

DCGL: Dual-Channel Graph Learning with Large Language Models for Knowledge-Aware Recommendation

Xinchi Zou

School of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan, China
xinchizou@hust.edu.cn

Tongzhenzhi Su

School of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan, China
tongzhenzhisu@hust.edu.cn

Jianjun Li*

School of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan, China
jianjunli@hust.edu.cn

Yuan Fu

School of Software Engineering, Huazhong University of Science and Technology
Wuhan, China
yuanfu@hust.edu.cn

Chang Liu

School of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan, China
diu@hust.edu.cn

Zhiying Deng

Laboratory for Artificial Intelligence and New Forms of Education, Central China Normal University
Wuhan, China
zhiyingdzy@ccnu.edu.cn

Zhiwei Shen

School of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan, China
shenzhiwei@hust.edu.cn

Abstract

Knowledge Graphs (KGs) have proven highly effective for recommendation systems by capturing latent item relationships, while recent integration of Large Language Models (LLMs) has further enhanced semantic understanding and addressed knowledge sparsity issues. Nevertheless, current KG&LLM-based methods still face three main limitations: 1) inadequate modeling of implicit semantics relationships beyond explicit KG links; 2) suboptimal single-channel fusion of ID and LLM embeddings, which often leads to signal interference and blurred representations; and 3) insufficient consideration of user-item interaction frequency variations in recommendation strategies. To address these challenges, we propose the **Dual-Channel Graph Learning (DCGL)** framework, featuring three key innovations: 1) a dual-channel architecture that structurally decoupling rich semantic information from user behavioral patterns, preventing early interference; 2) a multi-level contrastive learning mechanism that enhances robustness against KG noise through intra-view contrast and bridges semantic gaps between channels via inter-view alignment; and 3) a dynamic fusion mechanism that adaptively balances semantic generalization and behavioral specificity based on interaction frequency, resolving the cascading limitation. Extensive experiments on four real-world

datasets show that DCGL consistently outperforms state-of-the-art methods, yielding substantial improvements in sparse scenarios while maintaining precision for active users. Our code is available at <https://github.com/XinchiZou/DCGL>.

CCS Concepts

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

Keywords

Recommender Systems, Knowledge Graphs, Large Language Models, Graph Neural Networks

ACM Reference Format:

Xinchi Zou, Tongzhenzhi Su, Jianjun Li, Yuan Fu, Chang Liu, Zhiying Deng, and Zhiwei Shen. 2026. DCGL: Dual-Channel Graph Learning with Large Language Models for Knowledge-Aware Recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26)*, July 20–24, 2026, Melbourne, VIC, Australia. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3805712.3809546>

1 Introduction

The rapid proliferation of online platforms has exacerbated the long-standing problem of information overload. Modern recommender systems alleviate this issue by modeling user preferences from historical interactions [9]. Among them, Collaborative Filtering (CF) remains a foundational paradigm, valued for its simplicity and efficacy [11, 12, 22]. However, methods relying solely on user-item interactions are inherently susceptible to data sparsity and cold-start problems [23]. To mitigate these limitations, Knowledge Graphs (KGs) have been integrated as external side information, empowering recommendation models to capture diverse, high-order

*Corresponding author.



This work is licensed under a Creative Commons Attribution 4.0 International License. *SIGIR '26, Melbourne, VIC, Australia*

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2599-9/2026/07

<https://doi.org/10.1145/3805712.3809546>

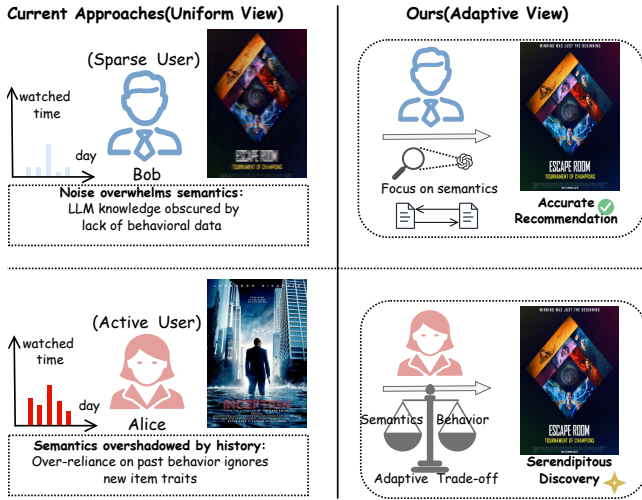


Figure 1: Motivating Example. Addressing frequency heterogeneity by shifting from a uniform view (Left) to an adaptive view (Right) that balances semantic and behavioral signals for different user groups.

relational dependencies among items [25, 26, 32, 33]. More recently, the semantic understanding and reasoning capabilities of Large Language Models (LLMs) have been leveraged to infer user interests [21, 37] and integrate them into collaborative learning [14], as well as to comprehend local and global KG structures to compensate for missing facts and sparsity [6].

Despite encouraging progress, we argue that state-of-the-art knowledge-aware recommenders [6, 21] are trapped in a vicious cycle rooted in frequency heterogeneity. User and item interaction frequencies exhibit a highly skewed distribution, yet mainstream methods forcibly employ uniform signal fusion within single-channel architectures. This practice blurs the fundamental distinction between frequency-sensitive behavioral embeddings (ID embeddings) and frequency-agnostic semantic embeddings (LLM embeddings). As a result, entities with rich interactions suffer from semantic dilution, while those with sparse interactions lack sufficient behavioral grounding. This suboptimal fusion subsequently undermines the model’s ability to capture implicit semantics—subtle relationships not explicitly encoded in the knowledge graph, such as the affinity between “mystery” and “detective” genres. Consequently, a chain reaction ensues: frequency skew complicates the fusion process, while poor fusion in turn hinders semantic discovery.

Motivating Example. Consider recommending a new mystery movie “Escape Room” to two users: Alice (a frequent viewer with hundreds of interactions) and Bob (a new user with few clicks) on a streaming platform, as shown in Figure 1. State-of-the-art models that fuse LLM semantics with collaborative IDs in a single channel struggle in both scenarios. For Bob, weak behavioral signals are overwhelmed by noise, drowning out clear semantic cues and leading to inaccurate recommendations. For Alice, strong historical patterns overshadow subtle semantic affinities (e.g., between the “mind-bending narrative” in her watched sci-fi movies and the new mystery film), leading to missed serendipitous discoveries.

This reveals a cascading limitation: the inability to adapt fusion to frequency heterogeneity (Bob vs. Alice) results in blurred representations, which in turn hinders the effective use of implicit semantics for robust recommendations across the sparsity spectrum.

To break this cycle, we propose **Dual-Channel Graph Learning (DCGL)**, a unified KG-enhanced recommendation framework designed to resolve the cascading problem through synergistic mechanisms. Unlike approaches that treat each challenge in isolation, DCGL employs a coherent strategy built on three pillars: (1) **A dual-channel architecture** that provides the structural basis by decoupling the learning of LLM semantics and ID collaborative signals into parallel pathways, thus preventing early signal interference. (2) **Multi-level contrastive learning** applied within this structure to explicitly shape robust and aligned representations; intra-view contrast enhances robustness within each channel, while inter-view contrast bridges the semantic and behavioral spaces, enabling the ID channel to internalize implicit semantic nuances. (3) **A frequency-aware gated fusion** mechanism that dynamically orchestrates the final integration. Crucially, this gating is not independent. Instead, it is informed by the clarified representations from the contrastive-learning-enhanced channels and is modulated by the interaction frequency, thereby enabling adaptive reliance on semantics or behavior according to sparsity levels. Together, these components work in concert: dual channels enable separation, contrastive learning refines and aligns within and across them, and the adaptive gate achieves intelligent, context-sensitive unification.

Our experiments on four real-world datasets show that DCGL consistently outperforms strong baselines. Ablation studies confirm the contribution of each component and their interdependencies, and detailed analyses demonstrate substantial gains in sparse and cold-start scenarios. Our main contributions are threefold:

- We identify and formalize a cascading limitation in knowledge-aware recommendation, where frequency heterogeneity, suboptimal single-channel fusion, and inadequate modeling of implicit semantics exacerbate one another.
- We propose DCGL, a unified framework that tackles this core problem through the synergistic design of a dual-channel architecture, multi-level contrastive learning, and a frequency-aware gated fusion mechanism.
- We provide comprehensive validation showing DCGL resolves the trade-off between semantic generalization and behavioral specificity, boosting sparse and long-tail performance while preserving precision for active users.

2 Related work

2.1 Knowledge-aware Recommendation

Knowledge-aware recommendation is a key research area within recommender systems, encompassing embedding-based, path-based, and graph neural network (GNN)-based methodologies in recent advancements. Embedding-based approaches [4, 24, 36] leverage relations and entities from knowledge graphs to enrich user and item representations.

Path-based methods [27, 35], on the other hand, focus on extracting meta-paths from knowledge graphs to capture user-item connectivity patterns.

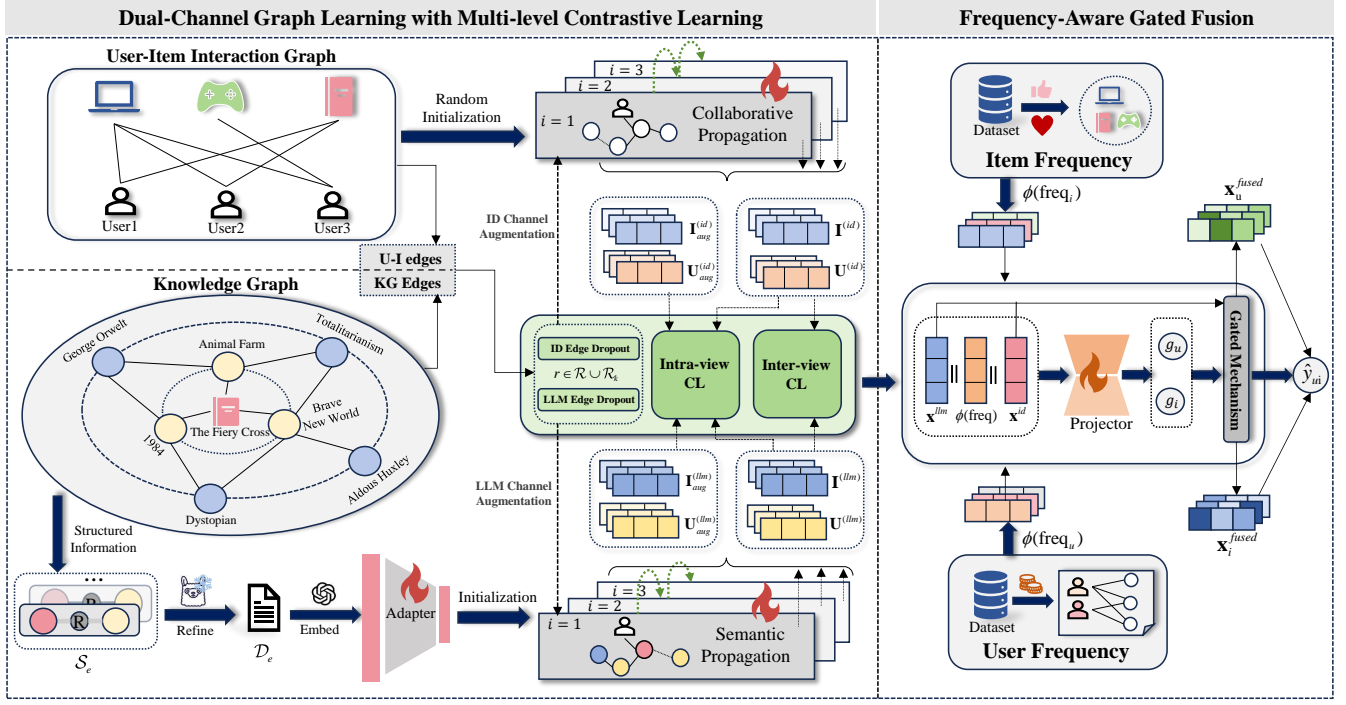


Figure 2: The framework of our proposed DCGL.

Progressing further, GNN-based methods [16, 18, 25, 26, 32, 33, 39, 40], at the research forefront, refine entity and relation representations by aggregating multi-hop neighbor embeddings. Notably, KGAT [25] employs graph attention mechanisms to propagate embeddings based on neighbor importance, enabling end-to-end recommendation score generation; KGRec [32] evaluates triplets with rationale scores, combining MAE-based reconstruction learning with contrastive objectives to improve recommendation performance. However, reliance on incomplete, explicit KG paths leaves these models brittle to missing facts and weak at capturing implicit semantics.

2.2 LLMs for Recommendation

In recent years, applying LLMs to recommender systems has emerged as a research hotspot [5, 30, 38], with LLMs primarily serving two roles: as recommenders or as enhancers. When acting as recommenders, LLMs integrate recommendation tasks into conversational contexts by tokenizing user and item IDs to directly generate recommendations [17]. Benefiting from their extensive world knowledge, LLMs perform well in zero-shot [13] and few-shot scenarios [2], while also offering benefits in fairness [8] and explainability [19]. However, when used as recommenders, LLMs face challenges such as high inference costs, poor top- K calibration, and underuse of collaborative filtering, hindering their scalable deployment. As enhancers, LLMs generate user-item interactions, item features [28], or semantic embeddings [15, 21] to augment traditional recommendation models. However, current enhancer methods typically fuse ID embeddings with LLM-generated semantic embeddings in a single-channel manner, failing to distinguish their unique roles

in capturing behavioral signals and semantic knowledge. This limits the effective integration of LLMs' rich semantic information with the behavioral signals captured by ID embeddings, thereby constraining the accurate modeling of complex user preferences.

2.3 KG&LLM-based Recommendation

Recently, hybrid approaches combining KGs and LLMs have gained popularity for alleviating knowledge sparsity and enriching semantic understanding. For example, CIKGRec [14] uses LLMs to infer user interests, builds a Collaborative Interest KG, and enhances user representations via interest reconstruction and cross-domain contrastive learning. CoLaKG [6] fills missing KG facts by extracting subtrees and using prompt engineering for local information, while incorporating global knowledge through semantic retrieval, optimizing recommendations via representation fusion and retrieval-augmented learning. Despite their potential, these methods still have some limitations: CIKGRec mainly enriches user-side semantics but ignores item-side ones, while CoLaKG heavily depends on LLM accuracy and retrieval quality, risking noise or hallucinations with large or noisy KGs. Moreover, both use static strategies to integrate semantic and collaborative signals, lacking dynamic adaptation to data sparsity or preference changes.

3 Preliminaries

User-Item Interaction Graph. Let \mathcal{U} and \mathcal{I} denote the user set and item set in the recommender system, respectively. We construct a bipartite user-item graph $\mathcal{G} = (\mathcal{U}, \mathcal{I}, \mathcal{R})$, where $\mathcal{R} = \{(u, i) \mid u \in \mathcal{U}, i \in \mathcal{I}, y_{ui} = 1\}$ represents the set of interactions between users and items, with $y_{ui} = 1$ indicating that user u has interacted with

item i , and $y_{ui} = 0$ otherwise. The neighbor sets of user u and item i are defined as $\mathcal{N}_u = \{i \in \mathcal{I} \mid (u, i) \in \mathcal{R}\}$ and $\mathcal{N}_i = \{u \in \mathcal{U} \mid (u, i) \in \mathcal{R}\}$, respectively.

Knowledge Graph. We incorporate real-world knowledge via a heterogeneous graph $\mathcal{G}_k = \{(h, r, t)\}$, defined over an entity set \mathcal{E} and a relation set \mathcal{R}_k . Here, (h, r, t) denotes a triplet where $h, t \in \mathcal{E}$ are entities and $r \in \mathcal{R}_k$ is a relation. Items in the interaction graph are linked to corresponding entities in \mathcal{E} . Formally, we define the item set \mathcal{I} as a subset of the entity set \mathcal{E} ($\mathcal{I} \subseteq \mathcal{E}$), which enables the modeling of rich relational dependencies. For an item $i \in \mathcal{I}$, its adjacent entities in \mathcal{G}_k are given by $\mathcal{N}_i^k = \{e \in \mathcal{E} \mid (i, r, e) \in \mathcal{G}_k\}$.

Task Definition. Given the user-item interaction graph \mathcal{G} and knowledge graph \mathcal{G}_k , our goal is to learn a recommendation model that predicts the preference score \hat{y}_{ui} of user u for item i . The model ranks items based on these scores and recommends the top- K items with the highest scores. The learning is performed by using the observed interactions in \mathcal{R} to infer users' potential preferences.

4 Methodology

We now present the detailed technical design of our proposed DCGL. The overall framework is illustrated in Figure 2, comprising two main components: (1) Dual-Channel Graph Learning with Multi-level Contrastive Learning (left), where the ID channel captures behavioral patterns through collaborative propagation while the LLM channel models semantic knowledge via semantic propagation, with multi-level contrastive learning enhancing both robustness and cross-channel alignment; (2) Frequency-Aware Gated Fusion (right), which adaptively integrates dual-channel representations based on interaction frequencies to produce the final prediction.

4.1 Dual-Channel Graph Learning Architecture

To break the vicious cycle of signal interference in single-channel designs, we introduce a dual-channel graph learning architecture. This structure provides the foundational separation needed to decouple the learning of frequency-sensitive behavioral patterns (ID channel) from frequency-agnostic semantic knowledge (LLM channel), preventing the semantic dilution and grounding issues illustrated in our motivating example.

4.1.1 Entity Embedding Generation. In line with our goal of distinct signal separation, we generate entity embeddings through two parallel, specialized pipelines.

LLM Semantic Channel: To capture rich, transferable semantics, we employ an entity-level description-to-embedding pipeline using Large Language Models (LLMs). As illustrated in Figure 3, structured fields of each entity e , denoted as \mathcal{S}_e (e.g., name, attributes), are injected into a reusable template. We guide the LLM to synthesize these attributes into a coherent, neutral-toned textual description \mathcal{D}_e , effectively capturing nuanced semantic relationships beyond superficial metadata. This description is then encoded by a pre-trained text embedding model to derive a high-dimensional semantic representation:

$$\mathbf{x}_e^{(\text{llm,raw})} = \text{TextEmbedding}(\mathcal{D}_e) \in \mathbb{R}^{d_{\text{llm}}} \quad (1)$$

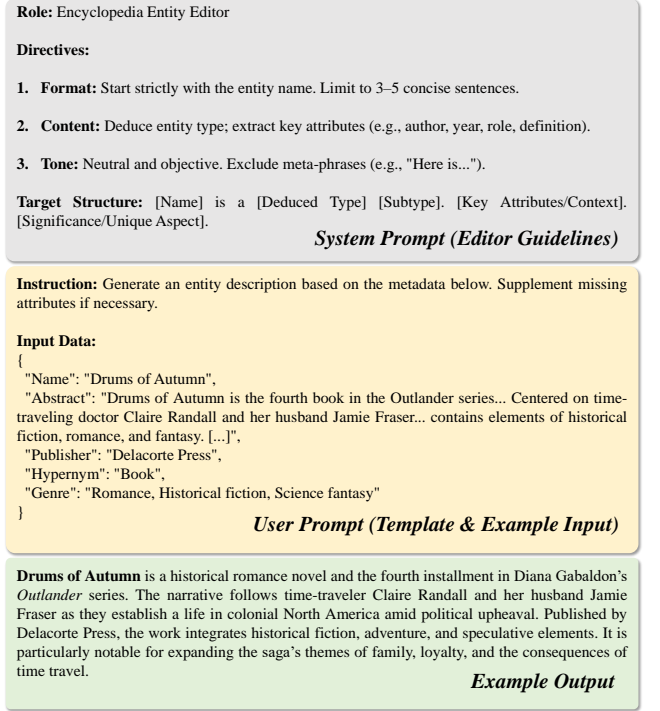


Figure 3: Prompt template for LLM-based entity description, including system prompt with editing rules, user prompt example and generated output.

A trainable adapter then projects this high-dimensional vector into the model's standard embedding space \mathbb{R}^d :

$$\mathbf{x}_e^{(\text{llm})} = \mathbf{W}_2 \cdot \text{LeakyReLU}(\mathbf{W}_1 \mathbf{x}_e^{(\text{llm,raw})} + \mathbf{b}_1) + \mathbf{b}_2, \quad (2)$$

where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2$ are learnable parameters. This process yields semantic embeddings that are independent of interaction frequency.

ID Information Channel: To preserve precise collaborative filtering signals from historical interactions, embeddings in this channel $\mathbf{x}_e^{(\text{id})} \in \mathbb{R}^d$ are randomly initialized and optimized purely on user-item interaction data, creating a dedicated pathway for learning behavioral patterns that are highly sensitive to interaction frequency.

This explicit separation at the input stage establishes the clear distinction between semantic and behavioral signals that is crucial for our subsequent adaptive fusion.

4.1.2 Knowledge Graph and Interaction Signal Modeling.

We model relational information and collaborative effects within each separate channel to maintain signal purity.

Relation-Aware Knowledge Graph Encoding: Inspired by recent advancements in knowledge graph-based methods [25, 32], we employ a relation-aware graph attention network (RGAT) in each channel $c \in \{\text{id}, \text{llm}\}$ to encode the knowledge graph \mathcal{G}_k . This approach dynamically models entity-relation dependencies through parameterized attention mechanisms, thereby obviating the need for manual path specification. Specifically, for an item i

with relational neighbors \mathcal{N}_i^k , the RGAT layer performs:

$$\begin{aligned} \mathbf{x}_i^{(c,l+1)} &= \mathbf{x}_i^{(c,l)} + \sum_{e \in \mathcal{N}_i^k} a_{e,i}^{(c,l)} \mathbf{x}_e^{(c,l)}, \\ a_{e,i}^{(c,l)} &= \frac{\exp\left(\phi\left(\mathbf{r}_{e,i}^{(c,l)}, \mathbf{x}_e^{(c,l)}, \mathbf{x}_i^{(c,l)}\right)\right)}{\sum_{e' \in \mathcal{N}_i^k} \exp\left(\phi\left(\mathbf{r}_{e',i}^{(c,l)}, \mathbf{x}_{e'}^{(c,l)}, \mathbf{x}_i^{(c,l)}\right)\right)}. \end{aligned} \quad (3)$$

where the attention coefficient ϕ is defined by a LeakyReLU-activated bilinear projection:

$$\phi(\mathbf{r}_{e,i}, \mathbf{x}_e, \mathbf{x}_i) = \text{LeakyReLU}\left(\mathbf{r}_{e,i}^\top \mathbf{W}[\mathbf{x}_e \parallel \mathbf{x}_i]\right), \quad (4)$$

with $\mathbf{W} \in \mathbb{R}^{d \times 2d}$ denoting the learnable projection matrix, and $\mathbf{r}_{e,i}^{(c)} \in \mathbb{R}^d$ representing the relation-specific embedding in channel c . The relation embeddings $\mathbf{r}_{e,i}^{(c)}$ in each channel are independently optimized: the ID channel learns behavior-aware relation representations from user-item interaction data, while the LLM channel incorporates semantic knowledge to enhance relational modeling. After L layers of RGAT aggregation, the item representation matrix $\mathbf{I}^{(c)} = \{\mathbf{x}_i^{(c,L)} \mid i \in \mathcal{I}\}$ encodes both semantic information and relational structures from \mathcal{G}_k .

Collaborative Signal Propagation: To further model the collaborative effects in user-item interactions, we employ a graph-based collaborative filtering framework to encode user and item representations. Specifically, we adopt LightGCN [12]’s efficient message propagation strategy to capture high-order collaborative signals through L' propagation layers:

$$\mathbf{x}_u^{(c,l+1)} = \sum_{i \in \mathcal{N}_u} \frac{\mathbf{x}_i^{(c,l)}}{\sqrt{|\mathcal{N}_u| \cdot |\mathcal{N}_i|}}, \quad (5)$$

$$\mathbf{x}_i^{(c,l+1)} = \sum_{u \in \mathcal{N}_i} \frac{\mathbf{x}_u^{(c,l)}}{\sqrt{|\mathcal{N}_i| \cdot |\mathcal{N}_u|}}, \quad (6)$$

where $\mathbf{x}_u^{(c,l)}$ and $\mathbf{x}_i^{(c,l)}$ represent the embeddings of user u and item i at the l -th graph propagation layer in channel c , respectively. Note here $\mathbf{x}_u^{(c,0)}$ is randomly initialized, while $\mathbf{x}_i^{(c,0)}$ is initialized with RGAT output $\mathbf{x}_i^{(c,L)}$. After stacking L' LightGCN layers, each channel c yields the user representation matrix $\mathbf{U}^{(c)} = \{\mathbf{x}_u^{(c)} \mid u \in \mathcal{U}\} \in \mathbb{R}^{|\mathcal{U}| \times d}$ and the item representation matrix $\mathbf{I}^{(c)} = \{\mathbf{x}_i^{(c)} \mid i \in \mathcal{I}\} \in \mathbb{R}^{|\mathcal{I}| \times d}$. The LLM channel thus provides semantically-oriented embeddings, whereas the ID channel focuses on behavioral signals.

4.1.3 Semantic Regularization via Translation-based Training. To further enhance the geometric consistency of the entity and relation embeddings used in RGAT, we incorporate an auxiliary translation-based training objective [3]. We adopt an alternating optimization strategy between the graph propagation and the TransE objective in each channel $c \in \{\text{id}, \text{llm}\}$. The core idea is to enforce the structure $\mathbf{h}^{(c)} + \mathbf{r}^{(c)} \approx \mathbf{t}^{(c)}$, where \mathbf{h}, \mathbf{r} and \mathbf{t} correspond to the embeddings of the head entity, relation, and tail entity in a triplet $(h, r, t) \in \mathcal{G}_k$, respectively. The distance score is defined using the L_1 norm:

$$f_d^{(c)}(h, r, t) = \|\mathbf{h}^{(c)} + \mathbf{r}^{(c)} - \mathbf{t}^{(c)}\|_1. \quad (7)$$

We employ a pairwise ranking loss to discriminate valid triplets from corrupted ones (h, r, t') , which are generated via negative

sampling by randomly replacing the tail entity. The regularization loss for channel c is then formulated as:

$$\mathcal{L}_{\text{TransE}}^{(c)} = \sum_{(h,r,t,t') \in \mathcal{G}_k} -\ln \sigma(f_d^{(c)}(h, r, t') - f_d^{(c)}(h, r, t)), \quad (8)$$

where $\sigma(\cdot)$ is the sigmoid function. This objective optimizes the embeddings such that valid triplets have significantly smaller distances than corrupted ones. Specifically, the loss in the ID channel refines behavior-aware structural embeddings, while in the LLM channel, it ensures that the semantic embeddings maintain valid relational dependencies. By alternating the optimization of $\mathcal{L}_{\text{TransE}}^{(c)}$ with the main recommendation task, DCGL effectively regularizes the representation space without interfering with the gradient flow of the collaborative filtering process.

4.2 Multi-level Contrastive Learning

After generating the semantically and behaviorally dominant representations $\mathbf{U}^{(c)}$ and $\mathbf{I}^{(c)}$, we introduce a multi-level contrastive learning mechanism to enhance the model’s robustness against noise, sparsity, and cross-channel heterogeneity. Here, we treat each representation variant as a distinct *view* of the same entity–graph augmentation produces augmented views within each channel, while the dual-channel outputs naturally form behavioral and semantic views. This mechanism comprises two complementary components: intra-view augmentation for structural stability preservation and inter-view alignment for cross-channel representation calibration.

4.2.1 Intra-View Augmentation. Building upon graph augmentation strategies [33], we implement a two-stage procedure that first perturbs the knowledge graph \mathcal{G}_k and then performs stability-aware perturbation on the user–item interaction graph \mathcal{G} . For \mathcal{G}_k , we generate augmented views through random edge dropout:

$$\mathcal{G}'_k \sim \text{DropEdge}(\mathcal{G}_k, \rho), \quad (9)$$

where $\rho \in (0, 1)$ denotes the edge dropout rate. Both original and augmented graphs are encoded by channel-specific RGAT encoders, yielding item-level stability scores $S_i \in [0, 1]$ via KG edge consistency measurement as defined in [33]. Conditioned on these stability scores, we perform stability-guided interaction graph augmentation:

$$\begin{aligned} \mathcal{G}' &\sim \text{StabAdaptDrop}(\mathcal{G}, p_{\text{drop}}), \\ p_{\text{drop}}(u, i) &= 1 - \mu S_i, \end{aligned} \quad (10)$$

where (u, i) represents an interaction edge and $\mu \in (0, 1]$ controls the stability guidance strength (i.e., higher S_i leads to lower drop probability).

The intra-view contrastive objective is formulated as the sum of losses over both channels:

$$\mathcal{L}_{\text{aug}} = \sum_{c \in \{\text{id}, \text{llm}\}} \left[\mathcal{L}_{\text{InfoNCE}}(\mathbf{U}_{\text{aug}}^{(c)}, \mathbf{U}^{(c)}) + \mathcal{L}_{\text{InfoNCE}}(\mathbf{I}_{\text{aug}}^{(c)}, \mathbf{I}^{(c)}) \right], \quad (11)$$

where $\mathbf{U}^{(c)} \in \mathbb{R}^{|\mathcal{U}| \times d}$ and $\mathbf{I}^{(c)} \in \mathbb{R}^{|\mathcal{I}| \times d}$ denote the original embeddings, and $\mathbf{U}_{\text{aug}}^{(c)}, \mathbf{I}_{\text{aug}}^{(c)}$ represent their augmented counterparts. The InfoNCE loss is defined using cosine similarity:

$$\mathcal{L}_{\text{InfoNCE}}(\mathbf{Z}_1, \mathbf{Z}_2) = - \sum_{n=1}^N \log \frac{\exp(\text{sim}(\mathbf{z}_{1,n}, \mathbf{z}_{2,n})/\tau)}{\sum_{m \neq n} \exp(\text{sim}(\mathbf{z}_{1,n}, \mathbf{z}_{2,m})/\tau)}, \quad (12)$$

where $\mathbf{Z}_1, \mathbf{Z}_2 \in \mathbb{R}^{N \times d}$ are embedding matrices from two distinct views, $\text{sim}(\cdot, \cdot)$ denotes cosine similarity, $\tau > 0$ is a temperature hyperparameter, and N corresponds to the batch size (i.e., $|\mathcal{U}|$ or $|\mathcal{I}|$). This design enhances representation robustness by preserving stable structural information across both \mathcal{G}_k and \mathcal{G} , thereby increasing resilience to noisy or sparse interactions.

4.2.2 Inter-View Alignment. To bridge the inherent heterogeneity between the ID and LLM representation spaces, we introduce an inter-view alignment mechanism. This module employs lightweight projection networks to map user embeddings from both channels into a unified latent space. Structurally, each projection network consists of a linear layer that preserves the embedding dimension, followed by a LeakyReLU activation to introduce non-linearity. Formally, the projection is defined as:

$$\mathbf{U}_{\text{proj}}^{(\text{id})} = \text{Proj}_1(\mathbf{U}^{(\text{id})}), \quad \mathbf{U}_{\text{proj}}^{(\text{llm})} = \text{Proj}_2(\mathbf{U}^{(\text{llm})}). \quad (13)$$

We optimize the alignment via an InfoNCE-based objective, consistent with the intra-view contrastive loss:

$$\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{InfoNCE}}(\mathbf{U}_{\text{proj}}^{(\text{id})}, \mathbf{U}_{\text{proj}}^{(\text{llm})}), \quad (14)$$

where $\mathcal{L}_{\text{InfoNCE}}$ follows the formulation in Eq.(12). Specifically, embeddings of the same user from the projected ID and LLM spaces constitute positive pairs, whereas pairings with different users serve as negatives. Maximizing the mutual information between these views not only reduces the semantic gap but also forces the ID channel to implicitly internalize semantic nuances (e.g., genre affinities) from the LLM channel. This ensures that the behavioral representations remain semantically aware even when the ID channel dominates the final fusion.

4.3 Frequency-Aware Gated Fusion

To adaptively integrate dual-channel representations, we propose a frequency-aware gating mechanism that dynamically modulates the contributions of the ID channel and the LLM channel according to item popularity. This design enables context-sensitive fusion by balancing behavioral patterns (captured by the ID channel) and semantic knowledge (provided by the LLM channel) through popularity-informed gating.

4.3.1 Gating Network Design. The gating network computes fusion weights by incorporating three complementary signals: ID representations, LLM representations, and interaction frequency statistics. For any item $i \in \mathcal{I}$, the gating weight is formulated as:

$$g_i = \sigma \left(\mathbf{W}_g \left[\mathbf{x}_i^{(\text{id})} \parallel \mathbf{x}_i^{(\text{llm})} \right] \phi(\text{freq}_i) \right) + \mathbf{b}_g, \quad (15)$$

where $\phi(\text{freq}_i) = \log(1 + \text{freq}_i) / \log(1 + \max_{j \in \mathcal{I}} \text{freq}_j)$ denotes the logarithmically normalized interaction frequency, $\text{freq}_i = |\mathcal{N}_i|$ is the number of interactions involving item i , $\mathbf{W}_g \in \mathbb{R}^{1 \times (2d+1)}$ and $\mathbf{b}_g \in \mathbb{R}$ are learnable parameters, and σ is the sigmoid activation. The user-side gating weight g_u is defined analogously, using user-specific features and interaction statistics.

4.3.2 Adaptive Fusion. Based on the derived gating weights, we produce fused representations through a weighted concatenation:

$$\mathbf{x}_i^{\text{fused}} = [g_i \cdot \mathbf{x}_i^{(\text{id})} \parallel (1 - g_i) \cdot \mathbf{x}_i^{(\text{llm})}], \quad (16)$$

Table 1: Statistics of the four evaluation datasets.

Statistics	Book-Crossing	DBbook	MovieLens	Amazon-Book
#Users	6,616	5,576	6,040	11,356
#Items	8,853	2,680	3,260	9,736
#Ratings	110,662	65,961	998,539	303,633
#Density	0.19%	0.44%	5.07%	0.27%
#Relations	4	13	20	10
#Entities	1,404	8,762	14,377	2,707
#Triplets	1,137	134,223	415,104	91,886

with an identical formulation for user representations $\mathbf{x}_u^{\text{fused}}$. This fusion strategy achieves three key objectives:

- **High-frequency regime:** it emphasises the ID channel ($\uparrow g_i$) to leverage precise behavioral patterns from abundant interaction data.
- **Low-frequency regime:** it shifts weight toward the LLM channel ($\downarrow g_i$) to mitigate data sparsity issues by relying on external knowledge.
- **Interpretability:** the explicit gating weights offer insights for model diagnosis, as visualized in Section 5.6.

To compute the preference score, we first consider the direct inner product of the fused representations:

$$\begin{aligned} \text{score}_{ui}^{\text{raw}} &= (\mathbf{x}_u^{\text{fused}})^\top \mathbf{x}_i^{\text{fused}} \\ &= g_u g_i \underbrace{(\mathbf{x}_u^{(\text{id})})^\top \mathbf{x}_i^{(\text{id})}}_{s_{\text{id}}} + (1 - g_u)(1 - g_i) \underbrace{(\mathbf{x}_u^{(\text{llm})})^\top \mathbf{x}_i^{(\text{llm})}}_{s_{\text{llm}}}. \end{aligned} \quad (17)$$

However, the total weight mass $g_u g_i + (1 - g_u)(1 - g_i)$ can vary substantially with the gating values, which may cause scale imbalance and destabilise training. To address this, we introduce a normalization factor $\alpha_{ui} = g_u g_i + (1 - g_u)(1 - g_i)$. The final preference score is then defined as:

$$\hat{y}_{ui} = \frac{(\mathbf{x}_u^{\text{fused}})^\top \mathbf{x}_i^{\text{fused}}}{\alpha_{ui}} = \frac{g_u g_i s_{\text{id}} + (1 - g_u)(1 - g_i) s_{\text{llm}}}{g_u g_i + (1 - g_u)(1 - g_i)}. \quad (18)$$

Letting $w = \frac{g_u g_i}{\alpha_{ui}}$, the score can be rewritten as a convex combination $\hat{y}_{ui} = w \cdot s_{\text{id}} + (1 - w) \cdot s_{\text{llm}}$ with $w \in (0, 1)$. This formulation ensures that the combined score remains on the same numerical scale as the individual channel scores regardless of the absolute values of g_u and g_i , thereby eliminating scale drift and maintaining smooth gradient flow. Finally, the top- K items with the highest \hat{y}_{ui} scores are selected as recommendations.

4.3.3 Gating Regularization. To prevent channel dominance and ensure balanced fusion, we add a gating regularisation term that penalises extreme weight distributions. For each training triple $(u, i, i') \in \mathcal{O}$ (user u , positive item i , and negative item i'), the regularization loss is:

$$\mathcal{L}_{\text{gate}} = \frac{1}{|\mathcal{O}|} \sum_{(u, i, i') \in \mathcal{O}} \sum_{x \in \{u, i, i'\}} [-\log g_x - \log(1 - g_x)], \quad (19)$$

where $g_x \in (0, 1)$ denotes the scalar gating weight. This loss maximizes the log-likelihood of g_x and $1 - g_x$, promoting a uniform weight distribution in the interval $(0, 1)$ and thus enhancing fusion stability and recommendation performance.

4.4 Model Training

The DCGL model is trained in a supervised manner by minimising a composite loss function that integrates the dynamic weighted BPR loss, intra-view contrastive loss, inter-view alignment loss, gating regularisation, and L_2 regularization:

$$\mathcal{L} = \mathcal{L}_{\text{BPR}} + \lambda_{\text{aug}} \mathcal{L}_{\text{aug}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}} + \lambda_{\text{gate}} \mathcal{L}_{\text{gate}} + \lambda_{\text{reg}} \|\Theta\|_2^2. \quad (20)$$

Where Θ denotes all trainable parameters, and λ_{aug} , λ_{align} , λ_{gate} , λ_{reg} are weighting coefficients. The dynamic weighted BPR loss is defined as:

$$\mathcal{L}_{\text{BPR}} = \sum_{(u,i,i') \in O} -\alpha_{ui} \log \sigma(\hat{y}_{ui} - \hat{y}_{ui'}), \quad (21)$$

where $O = \{(u, i, i') \mid (u, i) \in \mathcal{R}, (u, i') \notin \mathcal{R}\}$ is the set of training triples, \hat{y}_{ui} is the preference score, α_{ui} serves as a normalisation factor that stabilises the score scale, and σ is the sigmoid function.

4.5 Complexity Analysis

We analyze the time complexity of DCGL from three key components. (1) **Dual-channel graph propagation**: The RGAT module in the KG channel involves attention calculations with complexity $O(L|\mathcal{R}_k|d^2)$, while the LightGCN module in the interaction channel takes $O(L|\mathcal{R}|d)$, where $|\mathcal{R}_k|$ and $|\mathcal{R}|$ denote the number of triplets and interactions, L is the layer count, and d is the embedding dimension. (2) **Multi-level contrastive learning**: This component incurs $O(|\mathcal{R}|Bd)$ per epoch for calculating pairwise similarities within batch size B . (3) **Frequency-aware gating**: The gating operations are element-wise and incur $O((|\mathcal{U}| + |\mathcal{I}|)d)$. Consequently, the overall complexity grows linearly with the scale of the graphs. Compared with existing LLM-enhanced methods such as CIKGRec and CoLaKG, DCGL does not inflate the graph size nor perform computationally expensive online retrieval. Because the LLM-based semantic extraction is performed offline, DCGL maintains high training efficiency comparable to state-of-the-art GNN-based knowledge-aware recommendation models.

5 Experiment

5.1 Experimental Settings

5.1.1 Datasets. We conducted extensive experiments on four real world recommendation datasets: DBbook [4], Book-Crossing [7], MovieLens [20], and Amazon-Book [10]. All datasets contain auxiliary knowledge on the item side and cover diverse domains, scales, and densities. Following common practice [12, 31], we filtered out low-frequency users (users with fewer than 10 interactions in MovieLens and Amazon-Book, and fewer than 5 interactions in DBbook and Book-Crossing). The statistics of the processed datasets are summarised in Table 1.

5.1.2 Baseline. For a comprehensive evaluation, we compare DCGL with a wide range of baselines, grouped into five categories: (1) **Traditional CF**: LightGCN, SGL, SimGCL; (2) **Embedding based knowledge-aware**: CKE, CFKG; (3) **GNN-based knowledge-aware**: KGAT, KGIN, KGCL, MCCL, KGRec; (4) **LLM-enhanced GNN**: RLMRec; and (5) **LLM-enhanced KG**: CoLaKG, CIKGRec.

- **LightGCN** [12] simplifies GCN by removing feature transformation and nonlinear activation, retaining only linear neighbourhood aggregation.

- **SGL** [29] introduces self-supervised learning by creating contrastive views through node/edge dropout on user-item graph.
- **SimGCL** [34] discards graph-structure augmentations and instead adds uniform noise to embeddings to generate contrastive views.
- **CKE** [36] integrates collaborative filtering with structural, textual and visual knowledge embeddings via TransR and auto-encoders.
- **CFKG** [1] formulates recommendation as link prediction on a unified graph that includes users, items and entities.
- **KGAT** [25] employs graph attention networks to recursively propagate embeddings, with attention weights derived from relation semantics.
- **KGIN** [26] disentangles user intents and aggregates relational paths to capture fine-grained preferences..
- **KGCL** [33] adopts knowledge-graph augmentation and cross view contrastive learning to alleviate noise and sparsity.
- **MCCL** [39] proposes multi-level contrastive learning across collaborative, semantic and structural views.
- **KGRec** [32] is a self-supervised method that rationalises knowledge connections by masking and reconstruction.
- **RLMRec** [21] enriches item profiles with LLM-generated textual signals and aligns them with behavioural relations.
- **CoLaKG** [6] uses LLMs to comprehend local sub-graphs and retrieve global neighbours, fusing them to supplement missing KG facts.
- **CIKGRec** [14] infers user interests with LLMs and constructs a Collaborative Interest Knowledge Graph, employing reconstruction modules to reduce noise.

5.1.3 Implementation Details and Evaluation Metrics. For each user, the historical interactions were randomly split into training, validation and test sets following a 7:1:2 ratio. For DCGL, hyper-parameters were tuned as follows: learning rate in $\{0.0001, 0.0002, \dots, 0.005\}$; number of RGAT layers L and LightGCN propagation depth L' both set to 3; embedding dimension fixed to 64 (consistent with all baselines); gating regularisation weight λ_{gate} , intra-view contrastive weight λ_{aug} and inter-view contrastive weight λ_{align} searched over $\{0.01, \dots, 0.1\}$ with step 0.01; KG-edge dropout rate ρ for intra-view augmentation set to 0.5; stability-aware dropout coefficient μ tuned in $\{0.1, \dots, 1.0\}$ with step 0.1.

For the LLM semantic channel, we first extracted structured information \mathcal{S}_e (e.g., item names and attributes) from DBpedia¹. These structured entries were then fed to Meta-Llama-3-8B-Instruct² to generate textual descriptions. To minimize randomness, we set temperature $\tau = 0$ and top- $p = 0.001$. The generated texts were encoded into semantic embeddings using text-embedding-3-small³. The adapter's intermediate dimension was set to $d_{\text{mid}} = (d_{\text{llm}} + d)/2$, where $d_{\text{llm}} = 1536$ and $d = 64$. All experiments ran on an NVIDIA GeForce RTX 4090 GPU.

For all baselines, we used the authors' official implementations and tuned hyper-parameters according to their recommendations or validation performance. To ensure fairness, common settings were standardised: embedding dimension 64, evaluation interval

¹<https://dbpedia.org/sparql>

²<https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct>

³<https://api.openai.com/v1/embeddings>

Table 2: Performance comparison of different methods. The best results are bolded with * indicating statistical significance ($p < 0.05$), and the second best results are underlined.

Methods	Book-Crossing				DBbook				MovieLens				Amazon-Book			
	R@50	R@100	N@50	N@100	R@50	R@100	N@50	N@100	R@50	R@100	N@50	N@100	R@50	R@100	N@50	N@100
LightGCN	0.1607	0.2246	0.0765	0.0911	0.4214	0.5358	0.2230	0.2481	0.4086	0.5574	0.3977	0.4432	0.2982	0.4083	0.1573	0.1864
SGL	0.1613	0.2154	0.0860	0.0982	0.4258	0.5249	0.2350	0.2567	0.4094	0.5505	0.4004	0.4433	0.2866	0.3769	0.1583	0.1825
SimGCL	0.1653	0.2211	0.0860	0.0988	0.4280	0.5369	0.2376	0.2610	0.4131	0.5560	0.4048	0.4482	0.3099	0.4181	0.1651	0.1938
CKE	0.1284	0.1691	0.0708	0.0811	0.3419	0.4448	0.1832	0.2060	0.3948	0.5421	0.3841	0.4295	0.2834	0.3844	0.1541	0.1808
CFKG	0.1216	0.1771	0.0503	0.0617	0.3586	0.4559	0.1627	0.1816	0.3930	0.5439	0.3672	0.4142	0.2826	0.3941	0.1312	0.1669
KGAT	0.1564	0.2133	0.0865	0.1004	0.4174	0.5309	0.2110	0.2360	0.4059	0.5527	0.3928	0.4378	0.3045	0.4144	0.1595	0.1888
KGIN	0.1468	0.2075	0.0681	0.0819	0.4147	0.5337	0.2200	0.2462	0.4095	0.5581	0.4004	0.4453	0.3324	0.4478	0.1815	0.2123
KGCL	0.1562	0.2049	0.0859	0.0971	0.4308	0.5271	0.2453	0.2666	0.4083	0.5496	0.3996	0.4430	0.3340	0.4411	0.1854	0.2141
MCCLK	0.1476	0.2039	0.0656	0.0774	0.4376	0.5491	0.2054	0.2273	0.4176	0.5655	0.3876	0.4339	0.3285	0.4412	0.1614	0.1890
KGRec	0.1473	0.2075	0.0699	0.0836	0.4415	0.5497	0.2373	0.2611	0.4136	0.5633	0.4072	0.4524	0.3263	0.4429	0.1787	0.2096
RLMRec	0.1682	0.2308	0.0820	0.0963	0.4286	0.5380	0.2375	0.2615	0.4162	0.5618	0.4016	0.4452	0.3161	0.4320	0.1686	0.1991
CoLaKG	<u>0.1888</u>	<u>0.2512</u>	<u>0.0898</u>	<u>0.1044</u>	0.4494	0.5617	0.2400	0.2637	0.4232	0.5692	0.4062	0.4492	<u>0.3427</u>	<u>0.4582</u>	0.1849	0.2156
CIKGR	<u>0.1791</u>	0.2455	0.0889	0.1041	<u>0.4642</u>	<u>0.5756</u>	<u>0.2494</u>	<u>0.2739</u>	<u>0.4250</u>	<u>0.5718</u>	<u>0.4162</u>	<u>0.4609</u>	0.3391	0.4492	<u>0.1863</u>	<u>0.2157</u>
DCGL	0.1923*	0.2605*	0.0979*	0.1119*	0.4664*	0.5804*	0.2572*	0.2798*	0.4252*	0.5727*	0.4176*	0.4616*	0.3574*	0.4663*	0.2028*	0.2317*

Table 3: Ablation study of DCGL components on four datasets. R@100 and N@100 are reported.

Variants	Book-Crossing		DBbook		MovieLens		Amazon-Book	
	R@100	N@100	R@100	N@100	R@100	N@100	R@100	N@100
w/o LLM	0.2319	0.0985	0.5557	0.2655	0.5565	0.4444	0.4440	0.2145
w/o ID	0.2500	0.1017	0.5573	0.2599	0.5663	0.4495	0.4466	0.2154
CAT	0.2396	0.1045	0.5619	0.2697	0.5536	0.4458	0.4513	0.2263
w/o FREQ	0.2576	0.1116	0.5757	0.2745	0.5687	0.4588	0.4654	0.2308
w/o AUG	0.2425	0.1010	0.5778	0.2690	0.5706	0.4580	0.4496	0.2121
w/o ALIGN	0.2578	0.1100	0.5741	0.2770	0.5666	0.4551	0.4645	0.2292
DCGL	0.2605	0.1119	0.5804	0.2798	0.5727	0.4616	0.4663	0.2317

1, and early stopping with a maximum of 2,000 epochs. Reported results are the mean and standard deviation over 10 independent runs with different random seeds.

We adopt Recall@K (R@K) and NDCG@K (N@K), $K \in \{50, 100\}$, to evaluate Top-K recommendation accuracy and ranking quality, respectively. The same evaluation protocol is applied to all LLM- and text-embedding-based baselines.

5.2 Performance Comparison

Quantitative results in Table 2 show that DCGL consistently outperforms all baselines across the four datasets (statistically significant with $p < 0.05$). A key observation is DCGL’s exceptional robustness under sparse conditions. On low-density datasets such as Book-Crossing (0.19%) and Amazon-Book (0.27%), traditional GNN-based models (e.g., KGAT, KGCL) struggle because they heavily rely on explicit structural connectivity, which is scarce in such environments. DCGL effectively circumvents this “sparsity trap” by leveraging its decoupled LLM semantic channel. By treating LLM-derived semantics as an independent view, DCGL constructs implicit semantic bridges that compensate for missing structural

links, enabling preference inference based on content understanding even when behavioural paths are broken. This leads to substantial gains, notably surpassing the strongest baseline on Amazon-Book by a clear margin.

Moreover, DCGL exhibits superior adaptability compared with state-of-the-art LLM-enhanced baselines such as CoLaKG and CIKGR. These methods are fundamentally constrained by static fusion strategies that blend semantic and ID signals indiscriminately, often introducing semantic noise for popular items or failing to support cold-start ones. DCGL resolves this dichotomy through its frequency-aware gating mechanism, which acts as a dynamic arbiter between the two channels. As evidenced by its consistent lead on both dense (MovieLens) and sparse (DBbook) datasets, DCGL successfully prioritises the LLM channel to alleviate cold-start issues for tail items while shifting focus to the ID channel for head items. This dynamic calibration effectively balances semantic generalisation with behavioural specificity, validating the necessity of a dual-channel architecture for handling frequency heterogeneity.

5.3 Ablation Study

To evaluate the contribution of each component in DCGL, we conducted ablation studies using fixed hyperparameters, with results on all four core datasets presented in Table 3. The key findings are as follows: First, removing the LLM semantic channel (w/o LLM) leads to substantial performance drops across all datasets, underscoring the critical role of LLM-generated semantic representations in mitigating data sparsity and improving recommendation accuracy. Similarly, discarding the ID information channel (w/o ID) also results in performance degradation, indicating that interaction behavior signals captured by the ID channel complement those from the LLM channel to jointly enhance recommendation effectiveness. Replacing the dual-channel architecture with a simple concatenation of LLM and ID embeddings at the input (CAT) yields inferior results, demonstrating the necessity of separate channels for adaptive fusion. Likewise, replacing the frequency-aware gating

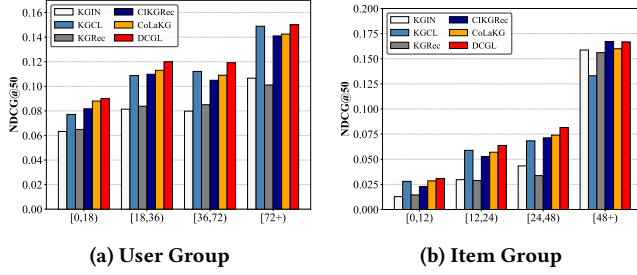


Figure 4: Performance of DCGL vs. Baselines under different user & item groups (Book-Crossing Dataset).

with a fixed-weight scheme (*w/o FREQ*) leads to clear performance decline, verifying that dynamic gating is essential for balancing semantic and behavioural information according to interaction frequency. Furthermore, removing either intra-view augmentation (*w/o AUG*) or inter-view alignment (*w/o ALIGN*) significantly reduces model performance, demonstrating that intra-view augmentation enhances representation robustness while inter-view alignment effectively bridges the heterogeneity between dual-channel representation spaces.

5.4 Group Analysis

To evaluate the robustness of DCGL across varying data densities, we conduct a group-wise analysis on the Book-Crossing dataset. Users are divided into four intervals based on interaction counts: [0, 18), [18, 36), [36, 72), and [72+), while items are grouped by interaction frequency: [0, 12), [12, 24), [24, 48), and [48+). Figure 4 illustrates the performance comparison against competitive baselines. The results indicate that DCGL consistently outperforms alternative methods across all user and item segments, validating its adaptability to both data-scarce and data-rich environments.

- **Sparse Scenarios:** In low-interaction regimes, specifically the user group [0, 18) and item group [0, 12), DCGL achieves substantial gains over traditional GNNs like KGIN and KGRec, while maintaining a clear lead over other LLM-enhanced baselines. This confirms that the dual-channel architecture effectively harnesses LLM-derived knowledge to mitigate cold-start challenges, substituting missing behavioral signals with rich implicit semantics.
- **Dense Scenarios:** In high-interaction regimes (e.g., [72+) users, [48+) items), DCGL maintains superior performance where other methods often exhibit volatility. Unlike purely semantic-enhanced methods that risk introducing noise, DCGL employs the frequency-aware gating mechanism to adaptively prioritize the ID channel ($\uparrow g$). This strategy effectively filters out generic semantic redundancy while fully exploiting the precise collaborative patterns captured by the ID embeddings, ensuring that the rich historical data of active users is leveraged without dilution.

In summary, DCGL effectively bridges the information gap for long-tail items while maintaining accuracy for popular ones, validating its adaptive fusion mechanism in balancing semantic generalization and behavioral specificity.

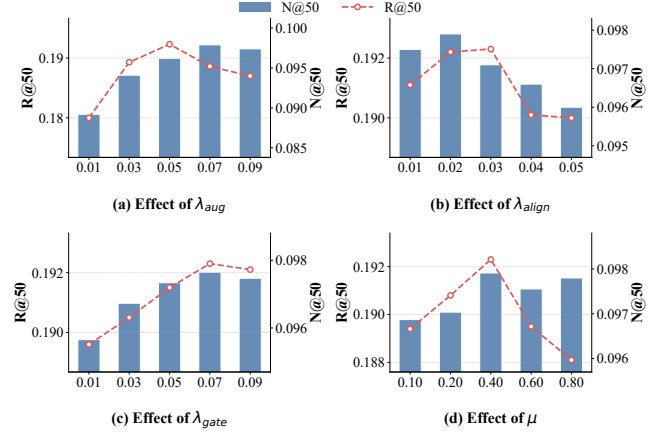


Figure 5: Hyper-parameter analysis on Book-Crossing.

5.5 Hyperparameter Sensitivity

We study DCGL’s sensitivity to four key hyperparameters on Book-Crossing: the intra-view contrastive weight λ_{aug} , the inter-view contrastive weight λ_{align} , the gating regularization weight λ_{gate} , and the stability-aware dropout coefficient μ . As illustrated in Figure 5(a) and (b), both contrastive weights exhibit a similar trend: performance first improves and then declines. Small values fail to provide sufficient self-supervised regularization to bridge the semantic gap or enhance robustness. Conversely, excessively large values risk overshadowing the main recommendation objective, leading to suboptimal rankings. A moderate range yields the best trade-off between representation distinctiveness and semantic alignment. Figure 5(c) indicates that λ_{gate} is crucial for preventing the gating mechanism from collapsing to trivial solutions (i.e., saturation at 0 or 1), thereby preserving its frequency-aware adaptivity. For the stability coefficient μ in Figure 5(d), an intermediate value achieves the most effective perturbation, confirming that stability-guided pruning effectively removes noise while preserving structural information, outperforming both random and aggressive dropout.

5.6 Visualization of Gating Interpretability

To verify whether DCGL effectively resolves the frequency heterogeneity challenge highlighted in our Motivating Example, we visualize the learned gating weights on the Book-Crossing dataset. Figure 6 illustrates the relationship between normalized interaction frequency and the learned gate weight g (which controls the contribution of the ID channel). As shown in Figure 6, the clear positive correlation between interaction frequency and gate weight g provides empirical evidence that our model successfully adapts its fusion strategy across the sparsity spectrum:

- **Response to the “Bob” Scenario (Sparse Regime):** In the low-frequency region (left side of the plots), the model consistently learns lower gate weights (lower g). This aligns perfectly with the needs of cold-start users like Bob. Since their behavioral signals are weak and noise-prone, DCGL automatically suppresses the unreliable ID channel and assigns a larger weight ($1 - g$) to the

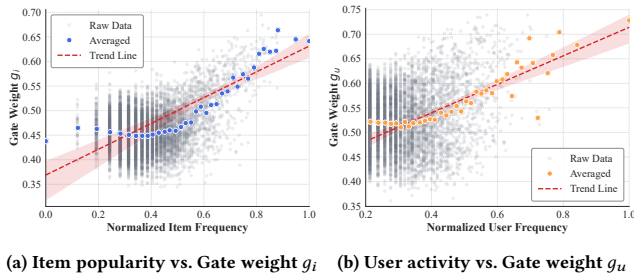


Figure 6: Visualization of the learned gating weights g w.r.t interaction frequency on Book-Crossing.

LLM channel. This mechanism effectively leverages clear semantic cues to compensate for the lack of interaction data, preventing the “noise overwhelming” issue described in the motivation.

- **Response to the “Alice” Scenario (Dense Regime):** In the high-frequency region (right side of the plots), the gate weights significantly increase, indicating a dominance of the ID channel. This mirrors the scenario of active users like Alice. Given her strong historical patterns, the model prioritizes the precise collaborative signals captured by the ID embeddings. Crucially, by limiting the weight of the LLM channel in this regime, DCGL avoids the “blurred representations” caused by indiscriminate semantic infusion, ensuring that Alice’s precise personal preferences are not diluted by generic semantic information.

This visualization confirms that DCGL breaks the vicious cycle of static fusion. Instead of a uniform view, it achieves a context-sensitive balance: relying on semantics for grounding in sparse cases while preserving behavioral precision in dense cases.

6 Conclusion

In this paper, we proposed DCGL, a novel dual-channel graph learning framework designed to break the cascading limitation in knowledge-aware recommendation. Unlike conventional methods that suffer from signal interference due to static fusion, DCGL structurally decouples frequency-agnostic LLM semantics from frequency-sensitive behavioral signals. Through a synergistic combination of multi-level contrastive learning and frequency-aware gated fusion, our approach effectively resolves the trade-off between semantic generalization and behavioral specificity. Extensive experiments on four real-world datasets demonstrate that DCGL not only consistently outperforms state-of-the-art baselines but also achieves a context-sensitive balance: leveraging external knowledge to alleviate sparsity for cold-start entities while preserving precise collaborative patterns for active ones. Furthermore, since LLM-based semantic extraction is performed offline, DCGL requires no real-time LLM inference, ensuring high efficiency for practical deployment.

References

- [1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms* 11, 9 (2018), 137.
- [2] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM conference on recommender systems*. 1007–1014.
- [3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).
- [4] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The world wide web conference*. 151–161.
- [5] Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. 2024. When large language models meet personalization: Perspectives of challenges and opportunities. *World Wide Web* 27, 4 (2024), 42.
- [6] Ziqiang Cui, Yunpeng Weng, Xing Tang, Fuyuan Lyu, Dugang Liu, Xiuqiang He, and Chen Ma. 2025. Comprehending knowledge graphs with large language models for recommender systems. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1229–1239.
- [7] Xin Dong, Lei Yu, Zhonghuo Wu, Yuxia Sun, Lingfeng Yuan, and Fangxi Zhang. 2017. A hybrid collaborative filtering model with deep structure for recommender systems. In *Proceedings of the AAAI Conference on artificial intelligence*, Vol. 31.
- [8] Chongming Gao, Ruijun Chen, Shuai Yuan, Kexin Huang, Yuanqing Yu, and Xiangnan He. 2025. Sprec: Self-play to debias llm-based recommendation. In *Proceedings of the ACM on Web Conference 2025*. 5075–5084.
- [9] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhuan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. 2023. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems* 1, 1 (2023), 1–51.
- [10] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.
- [11] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.
- [12] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.
- [13] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*. Springer, 364–381.
- [14] Zheng Hu, Zhe Li, Ziyun Jiao, Satoshi Nakagawa, Jiawen Deng, Shimin Cai, Tao Zhou, and Fuji Ren. 2025. Bridging the user-side knowledge gap in knowledge-aware recommendations with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 11799–11807.
- [15] Yangqin Jiang, Yuhao Yang, Lianghao Xia, Da Luo, Kangyi Lin, and Chao Huang. 2025. RecLM: Recommendation Instruction Tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, 15443–15459. <https://aclanthology.org/2025.acl-long.751/>
- [16] Anchen Li, Bo Yang, Huan Huo, Farookh Hussain, and Guandong Xu. 2025. Hypercomplex knowledge graph-aware recommendation. In *Proceedings of the 48th international ACM SIGIR conference on research and development in information retrieval*. 2017–2026.
- [17] Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. 2024. Large language models for generative recommendation: A survey and visionary discussions. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 10146–10159.
- [18] Yanhui Li, Dongxia Wang, Zhu Sun, Haonan Zhang, and Huizhong Guo. 2025. LightKG: Efficient Knowledge-Aware Recommendations with Simplified GNN Architecture. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 1577–1588.
- [19] Yuhuan Li, Xinni Zhang, Linhao Luo, Heng Chang, Yuxiang Ren, Irwin King, and Jia Li. 2025. G-refer: Graph retrieval-augmented large language model for explainable recommendation. In *Proceedings of the ACM on Web Conference 2025*. 240–251.
- [20] Tommaso Di Noia, Vito Claudio Ostuni, Paolo Tomeo, and Eugenio Di Sciascio. 2016. Sprank: Semantic path-based ranking for top-n recommendations using linked open data. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8, 1 (2016), 1–34.
- [21] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM web conference 2024*. 3464–3475.
- [22] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
- [23] Riku Togashi, Mayu Otani, and Shin’ichi Satoh. 2021. Alleviating cold-start problems in recommendation through pseudo-labelling over knowledge graph.

- In *Proceedings of the 14th ACM international conference on web search and data mining*. 931–939.
- [24] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*. 1835–1844.
- [25] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 950–958.
- [26] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the web conference 2021*. 878–887.
- [27] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable Reasoning over Knowledge Graphs for Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (Jul. 2019), 5329–5336. doi:10.1609/aaai.v33i01.33015329
- [28] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM international conference on web search and data mining*. 806–815.
- [29] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.
- [30] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2024. A survey on large language models for recommendation. *World Wide Web* 27, 5 (2024), 60.
- [31] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.
- [32] Yuhao Yang, Chao Huang, Lianghao Xia, and Chunzhen Huang. 2023. Knowledge graph self-supervised rationalization for recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*. 3046–3056.
- [33] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1434–1443.
- [34] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1294–1303.
- [35] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 283–292.
- [36] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.
- [37] Qian Zhao, Hao Qian, Ziqi Liu, Gong-Duo Zhang, and Lihong Gu. 2024. Breaking the barrier: utilizing large language models for industrial recommendation systems through an inferential knowledge graph. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 5086–5093.
- [38] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. 2024. Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering* 36, 11 (2024), 6889–6907.
- [39] Ding Zou, Wei Wei, Xian-Ling Mao, Ziyang Wang, Minghui Qiu, Feida Zhu, and Xin Cao. 2022. Multi-level cross-view contrastive learning for knowledge-aware recommender system. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1358–1368.
- [40] Ding Zou, Wei Wei, Feida Zhu, Chuanyu Xu, Tao Zhang, and Chengfu Huo. 2024. Knowledge enhanced multi-intent transformer network for recommendation. In *Companion proceedings of the ACM web conference 2024*. 1–9.