demonstrating fundamental improvements in edge composition and path connectivity that address preference gap and semantic distortion. Additionally, we present a case study visualizing the embedding space and provide computational complexity analysis to validate practical efficiency.

1) Impact of Refined Semantic Graph over Interaction Graph

We first examine how the refined semantic graph addresses the structural limitations of sparse interaction graphs. Figure 6 compares the degree distributions of users in the interaction graph (blue) and the refined semantic graph (green). The interaction graph exhibits a highly skewed distribution, with

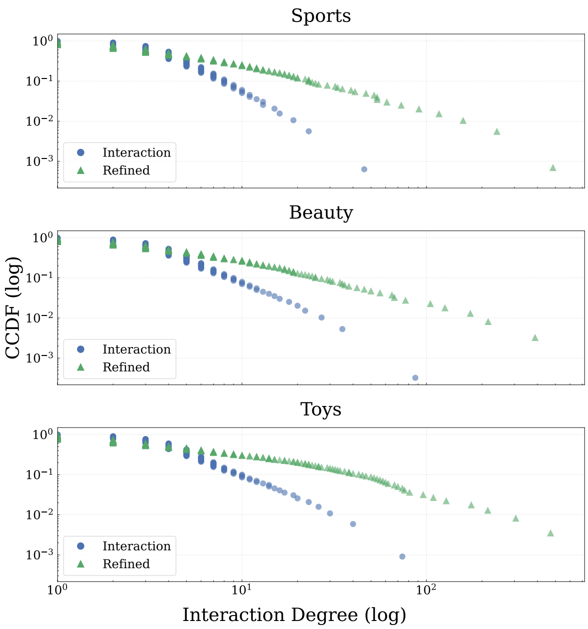


FIGURE 6. Comparison of interaction degree distributions. Interaction degree indicates the number of items each user has interacted with.

the majority of users having very few interactions—a characteristic manifestation of sparsity and cold-start problems in collaborative filtering. In contrast, the refined semantic graph shows a notably flatter slope, indicating that a larger proportion of users maintain sufficient connections. This structural densification is achieved not through arbitrary link addition, but by discovering semantically similar neighbors aligned with behavioral evidence, thereby enriching collaborative signals while maintaining connection quality.

This densification directly translates to improved cold-start robustness. Users with few historical interactions in the original graph can now leverage semantic connections to similar users and relevant items, enabling the model to infer preferences even with limited behavioral data. By complementing the sparse interaction structure with behaviorally grounded semantic relationships, the refined semantic graph provides a richer substrate for GNN propagation, allowing higher-order collaborative patterns to emerge even in data-scarce scenarios.

2) Impact of Refined Semantic Graph over Semantic Graph

We next analyze how interaction-grounded refinement transforms the naive top-k semantic graph constructed solely from profile similarity. As shown in Figure 8, the two approaches exhibit fundamentally different edge compositions. The top-k semantic graph, which selects neighbors based purely on textual similarity, is heavily dominated by user-user and item-item edges. This homogeneous clustering naturally arises from profile-based matching but completely disregards behavioral patterns and fails to establish meaningful cross-type connections. In contrast, the refined semantic graph produces a different edge distribution by aligning semantic neighborhoods with observed interactions. The refined graph introduces abundant user-item edges (93.1% in Sports, 93.6% in Beauty, and 38.4% in Toys) effectively bridging the semantic divide between users and items. This composition shift is crucial: while user-user and item-item edges capture homogeneous similarities, user-item edges directly model preference relationships essential for recommendation. By grounding these connections in behavioral signals rather than profile matching alone, the refined semantic graph addresses both the preference gap (C1) and semantic distortion (C2) challenges identified in Section 1.

Figure 7 further illustrates this refinement through shortest path distance distributions. In the top-k semantic graph (Figure 7(a)), path lengths concentrate around 6–8 hops, with substantial fractions extending beyond 10 hops. This elongated structure indicates that profile-based clusters operate largely independently of interaction patterns, requiring numerous intermediate nodes to bridge users and items. Conversely, the refined semantic graph (Figure 7(b)) exhibits tighter connectivity: over 70% of user-item paths concentrate within 2–4 hops across all datasets. This compression enables more efficient information propagation during GNN aggregation, allowing representations to incorporate richer neighborhood signals within fewer hops. Importantly, the refined semantic graph retains a meaningful fraction of long-range paths (>10 hops), particularly in Toys (18.3%), maintaining global semantic diversity alongside local behavioral alignment. This balance contributes to IGSRec's robust performance across diverse recommendation scenarios.

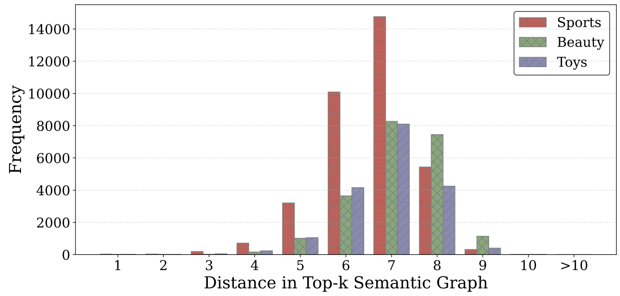
3) Directional Distance Encodings(DDE)

Figure 9 provides a visual illustration of Directional Distance Encoding (DDE). The leftmost column marks the anchor node in color, which serves as the diffusion. Moving from left to right, the figure depicts multiple rounds of graph diffusion along undirected edges. At each round, the diffusion weights are propagated outward from the anchor node to its neighbors, gradually expanding to more distant nodes. The color intensity of each node encodes the magnitude of its diffusion weight, with darker shades indicating stronger influence. Through this iterative process, DDE effectively captures both the proximity and directional structure of entities relative to the anchor, allowing the model to embed nodes with awareness of their multi-hop distances in the graph.

4) Computational Complexity Analysis

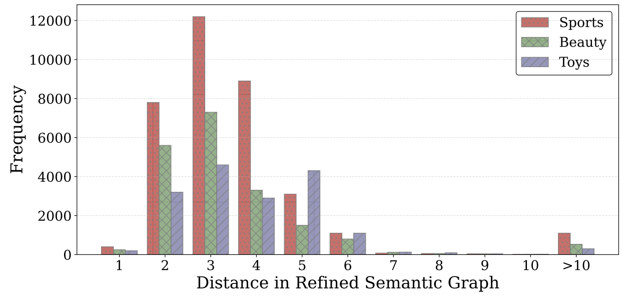
The computational complexity of IGSRec primarily stems from two components: (1) the LLM backbone for encoding prompts and user/item profiles, and (2) the refinement module for constructing the refined semantic graph.

Profile generation is a one-time processing step that requires exactly $|U| + |V|$ LLM calls (one per entity). Thus, the cost scales linearly with dataset size, avoiding any pairwise $O(|U| \times |V|)$ explosion. The subsequent embedding of profiles is performed via batch processing (batch size = 100),



Datasets	1	2	3	4	5	6	7	8	9	10	>10
Sports	31	35	196	714	3,206	10,087	14,767	5,437	321	2	0
Beauty	6	5	22	167	1,011	3,647	8,271	7,444	1,138	5	0
Toys	19	10	50	231	1,055	4,158	8,101	4,249	404	5	0

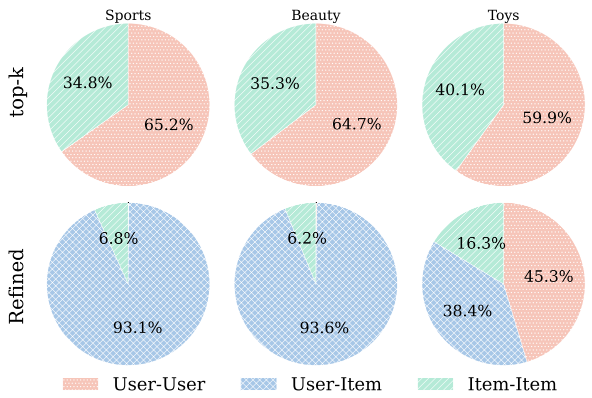
(a) Top-k Semantic Graph



Datasets	1	2	3	4	5	6	7	8	9	10	>10
Sports	391	7,812	12,087	8,941	3,078	1,092	83	64	41	17	1,190
Beauty	243	5,579	7,312	3,288	1,492	804	118	63	34	19	764
Toys	197	3,212	4,588	2,617	4,283	1,487	133	87	44	21	613

(b) Refined Semantic Graph

FIGURE 7. Comparison of shortest path distance distributions between user–item pairs. The refined semantic graph substantially shortens path lengths (mostly concentrated within 2–4 hops) compared to the top-k semantic graph (largely concentrated around 6–8 hops), indicating a closer alignment with observed interaction behaviors and more efficient connectivity between users and items.



(a) All edge types

	Sports	Beauty	Toys
Top-K Semantic Graph	117	20	39
Refined Semantic Graph	76,532	34,544	24,038

(b) User–Item edge type

FIGURE 8. Edge distribution comparison between the initial top-k semantic graph and the refined semantic graph.

reducing embedding API calls to $\lceil(|U| + |V|)/100\rceil$ rather than one per entity, making this stage negligible in both cost and latency.

Notably, the refinement operates on candidate subgraphs sampled from the top-k semantic graph rather than the full user–item interaction graph, ensuring that computational overhead scales with the number of retrieved entities K rather than the total graph size. During inference, the refined semantic graph is pre-constructed and remains static, eliminating repeated refinement costs. As a result, IGSRec's inference complexity is comparable to that of ELMRec [14], with only limited additional overhead from the initial semantic graph construction and refined semantic graph propagation over the enriched structure.

In short, IGSRec introduces small, amortizable pre-computation costs, while keeping online inference overhead negligible, making it practical for real-world deployment.

VII. CONCLUSION AND FUTURE WORK

In this paper, we presented IGSRec, an interaction-grounded semantic recommender that addresses the fundamental limitation of existing LLM-based recommendation systems in capturing higher-order semantic patterns from textual information. By constructing a refined semantic graph that aligns profile-based connections with actual interaction patterns through a semantic graph refinement module, our approach effectively mitigates the preference gap and semantic distortion inherent in naive similarity-based methods. Experimental results on Amazon Review benchmarks demonstrated that IGSRec achieves state-of-the-art performance in both direct and sequential recommendation tasks, with significantly improved robustness in sparse and cold-start scenarios where traditional interaction-based approaches struggle.

Future research could explore several promising directions. First, extending our framework to dynamic recommendation settings would provide insights into temporal semantic relationships and adaptive graph refinement. Second, applying refined semantic graphs to broader domains, including social network analysis and cross-domain recommendation, could validate the generalizability of aligning similarity-based structures with behavioral signals. Finally, extending our approach to incorporate multimodal information (e.g., images, videos) alongside textual profiles may enrich semantic representations and unlock new avenues for improving recommendation quality.

ACKNOWLEDGMENT

(Wooseok Jeong and Young-Jin Kim are co-first authors.)

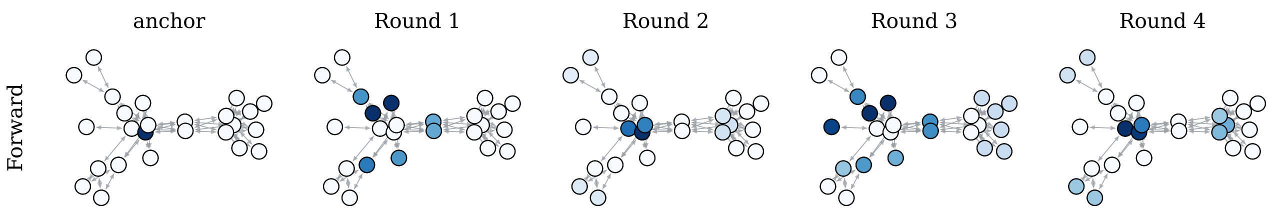


FIGURE 9. A visual illustration of Directional Distance Encodings(DDE), where the leftmost column highlights the anchor node in color.

This work was conducted while Wooseok Jeong and Young-Jin Kim were undergraduate interns at the Graph & Language Intelligence (GLI) Lab, Konkuk University.

REFERENCES

[1] B. Oh, Y. Kim, and B. Min, "Dual-oriented contrast for recommendation with a stop-gradient operation," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 1939–1948.

[2] N. Glisovic, D. Kragic, and M. Tegner, "Item cold start in e-commerce recommender systems: A survey," *IEEE Access*, 2025.

[3] J. Li, J. Zhu, Q. Bi, G. Cai, L. Shang, Z. Dong, X. Jiang, and Q. Liu, "Miner: Multi-interest matching network for news recommendation," in *Findings of the association for computational linguistics: ACL 2022*, 2022, pp. 343–352.

[4] X. Huang, J. Lian, Y. Lei, J. Yao, D. Lian, and X. Xie, "Recommender ai agent: Integrating large language models for interactive recommendations," *ACM Transactions on Information Systems*, vol. 43, no. 4, pp. 1–33, 2025.

[5] W. Wei, X. Ren, J. Tang, Q. Wang, L. Su, S. Cheng, J. Wang, D. Yin, and C. Huang, "Llmrec: Large language models with graph augmentation for recommendation," in *Proceedings of the 17th ACM international conference on web search and data mining*, 2024, pp. 806–815.

[6] J. Lee, Y. Jang, O.-W. Kwon, and H. Kim, "Does the emotional understanding of LVLMs vary under high-stress environments and across different demographic attributes?" in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 23 196–23 210.

[7] H. Won, B. Oh, H. Yang, and K.-H. Lee, "Cross-modal contrastive learning for aspect-based recommendation," *Information Fusion*, vol. 99, p. 101858, 2023.

[8] H. Joo, S. Hong, H. Han, Y. Yoon, and J. Lee, "Prototype-based explanation for semantic gap reduction with distributional embedding," *IEEE Access*, 2025.

[9] S. Geng, S. Liu, Z. Fu, Y. Ge, and Y. Zhang, "Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5)," in *Proceedings of the 16th ACM conference on recommender systems*, 2022, pp. 299–315.

[10] L. Li, Y. Zhang, and L. Chen, "Prompt distillation for efficient LLM-based recommendation," in *Proceedings of the 32nd ACM international conference on information and knowledge management*, 2023, pp. 1348–1357.

[11] S. Geng, J. Tan, S. Liu, Z. Fu, and Y. Zhang, "VIP5: Towards Multimodal Foundation Models for Recommendation," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 9606–9620.

[12] W. Hua, Y. Ge, S. Xu, J. Ji, and Y. Zhang, "UP5: Unbiased Foundation Model for Fairness-aware Recommendation," in *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, 2024, pp. 1692–1706.

[13] W. Wang, H. Bao, X. Lin, J. Zhang, Y. Li, F. Feng, S.-K. Ng, and T.-S. Chua, "Learnable item tokenization for generative recommendation," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 2400–2409.

[14] X. Wang, J. Cui, F. Fukumoto, and Y. Suzuki, "Enhancing high-order interaction awareness in LLM-based recommender model," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 11 696–11 711.

[15] X. Wang, J. Cui, Y. Suzuki, and F. Fukumoto, "RDRec: Rationale distillation for LLM-based recommendation," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2024, pp. 65–74.

[16] J. Tan, S. Xu, W. Hua, Y. Ge, Z. Li, and Y. Zhang, "IDGenRec: LLM-RecSys Alignment with Textual ID Learning," in *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, 2024, pp. 355–364.

[17] Y. Li, X. Zhai, M. Alzantot, K. Yu, I. Vulić, A. Korhonen, and M. Hammad, "CALRec: Contrastive Alignment of Generative LLMs for Sequential Recommendation," in *Proceedings of the 18th ACM Conference on Recommender Systems*, 2024, pp. 422–432.

[18] A. Damianou, F. Fabbri, P. Gigioli, M. De Nadai, A. Wang, E. Palumbo, and M. Lalmas, "Towards Graph Foundation Models for Personalization," in *Companion Proceedings of the ACM Web Conference 2024*, 2024, pp. 1798–1802.

[19] X. Huang, K. Han, Y. Yang, D. Bao, Q. Tao, Z. Chai, and Q. Zhu, "Can gnn be good adapter for llms?" in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 893–904.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[21] B. Oh, I. Suh, K. Cha, J. Kim, G. Park, and S. Jeong, "Pre-training local and non-local geographical influences with contrastive learning," *Knowledge-Based Systems*, vol. 259, p. 110016, 2023.

[22] C. Fan and S. Fujita, "Reinforcement learning-based recommender systems enhanced with graph neural networks," *IEEE Access*, 2025.

[23] Z. Lin, C. Tian, Y. Hou, and W. X. Zhao, "Improving graph collaborative filtering with neighborhood-enriched contrastive learning," in *Proceedings of the ACM web conference 2022*, 2022, pp. 2320–2329.

[24] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "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*, 2020, pp. 639–648.

[25] M. Li, S. Miao, and P. Li, "Simple Is Effective: The Roles of Graphs and Large Language Models in Knowledge-Graph-Based Retrieval-Augmented Generation," in *ICLR 2025 Workshop on Foundation Models in the Wild*, 2025.

[26] H. Cheng, S. Wang, W. Lu, W. Zhang, M. Zhou, K. Lu, and H. Liao, "Explainable recommendation with personalized review retrieval and aspect learning," *arXiv preprint arXiv:2306.12657*, 2023.

[27] Z. He, Z. Xie, R. Jha, H. Steck, D. Liang, Y. Feng, B. P. Majumder, N. Kallus, and J. McAuley, "Large language models as zero-shot conversational recommenders," in *Proceedings of the 32nd ACM international conference on information and knowledge management*, 2023, pp. 720–730.

[28] X. Yu, L. Zhang, X. Zhao, Y. Wang, and Z. Ma, "RA-Rec: An efficient id representation alignment framework for llm-based recommendation," *arXiv preprint arXiv:2402.04527*, 2024.

[29] J. Liao, S. Li, Z. Yang, J. Wu, Y. Yuan, and X. Wang, "Llara: Aligning large language models with sequential recommenders," *CoRR*, 2023.

[30] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[31] C.-Y. Hsieh, C.-L. Li, C.-K. Yeh, H. Nakhost, Y. Fujii, A. Ratner, R. Krishna, C.-Y. Lee, and T. Pfister, "Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes," *arXiv preprint arXiv:2305.02301*, 2023.

[32] J. Yu, X. Xia, T. Chen, L. Cui, N. Q. V. Hung, and H. Yin, "Xsimgcl: Towards extremely simple graph contrastive learning for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 2, pp. 913–926, 2023.

[33] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165–174.
[34] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. van den Driessche, J.-B. Lespiau, B. Damoc, A. Clark, D. de Las Casas, A. Guy, J. Menick, R. Ring, T. Hennigan, S. Huang, L. Maggiore, C. Jones, A. Cassirer, A. Brock, M. Paganini, G. Irving, O. Vinyals, S. Osindero, K. Simonyan, J. W. Rae, E. Elsen, and L. Sifre, "Improving language models by retrieving from trillions of tokens," in *International conference on machine learning*, 2022, pp. 2206–2240.
[35] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang, and S. Tang, "Graph retrieval-augmented generation: A survey," *arXiv preprint arXiv:2408.08921*, 2024.
[36] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, and J. Larson, "From Local to Global: A Graph RAG Approach to Query-Focused Summarization," *arXiv preprint arXiv:2404.16130*, 2024.
[37] P. Xu, W. Ping, X. Wu, L. McAfee, C. Zhu, Z. Liu, S. Subramanian, E. Bakhturina, M. Shoeybi, and B. Catanzaro, "Retrieval meets long context large language models," 2024. [Online]. Available: https://arxiv.org/abs/2310.03025
[38] L. Luo, Y.-F. Li, G. Haffari, and S. Pan, "Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning," in *The Twelfth International Conference on Learning Representations*, 2024.
[39] L. Luo, Z. Zhao, G. Haffari, Y.-F. Li, C. Gong, and S. Pan, "Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models," *arXiv preprint arXiv:2410.13080*, 2024.
[40] T. Guo, Q. Yang, C. Wang, Y. Liu, P. Li, J. Tang, D. Li, and Y. Wen, "Knowledgenavigator: Leveraging large language models for enhanced reasoning over knowledge graph," *Complex & Intelligent Systems*, vol. 10, no. 5, pp. 7063–7076, 2024.
[41] J. Kim, Y. Kwon, Y. Jo, and E. Choi, "Kg-gpt: A general framework for reasoning on knowledge graphs using large language models," *arXiv preprint arXiv:2310.11220*, 2023.
[42] X. Li, D. Lian, Z. Lu, J. Bai, Z. Chen, and X. Wang, "Graphadapter: Tuning vision-language models with dual knowledge graph," *Advances in Neural Information Processing Systems*, vol. 36, pp. 13 448–13 466, 2023.
[43] Y. Wen, Z. Wang, and J. Sun, "Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models," *arXiv preprint arXiv:2308.09729*, 2023.
[44] K. Mao, J. Zhu, J. Wang, Q. Dai, Z. Dong, X. Xiao, and X. He, "Simplex: A simple and strong baseline for collaborative filtering," 2023. [Online]. Available: https://arxiv.org/abs/2109.12613
[45] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2016. [Online]. Available: https://arxiv.org/abs/1511.06939
[46] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," 2018. [Online]. Available: https://arxiv.org/abs/1809.07426
[47] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE international conference on data mining (ICDM)*. IEEE, 2018, pp. 197–206.
[48] C. Ma, P. Kang, and X. Liu, "Hierarchical gating networks for sequential recommendation," 2019. [Online]. Available: https://arxiv.org/abs/1906.09217
[49] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450.
[50] T. Zhang, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, D. Wang, G. Liu, and X. Zhou, "Feature-level deeper self-attention network for sequential recommendation," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, ser. IJCAI'19. AAAI Press, 2019, p. 4320–4326.
[51] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, "S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. ACM, Oct. 2020, p. 1893–1902. [Online]. Available: http://dx.doi.org/10.1145/3340531.3411954
[52] Z. Chu, H. Hao, X. Ouyang, S. Wang, Y. Wang, Y. Shen, J. Gu, Q. Cui, L. Li, S. Xue, J. Y. Zhang, and S. Li, "Leveraging large language models for pre-trained recommender systems," 2023. [Online]. Available: https://arxiv.org/abs/2308.10837
[53] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2023. [Online]. Available: https://arxiv.org/abs/1910.10683

WOOSEOK JEONG is expected to receive the B.S. degree in Computer Science and Engineering from Konkuk University, Seoul, South Korea, in 2027. His current research interests graph neural networks, large language models and recommender systems.

YOUNG-JIN KIM is expected to receive the B.S. degree in Applied Statistics from Konkuk University, Seoul, South Korea, in 2027. Her current research interests include graph neural networks, large language models and recommendation systems.

HAE-YOON KOO received the B.S. degree in Computer Science and Engineering from Dong-A University, Busan, South Korea, in 2025. She is currently pursuing the M.S. degree in Computer Science and Engineering at Konkuk University, Seoul, South Korea. Her research interests include recommender systems and large language models.

JINHO CHOI is expected to receive the B.S. degree in Industrial Engineering from Konkuk University, Seoul, South Korea, in 2026. He is currently an undergraduate student in the Department of Industrial Engineering at Konkuk University. His current research interests include Graph Neural Networks, Large Language Models, and GraphRAG.

JIMYEUNG SEO received the B.S. degree in Computer Science and Engineering from Konkuk University, Seoul, South Korea, in 2025. He is currently pursuing the M.S. degree in Artificial Intelligence at Konkuk University, Seoul, South Korea. His current research interests include graph neural networks, large language model optimization, multimodal recommendation systems, and reinforcement learning.

BYUNGKOOK OH received the B.S., M.S. and Ph.D. degrees in Computer Science from Yonsei University, in 2015, 2017, and 2020, respectively. Previously, he worked as a research scientist at Samsung Research. He is currently working as an Assistant Professor with the Department of Computer Science and Engineering, Konkuk University, Seoul, South Korea, where he leads the Graph & Language Intelligence Lab. His current research interests focus on Human-like Artificial Intelligence, particularly in knowledge representation (e.g., knowledge and graph embedding) and knowledge-based applications (e.g., knowledge-enhanced NLP applications, information retrieval, and recommendation systems).