LAGCL4Rec: When LLMs Activate Interactions Potential in Graph Contrastive Learning for Recommendation

Leqi Zheng [1], Chaokun Wang* [1], Canzhi Chen [2], Jiajun Zhang [3], Cheng Wu [1], Zixin Song [1], Shanan Yan [1], Ziyang Liu [1], Hongwei Li [1]

[1]Tsinghua University

[2]Beijing Institute of Technology

[3]University of Science and Technology of China

{zhenglq24, wuc22, songzx24, ysn24, liu-zy21, lhw23}@mails.tsinghua.edu.cn

chencanzhi@bit.edu.cn, zhangjiajun519@mail.ustc.edu.cn

*** Correspondence:** chaokun@tsinghua.edu.cn

Abstract

A core barrier preventing recommender systems from reaching their full potential lies in the inherent limitations of user-item interaction data: (1) *Sparse user-item interactions*, making it difficult to learn reliable user preferences; (2) Traditional contrastive learning methods often *treat negative samples as equally hard or easy*, ignoring the informative semantic difficulty during training. (3) Modern LLM-based recommender systems, on the other hand, *discard all negative feedback*, leading to unbalanced preference modeling. To address these issues, we propose **LAGCL4Rec**, a framework leveraging **L**arge Language Models to **A**ctivate interactions in **G**raph **C**ontrastive **L**earning **for** **Rec**ommendation. Our approach operates through three stages: (i) Data-Level: augmenting sparse interactions with balanced positive and negative samples using LLM-enriched profiles; (ii) Rank-Level: assessing *semantic difficulty* of negative samples through LLM-based grouping for fine-grained contrastive learning; and (iii) Rerank-Level: reasoning over augmented historical interactions for personalized recommendations. Theoretical analysis proves that **LAGCL4Rec** achieves effective information utilization with minimal computational overhead. Experiments across multiple benchmarks confirm our method consistently outperforms state-of-the-art baselines.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in Natural Language Processing (NLP). The success of LLMs in NLP has encouraged researchers to explore their potential in recommendation tasks (Gao et al., 2023; Ren et al., 2023). Current LLM-based recommendation approaches generally fall into three categories: (a) Semantic Augmentation using LLMs: using LLMs to enhance textual descriptions of users and items to provide contextual information (Sun et al., 2025; Liu et al., 2024a); (b) Feature Extractions using LLMs: using LLMs as feature extractors to generate item and user embeddings (Hu et al., 2024; Liao et al., 2023) for downstream models; and (c) Decision Making using LLMs: directly produce recommendation lists through prompting or fine-tuning on foundation models (Li et al., 2023a; Bao et al., 2023).

Despite these advances, existing approaches fail to fully unleash LLMs' potential for recommendation due to three critical limitations: (1) Sparse User-Item Interactions: While (a) approaches augment positive interactions, they neglect the latent potential in negative interactions, leading to an imbalanced activation of interaction data (Wang et al., 2024a; Liu et al., 2024a). (2) Ignorance of Difficulty of Negative Samples in Contrastive Learning: (b) approaches face significant efficiency bottlenecks, while traditional Graph Contrastive Learning (GCL) methods fail to activate the semantic potential in interactions by treating all negative samples uniformly (Yu et al., 2023; Lin et al., 2022). (3) Neglect of Informative Negative Feedback in LLM Prompting: By discarding negative interactions, Current (c) methods fail to harness the complete spectrum of user preference signals (Yuan et al., 2023; Bao et al., 2023). Meanwhile, most of these research using prompting on general foundation models, ignoring the domain-specific nature of recommendation.

This leads us to a question: *How can we systematically unlock and activate the full potential of both positive and negative interactions throughout the entire recommendation pipeline to maximize preference modeling capabilities and recommendation performance?*

As an answer to this question, we introduce LAGCL4Rec, a novel approach that leverages **L**LMs to **A**ctivate interaction potential in **G**raph **C**ontrastive **L**earning through a progressive pipeline operating at each level of the recommen-

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 1163–1184
November 4-9, 2025

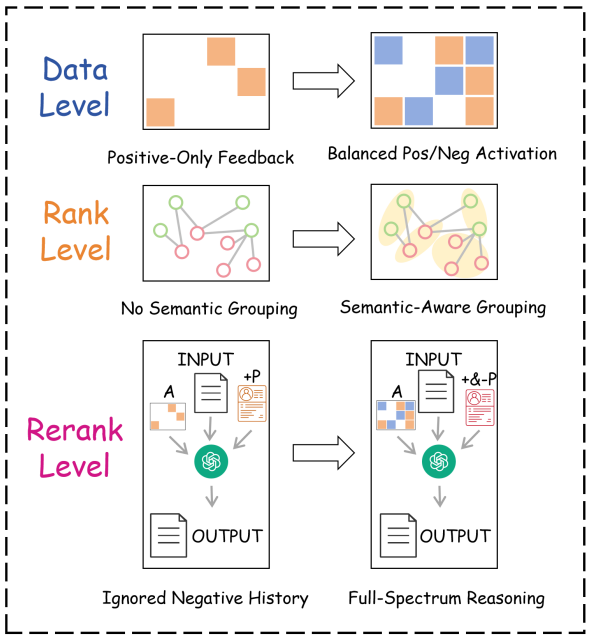


Figure 1: Comparison between conventional methods (left) and LAGCL4Rec (right). Data Level: Traditional methods rely on positive feedback only, while our approach activates potential in both positive and negative interactions. Rank Level: Conventional methods treat all negatives uniformly, whereas LAGCL4Rec implements semantic-aware grouping to differentiate hard/easy negatives, unlocking their potential for contrastive learning. Rerank Level: Existing approaches ignore negative interaction potential, while our method activates both positive and negative profiles through structured reasoning, enabling full-spectrum personalization.

dation process. Figure 1 contrasts our proposed solutions with existing research at each level. Our contributions include:

1. **Data-Level Activation**: We leverage LLMs to create semantically rich user and item profiles. Our approach activates both positive preferences and previously untapped negative interaction signals, transforming sparse data into a balanced representation of the complete user preference spectrum. (Section 2.1).

2. **Rank-Level Activation**: We propose a semantic grouping strategy to group negative samples based on their semantic relationships, which enables assessing their semantic difficulty. We then apply fine-grained contrastive learning to selectively emphasize informative negatives. By leveraging semantic grouping and efficient preprocessing instead of directly modeling negative interactions, our method improves computational efficiency while maintaining discriminative power (Section 2.2).

3. **Rerank-Level Activation**: We design a reranking approach that activates the full potential of historical interaction data by simultaneously leveraging both positive and negative signals. By employing LLMs to reason over these comprehensively activated interactions, we enable personalized and explainable recommendations that fully capitalize on all available preference information. (Section 2.3).

4. **Theoretical and Empirical Validation**: We provide rigorous theoretical analysis establishing how our pipeline successfully unlocks the potential in interaction data, along with extensive experiments across diverse domains demonstrating significant performance improvements. (Section 3 and Section 4).

2 Methodology

We introduce LAGCL4Rec, a novel framework that leverages LLMs to Activate Graph Contrastive Learning through a progressive activation pipeline operating at multiple levels of the recommendation process, as illustrated in Figure 2. Our approach systematically unleashes interaction potential via three complementary mechanisms: (1) Data-Level Activation (Section 2.1), (2) Rank-Level Activation (Section 2.2) and (3) Rerank-Level Activation (Section 2.3).

2.1 Data-Level Activation

As illustrated in Figure 2 (**Data-Level**), the data-level augmentation process involves (1) generating profiles for user and item to enhance semantic context for downstream recommendation, and (2) augmenting the interaction sequence to create a balanced representation with both positive and negative interactions.

User and Item Profile Generation. We generate semantically rich profiles for users and items using LLM:

$$\mathcal{P}_u = LLMs(\mathcal{S}_u, \mathcal{Q}_u), \quad \mathcal{P}_v = LLMs(\mathcal{S}_v, \mathcal{Q}_v) \tag{1}$$

where $\mathcal{P}_u$ and $\mathcal{P}_v$ are generated user and item profiles, $\mathcal{S}_u$ and $\mathcal{S}_v$ are system prompts (detailed in Appendix A.1.1 and A.1.2), and $\mathcal{Q}_u$ and $\mathcal{Q}_v$ are query inputs containing user/item information.

Positive Interaction Augmentation. For cold-start users with less than 30 interactions, we define candidate items $\mathcal{V}_u^{cand}$ (items with no interactions

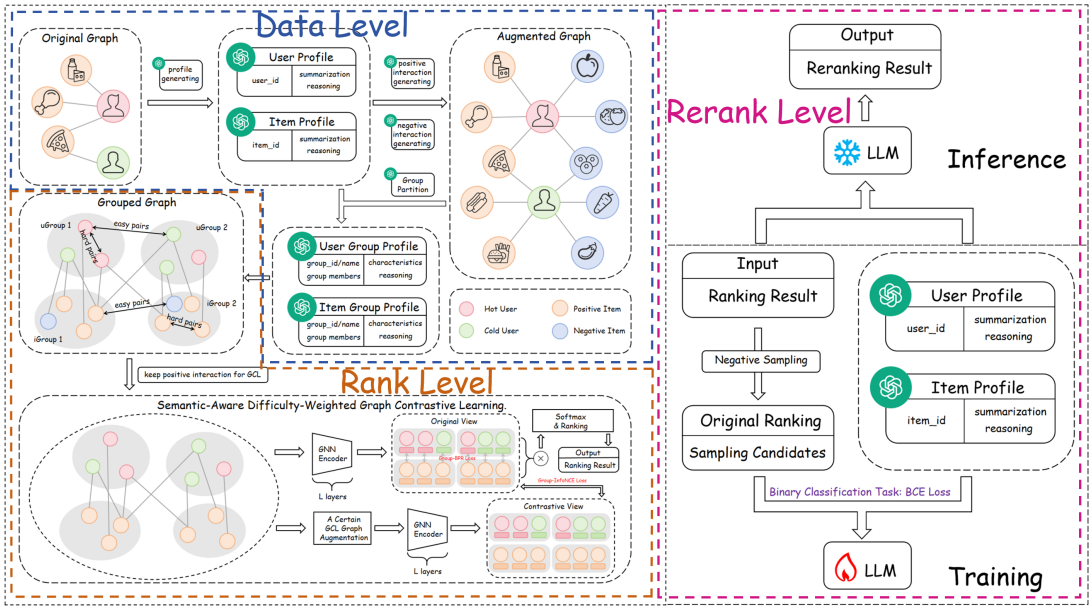


Figure 2: LAGCL4Rec's Progressive Activation Pipeline: Data-Level generates semantically-rich profiles to activate potential in user-item interactions. Rank-Level implements semantic-aware grouping for differentiating hard/easy negatives with adaptive contrastive learning. Rerank-Level leverages both positive and negative historical interactions through structured reasoning for refined recommendations.

for user u in dataset) and sample 1024 items as $\mathcal{V}_u^{sample}$. We score each item from $\mathcal{V}_u^{sample}$ using:

$$s_{uv}^{pos} = LLMs(\mathcal{P}_u, \mathcal{P}_v, \mathcal{S}_{pos}) \tag{2}$$

where s_{uv}^{pos} is the preference score and $\mathcal{S}_{pos}$ is the prompt template detailed in Appendix A.1.3. We select the top 2% of items to form positive interactions $\mathcal{R}_u^{pos}$ (likely positive preferences).

Negative Interaction Augmentation. We augment negative interactions by similarly constructing $\mathcal{V}_u^{neg-cand}$ and scoring items from candidate set:

$$s_{uv}^{neg} = LLMs(\mathcal{P}_u, \mathcal{P}_v, \mathcal{S}_{neg}) \tag{3}$$

where s_{uv}^{neg} is the dislike score and $\mathcal{S}_{neg}$ is the prompt template in Appendix A.1.4. We select the top 2% as negative interactions $\mathcal{R}_u^{neg}$ (likely negative preferences). This creates truly representative negative interactions rather than random items.

The final enhanced set combines both kinds of preference signals:

$$\mathcal{R}_u^{enhanced} = \mathcal{R}_u^{pos} \cup \mathcal{R}_u^{neg} \tag{4}$$

2.2 Rank-Level Activation

Unlike traditional sign-aware methods that directly model negative interactions with high computational overhead, we propose an efficient approach that indirectly leverages these signals through preprocessing and semantic grouping. This strategy significantly reduces computational complexity while maintaining recommendation effectiveness. Our key insight is that inferring dislikes within semantic categories (such as an orange lover disliking bananas) is harder and more informative than inferring dislikes for random objects. Therefore, we propose a contrastive learning approach based on *semantic difficulty*, as illustrated in Figure 2 (Rank-Level).

LLM-Guided Semantic Grouping. We use LLMs to identify semantically similar users and items:

$$\mathcal{G}_i^{user} = LLMs(\mathcal{P}_{\mathcal{U}_i}, \mathcal{S}_{group}) \tag{5}$$

where $\mathcal{G}_i^{user}$ represents semantic groups of users, $\mathcal{P}_{\mathcal{U}_i}$ is the set of user profiles, and $\mathcal{S}_{group}$ is the grouping prompt (Appendix A.2.1). We compute similarity between groups:

$$sim(G_a, G_b) = LLMs(G_a, G_b, \mathcal{S}_{sim}) \tag{6}$$

where $sim(G_a, G_b)$ quantifies group similarity and $\mathcal{S}_{sim}$ is detailed in Appendix A.2.2. Groups with similarity above threshold τ_{sim} are merged to form collections $\mathcal{G}^{user}$ and $\mathcal{G}^{item}$.

Semantic Difficulty Assessment. Hard negatives ($\mathcal{N}_h$) are negative samples belonging to the same semantic group as the positive item i, defined as

$$\mathcal{N}_h(u,i) = \{j \mid j \in \mathcal{V}, (u,j) \notin \mathcal{R}^{pos}, g(j) = g(i)\} \tag{7}$$

Easy negatives ($\mathcal{N}_e$) are negative samples from other groups, defined as

$$\mathcal{N}_e(u,i) = \{j \mid j \in \mathcal{V}, (u,j) \notin \mathcal{R}^{pos}, g(j) \neq g(i)\} \tag{8}$$

Here, $\mathcal{V}$ is the item set, $\mathcal{R}^{pos}$ denotes positive interactions, and $g(j)$ maps item j to its group. Similar definitions apply for item-user pairs.

Difficulty-Aware Contrastive Learning. To enable detailed and differentiated supervision using negative samples of varying difficulty levels, we propose two difficulty-aware loss functions.

For preference modeling, we propose LABPR loss, a difficulty-weighted version of BPR loss:

$$\begin{aligned}\mathcal{L}_{labpr} = &-\sum_{(u,i,j_e)\in D_e} \ln \sigma(r_{ui} - r_{uj_e}) \\ &-\sum_{(u,i,j_h)\in D_h} w_h \cdot \ln \sigma(r_{ui} - r_{uj_h})\end{aligned} \tag{9}$$

where D_e and D_h contain easy and hard negative triplets, r_{ui} is predicted preference, and $w_h > 1$ amplifies the emphasis on hard negatives.

For clearer and more informative signals, we propose an adaptive-temperature InfoNCE loss for difficulty-aware contrastive learning:

$$\mathcal{L}_{lainfo} = -\sum_{i=1}^{N} \log \frac{\exp(sim(v_i, v_i^+)/\tau_p)}{Z_i} \tag{10}$$

where $sim(v_i, v_j)$ is cosine similarity and the denominator Z_i is:

$$\begin{aligned}Z_i = \quad &\exp(sim(v_i, v_i^+)/\tau_p) + \\ &\sum_{j\neq i, g_j \neq g_i} \exp(sim(v_i, v_j)/\tau_e) + \\ &\sum_{j\neq i, g_j = g_i} w_h \cdot \exp(sim(v_i, v_j)/\tau_h)\end{aligned} \tag{11}$$

We set $\tau_h < \tau_e \leq \tau_p$ to enhance hard negatives, where τ_p, τ_e, and τ_h are temperature parameters for positive, easy negative, and hard negative pairs. The total loss is computed as:

$$\mathcal{L}_{total} = \mathcal{L}_{labpr} + \lambda_{cl} \cdot (\mathcal{L}^{user}_{lainfo} + \mathcal{L}^{item}_{lainfo}) + \lambda_{reg} \cdot \mathcal{L}_{reg} \tag{12}$$

where λ_{cl} controls contrastive learning strength, λ_{reg} is regularization weight, and $\mathcal{L}_{reg} = \sum_{\theta\in\Theta} ||\theta||_2^2$ is an L2 regularization term.

2.3 Rerank-Level Activation

To unleash the full potential of LLM in recommendation, we propose an LLM-based reasoning model as reranker, The reranker reason over the augmented historic interactions with positive and negative samples, and is optimized using a task-specific binary classification loss. as illustrated in Figure 2 (Rerank-Level).

Historical Interaction Integration. For each u, we assemble an input sequence reflecting their historical interactions:

$$\mathcal{I}_u = \{\mathcal{P}_u, \mathcal{C}_u, \mathcal{R}_u^{pos-hist}, \mathcal{R}_u^{neg-hist}\} \tag{13}$$

where $\mathcal{P}_u$ denotes the comprehensive user profile generated in the data augmentation stage, $\mathcal{C}_u$ represents the set of candidate items, $\mathcal{R}_u^{pos-hist}$ contains the user's historical positive interactions, $\mathcal{R}_u^{neg-hist}$ contains historical negative interactions.

Training Phase: Binary Classification. To effectively leverage both positive and negative historical interactions, we formulate reranking as a binary classification for individual item relevance:

$$p_{uv} = f_{\theta_{rer}}(\mathcal{P}_u, v, \mathcal{R}_u^{pos-hist}, \mathcal{R}_u^{neg-hist}, \mathcal{S}_{cot}) \tag{14}$$

where $f_{\theta_{rer}}$ represents our reranker parameterized by θ_{rer} (auto-regressive language model with binary classification head), v is a candidate item, and $\mathcal{S}_{cot}$ is the Chain-of-Thought prompt template to elicit reasoning in reranking (detailed in Appendix A.3).

We train the model using binary cross-entropy loss on a dataset of user-item pairs:

$$\begin{aligned}\mathcal{D}_{rer} = \{(u, v, y_{uv}) \mid &u \in \mathcal{U}_{\text{train}}, v \in \mathcal{C}_u, \\ &y_{uv} = \mathbb{1}[(u,v) \in \mathcal{R}_{train}]\},\end{aligned} \tag{15}$$

where $\mathcal{D}_{rer}$ is the training dataset, and y_{uv} is a binary indicator of whether the interaction (u, v) appears in the train set $\mathcal{R}_{train}$. The optimization objective is:

$$\begin{aligned}\mathcal{L}_{bce} = \frac{1}{|\mathcal{D}_{rer}|} \sum_{(u,v,y_{uv})\in\mathcal{D}_{rer}} &[-y_{uv} \log(p_{uv}) \\ &- (1 - y_{uv}) \log(1 - p_{uv})]\end{aligned} \tag{16}$$

$$\theta^*_{rer} = \arg\min_{\theta_{rer}} \mathcal{L}_{bce}(\theta_{rer}; \mathcal{D}_{rer}) \tag{17}$$

Inference Phase: List-wise Reranking. During inference, we apply the reranking to generate a personalized recommendation result $\mathcal{L}_u^{rerank}$ for each user. This process activates historical interaction signals through a structured reasoning process:

$$\mathcal{L}_u^{rerank} = g_{\theta_{rer}^*}(\mathcal{I}_u, \mathcal{S}_{cot}) \tag{18}$$

where $g_{\theta_{rer}^*}$ represents the application of our trained model to produce a reordered list of the candidate items.

Score Fusion Strategy. We combine the original collaborative filtering scores with the model-generated relevance scores:

$$s_{uj}^{final} = \alpha \cdot s_{uj}^{orig} + (1-\alpha) \cdot s_{uj}^{llm} \tag{19}$$

where s_{uj}^{orig} is the original score from the rank-level collaborative filtering model, s_{uj}^{llm} is the normalized score derived from the trained model, and $\alpha \in [0,1]$ is a weighting parameter that balances these two signals. The model score is normalized using softmax:

$$s_{uj}^{llm} = \frac{\exp(\tilde{s}_{uj}^{llm}/\tau)}{\sum_{v_l \in \mathcal{C}_u} \exp(\tilde{s}_{ul}^{llm}/\tau)} \tag{20}$$

where $\tilde{s}_{uj}^{llm}$ is the raw relevance score assigned by the model and τ is a temperature parameter controlling the sharpness of the distribution. The final ranked list is obtained by:

$$\mathcal{L}_u = \text{argsort}_{v_j \in \mathcal{C}_u}(s_{uj}^{final}) \tag{21}$$

where $\mathcal{L}_u$ represents the ordered list of items recommended to user u.

3 Theoretical Analysis

This section provides a theoretical analysis of our pipeline, focusing on the information gain guaranteed by hard negative samples.

3.1 Analysis of Rank-Level Activation

The separate treatment of easy and hard negative interactions based on semantic grouping enhances the effectiveness of ranker training. Proposition below theoretically demonstrates this point.

Proposition 1 (Hard-negative gradient dominance). Under Eqs. (9) and (10)–(11) with $w_h > 1$ and $\tau_h < \tau_e \leq \tau_p$, the absolute gradient w.r.t. a hard negative similarity exceeds that of an easy negative whenever $sim(v_i, v_{j_h}) > 0$. Consequently, hard negatives tighten the InfoNCE mutual-information lower bound faster than easy negatives.

This justifies our higher weight $w_h > 1$ for hard negative interactions. Our analysis in Appendix B.2 shows that under the LABPR loss, hard negative gradients are amplified by a factor of w_h, ensuring faster convergence for challenging cases. Similarly, with our adaptive temperature mechanism (Appendix B.3), the model converges to a more discriminative embedding space.

3.2 Analysis of Rerank-Level Activation

For the reranking component, we establish:

Proposition 2 (CoT features do not decrease task information). Let Y be the relevance label, X the base input, and $Z = \text{CoT}(X; \mathcal{S}_{cot})$. Then

$$I(Y; X, Z) = I(Y; X) + I(Y; Z \mid X) \geq I(Y; X), \tag{22}$$

with strict inequality if $I(Y; Z \mid X) > 0$.

In Appendix B.4, we prove there exists an optimal weighting parameter $\alpha^* \in [0,1]$ that minimizes expected ranking error by balancing collaborative filtering and LLM signals. Furthermore, our reranking model trained with BCE loss converges to a stable solution when the candidate set distribution remains consistent between training and inference (Appendix B.5).

4 Experiments

4.1 Experimental Settings

4.1.1 Datasets

We evaluate LAGCL4Rec on many public recommendation datasets: ML-1M, Amazon-book, Yelp, and Steam. Following standard practices (Seo et al., 2022; Wang et al., 2024b; Chen et al., 2024), we consider ratings more than 3 as positive interactions and ratings no more than 3 as negative interactions for datasets, while all interactions in Steam are treated as positive due to absence of explicit ratings. We apply 5-core filtering and divide each dataset into training, validation, and testing sets in a 7:1:2 ratio. Table 1 summarizes the dataset statistics. We adopt the all-rank protocol for comprehensive and unbiased evaluation. Two widely used ranking metrics are employed: Recall@N and NDCG@N, where N=20 for main results.

4.1.2 Baseline Models

We compare LAGCL4Rec with representative methods from four categories:

(1) CF Models: MF (Koren et al., 2009), NCF (He et al., 2017), NGCF (Wang et al., 2019),

Table 1: Statistics of the experimental datasets.

Dataset	#Users	#Items	#Interactions	Density
ML-1M	6,040	3,952	1,000,209	$4.2e^{-2}$
Amazon-book	11,000	9,332	120,464	$1.2e^{-3}$
Yelp	11,091	11,010	166,620	$1.4e^{-3}$
Steam	23,310	5,237	316,190	$2.6e^{-3}$

LightGCN (He et al., 2020), and SelfGNN (Liu et al., 2024b), representing classic and advanced collaborative filtering approaches.

(2) Sign-aware Models: SGFormer (Wu et al., 2024b), SIGFormer (Chen et al., 2024), and NFARec (Wang et al., 2024b), state-of-the-art methods that specifically model signed interactions.

(3) LLM-based Models: RLMRec (Ren et al., 2023), LLM4Rerank (Gao et al., 2025), and RankGPT (Sun et al., 2023), recent approaches leveraging large language models.

(4) GCL-based Models: SGL (Wu et al., 2021), LightGCL (Cai et al.), and XSimGCL (Yu et al., 2023).

4.1.3 Implementation Details

We implement all models with embedding dimension 64. For all experiments, we use Claude 3.7 API. For rerank-level model training, we fine-tune Qwen-2.5 7B Instruct to perform the reranking task. Experiments were conducted on 8 NVIDIA A800 GPUs (80GB memory each).

4.2 Overall Performance

Table 2 compares different recommendation models across four datasets. Our LAGCL4Rec consistently enhances all graph contrastive learning backbones, with improvements ranging from 6.79% to 17.42% on Recall@20 and 1.06% to 15.43% on NDCG@20. The most substantial gains appear on the Yelp dataset with XSimGCL backbone (17.42% R@20 improvement), suggesting our approach excels at activating sparse signals in diverse categorical datasets.

LAGCL4Rec variants outperform traditional CF methods by large margins and surpass specialized sign-aware and LLM-based approaches in most cases. XSimGCL+LAGCL4Rec achieves the best overall performance. The strong improvements in NDCG@20 for Amazon-book (consistently 15%) indicate enhanced ranking quality for sparse datasets through our systematic activation of dormant interaction signals.

4.3 Runtime Comparison

Time Complexity Analysis The per-epoch computational complexity of LAGCL4Rec with different baseline GCL models is summarized in Table 4. Here, $|E^+|$ denotes the number of positive edges, L the number of graph convolution layers, d the embedding dimension, M the total number of nodes, and B the batch size. Additional parameters include a (augmentation views in SGL) and q (singular values in LightGCL).

Faster Ranker Training As shown in Figure 3, LAGCL4Rec introduces minimal per-epoch overhead (1.61-2.24% increase) while substantially reducing total training time (58.7-63.2% reduction) for all tested architectures (XSimGCL, LightGCL, and SGL). This favorable efficiency profile aligns with our theoretical complexity analysis, which predicted that our rank-level activation would add minimal computational overhead while enabling more efficient signal extraction.

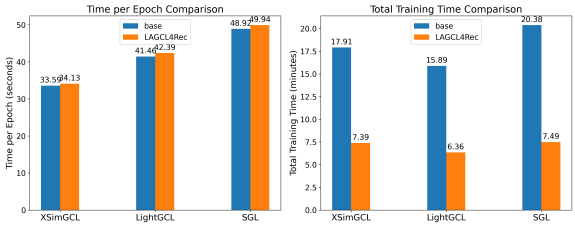


Figure 3: Rank-level runtime comparison between baseline GCL models and LAGCL4Rec variants (excluding rerank-level operations): (a) time per epoch showing minimal computational overhead; (b) total training time demonstrating significant efficiency improvements due to faster convergence.

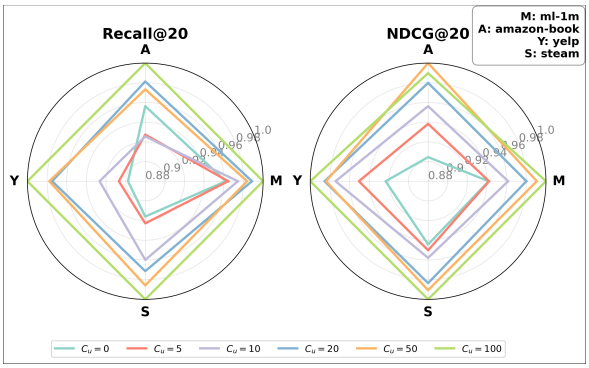


Figure 4: Impact of candidate set size ($\mathcal{C}_u$) on LAGCL4Rec performance across datasets, showing Recall@20 (left) and NDCG@20 (right).

4.4 Ablation Study

Component Analysis. We evaluate each component's contribution in LAGCL4Rec (Table 3). Results show that removing any component leads to performance degradation across all datasets. The

Table 2: Performance comparison of different recommendation models across four datasets. The best results are in **bold**, the strongest baseline performances are underlined, and * indicates statistical significance with $p < 0.01$. R@20 and N@20 represent Recall@20 and NDCG@20 metrics, respectively.

Category	Dataset		ML-1M		Amazon-book		Yelp		Steam	
	Model		R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20
CF	MF		0.1329	0.1988	0.0446	0.0397	0.0451	0.0259	0.0387	0.0213
	NCF		0.1501	0.2102	0.0693	0.0474	0.0637	0.0408	0.0603	0.0448
	NGCF		0.1630	0.2185	0.0809	0.0663	0.0744	0.0395	0.0532	0.0406
	LightGCN		0.1993	0.2632	0.0918	0.0699	0.0769	0.0512	0.0749	0.0632
	SelfGNN		0.2565	0.2810	0.1024	0.0757	0.0795	0.0603	0.0954	0.0739
Sign-RS	SGFormer		0.1877	0.2680	0.1102	0.0703	0.0658	0.0504	0.1265	0.0803
	SIGFormer		0.2995	0.3380	0.1428	0.1045	0.0959	0.0783	0.1448	0.0906
	NFARec		0.2840	0.3212	0.1496	0.1032	0.1143	0.0876	0.1369	0.0884
LLM-RS	RLMRec		0.2966	0.3294	0.1543	0.0976	0.1279	0.0885	0.1467	0.0897
	LLM4Rerank		0.3007	0.3316	0.1496	0.1033	0.1203	0.0892	0.1416	0.0916
	RankGPT		0.2515	0.2801	0.1327	0.0852	0.1154	0.0857	0.1402	0.0893
GCL-RS	Backbone	Variants								
	SGL	base	0.2798	0.3037	0.1438	0.0904	0.1068	0.0847	0.1401	0.0881
		LAGCL4Rec	0.2988	0.3279	0.1553	0.1041	0.1238	0.0856	0.1503	**0.0927***
		improv.	6.79%	7.97%	8.00%	15.15%	15.92%	1.06%	7.28%	5.22%
	LightGCL	base	0.273	0.3035	0.1484	0.0933	0.1092	0.0855	0.1322	0.0863
		LAGCL4Rec	0.3035	0.3301	0.1598	**0.1077***	0.1244	0.0882	0.1466	0.0899
		improv.	11.17%	8.76%	7.68%	15.43%	13.92%	3.16%	10.89%	4.17%
	XSimGCL	base	0.2729	0.3087	0.1477	0.0925	0.1108	0.0849	0.1397	0.0876
		LAGCL4Rec	**0.3106***	**0.3450***	**0.1604***	0.1065	**0.1301***	**0.0903***	**0.1507***	0.0924
		improv.	13.81%	11.76%	8.60%	15.14%	17.42%	6.36%	7.87%	5.48%

Table 3: Ablation study evaluating the contribution of each component in LAGCL4Rec. The table compares performance across four datasets when removing individual components: data-level, rank-level, and reranking-level. Checkmarks (✓) indicate included components while cross marks (✗) indicate removed components. R@20 and N@20 represent Recall@20 and NDCG@20, respectively.

Variant	Components			ML-1M		Amazon-book		Yelp		Steam	
	data-level	rank-level	rerank-level	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20
base	✗	✗	✗	0.2729	0.3087	0.1477	0.0925	0.1108	0.0849	0.1397	0.0876
w/o data-level	✗	✓	✓	0.2707	0.3111	0.1465	0.0944	0.1061	0.0873	0.1401	0.0899
w/o rank-level	✓	✗	✓	0.2976	0.3345	0.1526	0.0989	0.1137	0.0884	0.1412	0.0907
w/o rerank-level	✓	✓	✗	0.3011	0.3303	0.1581	0.0977	0.1232	0.0795	0.1479	0.0884
all	✓	✓	✓	0.3106	0.3450	0.1604	0.1065	0.1301	0.0903	0.1507	0.0924

full model consistently achieves the best performance, with substantial gains on the Yelp dataset (17.42% improvement in R@20 compared to the base model). Notably, removing data augmentation sometimes reduces performance below the base model, highlighting the critical importance of our LLM-activated sparse interactions. These findings confirm each component's essential contribution to addressing the sparse interaction challenge.

Candidate Set Size Analysis. Figure 4 illustrates how candidate set size ($\mathcal{C}_u$) affects reranker performance. Performance consistently improves as $\mathcal{C}_u$ increases from 0 to 100, with significant gains in the 0 to 20 range. The Amazon-book dataset shows highest sensitivity for NDCG@20, suggesting complex domains benefit most from comprehensive candidate consideration. While Recall@20 varies across datasets, NDCG@20 shows consistent improvements, confirming that candidate diversity maximizes the effectiveness of our interaction-history guided reranker.

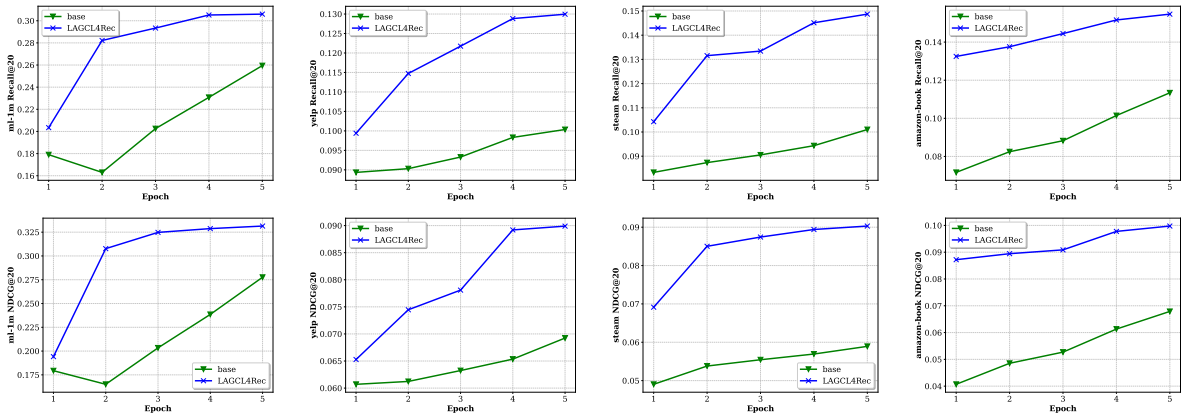
4.5 Empirical Analysis

Convergence Speed Analysis. Figure 5 demonstrates LAGCL4Rec's dramatically accelerated convergence across all datasets. This rapid optimization empirically validates Proposition 3.1, which shows hard negatives tighten the MI lower bound faster than easy negatives. By differentially amplifying these high-value signals through our semantic-aware grouping strategy, our Progressive Activation Pipeline effectively enhances learning efficiency at multiple levels.

Hyperparameter Analysis. Figure 6 shows sensitivity to key parameters, with optimal settings at fusion weight $\alpha \approx 0.2$, contrastive weight $\lambda_{cl} = 0.15$, and temperature ratios $\tau_e/\tau_p = 0.45$ and $\tau_h/\tau_p = 0.1$ across all datasets. The opti-

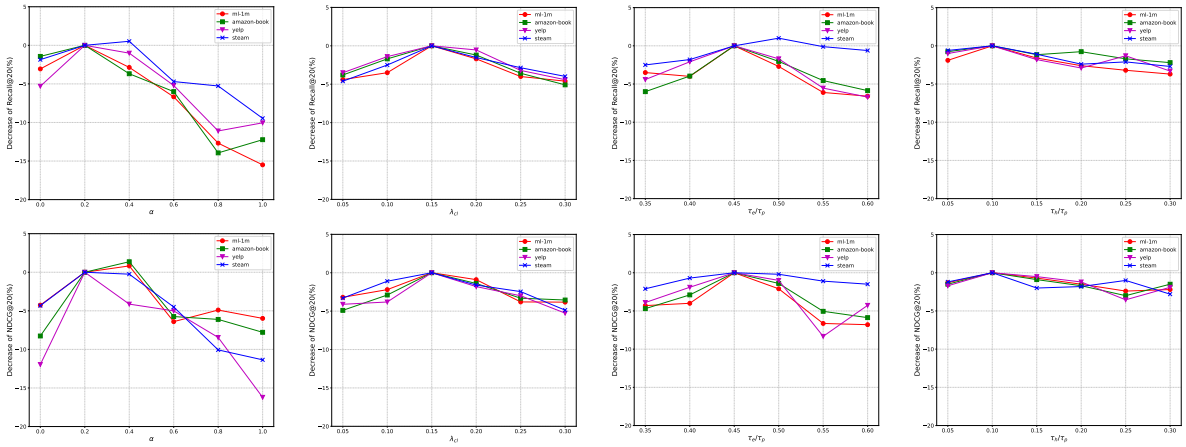
Table 4: Per-epoch time complexity of different LAGCL4Rec variants.

Component	LAGCL4Rec + SGL	LAGCL4Rec + LightGCL	LAGCL4Rec + XSimGCL
Baseline GCL Per-Epoch	$O(2\|E^+\|Ld + 4a\|E^+\|Ld + 3Md)$	$O(2\|E^+\|Ld + 2qMLd + 3Md)$	$O(2\|E^+\|Ld + 3Md)$
Rank-Level Addition	$O(Bd)$	$O(Bd)$	$O(Bd)$
Total Per-Epoch	$O(2\|E^+\|Ld + 4a\|E^+\|Ld + 3Md + Bd)$	$O(2\|E^+\|Ld + 2qMLd + 3Md + Bd)$	$O(2\|E^+\|Ld + 3Md + Bd)$



(a) ML-1M (b) Yelp (c) Steam (d) Amazon-book

Figure 5: Performance across training epochs for LAGCL4Rec versus baseline models on four datasets.



(a) Rerank weight α (b) Contrastive weight λ_{cl} (c) temp. ratio τ_e/τ_p (d) temp. ratio τ_h/τ_p

Figure 6: Impact of key hyperparameters on ranking quality (decrease in Recall@20 & NDCG@20 relative to optimal settings).

mal α value confirms our theoretical prediction in Proposition 3.2 that an intermediate fusion weight balancing collaborative filtering and LLM signals yields the best performance. Similarly, the temperature ratio settings align with our analysis of how adaptive temperatures enhance discriminative power.

User and Item Profiles Visualization. Figure 7 presents word clouds visualizing activated signals in user-item profiles. Across all domains, we observe the activation of both positive signals (terms like "enjoy," "appreciate") and negative signals (terms like "dislike," "negative review"), which aligns with our goal of awakening dormant preferences and illustrates how our approach captures domain-specific preference signals and activates relevant interactions.

4.6 Case Study

In Figure 8, we conducted detailed case analysis on the ML-1M dataset, examining a user with clear preferences for action/war films and aversions to horror. These bidirectional adjustments empirically validate Proposition 3.2, demonstrating how our approach effectively extracts higher information content from historical interactions compared to standard methods.

5 Related Work

Recommender systems have long been recognized as a canonical research problem and have attracted sustained attention from the academic

(a) ML-1M User Profile (b) ML-1M Item Profile (c) Yelp User Profile (d) Yelp Item Profile

(e) Steam User Profile (f) Steam Item Profile (g) Amazon-book User Profile (h) Amazon-book Item Profile

Figure 7: Word cloud visualizations of activated signals in user and item profiles from the datasets.

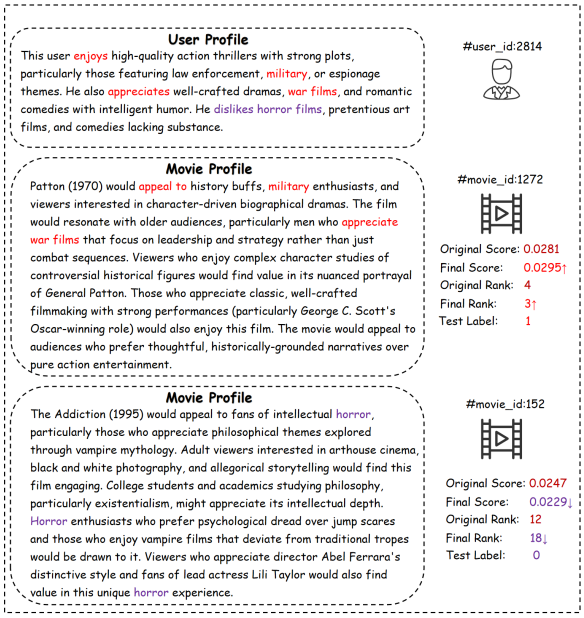


Figure 8: Case study demonstrating LAGCL4Rec's recommendation process for a user (ID: 2814) and two candidate movies: *Patton* (1970, ID: 1272) and *The Addiction* (1995, ID: 152). Red highlights indicate activated positive preference signals, while purple highlights indicate activated negative preference signals. The right panels show score and ranking changes throughout the recommendation pipeline.

community (Wu et al., 2024a, 2025; Di et al., 2025a,b,c), and the advent of large language models (LLMs) (Touvron et al., 2023; lla, 2023; Grattafiori et al., 2024; Qwen et al., 2025) has infused new vitality into the field, opening up fresh opportunities for innovation in recommendation (Wang and Lim, 2023; He et al., 2023; Li et al., 2025; Liang et al., 2024; Zhang et al., 2024; Feng et al., 2024; Fan et al., 2023; Chen et al., 2023; Zhang et al., 2023; Ortega et al., 2024; Li et al., 2023b; Yang et al., 2023; Wang et al., 2023). While LLM-based approaches have shown promise, they typically overlook dormant negative signals.

Graph Neural Networks (GNNs) have emerged as a highly active research frontier and are being applied across a wide spectrum of domains (Du et al., 2025a,b,c; Liu and Wang, 2025; Liu et al., 2024c), which have transformed recommender systems through their ability to model complex user-item relationships (Wu et al., 2023; Zheng et al., 2025a,b), but these methods treat all negative interactions uniformly rather than differentiating by difficulty level.

Our work addresses these limitations through a progressive activation pipeline operating at multiple levels. Unlike existing sign-aware systems (Derr et al., 2018; Huang et al., 2023; Wang and Cao, 2021; Gong and Zhu, 2022; Wu et al., 2020; Zhao et al., 2018) that lack systematic activation mechanisms, we explicitly awaken dormant signals in raw data, differentially amplify them based on semantic difficulty, and leverage historical interactions through structured reasoning. While recent reranking approaches (Xiong et al., 2023; Carraro and Bridge, 2024; Zhang et al., 2025; Wang et al., 2025a,b) have explored LLM integration, they fail to provide the comprehensive activation strategy we introduce.

6 Conclusion

We introduced LAGCL4Rec, a framework that activates the untapped potential in user-item interactions through our Progressive Activation Pipeline. By integrating LLMs with graph contrastive learning via Data-Level, Rank-Level, and Rerank-Level mechanisms, we addressed key limitations in existing approaches. Experiments across diverse datasets demonstrated significant performance improvements, highlighting the value of systematically unlocking and leveraging interaction potential.

Limitations

While our experimental results demonstrate the effectiveness of LAGCL4Rec across multiple public datasets, a notable limitation of our work is the absence of validation in industrial recommendation systems with real-world deployment. Industrial environments often present additional challenges including larger-scale data, more complex user-item interactions, and stricter latency requirements that may affect the practical applicability of our approach. Future work should focus on evaluating and adapting LAGCL4Rec for industrial deployment to comprehensively validate its effectiveness under production constraints.

Ethical Considerations

Our research exclusively uses public benchmark datasets with properly anonymized data. The recommendation domains explored (movies, books, businesses, and games) involve no sensitive content categories. Our LLM prompting approach focuses solely on preference patterns without introducing demographic or social biases. No human subjects were involved, and our work complies with standard ethical practices in recommendation systems research.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 92467203 and No. 62372264). Chaokun Wang is the corresponding author.

References

2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Keqin Bao, Jizhi Zhang, Yupeng Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. *arXiv preprint arXiv:2305.00447*.

Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. Lightgcl: Simple yet effective graph contrastive learning for recommendation. In *The Eleventh International Conference on Learning Representations*.

Diego Carraro and Derek Bridge. 2024. Enhancing recommendation diversity by re-ranking with large language models. *ACM Transactions on Recommender Systems*.

Jiawei Chen, Zhi Liu, Xin Huang, Chuan Wu, Qinzheng Liu, Guangyao Jiang, Yihuai Pu, Yunqi Lei, Xiuying Chen, Xin Wang, and 1 others. 2023. When large language models meet personalization: Perspectives of challenges and opportunities. *arXiv preprint arXiv:2307.16376*.

Sirui Chen, Jiawei Chen, Sheng Zhou, Bohao Wang, Shen Han, Chanfei Su, Yuqing Yuan, and Can Wang. 2024. Sigformer: Sign-aware graph transformer for recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1274–1284.

Tyler Derr, Yao Ma, and Jiliang Tang. 2018. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 929–934. IEEE.

Yicheng Di, Hongjian Shi, Ruhui Ma, Honghao Gao, Yuan Liu, and Weiyu Wang. 2025a. Fedrl: A reinforcement learning federated recommender system for efficient communication using reinforcement selector and hypernet generator. *ACM Transactions on Recommender Systems*, 4(1):1–31.

Yicheng Di, Hongjian Shi, Xiaoming Wang, Ruhui Ma, and Yuan Liu. 2025b. Federated recommender system based on diffusion augmentation and guided denoising. *ACM Transactions on Information Systems*, 43(2):1–36.

Yicheng Di, Xiaoming Wang, Hongjian Shi, Chongsheng Fan, Rong Zhou, Ruhui Ma, and Yuan Liu. 2025c. Personalized consumer federated recommender system using fine-grained transformation and hybrid information sharing. *IEEE Transactions on Consumer Electronics*.

Enjun Du, Xunkai Li, Tian Jin, Zhihan Zhang, Rong-Hua Li, and Guoren Wang. 2025a. Graphmaster: Automated graph synthesis via llm agents in data-limited environments. ArXiv preprint arXiv:2504.00711v2.

Enjun Du, Siyi Liu, and Yongqi Zhang. 2025b. Graphoracle: A foundation model for knowledge graph reasoning. ArXiv preprint arXiv:2505.11125.

Enjun Du, Siyi Liu, and Yongqi Zhang. 2025c. Mixture of length and pruning experts for knowledge graphs reasoning. ArXiv preprint arXiv:2507.20498.

Wenqi Fan, Zhiwei Zhao, Jianping Li, Yingxu Liu, Xiaokai Mei, Yupeng Wang, Jiliang Tang, and Qiang Li. 2023. Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046*.

Shanshan Feng, Haoming Lyu, Fan Li, Zhu Sun, and Caishun Chen. 2024. Where to move next: Zero-shot generalization of llms for next poi recommendation. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 1530–1535. IEEE.

Jingtong Gao, Bo Chen, Xiangyu Zhao, Weiwen Liu, Xiangyang Li, Yichao Wang, Wanyu Wang, Huifeng Guo, and Ruiming Tang. 2025. Llm4rerank: Llm-based auto-reranking framework for recommendations. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 228–239, New York, NY, USA. Association for Computing Machinery.

Yunfan Gao, Tao Sheng, Yi Xiang, Yihong Xiong, Hao Wang, and Jia Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524*.

Shansan Gong and Kenny Q Zhu. 2022. Positive, negative and neutral: Modeling implicit feedback in session-based news recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1185–1195.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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*, pages 639–648.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

Zhankui He, Zheng Xie, Rahul Jha, Harald Steck, Dawen Liang, Yansen Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. 2023. Large language models as zero-shot conversational recommenders. *arXiv preprint*.

Jiabin Hu, Weijun Xia, Xing Zhang, Chang Fu, Wei Wu, Zheng Huan, Anfeng Li, Zhengkun Tang, and Jianlong Zhou. 2024. Enhancing sequential recommendation via llm-based semantic embedding learning. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 103–111.

Junjie Huang, Ruobing Xie, Qi Cao, Huawei Shen, Shaoliang Zhang, Feng Xia, and Xueqi Cheng. 2023. Negative can be positive: Signed graph neural networks for recommendation. *Information Processing & Management*, 60(4):103403.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023a. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 1258–1267, New York, NY, USA. Association for Computing Machinery.

Xiangyu Li, Chen Chen, Xiangyu Zhao, Yicui Zhang, and Chao Xing. 2023b. E4srec: An elegant effective efficient extensible solution of large language models for sequential recommendation. *arXiv preprint arXiv:2312.02443*.

Yunzhe Li, Junting Wang, Hari Sundaram, and Zhining Liu. 2025. A zero-shot generalization framework for llm-driven cross-domain sequential recommendation. *arXiv preprint arXiv:2501.19232*.

Yueqing Liang, Liangwei Yang, Chen Wang, Xiongxiao Xu, Philip S Yu, and Kai Shu. 2024. Taxonomy-guided zero-shot recommendations with llms. *arXiv preprint arXiv:2406.14043*.

Junling Liao, Sichun Li, Ze Yang, Jonathan Wu, Yang Yuan, Xing Wang, and Xiangnan He. 2023. Llara: Aligning large language models with sequential recommenders. *arXiv preprint arXiv:2312.02445*.

Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM web conference 2022*, pages 2320–2329.

Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2024a. Once: Boosting content-based recommendation with both open- and closed-source large language models. WSDM '24, page 452–461, New York, NY, USA. Association for Computing Machinery.

Yuxi Liu, Lianghao Xia, and Chao Huang. 2024b. Self-gnn: Self-supervised graph neural networks for sequential recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1609–1618.

Ziyang Liu and Chaokun Wang. 2025. Terdy: Temporal relation dynamics through frequency decomposition for temporal knowledge graph completion. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9611–9622.

Ziyang Liu, Chaokun Wang, Hao Feng, and Ziyang Chen. 2024c. Efficient unsupervised graph embedding with attributed graph reduction and dual-level loss. *IEEE Transactions on Knowledge and Data Engineering*.

Gustavo Mendonça Ortega, Rodrigo Ferrari de Souza, and Marcelo Garcia Manzato. 2024. Evaluating zero-shot large language models recommenders on popularity bias and unfairness: A comparative approach to

traditional algorithms. In *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, pages 45–48. SBC.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Xuran Ren, Wenjie Wei, Longhui Xia, Lianghao Su, Sensen Cheng, Jiantao Wang, Dawei Yin, and Chao Huang. 2023. Representation learning with large language models for recommendation. *arXiv preprint arXiv:2310.15950*.

Changwon Seo, Kyeong-Joong Jeong, Sungsu Lim, and Won-Yong Shin. 2022. Siren: Sign-aware recommendation using graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):4729–4743.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agent. *arXiv preprint arXiv:2304.09542*.

Yuqi Sun, Qidong Liu, Haiping Zhu, and Feng Tian. 2025. Llmser: Enhancing sequential recommendation via llm-based data augmentation. *Preprint*, arXiv:2503.12547.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Chen Wang, Xiaokai Wei, Yexi Jiang, Frank Ong, Kevin Gao, Xiao Yu, Zheng Hui, Se-eun Yoon, Philip Yu, and Michelle Gong. 2025a. Solving the content gap in roblox game recommendations: Llm-based profile generation and reranking. *arXiv preprint arXiv:2502.06802*.

Chen Wang, Mingdai Yang, Zhiwei Liu, Pan Li, Linsey Pang, Qingsong Wen, and Philip Yu. 2025b. Automating personalization: Prompt optimization for recommendation reranking. *arXiv preprint arXiv:2504.03965*.

Jianling Wang, Haokai Lu, James Caverlee, Ed Chi, and Minmin Chen. 2024a. Large language models as data augmenters for cold-start item recommendation. *Preprint*, arXiv:2402.11724.

Lei Wang and Ee-Peng Lim. 2023. Zero-shot next-item recommendation using large pretrained language models. *arXiv preprint arXiv:2304.03153*.

Wei Wang and Longbing Cao. 2021. Interactive sequential basket recommendation by learning basket couplings and positive/negative feedback. *ACM Transactions on Information Systems (TOIS)*, 39(3):1–26.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174.

Xinfeng Wang, Fumiyo Fukumoto, Jin Cui, Yoshimi Suzuki, and Dongjin Yu. 2024b. Nfarec: A negative feedback-aware recommender model. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 935–945.

Yunfan Wang, Zeyu Liu, Jiangxia Zhang, Wei Yao, Stefan Heinecke, and Philip S Yu. 2023. Drdt: Dynamic reflection with divergent thinking for llm-based sequential recommendation. *arXiv preprint arXiv:2312.11336*.

Cheng Wu, Shaoyun Shi, Chaokun Wang, Ziyang Liu, Wang Peng, Wenjin Wu, Dongying Kong, Han Li, and Kun Gai. 2024a. Enhancing recommendation accuracy and diversity with box embedding: A universal framework. In *Proceedings of the ACM Web Conference 2024*, pages 3756–3766.

Cheng Wu, Liang Su, Chaokun Wang, Shaoyun Shi, Ziqian Zhang, Ziyang Liu, Wang Peng, Wenjin Wu, and Peng Jiang. 2025. Learning multiple user distributions for recommendation via guided conditional diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12845–12853.

Cheng Wu, Chaokun Wang, Jingcao Xu, Ziwei Fang, Tiankai Gu, Changping Wang, Yang Song, Kai Zheng, Xiaowei Wang, and Guorui Zhou. 2023. Instant representation learning for recommendation over large dynamic graphs. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 82–95. IEEE.

Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2020. Neural news recommendation with negative feedback. *CCF Transactions on Pervasive Computing and Interaction*, 2:178–188.

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*, pages 726–735.

Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. 2024b. Simplifying and empowering transformers for large-graph representations. *Advances in Neural Information Processing Systems*, 36.

Jing Xiong, Zixuan Li, Chuanyang Zheng, Zhijiang Guo, Yichun Yin, Enze Xie, Zhicheng Yang, Qingxing Cao, Haiming Wang, Xiongwei Han, and 1 others. 2023. Dq-lore: Dual queries with low rank approximation re-ranking for in-context learning. *arXiv preprint arXiv:2310.02954*.

Ze Yang, Juntao Wu, Yue Luo, Junhong Zhang, Yang Yuan, Aizhen Zhang, Xing Wang, and Xiangnan He. 2023. Large language model can interpret latent space of sequential recommender. *arXiv preprint arXiv:2310.20487*.

Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. 2023. Xsimgcl: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 36(2):913–926.

Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id- vs. modality-based recommender models revisited. SIGIR '23, New York, NY, USA. Association for Computing Machinery.

Haobo Zhang, Qiannan Zhu, and Zhicheng Dou. 2025. Enhancing reranking for recommendation with llms through user preference retrieval. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 658–671.

Jizhi Zhang, Keqin Bao, Yupeng Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation. *arXiv preprint arXiv:2305.07609*.

Xuan Zhang, Chunyu Wei, Ruyu Yan, Yushun Fan, and Zhixuan Jia. 2024. Large language model ranker with graph reasoning for zero-shot recommendation. In *International Conference on Artificial Neural Networks*, pages 356–370. Springer.

Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1040–1048.

Leqi Zheng, Chaokun Wang, Ziyang Liu, Canzhi Chen, Cheng Wu, and Hongwei Li. 2025a. Balancing self-presentation and self-hiding for exposure-aware recommendation based on graph contrastive learning. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2027–2037.

Leqi Zheng, Chaokun Wang, Zixin Song, Cheng Wu, Shannan Yan, Jiajun Zhang, and Ziyang Liu. 2025b. Negative feedback really matters: Signed dual-channel graph contrastive learning framework for recommendation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

A Progressive Activation Pipeline Prompts

This section provides comprehensive details of the prompts used in our Progressive Activation Pipeline. These prompts systematically activate dormant signals across the recommendation process, with particular focus on sparse interactions.

A.1 Data-Level Activation Prompts

A.1.1 User Profile Generation Prompt

This prompt generates comprehensive user profiles based on demographic information and ratings. It instructs the LLM to analyze both positive preferences (ratings 4-5) and negative preferences (ratings 1-3), explicitly activating dormant negative signals. Figure 9 shows the system instruction, Figure 10 demonstrates a sample input, and Figure 11 illustrates the resulting profile.

A.1.2 Item Profile Generation Prompt

This prompt analyses items and their potential appeal to different user demographics. Figure 12 shows the system instruction, Figure 13 illustrates a sample input, and Figure 14 displays the resulting profile.

A.1.3 Sparse Positive Interaction Activation Prompt

This prompt activates high-quality positive interactions for cold-start users. Figure 15 illustrates the prompt that guides the LLM through a systematic evaluation process to identify and activate relevant positive interactions.

A.1.4 Sparse Negative Interaction Activation Prompt

This prompt explicitly activates sparse negative preferences. Figure 16 guides the LLM to analyze patterns of aversion in user profiles and assign dislike scores with high discrimination, activating typically overlooked negative signals.

A.2 Rank-Level Activation Prompts

A.2.1 User/Item Grouping Prompt

This prompt implements a differentiated activation strategy based on interaction difficulty. Figure 17 shows how the LLM partitions users into semantically similar groups, enabling identification of hard and easy negative interactions requiring different activation levels.

A.2.2 Group Similarity Assessment Prompt

This prompt assesses similarity between groups to refine the differentiated activation strategy. Figure 18 illustrates the prompt that establishes the optimal grouping structure for differentiating between easy and hard negative interactions.

A.3 Rerank-Level Activation Prompts

This prompt activates historical user interactions during reranking. Figure 19 presents the structured Chain-of-Thought reasoning process that enables the LLM to systematically analyze preferences, compare items, and produce personalized rankings that effectively incorporate activated signals.

B Detailed Theoretical Proofs

B.1 Relationship Between Lemmas and Propositions

The theoretical analysis in Section 3 presents concise propositions that establish the effectiveness of our Progressive Activation Pipeline. These propositions are supported by a set of technical lemmas presented below, which provide more detailed mathematical properties and convergence guarantees.

Specifically, the first proposition on information gain from hard negative samples is supported by Lemma 1 and Lemma 2, which establish the gradient scaling properties and temperature-adaptive convergence behavior, respectively. The second proposition on Chain-of-Thought reasoning process is supported by Lemma 3 and Lemma 4, which prove the optimal weighting parameter existence and the convergence of our reranking approach. Together, these lemmas provide the mathematical foundation for our theoretical claims and explain why the Progressive Activation Pipeline effectively addresses the sparse interaction challenge in recommendation systems.

Lemma 1 (Difficulty-Based Gradient Scaling)**.** *Under the LABPR loss with $w_h > 1$, the expected gradient magnitude for hard negative samples is proportionally larger than for easy ones, ensuring accelerated learning for challenging cases.*

Lemma 2 (Temperature-Adaptive Convergence)**.** *With adaptive temperature parameters $\tau_h < \tau_e \leq \tau_p$ in our InfoNCE loss, the model converges to a more discriminative embedding space compared to uniform temperature settings.*

Lemma 3 (Score Fusion Optimality)**.** *There exists an optimal weighting parameter $\alpha^* \in [0, 1]$ that*

minimizes the expected ranking error:

$$\alpha^* = \arg\min_{\alpha \in [0,1]} \mathbb{E}_{(u,v)\sim\mathcal{D}}[L(s_{uv}^{final}, y_{uv})] \quad (23)$$

Lemma 4 (Reranking Convergence)**.** *The reranking model trained with BCE loss converges to a stable solution when the candidate set distribution remains consistent between training and inference.*

B.2 Detailed Proof of Lemma 1: Difficulty-Based Gradient Scaling

Considering the LABPR loss function 9, for a model parameterized by θ, the gradient with respect to parameters is:

$$\frac{\partial \mathcal{L}_{labpr}}{\partial \theta} = \quad (24)$$

$$-\sum_{(u,i,j_e)\in D_e} \frac{\sigma'(r_{ui} - r_{uj_e})}{\sigma(r_{ui} - r_{uj_e})} \cdot \frac{\partial r_{ui} - r_{uj_e}}{\partial \theta}$$

$$-\sum_{(u,i,j_h)\in D_h} w_h \cdot \frac{\sigma'(r_{ui} - r_{uj_h})}{\sigma(r_{ui} - r_{uj_h})} \cdot \frac{\partial r_{ui} - \partial r_{uj_h}}{\partial \theta} \quad (25)$$

Since $\sigma(x) = \frac{1}{1+e^{-x}}$, we have $\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$. Therefore:

$$\frac{\sigma'(x)}{\sigma(x)} = 1 - \sigma(x) \quad (26)$$

The gradient becomes:

$$\frac{\partial \mathcal{L}_{labpr}}{\partial \theta} = \quad (27)$$

$$-\sum_{(u,i,j_e)\in D_e} (1 - \sigma(r_{ui} - r_{uj_e})) \cdot \frac{\partial r_{ui} - r_{uj_e}}{\partial \theta}$$

$$-\sum_{(u,i,j_h)\in D_h} w_h \cdot (1 - \sigma(r_{ui} - r_{uj_h})) \cdot \frac{\partial r_{ui} - \partial r_{uj_h}}{\partial \theta} \quad (28)$$

For any pair of examples, one from D_e and one from D_h with identical $\sigma(r_{ui} - r_{uj})$ values, the ratio of their gradient contributions is exactly w_h. Since $w_h > 1$, the gradient contribution from hard negatives is proportionally larger, leading to faster adjustment of parameters to distinguish these challenging cases.

Furthermore, assuming that the distribution of $(r_{ui} - r_{uj})$ is similar for both easy and hard negative sets at initialization, the expected gradient magnitude ratio between hard and easy negatives is:

$$\frac{\mathbb{E}_{(u,i,j_h)\in D_h}[||\nabla_\theta \mathcal{L}_{(u,i,j_h)}||]}{\mathbb{E}_{(u,i,j_e)\in D_e}[||\nabla_\theta \mathcal{L}_{(u,i,j_e)}||]} \approx w_h \quad (29)$$

This ensures that the model allocates more learning capacity to hard negative examples, which is essential for developing fine-grained preference discrimination.

To establish the convergence implications, the overall loss function is as eq 12. Under standard gradient descent optimization with learning rate η, the parameter update rule is:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta \mathcal{L}_{total} \quad (30)$$

The component of this update attributable to the LABPR loss for hard negatives is approximately w_h times larger than for easy negatives, assuming similar initial conditions. This leads to faster convergence for hard negative cases, which is precisely what we desire for effective contrastive learning in sparse interaction scenarios.

B.3 Detailed Proof of Lemma 2: Temperature-Adaptive Convergence

Considering the activation-enhanced InfoNCE loss with adaptive temperature parameters in Eq 10 and Eq 11, let's denote $d_{ij} = sim(v_i, v_j)$ for simplicity. For any embedding pair (v_i, v_j), the contribution to the loss through the temperature parameter is determined by $\exp(d_{ij}/\tau)$.

With $\tau_h < \tau_e$, for any similarity value d_{ij}, we have:

$$\exp(d_{ij}/\tau_h) > \exp(d_{ij}/\tau_e) \quad (31)$$

assuming $d_{ij} > 0$ (which holds for cosine similarity of normalized embeddings with some positive correlation).

The gradient of the loss with respect to the similarity d_{ij} for a hard negative is:

$$\frac{\partial \mathcal{L}_{lainfo}}{\partial d_{ij}} = -\frac{1}{Z_i} \cdot w_h \cdot \frac{1}{\tau_h} \cdot \exp(d_{ij}/\tau_h) \quad (32)$$

and for an easy negative:

$$\frac{\partial \mathcal{L}_{lainfo}}{\partial d_{ij}} = -\frac{1}{Z_i} \cdot \frac{1}{\tau_e} \cdot \exp(d_{ij}/\tau_e) \quad (33)$$

For a given similarity value d_{ij}, the ratio of gradient magnitudes is:

$$\frac{|\frac{\partial \mathcal{L}_{lainfo}}{\partial d_{ij}}|_h}{|\frac{\partial \mathcal{L}_{lainfo}}{\partial d_{ij}}|_e} = w_h \cdot \frac{\tau_e}{\tau_h} \cdot \frac{\exp(d_{ij}/\tau_h)}{\exp(d_{ij}/\tau_e)} \tag{34}$$

$$= w_h \cdot \frac{\tau_e}{\tau_h} \cdot \exp\left(d_{ij}\left(\frac{1}{\tau_h} - \frac{1}{\tau_e}\right)\right) \tag{35}$$

Since $\tau_h < \tau_e$, we have $\frac{1}{\tau_h} > \frac{1}{\tau_e}$, so $\exp\left(d_{ij}\left(\frac{1}{\tau_h} - \frac{1}{\tau_e}\right)\right) > 1$ for positive d_{ij}. Additionally, $\frac{\tau_e}{\tau_h} > 1$. Therefore:

$$\frac{|\frac{\partial \mathcal{L}_{lainfo}}{\partial d_{ij}}|_h}{|\frac{\partial \mathcal{L}_{lainfo}}{\partial d_{ij}}|_e} > w_h > 1 \tag{36}$$

This demonstrates that the gradient magnitude for hard negatives is substantially larger than for easy negatives, leading to more aggressive optimization for challenging cases.

To establish convergence to a more discriminative embedding space, we need to analyze the equilibrium state of the optimization. Let us denote the embeddings after t iterations of gradient descent as $v_i^{(t)}$. The update rule for embeddings is:

$$v_i^{(t+1)} = v_i^{(t)} - \eta \cdot \nabla_{v_i} \mathcal{L}_{lainfo} \tag{37}$$

The gradient with respect to the embedding v_i depends on the similarities with all other embeddings and their contributions to the loss. Due to the larger gradient magnitudes for hard negatives (as established above), the embedding vectors will move more quickly to reduce similarities with hard negatives compared to easy negatives.

At convergence, the embeddings reach a state where the gradients approach zero. Due to the differential treatment of hard and easy negatives, this equilibrium state will have:

$$\mathbb{E}_{j \in \mathcal{N}_h(i)}[sim(v_i, v_j)] < \mathbb{E}_{j \in \mathcal{N}_e(i)}[sim(v_i, v_j)] \tag{38}$$

This indicates that hard negatives are pushed further away in the embedding space than easy negatives, creating a more discriminative representation where semantically similar but distinct items are well-separated. This property is crucial for fine-grained recommendation capabilities, especially in sparse interaction scenarios.

B.4 Detailed Proof of Lemma 3: Score Fusion Optimality

We establish the existence of an optimal weighting parameter α^* that minimizes the expected ranking error. The activation-weighted score is defined as:

$$s_{uv}^{final} = \alpha \cdot s_{uv}^{orig} + (1 - \alpha) \cdot s_{uv}^{llm} \tag{39}$$

Let $L(s_{uv}^{final}, y_{uv})$ be a ranking loss function measuring the discrepancy between the predicted score s_{uv}^{final} and the ground truth relevance y_{uv}. Common choices include pairwise ranking loss or listwise ranking loss.

Define the expected loss as:

$$\mathcal{R}(\alpha) = \mathbb{E}_{(u,v)\sim\mathcal{D}}[L(s_{uv}^{final}, y_{uv})] \tag{40}$$

We need to prove there exists an $\alpha^* \in [0, 1]$ that minimizes $\mathcal{R}(\alpha)$.

First, the continuity of $\mathcal{R}(\alpha)$ with respect to α can be established through the following observations: s_{uv}^{final} is a continuous function of α, the loss function $L(s_{uv}^{final}, y_{uv})$ is typically continuous in its first argument for common ranking losses, and the expectation operator preserves continuity.

To analyze the behavior of $\mathcal{R}(\alpha)$, we examine the extreme cases:

$$\mathcal{R}(0) = \mathbb{E}_{(u,v)\sim\mathcal{D}}[L(s_{uv}^{llm}, y_{uv})] \quad \text{(using only LLM scores)} \tag{41}$$

$$\mathcal{R}(1) = \mathbb{E}_{(u,v)\sim\mathcal{D}}[L(s_{uv}^{orig}, y_{uv})] \quad \text{(using only CF scores)} \tag{42}$$

If either $\mathcal{R}(0) \leq \mathcal{R}(\alpha)$ for all $\alpha \in [0, 1]$ or $\mathcal{R}(1) \leq \mathcal{R}(\alpha)$ for all $\alpha \in [0, 1]$, then α^* is trivially 0 or 1, respectively.

Otherwise, by the extreme value theorem, since $\mathcal{R}(\alpha)$ is continuous over the compact interval $[0, 1]$, there exists an $\alpha^* \in [0, 1]$ such that $\mathcal{R}(\alpha^*) \leq \mathcal{R}(\alpha)$ for all $\alpha \in [0, 1]$.

To prove that $\alpha^* \in (0, 1)$ is likely in practice (rather than at the extremes), we can construct a scenario where the original and LLM scores have complementary strengths. Let $\mathcal{U}_1$ and $\mathcal{U}_2$ partition the user space such that:

$$\mathbb{E}_{(u,v)\sim\mathcal{D}, u\in\mathcal{U}_1}[L(s_{uv}^{orig}, y_{uv})] < \mathbb{E}_{(u,v)\sim\mathcal{D}, u\in\mathcal{U}_1}[L(s_{uv}^{llm}, y_{uv})] \tag{43}$$

$$\mathbb{E}_{(u,v)\sim\mathcal{D}, u\in\mathcal{U}_2}[L(s_{uv}^{orig}, y_{uv})] > \mathbb{E}_{(u,v)\sim\mathcal{D}, u\in\mathcal{U}_2}[L(s_{uv}^{llm}, y_{uv})] \tag{44}$$

That is, collaborative filtering scores perform better for users in $\mathcal{U}_1$, while LLM scores perform better for users in $\mathcal{U}_2$.

In this case, neither $\alpha = 0$ nor $\alpha = 1$ would be optimal across all users. There exists an intermediate value $\alpha^* \in (0, 1)$ that minimizes the overall expected loss by balancing the strengths of both score sources.

To find this optimal value analytically, we can differentiate $\mathcal{R}(\alpha)$ with respect to α and set it to zero:

$$\frac{d\mathcal{R}(\alpha)}{d\alpha} = \mathbb{E}_{(u,v)\sim\mathcal{D}}\left[\frac{\partial L(s_{uv}^{final}, y_{uv})}{\partial s_{uv}^{final}} \cdot \frac{\partial s_{uv}^{final}}{\partial \alpha}\right] = 0 \tag{45}$$

Since $\frac{\partial s_{uv}^{final}}{\partial \alpha} = s_{uv}^{orig} - s_{uv}^{llm}$, we have:

$$\mathbb{E}_{(u,v)\sim\mathcal{D}}\left[\frac{\partial L(s_{uv}^{final}, y_{uv})}{\partial s_{uv}^{final}} \cdot (s_{uv}^{orig} - s_{uv}^{llm})\right] = 0 \tag{46}$$

This equation has a solution $\alpha^* \in (0, 1)$ when the original and LLM scores have complementary strengths across different regions of the user-item space. In practical recommendation scenarios, this is often the case, as collaborative filtering excels at capturing collaborative patterns while LLM-based models capture semantic relationships.

B.5 Detailed Proof of Lemma 4: Reranking Model Convergence

We analyze the convergence properties of the reranking model trained with binary cross-entropy (BCE) loss. The BCE loss is defined as:

$$\mathcal{L}_{bce} = \frac{1}{|\mathcal{D}_{rer}|} \sum_{(u,v,y_{uv})\in\mathcal{D}_{rer}} [-y_{uv}\log(p_{uv}) - (1 - y_{uv})\log(1 - p_{uv})] \tag{47}$$

where p_{uv} is the model's predicted probability that user u would interact with item v.

First, we establish that BCE loss is convex with respect to model predictions. For a single example (u, v, y_{uv}), the loss is:

$$\ell(p_{uv}, y_{uv}) = -y_{uv}\log(p_{uv}) - (1-y_{uv})\log(1-p_{uv}) \tag{48}$$

The second derivative with respect to p_{uv} is:

$$\frac{\partial^2 \ell}{\partial p_{uv}^2} = \frac{y_{uv}}{p_{uv}^2} + \frac{1 - y_{uv}}{(1 - p_{uv})^2} \tag{49}$$

Since $y_{uv} \in \{0, 1\}$ and $p_{uv} \in (0, 1)$ in practice, the second derivative is always positive, confirming that BCE loss is strictly convex in the model predictions.

For the reranking model with parameters θ_{rer}, assuming it maps inputs to probabilities via a function $f_{\theta_{rer}}$, the optimization problem is:

$$\theta_{rer}^* = \arg\min_{\theta_{rer}} \mathcal{L}_{bce}(\theta_{rer}; \mathcal{D}_{rer}) \tag{50}$$

When the model has sufficient capacity and the mapping from parameters to predictions is continuous, gradient-based optimization of convex loss functions converges to a global minimum under standard conditions (e.g., appropriate learning rate scheduling, sufficient iterations).

Now, we analyze the stability of the solution when the candidate set distribution remains consistent between training and inference. Let $\mathcal{D}_{train}$ be the distribution of examples in the training set and $\mathcal{D}_{inf}$ be the distribution during inference.

The training objective is:

$$\mathbb{E}_{(u,v,y_{uv})\sim\mathcal{D}_{train}}[\ell(f_{\theta_{rer}}(u, v), y_{uv})] \tag{51}$$

The optimal parameters under this objective are:

$$\theta_{rer}^* = \arg\min_{\theta_{rer}} \mathbb{E}_{(u,v,y_{uv})\sim\mathcal{D}_{train}}[\ell(f_{\theta_{rer}}(u, v), y_{uv})] \tag{52}$$

At inference time, we want these parameters to perform well under $\mathcal{D}_{inf}$. The performance gap can be bounded by:

$$\begin{aligned} &|\mathbb{E}_{(u,v,y_{uv})\sim\mathcal{D}_{inf}}[\ell(f_{\theta_{rer}^*}(u, v), y_{uv})] \\ &\quad - \mathbb{E}_{(u,v,y_{uv})\sim\mathcal{D}_{train}}[\ell(f_{\theta_{rer}^*}(u, v), y_{uv})]| \\ &\leq TV(\mathcal{D}_{inf}, \mathcal{D}_{train}) \cdot C \end{aligned} \tag{53}$$

where TV is the total variation distance between distributions and C is a constant related to the maximum possible loss value.

When the candidate set distribution remains consistent between training and inference, $TV(\mathcal{D}_{inf}, \mathcal{D}_{train})$ is small, ensuring that the model's performance generalizes well from training to inference.

Furthermore, since the collaborative filtering model is fixed during reranking optimization, any

instability in the overall system would have to originate from the reranking model itself. By ensuring stable convergence of the reranking model through proper regularization and training procedures, the overall system maintains stability.

To analyze the convergence rate, we can use standard results from convex optimization. For a strongly convex loss function with parameter μ and Lipschitz continuous gradients with constant L, gradient descent with learning rate $\eta = \frac{1}{L}$ achieves linear convergence:

$$\mathcal{L}_{bce}(\theta_t) - \mathcal{L}_{bce}(\theta^*) \leq \left(1 - \frac{\mu}{L}\right)^t \cdot [\mathcal{L}_{bce}(\theta_0) - \mathcal{L}_{bce}(\theta^*)] \tag{54}$$

where θ_t is the parameter vector at iteration t, and θ^* is the optimal parameter vector.

In practice, while the BCE loss itself is convex in model predictions, the overall loss landscape may not be strongly convex in model parameters due to the non-linear mapping from parameters to predictions in neural networks. However, empirical evidence suggests that well-regularized models with appropriate architectures converge reliably to good solutions, even if they are local rather than global minima.

The stability of the reranking process is further enhanced by the activation-weighted score computation, which combines the collaborative filtering scores with the LLM-generated scores as eq 19. This weighted combination acts as a form of ensemble learning, which is known to improve stability and generalization. Even if one component (either the collaborative filtering model or the LLM reranking model) experiences instability, the weighted combination can mitigate its effects on the final recommendations.

You will serve as an assistant to help me determine which types of movies a specific user is likely to enjoy. Here are the instructions:
1. The user data will be provided in JSON format with the following structure:

```
{
  "user_id": "unique identifier for the user",
  "user_information": {
    "gender": "M for male, F for female",
    "age": "age range (Under 18, 18-24, 25-34, 35-44, 45-49, 50-55, 56+)",
    "occupation": "occupation description (e.g., programmer, student, etc.)"
  },
  "rated_movies": [
    {
      "movie_id": "unique identifier for the movie",
      "title": "movie title with release year",
      "genres": "pipe-separated list of genres",
      "rating": "rating on a 5-star scale"
    },
    ...
  ]
}
```

2. Rating interpretation: Ratings 4-5 represent positive preferences (the user likes these movies). Ratings 1-3 represent negative preferences (the user dislikes these movies).

REQUIREMENTS:

1. Consider both positive and negative ratings to understand overall preferences.
2. Please provide your analysis in JSON format, following this structure:

```
{
  "user_id": "the user ID from the input data",
  "summarization": "A summarization of what types of movies this user is likely to enjoy",
  "reasoning": "briefly explain your reasoning for the summarization, including analysis of demographic factors and rating patterns"
}
```

3. Please ensure that the "summarization" is no longer than 100 words.
4. Please ensure that the "reasoning" is no longer than 200 words.
5. Do not provide any other text outside the JSON string.

User Profile Generation Instruct

Figure 9: User Profile Generation Instruct

Here is the user profile to analyze:


```
{
  "user_id": "1683",
  "user_information": {
    "gender": "F",
    "age": "25-34",
    "occupation": "executive/managerial"
  },
  "rated_movies": [
    {
      "movie_id": "2987",
      "title": "Who Framed Roger Rabbit? (1988)",
      "genres": "Adventure|Animation|Film-Noir",
      "rating": "4"
    },
    {
      "movie_id": "2054",
      "title": "Honey, I Shrunk the Kids (1989)",
      "genres": "Adventure|Children's|Comedy|Fantasy|Sci-Fi",
      "rating": "2"
    },
    ...
    ]
}
```


User Profile Prompt Input

Figure 10: User Profile Prompt Input


```
[
  {
    "user_id": "1683",
    "summarization": "This user enjoys sophisticated comedies, romantic films, and thoughtful dramas from the 1980s-1990s. She prefers character-driven stories with emotional depth, particularly those featuring strong performances. She strongly dislikes horror movies but appreciates quality thrillers with compelling narratives.",
    "reasoning": "The user is a female executive/managerial professional in the 25-34 age range, which suggests she may appreciate films with mature themes and sophisticated storytelling. Her rating patterns reveal clear preferences and dislikes:\n\nStrong preferences (rated 4-5):\n1. Comedies with smart dialogue and adult themes (\"American Beauty\", \"When Harry Met Sally\", \"Moonstruck\", \"Sixteen Candles\", \"Ruthless People\")\n2. Romantic films, particularly those with depth (\"Ghost\", \"Room with a View\", \"Dangerous Liaisons\")\n3. Thoughtful dramas, especially those with strong performances (\"Gandhi\", \"On Golden Pond\", \"Children of a Lesser God\", \"Sophie's Choice\")\n4. Classic film noir and mysteries (\"The Big Sleep\", \"Who Framed Roger Rabbit?\")\n5. Action-comedies with romantic elements (\"Romancing the Stone\", \"Lethal Weapon\")\n\nStrong dislikes (rated 1-2):\n1. Horror movies (consistently rated all horror films as 1, including \"Hellraiser\", \"Friday the 13th\" series, \"Nightmare on Elm Street\" series)\n2. Children's fantasy adventures (\"NeverEnding Story\", \"Willow\")\n3. Some sci-fi action films (\"Superman III\", \"Star Trek: The Wrath of Khan\")\n\nThe user shows a particular affinity for 1980s films, which may reflect nostalgia for films from her formative years. She rates comedies and dramas with emotional depth highly, while consistently rejecting horror films. Her preferences suggest she values character development, emotional resonance, and sophisticated storytelling over special effects or fantastical elements. The high ratings for films like \"Dead Poets Society\", \"Steel Magnolias\", and \"Terms of Endearment\" further support her preference for emotionally impactful storytelling."
  },
  ...
]
```


User Profile Prompt Output

Figure 11: User Profile Prompt Output

You will serve as an assistant to help me summarize which types of users would enjoy a specific movie. Here are the instructions:
1. I will provide you with information in the form of a JSON string that describes the movie:
```
{
  "movie_id": "unique identifier for the movie",
  "title": "movie title with release year",
  "genres": "pipe-separated list of genres"
},
```
2. Use your knowledge about this movie to analyze what types of viewers might enjoy it.
REQUIREMENTS:
1. Please provide your analysis in JSON format, following this structure:
```
{
  "movie_id": "the movie ID from the input data",
  "summarization": "A summarization of what types of users would enjoy this movie",
  "reasoning": "briefly explain your reasoning for the summarization"
}
```
2. If you don't have enough information about the movie or cannot recognize it, please still provide a reasonable estimation based on the genres and title.
3. Please ensure that the "summarization" is no longer than 100 words.
4. Please ensure that the "reasoning" is no longer than 200 words.
5. Do not provide any other text outside the JSON string.

Item Profile Generation Instruct

Figure 12: Item Profile Generation Instruct

Here is the movie data to analyze:
```
[
  {
    "movie_id": "1",
    "title": "Toy Story (1995)",
    "genres": "Animation|Children's|Comedy"
  },
  ...
]
```

Item Profile Prompt Input

Figure 13: Item Profile Prompt Input

```
[
  {
    "movie_id": "1",
    "summarization": "Toy Story (1995) would appeal to a diverse audience, with children ages 4-12 forming the primary demographic. Families seeking wholesome entertainment would enjoy its universal themes of friendship, loyalty, and growing up. Adults, particularly those with nostalgia for childhood toys or appreciation for Pixar's pioneering animation, would find deeper emotional resonance and humor. Animation enthusiasts would value its groundbreaking technical achievements as the first fully computer-animated feature film. The movie's blend of slapstick humor for children and clever wit for adults makes it ideal for multi-generational viewing. Fans of imaginative storytelling that anthropomorphizes everyday objects would connect with its premise of toys coming to life.",
    "reasoning": "As Pixar's debut feature and the first fully computer-animated film, Toy Story revolutionized animation while delivering a story with multi-generational appeal. The genre combination of Animation, Children's, and Comedy indicates its primary target audience (children) while suggesting broader appeal through humor. The film balances child-friendly adventure with sophisticated emotional themes and clever dialogue that adults appreciate. Its innovative animation would attract film enthusiasts and technology fans. The universal themes of friendship, jealousy, and belonging transcend age barriers, while its portrayal of toys from different eras creates nostalgia for various generations. The film's G rating, adventurous plot, and emotional depth make it accessible to young viewers while engaging enough for adults, explaining why it became a beloved family classic that launched a successful franchise."
  },
  ...
]
```

Item Profile Prompt Output

Figure 14: Item Profile Prompt Output

You are a recommendation expert. Score how likely this user would enjoy each movie (1-10 scale).
USER PROFILE:
{json.dumps(user_profile, indent=2)}
ITEM PROFILE:
{json.dumps(item_profiles_list, indent=2)}
ANALYTICAL PROCESS:
STEP 1: Analyze user profile: Identify top genres, creators, and time periods user prefers.
STEP 2: Score each item on key factors (1-10): Genre match, Creator match, Content match.
STEP 3: Assign final score with high discrimination: Use FULL 1-10 range with meaningful differentiation.
- 9-10: Perfect match (rare, <5% of recommendations)
- 7-8: Strong match with minor misalignments
- 5-6: Moderate match with notable mismatches
- 3-4: Weak match, significant incompatibility
- 1-2: Poor match, fundamental misalignment
REQUIREMENTS:
1. Please provide your analysis in JSON format, following this structure:
```
{
  "ratings": [
    {
      "movie_id": "movie_id_value",
      "like_score": "score_value",
      "reasoning": "Detailed explanation of scores"
    }
  ]
}
```
2. Use precise, highly differentiated scores
3. Ensure movie_id matches exactly as provided

IMPORTANT: Return ONLY the JSON object directly, with no additional text, comments, or markdown formatting before or after it.

Positive Interaction Generation Instruct

Figure 15: Cold-Start Positive Interaction Activation Instruct

```
You are an anti-recommendation expert. Score how likely this user would DISLIKE each movie (1-10 scale).
USER PROFILE:
{json.dumps(user_profile, indent=2)}
ITEM PROFILE:
{json.dumps(item_profiles_list, indent=2)}
ANALYTICAL PROCESS:
STEP 1: Analyze user profile: Identify genres, creators, and time periods user clearly avoids or dislikes and viewing patterns that suggest strong aversions.
STEP 2: Score each movie on key mismatch factors (1-10): Genre match, Creator match, Content match.
STEP 3: Assign final dislike score with high discrimination: Use FULL 1-10 range with meaningful differentiation.
- 9-10: Complete mismatch (high confidence user would dislike)
- 7-8: Strong mismatch with few redeeming qualities
- 5-6: Moderate mismatch with some compatibility
- 3-4: Minor mismatch, mostly compatible
- 1-2: Minimal mismatch, likely enjoyable
REQUIREMENTS:
1. Please provide your analysis in JSON format, following this structure:
{
  "ratings": [
    {
      "movie_id": "movie_id_value",
      "dislike_score": "score_value",
      "reasoning": "Detailed explanation of scores"
    }
  ]
}
2. Use precise, highly differentiated scores
3. Ensure movie_id matches exactly as provided

IMPORTANT: Return ONLY the JSON object directly, with no additional text, comments, or markdown formatting before or after it.
```

Negative Interaction Generation Instruct

Figure 16: Sparse Negative Interaction Activation Instruct

```
You will be given {len(groups)} group profiles, each containing information about a group's movie preferences and the reasoning behind these preferences.
Your task is to calculate an estimated similarity score between two groups.
INPUT
Group 1:
- ID: {group1['group_id']}
- Name: {group1.get('group_name', 'Unnamed Group')}
- Key Characteristics: {json.dumps(group1.get('key_characteristics', []))}
- Reasoning: {group1.get('reasoning', '')}
Group 2:
- ID: {group2['group_id']}
- Name: {group2.get('group_name', 'Unnamed Group')}
- Key Characteristics: {json.dumps(group2.get('key_characteristics', []))}
- Reasoning: {group2.get('reasoning', '')}
ANALYTICAL PROCESS:
STEP 1: Analyze the key characteristics of both groups. Identify shared characteristics between both groups.
STEP 2: Analyze the overall themes, interests, and behaviors represented in each group.
STEP 3: Calculate an estimated similarity score (0-100) based on these criteria:
  - 0-20: Fundamentally different groups with almost no overlapping characteristics
  - 21-40: Mostly different with a few minor similarities
  - 41-60: Moderate similarity with significant differences
  - 61-80: Substantial similarity with some notable differences
  - 81-100: Extremely similar groups with only minimal differences
STEP 4: Provide clear justification for your similarity score, explaining specific similarities and differences.
REQUIREMENTS:
1. Please provide your analysis in JSON format, following this structure:
{
  "similarity_score": [your calculated score between 0-100],
  "similarity_explanation": "Detailed explanation of similarities and differences..."
}
2. All property names and string values must use double quotes.
IMPORTANT: Do not include any additional text, markdown formatting, code blocks, or explanations outside the JSON.
```

Group Merge Instruct

Figure 17: Group Merge Instruct

You will be given {len(users)} user profiles, each containing information about a user's movie preferences and the reasoning behind these preferences.

Your task is to:
1. Create logical groups of users with similar preferences and tastes
2. Name and characterize each group
3. Assign each user to their most appropriate group
4. IMPORTANT: Each user must be assigned to EXACTLY ONE group
5. CRITICAL: Only use valid user_ids exactly as provided in the input user profiles

USER PROFILE:
{json.dumps(users, indent=2)}
REQUIREMENTS:
1. Please provide your analysis in JSON format, following this structure:

```
{
  "groups": [
    {
      "group_id": "group_1",
      "group_name": "Descriptive name for the group",
      "key_characteristics": ["characteristic 1", "characteristic 2", ...],
      "members": ["user_id_1", "user_id_3", ...],
      "reasoning": "Explanation of what defines this group"
    },
    ...
  ]
}
```

2. You MUST use the exact user_ids provided in the input - do not modify, reformat, or make up user_ids
3. Every user_id in your response must exactly match one from the provided user profiles
4. Every user must be assigned to exactly one group
5. User_ids are case-sensitive strings - preserve their exact format
6. Your ENTIRE response must be a valid JSON object with the exact structure specified above.
IMPORTANT: Do not include any additional text, markdown formatting, code blocks, or explanations outside the JSON.

Group Partition Instruct

Figure 18: Group Partition Instruct

You are a Precision Reranking Engine.
Your sole task is to reorder the provided list of candidates strictly and exclusively based on the user preferences detailed below.
USER INFORMATION
User Information: User {user_id} with explicit preferences: {summarization}
CANDIDATE ITEMS
Candidate Items List: {item_id}
Original Ranking: {item_id}
USER INTERACTION HISTORY
Items the user previously liked (strong relevance signals): Item {pos_id}: {summarization}
Items the user previously disliked (negative signals): Item {neg_id}: {summarization}
CANDIDATE ITEM DETAILS
Item {item_id} {item_signal}: {summarization}
RANKING REASONING PROCESS
Follow this reasoning process:
Step 1: Prioritize items that closely match the user's historical preferences and explicit likes.
Step 2: Demote items similar to those the user has historically disliked.
Step 3: Make minimal adjustments to the original ranking, changing positions only when you have high confidence.
RANKING GUIDELINES
1. Promote the most relevant items to top positions
2. Make conservative adjustments - only move items when necessary
3. The original ranking is usually correct - only adjust with clear evidence

[TRAINING PHASE ONLY]
For this single-item evaluation:
- If this item should be ranked highly based on user preferences, classify as relevant (1)
- If this item should be ranked lower, classify as less relevant (0)

Rerank Prompt Instruct

Figure 19: Chain-of-Thought Activation Process Instruct