Hard vs. Noise: Resolving Hard-Noisy Sample Confusion in Recommender Systems via Large Language Models

Tianrui Song[1], **Wen-Shuo Chao**[1], **Hao Liu**[1,2*]

[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology
tsong847@connect.hkust-gz.edu.cn, wschao829@connect.hkust-gz.edu.cn, liuh@ust.hk

Abstract

Implicit feedback, employed in training recommender systems, unavoidably confronts noise due to factors such as misclicks and position bias. Previous studies have attempted to identify noisy samples through their diverged data patterns, such as higher loss values, and mitigate their influence through sample dropping or reweighting. However, we observed that noisy samples and hard samples display similar patterns, leading to hard-noisy confusion issue. Such confusion is problematic as hard samples are vital for modeling user preferences. To solve this problem, we propose LLMHNI framework, leveraging two auxiliary user-item relevance signals generated by Large Language Models (LLMs) to differentiate hard and noisy samples. LLMHNI obtains user-item semantic relevance from LLM-encoded embeddings, which is used in negative sampling to select hard negatives while filtering out noisy false negatives. An objective alignment strategy is proposed to project LLM-encoded embeddings, originally for general language tasks, into a representation space optimized for user-item relevance modeling. LLMHNI also exploits LLM-inferred logical relevance within user-item interactions to identify hard and noisy samples. These LLM-inferred interactions are integrated into the interaction graph and guide denoising with cross-graph contrastive alignment. To eliminate the impact of unreliable interactions induced by LLM hallucination, we propose a graph contrastive learning strategy that aligns representations from randomly edge-dropped views to suppress unreliable edges. Empirical results demonstrate that LLMHNI significantly improves denoising and recommendation performance.

Code — https://github.com/TianRui-Song717/LLMHNI
Extended version — http://arxiv.org/abs/2511.07295

Introduction

Recommender Systems (RS) rely on implicit feedback, such as clicks and purchases, to model user preferences (He et al. 2020; Luo et al. 2020). Traditionally, these interactions are labeled positively if observed and negatively if not (Ding et al. 2020; Wang et al. 2021a). However, this schema is questioned due to the false-positive noise from misclicks and false-negative noise from position bias (Wang et al. 2021b). To address such noise issues, denoising recommendation strategies have emerged, including sample dropping and sample reweighting. Sample dropping mitigates the impact of noise by removing noisy interactions during training (Chen et al. 2021), while reweighting assigns lower weights to noisy interactions (Wang et al. 2023; Gao et al. 2022). These techniques hinge on accurately distinguishing between clean and noisy samples by their divergent patterns in loss value (Ding et al. 2020), prediction scores (Wang et al. 2021a), and gradients (Wang et al. 2023).

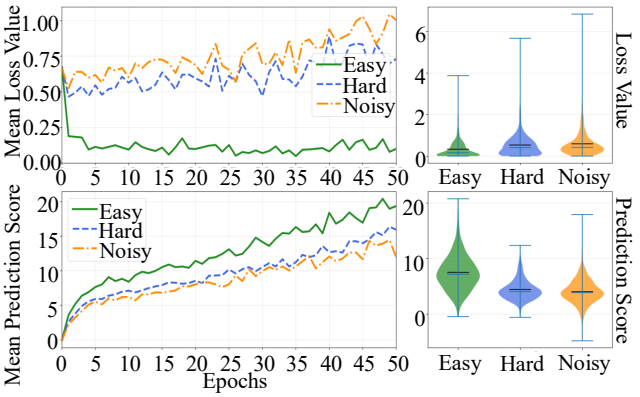
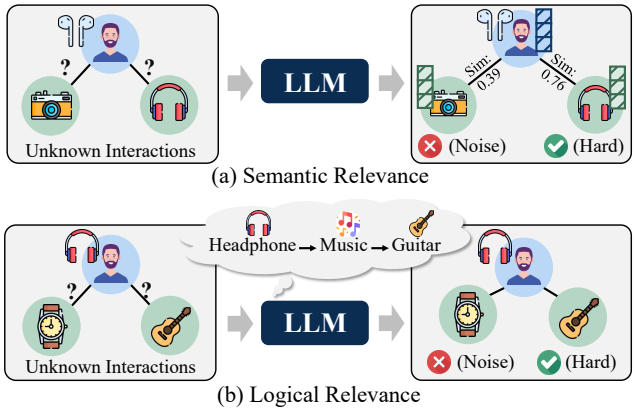


Figure 1: On the left, we demonstrate that hard and noisy samples display similar patterns in both loss values and prediction scores throughout the training process. On the right, we take the results from the 5th epoch as an example to illustrate how the prediction scores and loss values of hard and noisy samples overlap in distribution. Additional details about this figure can be found in the appendix.

Despite their advancements, these denoising methods often face the challenge of misidentifying hard samples as noisy ones. As illustrated in Figure 1, while noisy samples exhibit distinct patterns compared to easy samples, we observed that hard samples and noisy samples tend to present similar patterns in both prediction scores and loss values. Consequently, previous denoising approaches that rely solely on data patterns *struggle to distinguish between hard and noisy samples*. This misidentification is problematic because hard samples have been shown to be beneficial, both empirically (Gantner et al. 2012) and theoretically (Shi

*Corresponding author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.



(a) Semantic Relevance

(b) Logical Relevance

Figure 2: Semantic Relevance and Logical Relevance.

et al. 2023). Mistakenly treating hard samples as noise during training ultimately leads to suboptimal results.

Since distinguishing hard and noisy samples based solely on numerical patterns derived from user-item collaborative information is insufficient, addressing this issue necessitates auxiliary signals. Recently, Large Language Models (LLMs) have emerged as powerful tools to enhance recommender systems. Existing approaches take the knowledge generated by LLMs as supplementary information beyond collaborative signals in recommender systems (Lin et al. 2025). Inspired by their promising performance, we employ LLMs to provide auxiliary information for distinguishing hard samples from noisy ones. Specifically, we exploit two types of user-item relevance signals that distinct from those captured by user-item interactions. 1) Semantic Relevance in LLM-encoded embedding: As shown in Fig. 2(a), LLM-encoded user and item embeddings offer semantic relevance between users and items, which helps identify hard and noisy samples with relevance scores. 2) Logical Relevance in LLM-inferred interactions: LLMs possess reasoning capabilities that can infer logical relevance within user-item interactions and distinguish hard samples from noisy ones. As shown in Fig. 2(b), LLM deduces that a user bought headphones enjoys music and might therefore be interested in a guitar.

However, leveraging these two auxiliary relevance signals to distinguish hard and noisy samples in recommender systems faces two challenges: 1) **Objective-Mismatched Embeddings**: LLM-encoded embeddings, trained for general language tasks rather than user preference modeling, suffer an objective mismatch for recommendation tasks. Consequently, the user-item similarity values derived from these objective-mismatched embeddings can mislead the identification between hard and noisy samples and hinder recommendation model performance. 2) **Hallucination-Induced Interactions**: LLMs suffer from hallucination, which undermines the reliability of their inferred user-item interactions. Including these hallucination-induced interactions during training may amplify label noise and propagate hallucination errors into the recommendation model.

To overcome aforementioned challenges, we introduce the **L**arge **L**anguage **M**odels enhanced **H**ard-**N**oisy sample **I**dentification framework (**LLMHNI**). It comprises two modules that take auxiliary signals generated by LLMs to differentiate hard and noisy samples, improving the denoising process. The first module, Semantic Relevance Guided Hard Negative Mining, harnesses LLMs to encode text profiles of users and items. Semantic relevance (i.e., embedding similarities) between users and items are used to guide negative sampling, facilitating the selection of hard negatives while avoiding the introduction of false negatives. To further mitigate the *objective-mismatched embedding*, we design an objective alignment strategy that projects raw LLM-encoded embeddings into a tailored representation space optimized for preference modeling. The second module, Logical Relevance Guided Interaction Denoising, employs LLMs to infer logical relevance within user-item interactions, identifying hard and noisy ones. These interactions are integrated into interaction graph and guide interaction denoise. Specifically, we design a cross-graph contrastive alignment that suppresses interactions inconsistent between the original graph and the one enhanced with LLM-inferred hard and noisy interactions. To mitigate *hallucination-induced interactions* within interaction graph, a graph contrastive learning strategy is incorporated, which suppresses hallucination-induced edges by aligning representations from two randomly edge-dropped views of the interaction graph.

Our main contributions are summarized as follows.

- We propose **LLMHNI**, a novel framework that takes semantic relevance signals in LLM-encoded embeddings and logical relevance signals in LLM-inferred interactions to guide negative sampling and interaction denoising, addressing the noisy-hard sample confusion in RS.
- LLMHNI addresses the objective mismatch of LLM-encoded embeddings by projecting the raw embedding into an aligned representation space. It also reduces the influence of hallucination-induced interactions inferred by LLM with a graph contrastive learning strategy.
- Extensive experiments on three real-world datasets and two backbone recommenders demonstrate the effectiveness of our method. Results show that LLMHNI delivers impressive performance and robust noise resilience.

Preliminary

The objective of training a recommender system is to learn a scoring function $\hat{y}_{u,i} = f_\theta(u, i)$ from interactions between users $u \in \mathcal{U}$ and items $i \in \mathcal{I}$. We assume that user-interacted items $y^*_{u,i} = 1$ are preferred by the user, while those not interacted $y^*_{u,i} = 0$ are not. To optimize the scoring function $f_\theta(u, i)$, we employ Bayesian Personalized Ranking (BPR) loss as loss function $\mathcal{L}_{rec}$, which are formulated as follows:

$$\mathcal{L}_{\text{BPR}}(\mathcal{D}^*) = \underset{(u,i,j)\sim\mathbf{P}_{\mathcal{D}^*}}{-\mathbb{E}[\log(\sigma\,(\hat{y}_{u,i} - \hat{y}_{u,j}))]}, \tag{1}$$

where j denotes negative items sampled according to the distribution $\mathbf{P}_{\mathcal{D}^*}$, and $\mathcal{D}^* = \{(u, i, y^*_{u,i}) \mid u \in \mathcal{U}, i \in \mathcal{I}\}$ represents the dataset. σ denotes the sigmoid. The optimal parameter θ^* is obtained by minimizing the $\mathcal{L}_{rec}$:

$$\theta^* = \arg\min_\theta \mathcal{L}_{rec}(\mathcal{D}^*), \tag{2}$$

But this assumption is unreliable for two reasons: *(1) False positive issue*, user-interacted items might not reflect real user preference due to factors such as accidental clicks and position bias. *(2) False negative issue*, non-interacted items are not necessarily user dislikes, they may have been overlooked due to factors such as suboptimal display positions. These issues introduce noisy interactions, formally defined as $\tilde{\mathcal{D}} = \{(u, i, \tilde{y}) \mid \tilde{y} \neq y^*\}$. To address this, in this work, we formulate *denoising recommender training task* as:

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{rec}(\mathcal{D}^* \cup \tilde{\mathcal{D}}, \mathcal{P}_{\mathcal{U}}, \mathcal{P}_{\mathcal{I}}), \tag{3}$$

where $\mathcal{P}_{\mathcal{U}} = \{\mathcal{P}_u | u \in \mathcal{U}\}, \mathcal{P}_{\mathcal{I}} = \{\mathcal{P}_i | i \in \mathcal{I}\}$ are the text profiles of users and items that describe user prefereces and item characteristics, respectively.

Proposed Method

We present the **LLMHNI**, a novel framework that harnesses LLM-generated auxiliary signals to resolve the *hard-noisy sample confusion*. As illustrated in Fig. 3, LLMHNI integrates two modules: (1) *Semantic Relevance Guided Hard Negative Mining* leverages user and item embeddings encoded by LLMs to capture semantic relevance, facilitating effective sampling of hard negatives while minimizing the risk of false negatives. (2) *Logical Relevance Guided Interaction Denoising* employs LLM-inferred logical user-item relevance to identify hard and noisy interactions, thereby refining the interaction graph and samples for noise mitigation.

Semantic Relevance Guided Hard Negative Mining

LLM-encoded textual embeddings inherently capture semantic relevance between users and items, which are critical auxiliary information benefiting hard-noisy sample differentiation. In this part, we leverage these inherent semantic relevance signals to select hard negative items while filtering out false negatives for recommender system.

Objective-Aligned Embedding Generation. User and item text embeddings encoded by LLMs reflect user-item semantic relevance. However, LLMs are trained for language modeling, which hinders the embeddings' effectiveness in reflecting the user-item correlation required for the recommendation. To address this issue, we project embeddings to an optimized representation space. We utilize the LLM embedding model LLM_{enc} to encode the text profiles of user ($\mathcal{P}_u$) and item ($\mathcal{P}_i$). An MLP projects embeddings to a low-dimensional representation space for objective alignment

$$\mathbf{e}_u^{llm} = \text{LLM}_{\text{enc}}(\mathcal{P}_u), \quad \mathbf{e}_i^{llm} = \text{LLM}_{\text{enc}}(\mathcal{P}_i), \tag{4}$$

$$\mathbf{z}_u^{llm} = \text{MLP}(\mathbf{e}_u^{llm}), \quad \mathbf{z}_i^{llm} = \text{MLP}(\mathbf{e}_i^{llm}) \tag{5}$$

where $\mathbf{e}_u^{llm} \in \mathbb{R}^{d_{llm}}$ and $\mathbf{e}_i^{llm} \in \mathbb{R}^{d_{llm}}$ represent the text embeddings for user u and item i, $\mathbf{z}_u^{llm} \in \mathbb{R}^{d_{rec}}$ and $\mathbf{z}_i^{llm} \in \mathbb{R}^{d_{rec}}$ are the projected embeddings ($d_{rec} << d_{llm}$).

To train the projector, we then construct pseudo labels: for each user, items that (1) occupy top-ranked textual embedding similarity scores and (2) have prior interaction with the user are considered reliable labels. Formally, for each user u, we define a set of reliable positive items as follows:

$$\mathcal{I}_u^{al+} = \left\{i | y_{u,i}^* = 1\right\} \cap \left\{i | \text{Top-}N(\hat{y}_{u,i}^{llm})\right\}, \tag{6}$$

where $\hat{y}_{u,i}^{llm} = \text{sim}(\mathbf{e}_u^{llm}, \mathbf{e}_i^{llm})$ denote the cosine similarity of LLM-encoded embeddings, N is a hyperparameter controlling sample quality (typical $N = 50$). The MLP projector is then trained with the following objective,

$$\mathcal{L}_{al} = -\log \frac{\exp(\mathbf{z}_u \cdot \mathbf{z}_{i^{al+}}/\tau)}{\exp(\mathbf{z}_u \cdot \mathbf{z}_{i^{al+}}/\tau) + \sum_{k=1}^{N} \exp(\mathbf{z}_u \cdot \mathbf{z}_{i_k^{al-}}/\tau)}, \tag{7}$$

where $i^{al+} \in \mathcal{I}_u^{al+}$ and i_k^{al-} are random sampled negatives from $\{\mathcal{I} \setminus \mathcal{I}_u^{al}\}$, $\tau = 0.5$ is a temperature hyperparameter. After training the MLP projector, the resulting aligned text embeddings can be formulated as $\mathbf{z'}_u^{llm} = \text{MLP}'(\mathbf{e}_u^{llm})$, $\mathbf{z'}_i^{llm} = \text{MLP}'(\mathbf{e}_i^{llm})$, where $\mathbf{z'}_u^{llm}, \mathbf{z'}_i^{llm} \in \mathbb{R}^{d_{rec}}$ and MLP' denote the traiend projector.

Semantic-Guided Hard Negative Sampling. We leverage the objective-aligned $\mathbf{z'}_u^{llm}$ and $\mathbf{z'}_i^{llm}$ in negative sampling to select hard negatives and filter out noisy false negatives. For each u, we randomly initialize a hard negative pool $\mathbf{HN}_u^- = \{j \mid \forall j \in \mathcal{I}_u^-\}_{k=1}^K$ with K negative items. When training Rec_θ, for each positive $(u, i) \in \mathcal{B}$, we uniformly sample M new negative items by

$$\mathbf{N}_u^- = \{j_m \mid j_m \sim \text{Uniform}(\mathcal{I}_u^-)\}_{m=1}^M. \tag{8}$$

The $\mathbf{N}_u^-$ is adopted to update $\mathbf{HN}_u^-$ dynamically according to the recommender system prediction scores $\hat{y}_{u,i} = \text{Rec}_\theta(u, i)$, formally represented as

$$\mathbf{HN}_u^- = \{j_k | j_k \sim P(j) \propto \hat{y}_{u,j}, \forall j \in \mathbf{HN}_u^- \cup \mathbf{N}_u^-\}_{k=1}^K, \tag{9}$$

where $P(j)$ denotes the sampling distribution. Considering false negatives might exhibit both high $\hat{y}_{u,j}$ and high semantic similarity, we select the negative item j from $\mathbf{HN}_u^-$ with the lowest semantic similarity score,

$$j = \arg\min_{k \in \mathbf{HN}_u^-} (s(\mathbf{z'}_u^{llm}, \mathbf{z'}_{j_k}^{llm})), \forall j_k \in \mathbf{HN}_u^- \tag{10}$$

where $s(\cdot)$ denotes the cosine similarity. We then take the hard negative j and positive (u, i) interaction pair to optimize the Rec_θ with recommendation loss (i.e., BPR loss),

$$\mathcal{L}_{\text{rec}} = -\frac{1}{|\mathcal{B}|} \sum_{(u,i) \in \mathcal{B}} \log\left(\sigma\left(\hat{y}_{u,i} - \hat{y}_{u,j}\right)\right), \tag{11}$$

where the negative item j is selected via Eq.10.

Logical Relevance Guided Interaction Denoising

With powerful reasoning capability, Large Language Models can infer the logical relationships between users and items that reveal users' potential interest in items. Therefore, we design the following strategies that take into account these LLM-inferred logical relevance within user-item interactions to identify noisy and hard samples.

Logical Relevance Inference. We first obtain the logical relevance between users and items in RS with LLM. Given the enormous number of u and i, employing LLMs to scrutinize every user-item pair is infeasible. Therefore, we select potential hard and noisy interactions before subjecting them

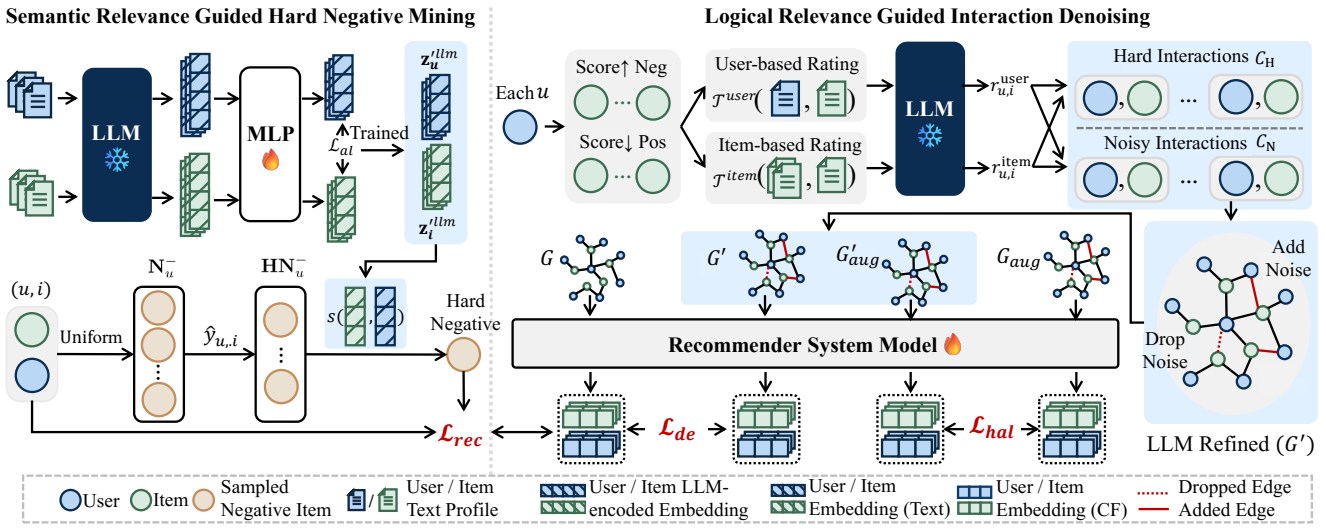


Figure 3: The overview of our proposed LLMHNI framework.

to LLM for logical relevance analysis. Specifically, we leverage a trained recommender system Rec_{pre} to sample user-item pairs. For each user u, two subsets are constructed: (1) High-Score Negatives: n_1 items that have higher prediction scores $\hat{y}_{u,i}^{pre} = \text{Rec}_{pre}(u,i)$, where $y_{u,i}^{*} = 0$. (2) Low-Score Positives: n_2 items that have lower prediction scores $\hat{y}_{u,i}^{pre}$ and $y_{u,i}^{*} = 1$. The unified candidate set $\mathcal{C}$ is defined as:

$$\mathcal{C} = \bigcup_{u\in\mathcal{U}} (\underbrace{\{(u,i)|i \sim P_{\mathcal{I}_u^-}(i) \propto \hat{y}_{u,i}^{pre}\}}_{\text{High-score false negatives}} \cup \underbrace{\{(u,i)|i \sim P_{\mathcal{I}_u^+}(i) \propto -\hat{y}_{u,i}^{pre}\}}_{\text{Low-score false positives}}), \tag{12}$$

where $\mathcal{I}_u^- = \{i \mid y_{u,i}^{*} = 0\}$, $\mathcal{I}_u^+ = \{i \mid y_{u,i}^{*} = 1\}$, $P_{\mathcal{S}}(i)$ denotes the sampling distribution on the set $\mathcal{S}$.

For each candidate user-item pair $(u,i) \in C$, we assess their relevance from two aspects: (1) User-based Rating: This employs the user text profile to describe user u's preference. LLM is prompted with a predefined prompt template $\mathcal{T}^{\text{user}}$ to rate the logical relevance between u and i.

$$r_{u,i}^{\text{user}} = \text{LLM}(\mathcal{T}^{\text{user}}(V_u^{\text{user}}, \mathcal{P}_i)), \tag{13}$$

where $r_{u,i}^{\text{user}} \in$ [High, Mid, Low] and $V_u^{\text{user}} = \mathcal{P}_u$. (2) Item-based Rating: This takes the profiles of item that u has interacted and has high $\hat{y}_{u,i}^{pre}$ as u's preference descriptions. Using predefined prompt templates $\mathcal{T}^{\text{item}}$, LLM rates the logical relevance between u and i as follows,

$$r_{u,i}^{\text{item}} = \text{LLM}(\mathcal{T}^{\text{item}}(V_u^{\text{item}}, \mathcal{P}_i)), \tag{14}$$

where $r_{u,i}^{\text{item}} \in$ [High, Mid, Low], $V_u^{\text{item}} = \{\mathcal{P}_j | j \in \mathcal{I}, y_{u,j}^{*} = 1, \hat{y}_{u,j}^{pre} \in \text{top-K}(\hat{\mathbf{y}}_u^{pre})\}$, $\hat{\mathbf{y}}_u^{pre}$ denotes the preference scores of u with all items predicted by Rec_{pre}.

Interaction Denoising. Building on the logical relevance rates $r_{u,i}^{\text{user}}$ and $r_{u,i}^{\text{item}}$, we identify hard and noisy samples within the candidate set $\mathcal{C}$. To preserve performance-enhancing hard samples while conservatively filtering noise, we define the noise subset $\mathcal{C}_{\text{N}}$ and hard subset $\mathcal{C}_{\text{H}}$ as follows,

$$\mathcal{C}_{\text{H}} = \{(u,i) \in \mathcal{C} \mid r_{u,i}^{\text{user}} = \text{High} \wedge r_{u,i}^{\text{item}} = \text{High}\} \tag{15}$$

$$\mathcal{C}_{\text{N}} = \mathcal{C} \setminus \mathcal{C}_{\text{H}} \tag{16}$$

That is, (u,i) interactions are considered as hard samples if both the User-Centric and Item-Centric ratings yield High logical relevance scores. All remaining samples in $\mathcal{C}$ are treated as noisy samples. As integrate $\mathcal{C}_{\text{H}}$ and $\mathcal{C}_{\text{N}}$ to train recommenders might introduce label noise. We construct the user-item interaction graph $G' = \{\mathcal{U}, \mathcal{I}, \mathcal{E}'\}$ with the original interaction graph $G = \{\mathcal{U}, \mathcal{I}, \mathcal{E}\}$ to guide interaction denoise. Here, the edge $\mathcal{E}'$ can be formally formulated as

$$\mathcal{E}' = \mathcal{E} \setminus \{e_{u,i} \mid (u,i) \in \mathcal{C}_{\text{N}}\} \cup \{e_{u,i} \mid (u,i) \in \mathcal{C}_{\text{H}}\}, \tag{17}$$

where $e_{u,i}$ denotes an edge between user u and item i. We obtain user $(\mathbf{z}_u, \mathbf{z}'_u)$ and item $(\mathbf{z}_i, \mathbf{z}'_i)$ representations from both G and G' with the recommender Rec_θ, all in $\mathbb{R}^{d_{rec}}$,

$$\mathbf{z}_u, \mathbf{z}_i = \text{Rec}_\theta(G); \quad \mathbf{z}'_u, \mathbf{z}'_i = \text{Rec}_\theta(G'). \tag{18}$$

A cross-graph contrastive alignment strategy is designed to enhance (u,i) interactions that are consistent on G and G',

$$\mathcal{L}_{\text{de}}^{u'i} = -\frac{1}{|\mathcal{B}|} \sum_{(u,i)\in\mathcal{B}} \log \frac{\exp(s(\mathbf{z}'_u, \mathbf{z}_i)/\tau_{\text{de}})}{\sum_{(u,j)\in\mathcal{B}} \exp(s(\mathbf{z}'_u, \mathbf{z}_j)/\tau_{\text{de}})} \tag{19}$$

where $s(\cdot,\cdot)$ is cosine similarity, and τ_{de} is temperature hyperparameter. Together with the item side loss, the denoise loss is $\mathcal{L}_{\text{de}} = \mathcal{L}_{\text{de}}^{u'i} + \mathcal{L}_{\text{de}}^{ui'}$. As all positive (u,i) pairs ($i \in \mathcal{I}_u^+$) might appear in both the numerator and the denominator of $\mathcal{L}_{\text{de}}$, the embedding of (u,i) pairs that consistent with G and G' (i.e., high similarity between $(\mathbf{z}_u, \mathbf{z}'_i)$ and $(\mathbf{z}'_u, \mathbf{z}_i)$) will be aligned better, while those inconsistent are suppressed.

Hallucination-Robust Contrastive Learning. Although G' are constructed based on LLM-inferred interaction, leveraging $\mathcal{C}_{\text{H}}$ and $\mathcal{C}_{\text{N}}$ risks propagate hallucination-induced interactions. Therefore, we design a graph contrastive learning strategy to reduce the negative impact of hallucination-induced edges in G'. In each training batch, we generate two augmented views by stochastic edge drop to G' and G:

$$G_{\text{aug}} = (\mathcal{U}, \mathcal{I}, \mathcal{E} \backslash \mathcal{M}); \quad G'_{\text{aug}} = (\mathcal{U}, \mathcal{I}, \mathcal{E}' \backslash \mathcal{M}'), \tag{20}$$

where $\mathcal{M}', \mathcal{M} \sim \text{Bernoulli}(\rho; |\mathcal{E}'|)$ denotes the set of randomly masked edges. Each edge is dropped independently with probability ρ. Both G'_{aug} and G_{aug} are processed by the same graph-based recommendation models Rec_θ with parameters θ, generating user and item representations:

$$\mathbf{z}_u^{(1)}, \mathbf{z}_i^{(1)} = \text{Rec}_\theta(G'_{\text{aug}}); \mathbf{z}_u^{(2)}, \mathbf{z}_i^{(2)} = \text{Rec}_\theta(G_{\text{aug}}). \quad (21)$$

where $\mathbf{z}_u^{(k)}, \mathbf{z}_i^{(k)}$ are the user and item representations ($k = [1, 2]$). We adopt contrastive loss to maximize the agreement of positive pairs and minimize that of negative pairs,

$$\mathcal{L}_{\text{hal}}^{\text{user}} = \sum_{u \in \mathcal{U}} -\log \frac{\exp(s(\mathbf{z}_u^{(1)}, \mathbf{z}_u^{(2)})/\tau_{\text{hal}})}{\sum_{v \in \mathcal{U}} \exp(s(\mathbf{z}_u^{(1)}, \mathbf{z}_v^{(2)})/\tau_{\text{hal}})}, \quad (22)$$

where $s(\cdot)$ denotes the cosine similarity; τ_{hal} is the temperature hyperparameter. We get the objective function by combining the item side loss $\mathcal{L}_{\text{hal}} = \mathcal{L}_{\text{hal}}^{\text{user}} + \mathcal{L}_{\text{hal}}^{\text{item}}$.

Joint Optimization. We optimize the recommender system model with the total loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \lambda_1 \mathcal{L}_{\text{de}} + \lambda_2 \mathcal{L}_{\text{hal}}, \quad (23)$$

where λ_1, λ_2 are hyperparameters that balance the weight.

Experiment

We compare LLMHNI with state-of-the-art denoise methods on two backbones and three real-world datasets. Experiments are directed by following research questions (RQs):

- **RQ1:** How does LLMHNI performs compared with other state-of-the-art denoise methods across datasets?
- **RQ2:** Does the LLMHNI demonstrate robustness when tackling different levels of noisy data?
- **RQ3:** What is the effect of different components within the LLMHNI on performance?
- **RQ4:** How do hyperparameters in LLMHNI influence the effectiveness?
- **RQ5:** What is the training efficiency of LLMHNI?

Experiment Settings

Datasets. We conduct all experiments on three datasets: (1) **Amazon-Books** collected from the Amazon platform. We conduct experiments on the book subcategories. (2) **Yelp** is a large-scale dataset with real check-in history. (3) **Steam** consists of users and games on the Steam platform. Since we adopt the item and user profile provided in (Ren et al. 2024), we process these datasets following their settings.

Evaluation Metrics. Following existing works on recommender system denoising (Wang et al. 2021c; He et al. 2024), we report the results w.r.t. two widely used metrics: NDCG@K and Recall@K (K= $[10, 20]$).

Baselines. We conduct experiments with the NGCF (Wang et al. 2019) and LightGCN (He et al. 2020) backbones. Three types of denoising approaches are compared: (1) Instance-level approaches, including WBPR (Gantner et al. 2012), T-CE (Wang et al. 2021a) and BOD (Wang et al. 2023). (2) Representation-level approaches, including SGL (Wu et al. 2021), SimGCL (Yu et al. 2022) and XSimGCL (Yu et al. 2023). (3) LLM Enhanced approaches, including RLMRec (Ren et al. 2024) and LLaRD (Wang et al. 2025).

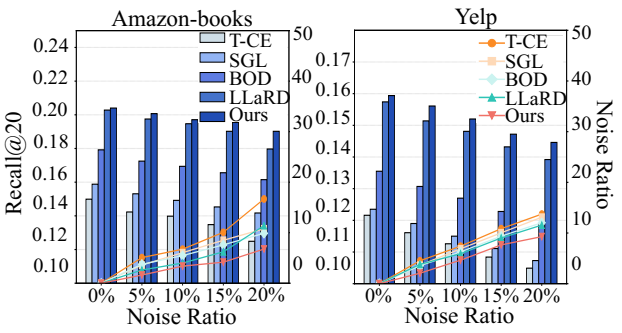


Figure 4: Model performance $w.r.t$ different noise ratio. The bar chart represents Recall values (see left y-axis), while the line chart shows Drop Rate (see right y-axis) All denoise methods are trained with the LightGCN backbone.

Implementation Details. For all models, the embedding size is set to 64, the batch size is 1024, and the learning rate is 1e-3. All models are trained with the Adam optimizer. For baseline models, we refer to their best parameter setups reported in original papers. For our model, we set "gpt-4o-2024-08-06" as LLM and "text-embedding-ada-002" as LLM_{enc}. We set the uniform sampled negative item number M at 30 and the hard negative candidate number K at 10.

Performance Comparison (RQ1)

To evaluate the effectiveness and generalizability of our propsoed framework, we compared our LLMHNI with existing denoising baselines across three datasets and two backbone models. The result is shown in Table 1. Our LLMHNI consistently exceeds desnoising baselines in all three datasets and both backbone models. On average, LLMHNI achieves 46.55% improvements on the vanilla NGCF backbone and 45.31% on the original LightGCN backbone. Compared with previous instance-level denoising approaches (i.e., T-CE and BOD) and representation-level techniques (i.e., SGL, SimGCL, and XSimGCL), LLMHNI exhibits a substantial performance improvement from 11.78% to 37.73%. This significant enhancement is attributed to our utilization of LLMs to provide auxiliary relevance signals beyond the original interaction data. Regarding LLM-enhanced denoising techniques (i.e., RLMRec and LLaRD), LLM outperforms them by roughly 2.47% to 33.86%. Although RLMRec and LLaRD incorporate supplementary information generated by LLMs, they lack the capabilities of identifying hard samples. Our LLMHNI, in comparison, extends the LLM-provided signals in hard sample identification, thereby excelling both baselines.

Noise Robustness (RQ2)

To assess the robustness of LLMHNI's noise resistance capabilities, following previous methods (Ren et al. 2024; Wang et al. 2023), we add certain levels of non-existent interactions to the training set (i.e., 5%, 10%, 15%, 20% negative interactions) and keep the test set unchanged. Fig. 4 shows the results in the Amazon-books and Yelp. Our

Dataset		Amazon-book				Yelp				Steam			
Backbone	**Method**	**R@10**	**N@10**	**R@20**	**N@20**	**R@10**	**N@10**	**R@20**	**N@20**	**R@10**	**N@10**	**R@20**	**N@20**
NGCF	Normal	0.0763	0.0584	0.1204	0.0726	0.0634	0.0527	0.1045	0.0664	0.0795	0.0658	0.1271	0.0814
	WBPR	0.0765	0.0587	0.1212	0.0729	0.0636	0.0530	0.1048	0.0669	0.0796	0.0657	0.1270	0.0813
	T-CE	0.0844	0.0648	0.1288	0.0789	0.0650	0.0543	0.1071	0.0683	0.0808	0.0671	0.1290	0.0828
	BOD	0.1199	0.0959	0.1666	0.1109	0.0721	0.0615	0.1193	0.0770	0.0921	0.0767	0.1461	0.0944
	SGL	0.0902	0.0707	0.1362	0.0856	0.0667	0.0557	0.1103	0.0704	0.0832	0.0689	0.1305	0.0843
	SimGCL	0.0863	0.0667	0.1304	0.0807	0.0680	0.0577	0.1140	0.0732	0.0819	0.0680	0.1274	0.0830
	XSimGCL	0.0963	0.0746	0.1431	0.0894	0.0701	0.0598	0.1153	0.0744	0.0957	0.0710	0.1352	0.0871
	RLMRec	0.0855	0.0635	0.1323	0.0815	0.0697	0.0568	0.1150	0.0723	0.0865	0.0690	0.1370	0.0862
	LLaRD	0.1265	0.1021	0.1834	0.1203	0.0845	0.0758	0.1347	0.0903	0.0997	0.0821	0.1565	0.1012
	LLMHNI	**0.1290**	**0.1038**	**0.1852**	**0.1230**	**0.0880**	**0.0823**	**0.1367**	**0.0938**	**0.1037**	**0.0864**	**0.1588**	**0.1057**
LightGCN	Normal	0.0950	0.0746	0.1415	0.0893	0.0617	0.0529	0.1011	0.0659	0.0838	0.0701	0.1317	0.0858
	WBPR	0.0956	0.0749	0.1419	0.0901	0.0618	0.0531	0.1015	0.0662	0.0840	0.0704	0.1312	0.0850
	T-CE	0.0990	0.0779	0.1499	0.0939	0.0740	0.0623	0.1216	0.0779	0.0877	0.0731	0.1376	0.0893
	BOD	0.1273	0.0996	0.1792	0.1162	0.0843	0.0714	0.1355	0.0882	0.0955	0.0791	0.1484	0.0966
	SGL	0.1091	0.0872	0.1588	0.1030	0.0762	0.0648	0.1235	0.0803	0.0890	0.0743	0.1378	0.0903
	SimGCL	0.1172	0.0940	0.1681	0.1102	0.0785	0.0669	0.1265	0.0827	0.0887	0.0738	0.1373	0.0899
	XSimGCL	0.1153	0.0931	0.1637	0.1084	0.0769	0.0663	0.1277	0.0830	0.0884	0.0736	0.1385	0.0903
	RLMRec	0.1034	0.0788	0.1601	0.0960	0.0794	0.0652	0.1275	0.0815	0.0926	0.0746	0.1452	0.0924
	LLaRD	0.1408	0.1126	0.2028	0.1326	0.0975	0.0809	0.1574	0.1008	0.1054	0.0868	0.1631	0.1059
	LLMHNI	**0.1423**	**0.1168**	**0.2040**	**0.1369**	**0.0981**	**0.0837**	**0.1594**	**0.1047**	**0.1065**	**0.0893**	**0.1646**	**0.1087**

Table 1: Performance comparison of backbone recommenders trained with different denoising approaches. R and N refer to Recall and NDCG, respectively. The highest scores are in **bold**, and the runner-ups are with underline. All results are statistically significant according to the t-tests with a significance level of $p < 0.01$.

LLMHNI consistently outperforms other baseline models across all noise levels. While performance drops as noise levels rise, the rate at which LLMHNI's performance declines remains relatively stable compared to other baselines, demonstrating that LLMHNI is the least impacted by noise. This indicates that LLMHNI effectively identifies noisy and hard samples, even in the presence of significant noise.

In-depth Analysis of LLMHNI (RQ3 - RQ5)

Ablation Study (RQ3). To assess the impact of each component within LLMHNI, we conducted ablation studies with four variants. Here, SR represents components associated with semantic relevance, and LR pertains to logical relevance: **(1) w/o $\mathbf{SR_{lmns}}$**: Replaces LLM-embedding guided hard negative sampling (Equation 10) with uniform sampling. **(2) w/o $\mathbf{SR_{al}}$**: Excludes objective alignment strategy applied to LLM-encoded embeddings (Equation 7). **(3) w/o $\mathbf{LR_{hal}}$**: Removes the graph contrastive loss $\mathcal{L}_{\text{hal}}$ aimed at mitigating unreliable interactions. **(4) w/o $\mathbf{LR_{de}}$**: Removes the graph contrastive loss $\mathcal{L}_{\text{de}}$ for cross-graph user-item alignment. Table 2 shows varying performance degradation when specific modules are removed. The drop in performance for **w/o $\mathbf{TR_{lmns}}$** underscores the crucial role of auxiliary semantic relevance in distinguishing hard and false negatives. Similarly, the reduction in performance for **w/o $\mathbf{TR_{al}}$** illustrates the significance of objective-aligned LLM-encoded embeddings in selecting hard negative items. Furthermore, performance declines in **w/o $\mathbf{LR_{de}}$** highlight the effectiveness of logical relevance in interaction denoising. While the performance drop in **w/o $\mathbf{LR_{hal}}$** demonstrates the importance of eliminating hallucination-induced interaction graph edges.

Variants	w/o TR_{lmns}	w/o TR_{ssl}	w/o LR_{nie}	w/o LR_{uis}	**LLMHNI**
R@10	0.1199	0.1248	0.1125	0.1294	0.1423
R@20	0.1799	0.1848	0.1772	0.1854	0.2040
N@10	0.0937	0.0982	0.0855	0.1047	0.1168
N@20	0.1112	0.1174	0.1060	0.1230	0.1369

Table 2: The effect of components in LLMHNI with the LightGCN on Amazon-books datasets.

Hyperparameters Analysis (RQ4). To assess LLMHNI's sensitivity to hyperparameter changes, we varied the hyperparameters λ_1, λ_2, τ_{de}, and τ_{hal} within the range of $[0.1, 0.3, 0.5, 0.7, 1.0]$. Due to space constraints, we present only the results from the Amazon-books dataset in Figure 5 as results from other datasets show similar patterns. Our analysis reveals that modifications in temperature parameters τ_{hal} and τ_{de} lead to significant performance variations. The performance of both τ_{hal} and τ_{de} shows an upward trend first and then drops steadily. This highlights the importance of choosing a suitable temperature in contrastive learning. Furthermore, alterations in λ_1 and λ_2 have minimal impact on performance, demonstrating the stability of

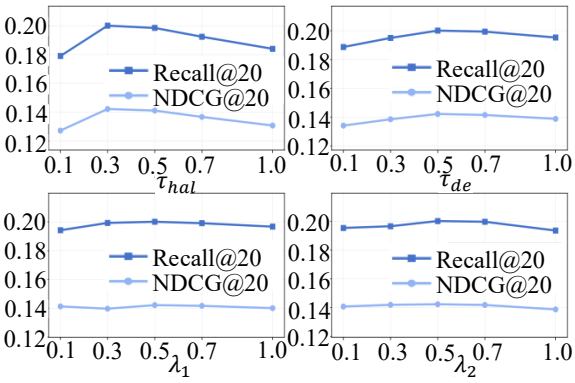


Figure 5: Hyper-parameter analysis of LLMHNI with LightGCN backbone on the Amazon-books datasets.

Method	Amazon-Book	Yelp	Steam
Normal	1.0307	1.3568	4.7756
T-CE	1.2369	1.4639	5.0580
XSimGCL	1.5066	1.8678	7.0733
SimGCL	3.0946	4.0361	13.7971
SGL	3.5637	4.4746	14.7370
BOD	5.7516	6.4349	19.5580
LLMHNI	6.9677	8.7488	26.6929

Table 3: Comparison of training time in seconds per epoch across different datasets and baseline denoise methods.

LLMHNI with these two hyperparameters.

Training Efficiency Analysis (RQ5). While LLMHNI includes multiple components, its overall time complexity remains comparable to mainstream denoising methods. The relevance signal generation is conducted before training the recommender system; thus is excluded from this analysis. The calculation of $\mathcal{L}_{\text{hal}}$ introduce an additional complexity of $O(|\mathcal{E}|d_{rec}(2+|\mathcal{V}|))$, where $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$. While $\mathcal{L}_{\text{de}}$ takes in-batch negatives, resulting in $O\left(|\mathcal{E}|d_{rec}(2+2\mathcal{B})\right)$ time cost. In addition to the inference on interaction graph G, our method introduces additional inference on G'_{aug} and G_{aug}, resulting in addition complexity of $O(|\mathcal{E}'_{aug}|d_{rec}L)$, and $O(|\mathcal{E}_{aug}|d_{rec}L)$, respectively. The L denotes the number of layers in the backbone recommenders. We also provide a running time comparison with the baselines on the LightGCN backbone. Table 3 presents the results obtained on a server with two Intel(R) Xeon(R) Gold 5118 CPUs (12 cores each) and an NVIDIA GeForce RTX 3090 GPU.

Related Works

Denoise Recommendation

Recommenders are pointed out to be affected by users' unconscious behaviors (Wang et al. 2021b), leading to noisy data. As a result, many efforts are designed to alleviate the problem. These approaches can be categorized into two paradigms: sample dropping (Gantner et al. 2012; Lin et al. 2023) and sample re-weighting (Wang et al. 2023; Gao et al. 2022). Sample dropping methods aim to keep clean samples and discard noisy ones. For instance, T-CE (Wang et al. 2021a) observes that noisy samples exhibit high loss values and removes them during training. DCF (He et al. 2024) introduces a double correction method that drops samples based on loss values and prediction score variances. Sample re-weighting methods try to mitigate the impact of noisy samples by assigning lower weights to them. Typically, R-CE (Wang et al. 2021a) assigns lower weights to noisy samples according to the prediction score. BOD (Wang et al. 2023) considers weight assignment as a bi-level optimization problem. Despite their promising results, they rely on data patterns to recognize noisy samples (e.g., loss values and prediction scores), resulting in the hard-noisy sample confusion issue.

LLMs for Recommendation

LLMs are effective tools for NLP tasks and have gained significant attention in the domain of Recommendation Systems (RS). For the adaption of LLMs in recommendations, existing works can be divided into three categories (Wu et al. 2024): LLM as RS, LLM Embedding for RS, and LLM token for RS. The LLM as RS aims to transform LLMs into effective recommendation systems (Chao et al. 2024), such as LC-Rec (Zheng et al. 2024a) and LLM-TRSR (Zheng et al. 2024b). In contrast, the LLM embedding for RS and LLM token for RS views the language model as an enhancer. The former typically adopts embeddings related to users and items, incorporating semantic information in the recommender (Ren et al. 2024). While the latter generates text tokens to capture potential preferences between user and items (Wei et al. 2024; Xi et al. 2023). Recent studies also leverage LLMs in recommender system denoise, where RLMRec (Ren et al. 2024) and DALR (Peng et al. 2025) implicitly eliminate noise at the representation-level. The LLaRD (Wang et al. 2025) takes LLMs to generate preference knowledge and relationship knowledge to denoise. However, none of them discuss the potential of LLMs in supporting the identification of hard and noisy samples.

Conclusion

In this work, we investigate the potential of Large Language Models in solving the hard and noisy sample confusion in recommender systems. We discvoered that LLMs can offer valuable auxiliary signals for addressing hard-noisy sample confusion, including the user-item semantic relevance from LLM-encoded embeddings and the user-item logical relevance from LLM-inferred interactions. To take advantage of these two signals, we introduce the **L**arge **L**anguage **M**odel Enhanced **H**ard-**N**oisy Sample **I**dentification framework (**LLMHNI**). LLMHNI generates both relevance signals, leveraging them to resolving hard-noisy confusion issues in both hard negative sampling and interaction denosing. More importantly, LLMHNI enhances the utilization of these two signals in recommender systems by effectively addressing the objective mismatch of LLM-encoded embeddings and hallucinations in LLM-inferred interactions. Experiments on three real-world datasets and two backbone recommenders confirm the efficacy of our approach.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62572417, No.92370204), National Key R&D Program of China (Grant No.2023YFF0725004), the Guangzhou Basic and Applied Basic Research Program under Grant No. 2024A04J3279, and Education Bureau of Guangzhou Municipality.

References

Chao, W.; Zheng, Z.; Zhu, H.; and Liu, H. 2024. Make Large Language Model a Better Ranker. arXiv:2403.19181.

Chen, C.; Zheng, S.; Chen, X.; Dong, E.; Liu, X.; Liu, H.; and Dou, D. 2021. Generalized Data Weighting via Class-level Gradient Manipulation. In *Neural Information Processing Systems*.

Ding, J.; Quan, Y.; Yao, Q.; Li, Y.; and Jin, D. 2020. Simplify and robustify negative sampling for implicit collaborative filtering. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

Gantner, Z.; Drumond, L.; Freudenthaler, C.; and Schmidt-Thieme, L. 2012. Personalized ranking for non-uniformly sampled items. In *Proceedings of KDD Cup 2011*, 231–247. PMLR.

Gao, Y.; Du, Y.; Hu, Y.; Chen, L.; Zhu, X.; Fang, Z.; and Zheng, B. 2022. Self-guided learning to denoise for robust recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1412–1422.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 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*, SIGIR '20, 639–648. New York, NY, USA: Association for Computing Machinery. ISBN 9781450380164.

He, Z.; Wang, Y.; Yang, Y.; Sun, P.; Wu, L.; Bai, H.; Gong, J.; Hong, R.; and Zhang, M. 2024. Double Correction Framework for Denoising Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Lin, J.; Dai, X.; Xi, Y.; Liu, W.; Chen, B.; Zhang, H.; Liu, Y.; Wu, C.; Li, X.; Zhu, C.; et al. 2025. How can recommender systems benefit from large language models: A survey. *ACM Transactions on Information Systems*, 43(2): 1–47.

Lin, W.; Zhao, X.; Wang, Y.; Zhu, Y.; and Wang, W. 2023. Autodenoise: Automatic data instance denoising for recommendations. In *Proceedings of the ACM Web Conference 2023*, 1003–1011.

Luo, H.; Zhou, J.; Bao, Z.; Li, S.; Culpepper, J. S.; Ying, H.; Liu, H.; and Xiong, H. 2020. Spatial Object Recommendation with Hints: When Spatial Granularity Matters. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Peng, Y.; Gao, C.; Zhang, Y.; Dan, T.; Du, X.; Luo, H.; Li, Y.; and Meng, X. 2025. Denoising alignment with large language model for recommendation. *ACM Transactions on Information Systems*, 43(2): 1–35.

Ren, X.; Wei, W.; Xia, L.; Su, L.; Cheng, S.; Wang, J.; Yin, D.; and Huang, C. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM on Web Conference 2024*, 3464–3475.

Shi, W.; Chen, J.; Feng, F.; Zhang, J.; Wu, J.; Gao, C.; and He, X. 2023. On the theories behind hard negative sampling for recommendation. In *Proceedings of the ACM Web Conference 2023*, 812–822.

Wang, S.; Zheng, Z.; Sui, Y.; and Xiong, H. 2025. Unleashing the Power of Large Language Model for Denoising Recommendation. In *Proceedings of the ACM on Web Conference 2025*, 252–263.

Wang, W.; Feng, F.; He, X.; Nie, L.; and Chua, T.-S. 2021a. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, 373–381.

Wang, W.; Feng, F.; He, X.; Zhang, H.; and Chua, T.-S. 2021b. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1288–1297.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 165–174.

Wang, Z.; Gao, M.; Li, W.; Yu, J.; Guo, L.; and Yin, H. 2023. Efficient bi-level optimization for recommendation denoising. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2502–2511.

Wang, Z.; Xu, Q.; Yang, Z.; Cao, X.; and Huang, Q. 2021c. Implicit feedbacks are not always favorable: Iterative relabeled one-class collaborative filtering against noisy interactions. In *Proceedings of the 29th ACM International Conference on Multimedia*, 3070–3078.

Wei, W.; Ren, X.; Tang, J.; Wang, Q.; Su, L.; Cheng, S.; Wang, J.; Yin, D.; and Huang, C. 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.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 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.

Wu, L.; Zheng, Z.; Qiu, Z.; Wang, H.; Gu, H.; Shen, T.; Qin, C.; Zhu, C.; Zhu, H.; Liu, Q.; et al. 2024. A Survey on Large Language Models for Recommendation. arXiv:2305.19860.

Xi, Y.; Liu, W.; Lin, J.; Cai, X.; Zhu, H.; Zhu, J.; Chen, B.; Tang, R.; Zhang, W.; Zhang, R.; and Yu, Y. 2023. Towards Open-World Recommendation with Knowledge Augmentation from Large Language Models. arXiv:2306.10933.

Yu, J.; Xia, X.; Chen, T.; Cui, L.; Hung, N. Q. V.; and Yin, H. 2023. XSimGCL: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 36(2): 913–926.

Yu, J.; Yin, H.; Xia, X.; Chen, T.; Cui, L.; and Nguyen, Q. V. H. 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.

Zheng, B.; Hou, Y.; Lu, H.; Chen, Y.; Zhao, W. X.; Chen, M.; and Wen, J.-R. 2024a. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 1435–1448. IEEE.

Zheng, Z.; Chao, W.; Qiu, Z.; Zhu, H.; and Xiong, H. 2024b. Harnessing large language models for text-rich sequential recommendation. In *Proceedings of the ACM on Web Conference 2024*, 3207–3216.