Diffusion-guided Contrastive Learning with Large Language Model-enhanced Knowledge Graph Recommendation

Yichen Zhang[1], Huaxiong Zhang[1,*], Zhijian Fang[1], Lei Zhou[2]
[1]*School of Computer Science and Technology (School of Artificial Intelligence), Zhejiang Sci-Tech University*, Hangzhou, China.
[2]*Zhejiang Talent Development Group.*
*Corresponding author. Email: zhxhz@zstu.edu.cn

***Abstract*—Knowledge graphs (KGs) provide rich factual and structural information for recommendation, yet real-world KGs often contain noisy links and long-tail sparsity. Most existing methods learn on a fixed KG, making it difficult to explicitly remove task-irrelevant links from a structural perspective. Meanwhile, inner-product-based pointwise scoring has limited ability to jointly exploit multiple structural cues when candidate structures are highly similar or evidence is scarce, limiting fine-grained ranking at the top positions. To address these issues, we propose DiCoLL-KGRec, a unified framework that integrates diffusion-based KG denoising, diffusion-guided multi-view contrastive learning, and large language model (LLM)-enhanced two-stage reranking. Specifically, we perform diffusion-based generative modeling in the item-entity relation space to denoise noisy KGs, deriving a task-relevant clean KG and multiple diffusion views. We then quantify discrepancies across diffusion views as structural reliability signals, which serve as a consistency prior. Guided by this signal, we conduct biased augmentations on the user-item graph and impose multi-view contrastive constraints on the KG side, enabling robust representation learning via generated views and consistency alignment. At inference time, we rewrite the top candidates and their structural features into structured prompts and use an LLM to rerank them, further refining the final list. Experiments on three public datasets show that our method yields consistent improvements in overall performance, robustness, and decision enhancement.**

***Index Terms*—Knowledge graph recommendation, diffusion model, contrastive learning, large language model.**

I. Introduction

Recommender systems alleviate information overload by modeling users' historical behaviors to deliver personalized content. Among them, collaborative filtering (CF) [1] has achieved broad success, but its performance is often limited by sparse interactions, especially in cold-start scenarios. To address this issue, knowledge graphs (KGs) have been widely introduced as structured side information for recommendation [2], [3]. In particular, graph neural network (GNN)-based KG recommendation methods have shown a strong ability to capture high-order semantic dependencies through multi-hop propagation [4], [5]. However, real-world KGs are often incomplete and noisy, containing topic-irrelevant links, erroneous edges, and long-tail entities. Such noise can be amplified during graph propagation and mislead preference learning [6].

Recent studies have improved robustness through self-supervised contrastive learning, which constructs augmented graph views and aligns them at the representation level [7], [8]. Nevertheless, two limitations remain. First, task-agnostic random augmentations cannot effectively distinguish preference-relevant structures from noisy or irrelevant ones. Second, although contrastive learning improves representation quality, final decisions are still commonly based on inner-product scoring, which limits fine-grained discrimination when candidates are highly similar or evidence is sparse.

Diffusion-based generative modeling offers a promising way to denoise KGs by recovering task-relevant structures under collaborative signals [6]. However, existing methods mainly use diffusion views as denoised outputs, without explicitly transforming cross-view discrepancies into reliability signals that can guide contrastive view construction and interaction augmentation. Meanwhile, large language models (LLMs) have shown strong reasoning and analogy capabilities over structured knowledge [9], [10], which suggests their potential for decision-level enhancement in recommendation. However, for ID-only recommendation data, how to build stable and interpretable structural prompts for candidate reranking remains underexplored [11].

To address the above limitations, we propose DiCoLL-KGRec (Diffusion-guided Contrastive Learning with Large Language Model-enhanced Knowledge Graph Recommendation), a unified framework that integrates structural denoising, representation alignment, and decision enhancement. Specifically, we first perform diffusion-based KG denoising to obtain a task-relevant clean KG and multiple diffusion views. We then develop diffusion-guided multi-view contrastive learning, where diffusion-derived structural consistency guides biased augmentation on the user–item interaction graph, while view-level contrastive alignment is imposed on the KG side to improve representation robustness. Finally, during inference, we introduce an LLM-enhanced two-stage reranking module that converts structural features of the Top-m candidates into structured prompts and performs structure-aware ranking with in-context demonstrations.

The main contributions of this work are summarized as

follows:

- We propose DiCoLL-KGRec, a unified recommendation framework that combines diffusion-based KG denoising, multi-view contrastive learning, and LLM-enhanced reranking.
- We design a diffusion-guided multi-view contrastive learning strategy, where cross-view structural consistency is converted into guidance for interaction augmentation and KG-side contrastive alignment.
- We propose a structured two-stage reranking strategy for ID-based recommendation data, enabling LLMs to make structure-aware ranking decisions within the retrieved candidate set.
- We conduct extensive experiments on Yelp2018, Amazon-Book, and MIND, demonstrating the effectiveness of DiCoLL-KGRec in overall recommendation performance, robustness, and decision enhancement on challenging instances.

II. Related Work

A. Robustness of KG Recommendation

Real-world KGs often contain topic-irrelevant links, erroneous edges, and long-tail entities; directly propagating messages over such graphs may amplify noise and mislead preference learning. To improve robustness, prior work mainly follows two lines. The first line performs explicit denoising from a generative-modeling perspective. For example, DiffKG [6] introduces a diffusion noising–denoising process in the item–entity relation space to recover task-relevant structural views. The second line adopts self-supervised contrastive learning. For example, KGCL [7] constructs multi-view structural augmentations and enforces cross-view consistency to reduce reliance on noisy links and alleviate sparse supervision. Overall, diffusion-based methods primarily aim to enhance the quality of structural views, whereas contrastive learning focuses on strengthening representation robustness. Nevertheless, the two lines are often developed in separate frameworks, and a unified mechanism that leverages diffusion-induced multi-view structures to systematically guide contrastive view construction and interaction-graph augmentation is still lacking.

B. Large Language Models and Knowledge Graphs

LLMs have shown promise in understanding structured knowledge and performing analogical reasoning [9], [12]. Typical two-stage frameworks (e.g., KICGPT [11]) first retrieve candidate structural facts and then use structured prompts to guide an LLM to make read-only judgments within the candidate set, thereby improving long-tail performance without fine-tuning. Although recent studies and survey works have shown the potential of LLMs in structured knowledge understanding and related downstream tasks [13], recommendation data often lack high-quality textual descriptions of entities, and letting an LLM directly perform full ranking is usually costly, low-throughput, and difficult to couple with graph-based recommendation pipelines. Therefore, a more practical direction is to let the backbone model provide the Top-K candidates first, and then rewrite the structural evidence into compact prompts so that the LLM can deliver "decision-level enhancement" with modest additional overhead.

III. Problem Definition

This section introduces our notation and formalizes the studied task.

User-Item Interaction Graph: Let $\mathcal{U}$ denote the set of users and $\mathcal{I}$ the set of items. The observed interaction set is $\mathcal{O} = \{(u, i) \mid u \in \mathcal{U},\ i \in \mathcal{I}\}$. Based on $\mathcal{O}$, we construct a user-item interaction graph $\mathcal{G}_u = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ is the node set and $\mathcal{E}$ is the edge set. If user u interacts with item i, we set $y_{u,i} = 1$; otherwise $y_{u,i} = 0$.

Knowledge Graph: We denote the knowledge graph as $\mathcal{G}_k = \{(h, r, t)\}$, where $h, t \in E$ are entities and $r \in R$ is a relation type. Each triple (h, r, t) describes a semantic association between entities, providing structured external knowledge for items.

Task Formulation: Given the interaction graph $\mathcal{G}_u$ and the knowledge graph $\mathcal{G}_k$, we learn a recommendation model $\mathcal{F}(u, i \mid \mathcal{G}_u, \mathcal{G}_k, \Theta)$ with parameters Θ to predict a preference score $\hat{y}_{u,i}$. Based on these scores, we retrieve a candidate set from all items as $\mathcal{R}_{\text{retriever}}(u) = \text{Top-}K(\hat{y}_{u,i})$. At inference time, we employ an LLM to rerank the top m candidates in $\mathcal{R}_{\text{retriever}}(u)$, yielding the final recommendation list $\mathcal{R}_{\text{final}}(u)$.

IV. Methodology

In this section, we present our proposed DiCoLL-KGRec framework. As illustrated in Fig. 1, the framework comprises three modules organized into a two-stage pipeline: the first two modules jointly form a graph-based retriever, and the third module serves as an LLM-based reranker. Specifically, DiCoLL-KGRec consists of a diffusion-based KG denoising module, a diffusion-guided multi-view contrastive learning module, and an LLM-enhanced two-stage reranking module.

A. Diffusion-based KG Denoising Module

Following the modeling paradigm of diffusion-based KG denoising [6], we perform diffusion modeling over the relation matrix χ_0 in the item–entity relation space to generate task-relevant denoised structural views. These views serve as the view foundation for subsequent contrastive learning and as a source of consistency signals. In the forward process, Gaussian noise is injected step by step:

$$q(\chi_t \mid \chi_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}\,\chi_{t-1}, (1-\alpha_t)I)\,,\ \chi_0 \to \chi_T \quad (1)$$

In the reverse process, we learn a denoising network $\hat{\chi}_\theta(\chi_t, t)$ to approximate $p_\theta(\chi_{t-1} \mid \chi_t)$. During inference, we initialize from $\hat{\chi}_T = \chi_T$ and iteratively apply the deterministic update $\hat{\chi}_{t-1} = \mu_\theta(\hat{\chi}_t, t)$ until $\hat{\chi}_0$, so as to reduce sampling variance and obtain the denoised relation matrix. The training objective is the step-wise reconstruction ELBO:

$$\mathcal{L}_{\text{elbo}} = \mathbb{E}_{t\sim\mathcal{U}(1,T)}\big[\mathcal{L}_t\big] \quad (2)$$

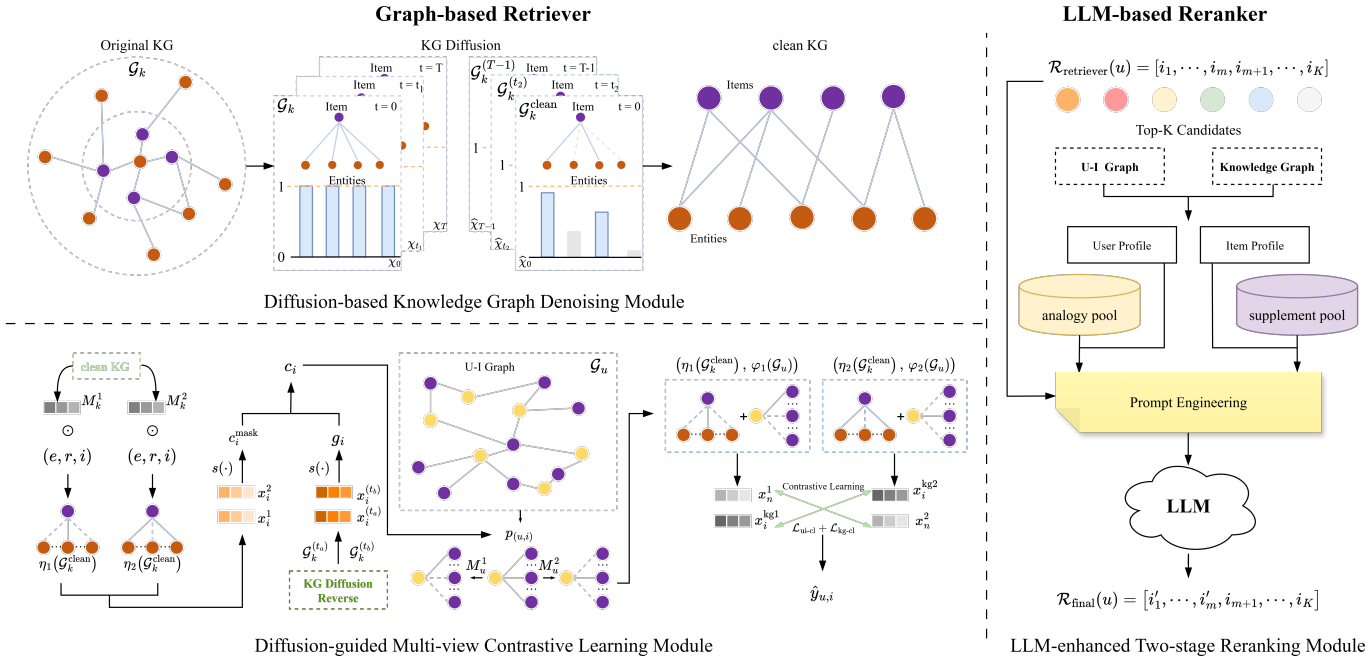

Fig. 1. Overview of the DiCoLL-KGRec framework (two-stage retrieval–reranking).

Since structure-only denoising may be misaligned with the recommendation objective, we further introduce a collaborative constraint (Collaborative KG Convolution, CKGC) that fuses the interaction matrix A with $\hat{\chi}_0$ to align denoising with preference learning:

$$\mathcal{L}_{\text{ckgc}} = \left\| A \cdot \hat{\chi}_0^{\top} x_u - x_i \right\|_2^2, \quad \mathcal{L}_{\text{kgdm}} = (1-\lambda_0)\mathcal{L}_{\text{elbo}} + \lambda_0 \mathcal{L}_{\text{ckgc}} \tag{3}$$

After denoising, we keep the Top-k entity links for each item on $\hat{\chi}_0$ to construct the task-relevant subgraph $\mathcal{G}_k^{\text{clean}}$. Meanwhile, we repeatedly export graphs at several intermediate steps $t \in \mathcal{T}$ to obtain multiple diffusion views $\{\mathcal{G}_k^{(t)}\}$, providing "multi-perspective structural views" for subsequent contrastive learning.

B. Diffusion-guided Multi-view Contrastive Learning Module

Under the multi-view contrastive learning paradigm [7], we further leverage the multiple diffusion views exported from the diffusion stage, and explicitly transform the "view discrepancy" into a global structural reliability signal g_i. We then fuse it with the masked consistency c_i^{mask} to obtain c_i, so as to enable task-aware, adaptive control of the augmentation strength on the interaction graph at the knowledge-aware injection point. Different from contrastive learning that relies solely on random augmentations, we schedule the augmentation strength using the diffusion consistency signal, allowing the gains brought by structural denoising to be explicitly propagated to the contrastive learning stage. Meanwhile, we introduce an additional KG-side contrastive branch $\mathcal{L}_{\text{kg-cl}}$ to directly align knowledge representations, improving stability under residual noise. The module is jointly trained on $\mathcal{G}_u = \{\mathcal{V}, \mathcal{E}\}$ and $\mathcal{G}_k^{\text{clean}}$, outputs $\hat{y}_{u,i}$, and produces the candidate set $\mathcal{R}_{\text{retriever}}(u) = \text{Top-}K(\hat{y}_{u,i})$.

a) KG-side augmentation and consistency signals: On $\mathcal{G}_k^{\text{clean}}$, we apply a random masking operator $\eta(\cdot)$ to obtain two KG views, $\eta_1(\mathcal{G}_k^{\text{clean}})$ and $\eta_2(\mathcal{G}_k^{\text{clean}})$. To characterize the stability of the same item under local perturbations, we define the masked consistency as

$$c_i^{\text{mask}} = s\left(f_k\left(x_i, \eta_1(\mathcal{G}_k^{\text{clean}})\right), f_k\left(x_i, \eta_2(\mathcal{G}_k^{\text{clean}})\right)\right) \tag{4}$$

Besides $\mathcal{G}_k^{\text{clean}}$, the diffusion module also outputs multiple diffusion views $\{\mathcal{G}_k^{(t)} \mid t \in \mathcal{T}\}$. On each view, we obtain item representations using the same knowledge aggregator, and define the cross-diffusion-view stability as the mean of pairwise cosine similarities:

$$g_i = \frac{2}{|\mathcal{T}|(|\mathcal{T}|-1)} \sum_{t_a < t_b} s\left(x_i^{(t_a)}, x_i^{(t_b)}\right) \tag{5}$$

A larger g_i indicates that the item exhibits more consistent structural representations across different diffusion views. Therefore, we interpret g_i as a global structural reliability measure, and fuse it with c_i^{mask} to obtain the final structural consistency:

$$c_i = \alpha \cdot c_i^{\text{mask}} + (1-\alpha) \cdot g_i, \ \alpha \in [0,1] \tag{6}$$

which is used to control the augmentation strength in the subsequent interaction-graph augmentation.

b) Diffusion-guided edge dropping on the interaction graph: We treat c_i as a structural reliability signal and map it to the edge (u,i)'s drop probability $p_{u,i}$. We then perform Bernoulli sampling with $p_{u,i}$ to obtain two interaction-augmented views $\varphi_1(\mathcal{G}_u)$ and $\varphi_2(\mathcal{G}_u)$. Here, g_i indirectly

controls the augmentation strength by affecting c_i: items that are inconsistent across diffusion views (small g_i) are regarded as less reliable, and thus their associated interaction edges are more likely to be dropped; conversely, they are more likely to be preserved.

c) Joint objective: On $(\eta_1(\mathcal{G}_k^{\text{clean}}), \varphi_1(\mathcal{G}_u))$ and $(\eta_2(\mathcal{G}_k^{\text{clean}}), \varphi_2(\mathcal{G}_u))$, we apply LightGCN propagation to obtain two sets of representations (x_u^1, x_i^1) and (x_u^2, x_i^2), and enforce cross-view consistency via an InfoNCE loss:

$$\mathcal{L}_{\text{ui-cl}} = \sum_{n \in \mathcal{V}} -\log \frac{\exp\big(s(x_n^1, x_n^2)/\tau\big)}{\sum_{n' \in \mathcal{V},\, n' \neq n} \exp\big(s(x_n^1, x_{n'}^2)/\tau\big)} \tag{7}$$

We further explicitly introduce KG-view contrast by letting x_i^{kg1} and x_i^{kg2} be generated directly by the knowledge aggregator on $\eta_1(\mathcal{G}_k^{\text{clean}})$ and $\eta_2(\mathcal{G}_k^{\text{clean}})$, and imposing the following constraint in the KG embedding space:

$$\mathcal{L}_{\text{kg-cl}} = \sum_{i \in \mathcal{I}} -\log \frac{\exp\big(s(x_i^{\text{kg1}}, x_i^{\text{kg2}})/\tau_{\text{kg}}\big)}{\sum_{j \in \mathcal{I},\, j \neq i} \exp\big(s(x_i^{\text{kg1}}, x_j^{\text{kg2}})/\tau_{\text{kg}}\big)} \tag{8}$$

The main recommendation objective still adopts BPR ranking supervision, and the final optimization target is:

$$\mathcal{L}_{\text{rec}} = \mathcal{L}_{\text{bpr}} + \lambda_{\text{ui-cl}}\mathcal{L}_{\text{ui-cl}} + \lambda_{\text{kg-cl}}\mathcal{L}_{\text{kg-cl}} + \lambda_2\|\Theta\|_2^2 \tag{9}$$

Here, $\mathcal{L}_{\text{ui-cl}}$ primarily aligns the final user/item representations under knowledge-aware interaction views, while $\mathcal{L}_{\text{kg-cl}}$ directly aligns the knowledge representations produced by KG aggregation; the two losses complement each other to improve robustness.

C. LLM-enhanced Two-stage Reranking Module

The graph-based model can be viewed as a structure-aware retriever: for a user u, it computes preference scores $\hat{y}_{u,i}$ over all items and takes the Top-K results to form a ranked list $\mathcal{R}_{\text{retriever}}(u)$. When candidate items exhibit highly similar structures or when evidence is sparse, inner-product-based scoring struggles to make fine-grained distinctions. To improve the retriever's decision resolution under highly similar candidates, we introduce an LLM to rerank the top m ($m \leq K$) candidates, leveraging structured evidence to conduct discriminative reasoning among candidates. This yields the final list $\mathcal{R}_{\text{final}}(u)$.

a) Structured Rewriting without Textual Descriptions: Given that many datasets provide only entity/relation IDs, directly relying on textual descriptions is difficult to control and highly dataset-dependent. We rewrite the structural signals of u and i on the KG into semi-structured fields, so as to stably inject generalizable structural evidence. We define two structural schemas:

User Profile. It is used to characterize the overall structural preference on the KG induced by a user's historical interactions, using the following four types of statistics: Interaction volume: the size of the historically interacted item set N_u, i.e., $|N_u|$; One-hop reachability: the number of distinct entities $|N_e(N_u)|$ and the number of distinct relation types $|N_r(N_u)|$ in the one-hop neighborhood starting from N_u, where $N_e(\cdot)$ denotes the neighbor-entity set and $N_r(\cdot)$ denotes the set of relation types connecting to the neighbors; Historical structural complexity: the mean and a high quantile of the degrees $d(\cdot)$ of historically interacted items, denoted as $\text{mean}(d)$ and $\text{p90}(d)$, respectively; Historical stability: the proportion of high-consistency items, $\rho_u = \frac{1}{|N_u|}\sum_{j \in N_u} \mathbf{1}(c_j \geq \tau_c)$, where c_j is the item structural consistency indicator and τ_c is the stability threshold.

Item Profile. This schema characterizes the structural match between an item and the user's historical preferences. For an item i, we extract the following five structural features: Node degree d_i (computed on the clean KG); Binned structural consistency: we use the consistency indicator c_i and bin it by fixed thresholds as $\text{bin}(c_i) \in \{\text{Low}, \text{Mid}, \text{High}\}$; Number of shared neighbors: $s_i = |N_e(i) \cap N_e(N_u)|$; Neighbor overlap ratio: $o_i = \frac{|N_e(i) \cap N_e(N_u)|}{|N_e(i) \cup N_e(N_u)|}$; overlap ratio: $r_i = \frac{|N_r(i) \cap N_r(N_u)|}{|N_r(i)|}$.

To improve the LLM's stable understanding of these structural quantities and to control prompt length, we serialize the above statistics into semi-structured fields with a fixed template, e.g., "item_i profile: $d_i = 5$, $\text{bin}(c_i) = \text{High}$, $s_i = 3$, $o_i = 0.18$, $r_i = 0.25$".

b) Analogy Pool and Supplement Pool: To fully exploit the LLM's analogical reasoning capability and enable stable inference under read-only structural conditions, we construct two types of demonstration pools:

Analogy Pool. We directly use the target user's historically interacted item set N_u as the source of analogies, allowing the LLM to induce, from the user's own preferences, the "item structural patterns that are more likely to match this user," denoted as $D^a(u)$.

Supplement Pool. For each candidate item i, we collect its historically interacted users from the training interactions and sample S_u prototypical users. The supplement pool $D^s(u)$ is formed by all candidate items together with their corresponding prototypical users. By comparing the User Profile of the target user with those of the prototypical users, the LLM estimates the structural compatibility of u-i, thereby assisting discrimination among candidates.

Both pools are constructed in the preprocessing stage; at online inference, only lightweight retrieval and concatenation are required.

c) Prompt Design: To ensure valid and reproducible outputs, we adopt staged prompting to decouple constraints, demonstration injection, and the final query. For each reranking request we construct an independent dialogue-style prompt and organize the inputs into the following four stages (where Stage 3 can be repeated multiple times to inject more demonstrations):

Stage 1: Responsibility Description. We specify the LLM's role as a candidate reranker, and strictly constrain the output to be a single-line sequence consisting of the m item_ids within the given candidate set.

Stage 2: Question & Demonstration Description. We state that we will provide analogy demonstrations (the user's historical preferences) and supplement demonstrations (the candidates' prototypical users).

TABLE I
STATISTICS OF EXPERIMENTED DATASETS.

Stats	Yelp2018	Amazon-Book	MIND
# Users	45,919	70,679	300,000
# Items	45,538	24,915	48,957
# Interactions	1,183,610	846,434	2,545,327
Density Degree	5.7×10^{-4}	4.8×10^{-4}	1.7×10^{-4}
Knowledge Graph			
# Relations	42	39	90
# Entities	47,472	29,714	106,500
# Triples	869,603	686,516	746,270

Stage 3: Multiple Demonstrations (Repeatable). We sample S_a and S_s demonstrations from the analogy pool $D^a(u)$ and the supplement pool $D^s(u)$, respectively, and organize them into unified-format demonstration blocks. We first present the target user's User Profile. Then, each analogy demonstration includes the Item Profile of one historically interacted item of this user; each supplement demonstration includes the Item Profile of each candidate item and the User Profile of its paired prototypical user. This stage can be repeated to inject multiple batches of demonstrations as long as the token budget allows.

Stage 4: Final Query. We input the structural schemas of the current user and the Top-m candidates, and require the LLM to output only the ranking; ties are broken by the original order.

If the output is in an illegal format or cannot be parsed, we trigger a fallback by reverting to $\mathcal{R}_{\text{retriever}}(u)$, ensuring end-to-end stability. Accordingly, the final list is: $\mathcal{R}_{\text{final}}(u) = [i'_1, \ldots, i'_m, i_{m+1}, \ldots, i_K]$.

V. EXPERIMENTS AND ANALYSIS

In this section, we conduct systematic evaluations of DiCoLL-KGRec on three real-world datasets, focusing on recommendation performance, the contribution of key components, robustness, and the LLM's decision enhancement capability. We specifically address the following research questions:

RQ1: Compared with existing recommendation methods, how does DiCoLL-KGRec perform in terms of overall recommendation quality?

RQ2: How much improvement is contributed by each key module, including diffusion-based KG denoising, multi-view contrastive learning, and the LLM-based two-stage reranking?

RQ3: Does DiCoLL-KGRec exhibit better robustness under challenging settings such as sparse user interactions, long-tail items, and heavy KG noise?

RQ4: How effective is the decision enhancement brought by the LLM?

A. Datasets

We use the publicly available preprocessed versions of Yelp2018, Amazon-Book, and MIND [7]. The dataset statistics are reported in Table I.

B. Evaluation metrics and protocol

We adopt an all-ranking (full-ranking) evaluation protocol. For metrics, we choose two representative measures in recommender systems: Recall@N and NDCG@N. Unless otherwise specified, we set $N = 20$.

C. Baselines for Comparison

We select representative recommendation methods from different research paradigms as baselines, covering multiple technical routes:

- BPR [14]: Pairwise ranking with Bayesian Personalized Ranking loss.
- NCF [15]: MLP-based collaborative filtering for nonlinear interaction modeling.
- LightGCN [16]: Simplified graph CF with neighborhood aggregation and layer weighting.
- SGL [17]: Graph CF with structural augmentation and self-supervised contrastive learning.
- CKE [18]: TransR-based KG embeddings integrated into a denoising autoencoder.
- RippleNet [19]: Path-based preference propagation over the KG to enhance user representations.
- KGCN [20]: Neighbor-aware aggregation on KG for higher-order semantic context modeling.
- KGAT [4]: Attention-based message passing for embedding fusion on the knowledge-aware graph.
- DiffKG [6]: Diffusion modeling on item–entity relations to denoise KGs and generate task-aligned KG views.
- MCCLK [21]: Hierarchical contrastive learning on the user–item–entity graph and subgraphs.
- KGCL [7]: Multi-view augmentations on KG and user–item graph with InfoNCE to handle noise and long-tail entities.
- CRHCL [5]: Context-aware relational semantics with multi-perspective contrastive learning to mitigate noise and ambiguity. It represents one of the recent advances in KG-enhanced recommendation.

D. Parameter settings

We implement DiCoLL-KGRec in PyTorch. Most baselines are reproduced using RecBole [22] or their officially released code. For each dataset, we adopt a chronological leave-one-out split: for each user, the most recent interaction is held out for testing and the second most recent for validation, and the remaining interactions are used for training. We set the embedding dimension to $d = 64$, use the Adam optimizer with a learning rate of 10^{-3}, and tune the L_2 regularization coefficient in $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ on the validation set. For the diffusion module, we set the number of diffusion steps to $T = 5$, while keeping other hyperparameters consistent with the empirical settings of DiffKG [6]. For the contrastive learning component, we uniformly sample intermediate steps from the reverse denoising trajectory to compute the cross-diffusion-view stability signal g_i, and tune $\lambda_{\text{kg-cl}}$ and τ_{kg} in $\{0.1, 0.2, \ldots, 0.9, 1.0\}$ (other settings follow KGCL [7]).

TABLE II
PERFORMANCE COMPARISON OF ALL METHODS ON YELP, AMAZON, AND MIND.

Model	Yelp2018		Amazon-book		MIND	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPR	5.55%	0.0375	12.44%	0.0658	9.38%	0.0469
NCF	5.35%	0.0346	10.33%	0.0532	8.93%	0.0436
LightGCN	6.82%	0.0443	13.98%	0.0736	10.33%	0.0520
SGL	7.19%	0.0475	14.45%	0.0766	10.32%	0.0539
CKE	6.86%	0.0431	13.75%	0.0685	9.01%	0.0382
RippleNet	4.22%	0.0251	10.58%	0.0549	8.58%	0.0407
KGCN	5.32%	0.0338	11.11%	0.0569	8.87%	0.0431
KGAT	6.75%	0.0432	13.90%	0.0739	9.07%	0.0442
DiffKG	7.71%	0.0498	14.87%	0.0782	10.85%	0.0562
MCCLK	5.47%	0.0356	12.17%	0.0641	8.96%	0.0439
KGCL	7.56%	0.0493	14.96%	0.0793	10.73%	0.0551
CRHCL	7.79%	0.0502	14.72%	0.0771	10.46%	0.0543
Ours	**8.18%**	**0.0522**	**15.22%**	**0.0807**	**11.61%**	**0.0593**

For the LLM, we use gpt-3.5-turbo with temperature $= 0$ and rerank only the Top-m ($m \leq K$) candidates, where $m \in \{10, 20, 30, 40, 50\}$ is selected on the validation set. All experiments are conducted on the same hardware, repeated multiple times, and we report the averaged results.

E. Overall Performance Comparison(RQ1)

Table II reports the overall results on the three datasets. We make the following observations:

- DiCoLL-KGRec achieves the best performance on all three datasets, indicating that integrating diffusion-based KG denoising, multi-view contrastive learning, and LLM-based reranking within a unified framework is effective. Despite substantial differences in interaction sparsity and KG structures, DiCoLL-KGRec consistently maintains an advantage, suggesting strong generalization across datasets. The gains can be attributed to two factors: (i) diffusion denoising yields a task-relevant clean KG and multiple structural views, while contrastive learning further suppresses residual noise and stabilizes representations; (ii) the LLM conducts fine-grained reranking within the Top-m candidates using structured prompts, improving ranking quality at the top positions.
- Diffusion-based denoising and contrastive-learning-based KG recommenders are among the most competitive baselines. Diffusion-based approaches improve KG quality through a noising–denoising process, while contrastive methods enhance representation consistency and robustness via multi-view augmentation and InfoNCE objectives. These two lines address noise and long-tail challenges from the data and representation levels, respectively, and therefore tend to outperform earlier KG-enhanced paradigms. Overall, the results suggest that combining diffusion-derived structural reliability (data level), dual-side alignment on the KG and user–item graphs (representation level), and LLM reranking within Top-m (decision level) yields stable gains across heterogeneous datasets.

TABLE III
ABLATION STUDY OF KEY COMPONENTS IN DICOLL-KGREC.

Model	Yelp2018		MIND	
	Recall	NDCG	Recall	NDCG
DiCoLL-KGRec	8.18%	0.0522	11.61%	0.0593
w/o Diffusion	7.67%	0.0494	10.80%	0.0559
w/o Middle Diffusion Views	7.85%	0.0506	11.17%	0.0574
w/o KG-CL	7.97%	0.0512	11.23%	0.0584
w/o LLM	8.06%	0.0514	11.38%	0.0585

F. Ablation Study(RQ2)

To quantitatively analyze the contribution of each key component, we construct four variants of DiCoLL-KGRec and compare them under the same settings. The variants are listed as follows:

- w/o Diffusion: Remove the diffusion-based KG denoising module. The original KG is directly used as the knowledge input, while the multi-view contrastive learning module is retained.
- w/o Middle Diffusion Views: We keep only the diffusion-generated $\mathcal{G}_k^{\text{clean}}$ as the denoised structural foundation, while removing the intermediate diffusion views and the resulting cross-diffusion-view stability g_i. As a result, diffusion consistency is no longer used to adaptively schedule the augmentation strength.
- w/o KG-CL: Remove the KG-side view-contrastive branch and apply the contrastive loss only on the user-item graph.
- w/o LLM: Keep only the graph-based model without the LLM-based two-stage reranking.

Table III reports the performance of these variants on two representative datasets, Yelp2018 and MIND. We make the following observations:

- Diffusion denoising is a key source of gains. Removing the diffusion module leads to the largest performance drop on both datasets, indicating that contrastive learning alone is insufficient to filter task-irrelevant links. In contrast, the diffusion-derived clean KG provides a more reliable structural foundation for downstream augmentation and representation learning.
- Components in the multi-view contrastive learning module exhibit synergy. The variant w/o Middle Diffusion Views retains $\mathcal{G}_k^{\text{clean}}$ and outperforms w/o Diffusion, suggesting that offline denoising alone yields stable benefits. Further incorporating intermediate diffusion views and using the cross-diffusion-view stability signal g_i to adaptively schedule augmentation strength brings additional improvements, indicating that diffusion consistency serves as a task-relevant reliability prior for contrastive learning. Removing the KG-side view contrast (w/o KG-CL) results in further degradation, highlighting the complementary effects of diffusion-consistency-guided augmentation and KG-side alignment.
- LLM reranking mainly improves the top ranks. Without two-stage reranking (w/o LLM), the graph-based retriever remains competitive but consistently underperforms the

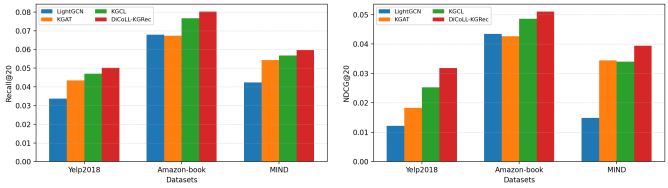

Fig. 2. Performance on cold-start users.

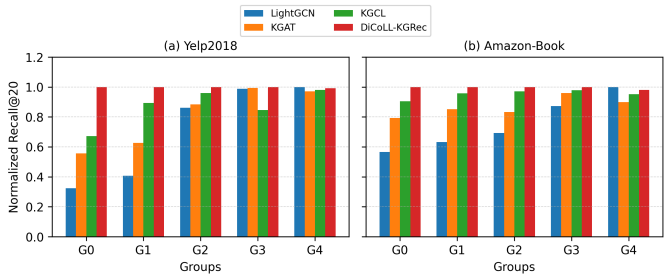

Fig. 3. Comparison across item groups with different interaction densities.

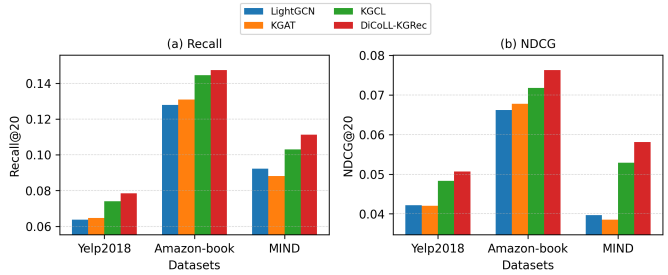

Fig. 4. Performance under KG noise.

full model. This is consistent with our design choice that the LLM refines the ordering only within Top-m, thereby improving head ranking quality.

G. Robustness Analysis under Sparsity, Long-tail, and Noise(RQ3)

We assess the robustness of DiCoLL-KGRec under three challenging settings: sparse user interactions, long-tail recommendation, and KG noise. We compare against LightGCN, KGAT, and KGCL (by default reporting the graph-based retriever only to avoid LLM randomness).

Sparse user interactions. We evaluate on cold-start users with < 20 interactions in Yelp2018/Amazon-Book and < 5 in MIND, and report Recall@20 and NDCG@20 (Fig. 2).

Long-tail item recommendation. Items are ranked by frequency and evenly split into five groups (G0: tail, G4: head). We compute group-wise Recall@20 with within-group normalization (Fig. 3).

KG noise. We inject 10% random noisy triples into the KG while keeping the test set unchanged (Fig. 4).

From the results under the three aspects above, we have the following observations:

- DiCoLL-KGRec is more robust under sparse-user and long-tail settings. On the cold-start user subsets, it consistently outperforms LightGCN and KGAT, and in most cases surpasses KGCL. In the long-tail grouping, the advantage is most pronounced in the $G0$–$G2$ ranges, suggesting that the diffusion-denoised clean KG and multi-view structural priors help compensate for sparse supervision, while the consistency-guided multi-view constraints further stabilize the learned representations.
- Performance degradation is more controlled under noisy-KG settings. Although all methods experience drops, DiCoLL-KGRec remains the best on Recall@20 and achieves NDCG@20 that is comparable to or slightly better than strong baselines. This indicates that combining generative denoising with dual-side contrastive constraints on the KG and user–item graphs helps mitigate the misleading effects of corrupted structures.

TABLE IV
DECISION-LEVEL REASONING EFFECT OF THE LLM MODULE ON YELP2018.

Subset	Model	NDCG@5
Easy	w/o LLM	0.0612
	w/ LLM	0.0625
Hard	w/o LLM	0.0317
	w/ LLM	0.0403

H. Analysis of the LLM-enhanced Module(RQ4)

In this subsection, we analyze the role of the LLM-based two-stage ranking in decision-level reasoning, with a focus on its ranking improvements on "hard instances". Unless otherwise specified, we use the graph-based model (w/o LLM) as the reference.

Decision enhancement on hard instances. We focus on users for whom the graph-based model is most uncertain. For each test user u, we take the Top-1 and Top-2 items i_1 and i_2 along with their scores $\hat{y}_{u,i_1}$ and $\hat{y}_{u,i_2}$, and define a difficulty measure $\Delta(u) = \hat{y}_{u,i_1} - \hat{y}_{u,i_2}$. After sorting users by $\Delta(u)$, we regard the top 30% as "easy instances" and the bottom 30% as "hard instances". We then compare w/o LLM and w/ LLM on these two subsets, and use NDCG@5 to evaluate head ranking quality; the results are reported in Table IV.

- Decision enhancement is concentrated on hard instances. The gains are modest on easy instances but substantially larger on hard instances, indicating that the LLM mainly contributes when the graph-based retriever is most uncertain. This is consistent with our design: when the score gap between top candidates is small (i.e., the model is "hesitating"), inner-product-based scoring provides limited resolution for fine-grained discrimination. By leveraging structured structural evidence in the prompt, the LLM can compare candidates more explicitly and refine their relative ordering within a constrained candidate set, leading to a larger improvement on hard instances.

VI. Conclusion and Future Work

In this paper, we propose DiCoLL-KGRec, a unified KG-based recommendation framework that employs diffusion-based denoising to derive a task-relevant clean KG and multiple structural views. Building on these views, we jointly perform diffusion-guided multi-view contrastive learning and LLM-enhanced two-stage reranking, forming a pipeline that progresses from structural purification to robust representation learning and finally to candidate refinement. Extensive experiments on three public datasets—Yelp2018, Amazon-Book, and MIND—show that DiCoLL-KGRec consistently outperforms a wide range of collaborative filtering and KG-based baselines in terms of Recall@20 and NDCG@20, while retaining stronger advantages under challenging conditions and on hard instances. A key takeaway is that converting diffusion multi-view discrepancies into structural reliability signals for augmentation scheduling, and coupling them with dual-side alignment on the KG and user–item graphs as well as LLM-based decision refinement, can improve both structural robustness and ranking quality. In future work, we will extend DiCoLL-KGRec to dynamic graphs and more complex recommendation scenarios, and explore integrating LLM capabilities more tightly into training (e.g., via distillation).

Acknowledgment

This work was supported in part by the Key Research and Development Program of Zhejiang Province under Grant Nos. 2026C02A1242 and 2026C01015.

References

[1] Y. Hao, F. Zhuang, D. Wang, G. Liu, V. S. Sheng, and P. Zhao, "A general strategy graph collaborative filtering for recommendation unlearning," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 799–808.

[2] E. Elahi, S. Anwar, M. Al-kfairy, J. J. Rodrigues, A. Ngueilbaye, Z. Halim, and M. Waqas, "Graph attention-based neural collaborative filtering for item-specific recommendation system using knowledge graph," *Expert Systems with Applications*, vol. 266, p. 126133, 2025.

[3] H. Zhang, X. Shen, B. Yi, J. Liu, and Y. Xie, "A plug-in critiquing approach for knowledge graph recommendation systems via representative sampling," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 322–333.

[4] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 950–958.

[5] Q. Ye, X. Feng, H. Xu, C. Wang, B. Zhang, P. Wang, C. Liu, and K. Wang, "Knowledge graph enhanced recommendation via context-aware dynamic relation perception and dual-view hierarchical contrastive learning," *Expert Systems with Applications*, p. 130636, 2025.

[6] Y. Jiang, Y. Yang, L. Xia, and C. Huang, "Diffkg: Knowledge graph diffusion model for recommendation," in *Proceedings of the 17th ACM international conference on web search and data mining*, 2024, pp. 313–321.

[7] Y. Yang, C. Huang, L. Xia, and C. Li, "Knowledge graph contrastive learning for recommendation," in *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 2022, pp. 1434–1443.

[8] Y. Yang, C. Huang, L. Xia, and C. Huang, "Knowledge graph self-supervised rationalization for recommendation," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 3046–3056.

[9] F. Meng, Z. Meng, R. Jin, R. Lin, and B. Wu, "Doge: Llms-enhanced hyper-knowledge graph recommender for multimodal recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 12, 2025, pp. 12 399–12 407.

[10] J. Gao, B. Chen, X. Zhao, W. Liu, X. Li, Y. Wang, W. Wang, H. Guo, and R. Tang, "Llm4rerank: Llm-based auto-reranking framework for recommendations," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 228–239.

[11] Y. Wei, Q. Huang, Y. Zhang, and J. Kwok, "Kicgpt: Large language model with knowledge in context for knowledge graph completion," in *Findings of the association for computational linguistics: EMNLP 2023*, 2023, pp. 8667–8683.

[12] K. Bao, J. Zhang, X. Lin, Y. Zhang, W. Wang, and F. Feng, "Large language models for recommendation: Past, present, and future," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 2993–2996.

[13] H. Zhou, C. Hu, Y. Yuan, Y. Cui, Y. Jin, C. Chen, H. Wu, D. Yuan, L. Jiang, D. Wu *et al.*, "Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 3, pp. 1955–2005, 2024.

[14] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[15] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 173–182.

[16] 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.

[17] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 2021, pp. 726–735.

[18] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 353–362.

[19] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems," in *Proceedings of the 27th ACM international conference on information and knowledge management*, 2018, pp. 417–426.

[20] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *The world wide web conference*, 2019, pp. 3307–3313.

[21] D. Zou, W. Wei, X.-L. Mao, Z. Wang, M. Qiu, F. Zhu, and X. Cao, "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*, 2022, pp. 1358–1368.

[22] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Lu, H. Wang, C. Tian *et al.*, "Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms," in *proceedings of the 30th acm international conference on information & knowledge management*, 2021, pp. 4653–4664.