SHORT-PAPER

Leveraging Refined Negative Feedback with LLM for Recommender Systems

CHANWOO JEONG, Chung-Ang University, Seoul, South Korea

YUJIN KANG, Chung-Ang University, Seoul, South Korea

YOON-SIK CHO, Chung-Ang University, Seoul, South Korea

Open Access Support provided by:

Chung-Ang University

Published: 08 May 2025

Citation in BibTeX format

WWW '25: The ACM Web Conference 2025
April 28 - May 2, 2025
Sydney NSW, Australia

Conference Sponsors:
SIGWEB

https://doi.org/10.1145/3701716.3715538
ISBN: 9798400713316

Leveraging Refined Negative Feedback with LLM for Recommender Systems

Chanwoo Jeong
Chung-Ang University
Seoul, Republic of Korea
chwchong@cau.ac.kr

Yujin Kang
Chung-Ang University
Seoul, Republic of Korea
zinzin32@cau.ac.kr

Yoon-Sik Cho*
Chung-Ang University
Seoul, Republic of Korea
yoonsik@cau.ac.kr

Abstract

Recently, there has been growing research on negative feedback in recommender systems. These studies use a fixed threshold to binarize feedback into positive or negative. However, such an approach bears limitations when the rating habits for expressing disappointment differ across users or when ratings are noisy. Motivated by the remarkable success of Large Language Models (LLMs), we investigate how LLM can address this challenge on the fly. To this end, we present **ReFINe**, **Re**surrecting **F**alsely **I**dentified **Ne**gative feedback with LLM. ReFINe classifies the negative feedback into two distinct types: *Falsely identified* negative with positive signals and *Confirmed* negative with only negative signals. To the best of our knowledge, our work is the first to propose and demonstrate the distinction between two perspectives on negative feedback. We first leverage LLM to better separate between positive and negative sets for each user, and implement *Re-Weighted* BPR, a dedicated Bayesian Personalized Ranking loss function tailored to our perspective on negative feedback. Experimental results show that our model outperforms strong baseline models. The code is available at https://github.com/Chanwoo-Jeong-2000/ReFINe.

CCS Concepts

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

Keywords

Recommender Systems, Negative Feedback, Large Language Models, Graph Neural Networks

ACM Reference Format:
Chanwoo Jeong, Yujin Kang, and Yoon-Sik Cho. 2025. Leveraging Refined Negative Feedback with LLM for Recommender Systems. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3701716.3715538

1 Introduction

The increasing reliance on online platforms and services has heightened the demand for personalized recommendations that align with users' unique preferences. While consistent improvement is being shown in recommender systems, one challenge comes from the data preprocessing. Most existing recommender systems employ simplified implicit data representations that compress user feedback to binary states. The binary transformation obscures the granularity of user preference, thereby resulting in significant information loss. In implicit recommender systems, ratings from 1 to 5 are transformed; ratings of 4 and 5 (*positive feedback*) are mapped to 1, while ratings of 1, 2, and 3 (*negative feedback*) are assigned a value of 0, the same score value for unrated data. Specifically, implicit recommender systems treat both unrated data and data with negative feedback identically, resulting in a problem where these cannot be distinguished at all.

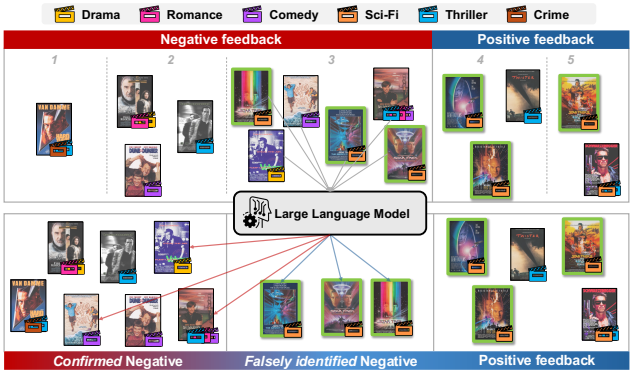

Figure 1: Illustration of the movie ratings of *user_394* in the ML-100K dataset. The *Star Trek* series are highlighted with green borders. The upper shows classifications from conventional negative feedback, while the lower part represents our refined results.

Previous works have pointed out that the negative feedback is often overlooked and have emphasized the importance of leveraging such feedback. These studies distinguish the low-rated feedback from unknown ratings, and utilize negative feedback to model user dispreferences [1, 5–7, 9]. Specifically, Huang et al. [5] identify the differences in negative feedback intensity across users, with some instances showing similarities to positive feedback. Although the study explores the diversity of negative feedback, it still neglects subtle distinctions between low-rated responses.

Figure 1 illustrates the movies rated by *user_394* in the MovieLens-100K (ML-100K) dataset, grouped by rating. The user is a big fan of Sci-Fi, particularly *Star Trek*, having watched the entire *Star Trek* series, as indicated by the posters with a green border in Figure 1. Despite this clear enthusiasm, three *Star Trek* movies rated 3 are treated as negative feedback under traditional perspective of negative feedback (See the upper part of figure). However, we

*Corresponding author.

WWW Companion '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1331-6/2025/04
https://doi.org/10.1145/3701716.3715538

can observe from the user's rating history the strong interest for *Star Trek* and clear preference for the Sci-Fi genre. Therefore, we suggest the possibility that the *Star Trek* movies categorized as negative feedback may exhibit positive signals. These observations highlight the need for a more detailed classification and utilization of negative feedback.

To address this problem, we propose **ReFINe**, **Re**surrecting **F**alsely **I**dentified **Ne**gative feedback. The conventional approach uniformly classifies all ratings of 3 or lower as negative feedback. We raise doubts about the effectiveness of this definition. Thus, ReFINe redefines negative feedback, considering the intrinsic attributes of individual instances, into two categories: *Falsely identified* negative and *Confirmed* negative. *Falsely identified* negative feedback occurs when a user dislikes an item due to specific aspects but still finds it relevant to their interests and preferences. In contrast, *confirmed* negative feedback represents a case where the user strongly dislikes the item and does not wish to receive similar recommendations again. As far as we know, our work is the first study to propose a novel perspective on negative feedback. Applying this new perspective, we refine existing data by calibrating user responses to better reflect their actual preferences. In this process, ratings are reassigned based on personalized criteria that reflect individual preferences and history.

Our approach leverages Large Language Model (LLM) to infer user preferences from their positive feedback and then distinguishes their negative feedback into *falsely identified* negative and *confirmed* negative. For *user_394* mentioned in Figure 1 in the earlier example, we confirm that the LLM identifies all three *Star Trek* movies rated 3 as containing positive signals and distinguishes them as falsely identified negative. Additionally, the other three movies rated 3, which belong to genres different from movies rated 4 and 5 but similar to genres of movies rated 1 and 2, are identified as confirmed negatives. These results confirm that the LLM effectively distinguishes the two types of negative feedback we propose (refer to the lower part of the Figure 1). The falsely identified negatives are no longer considered negative feedback and should be treated as equivalent to positive feedback. We use this refined feedback to train the Graph Convolutional Networks (GCNs) model. On the other hand, confirmed negatives, which contain only negative signals, are processed using our modified Bayesian Personalized Ranking (BPR) [8], *Re-Weighted* BPR, to drive the embeddings of the user and confirmed negative items further apart in the embedding space. Our model achieves state-of-the-art performance compared to existing baselines on three real-world benchmark datasets.

2 Proposed Method

In this section, we introduce our model, **ReFINe**, which consists of two modules—data refinement with LLM and training strategies—both leveraging our refined perspective on negative feedback.

2.1 LLM-Driven Strategies for Refining Negative Feedback

Overlooking subtle distinctions within negative feedback significantly constrains the model's ability to accurately capture users' preferences. We shed light on such traditional negative feedback

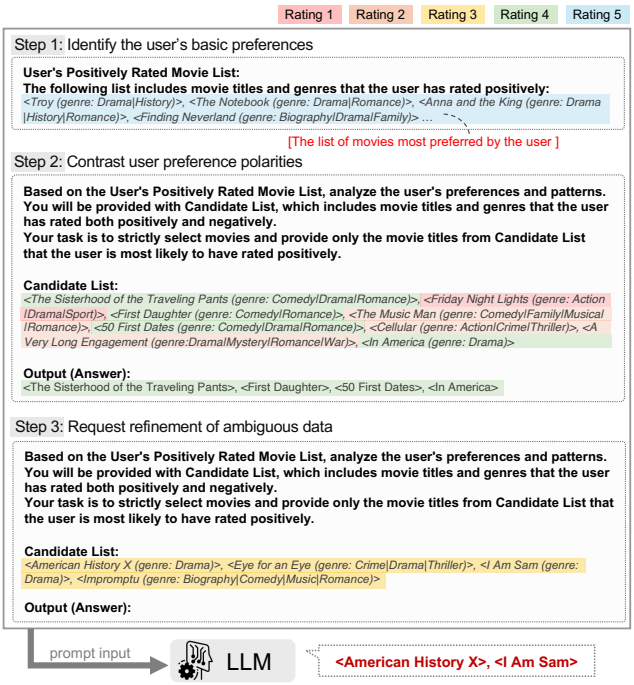

Figure 2: The prompt and inference for *user_1502605* from the Netflix-1M dataset. The prompt consists of three steps, with bold text indicating the prompt itself and *italic text* serving as the input.

through refined categories: *falsely identified* negatives and *confirmed* negatives. To effectively distinguish these negatives, we leverage the versatile LLM, LLaMA2-7B [10]. We design a three-step prompt strategy to enable LLM to effectively refine negative feedback.

In the first step, we provide the LLM with movies rated highest by the user, enabling the model to comprehend the user's strongest preferences. In the second step, we present an example consisting of an instruction, a candidate list containing both preferred and non-preferred movies, and the corresponding output. By constraining the output to include only preferred movies among a diverse set of contrasting options, the LLM effectively discerns detailed user preferences. In the third step, we employ a similar approach to Step 2, specifically targeting movies within the ambiguous scoring range of negative feedback as the candidate list for this step. We prompt the LLM to distinguish between falsely identified negative and confirmed negative in the challenging set, using the user preferences. See Figure 2 for detailed prompt. When prompted, the LLM identifies samples from the candidate list that exhibit relatively positive signals and classifies them as falsely identified negative. Items not selected as falsely identified negative are automatically assigned as confirmed negative. Since these falsely identified negative items are considered to contain positive aspects, we equate them with positive feedback.

Some users approach ratings generously, while others apply much stricter rating criteria; however, this variability has often

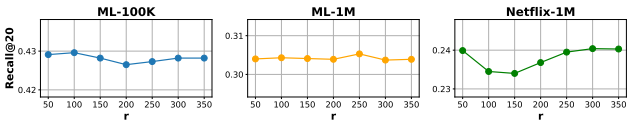

Figure 3: Recall@20 variation across three datasets as the hyperparameter r varies.

been overlooked in many studies. We acknowledge the significant variations in rating tendencies among different users by grouping them based on average ratings: generous raters, who rate 3 or above, and strict raters, who rate below 3. We tailor the composition of the candidate lists within prompts according to user rating tendencies. For generous users, ratings of 2 or lower are considered significantly low. Therefore, in Step 2, the candidate list includes non-preferred movies rated 1 or 2. In Step 3, the candidate list is composed solely of movies rated 3. For strict users, only movies rated 1 are included in the non-preferred movies for Step 2. Since their average ratings usually fall in the 2-rating range, we set ambiguous ratings to 2 and 3. Consequently, movies rated 2 and 3 comprise the candidate list for Step 3. The user in Figure 2 tends to give generous ratings. This approach significantly enhances the accuracy of negative feedback refinement.

2.2 Training with Re-Weighted BPR

Our approach to granularizing negative feedback is novel in this field, therefore requiring a tailored training methodology that can optimally leverage this perspective. Thus, we design a novel loss function and apply it to LightGCN [4], a representative GCNs model for recommender systems. LightGCN uses the following BPR loss as its loss function:

$$\mathcal{L}_{BPR} = \sum_{(u,i,j)\in O} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|\mathbf{E}^{(0)}\|^2. \tag{1}$$

Here, $\hat{y}_{ui}$ and $\hat{y}_{uj}$ represent the inner products of the embeddings between user u and item i and j, respectively. $\mathbf{E}^{(0)}$ denotes the embedding of the 0-th layer, and λ is the regularization weight. $O = \{(u,i,j) \mid (u,i) \in \mathcal{R}^+, (u,j) \in \mathcal{R}^-\}$ represents the pairwise training data. $\mathcal{R}^+$ consists of items that are positively rated by user u. However, BPR loss in this model treats all items as $\mathcal{R}^-$, the target set for negative sampling. For each positive item i, its corresponding negative item j is uniformly sampled from $\mathcal{R}^-$, regardless of user preferences, hence approach hinders accurate preference modeling. Therefore, in line with our refined feedback interpretation, we propose *Re-Weighted* BPR, assigning distinct sampling weight to items. Unlike the BPR loss that assigns uniform weights to all items, *Re-Weighted* BPR assigns a greater weight p to confirmed negative items and their similar counterparts. Conversely, for users' positive feedback and falsely identified negative items, we set the weight to 0. Moreover, for items similar to falsely identified negative and positive items—those unrated but with high potential user preference—*Re-Weighted* BPR allocate a lower weight q. Finally, the adjusted weights are converted to probability through softmax function in order to enable weighted negative sampling.

To facilitate learning, we select the top-r items that are unrated but similar to confirmed negative items, and the top-s items that are unrated but similar to falsely identified negative and positive items. To ensure the quality of similar items, we first train our model for t epochs until it becomes reliable, and then integrate these additional items into the training process. We set the hyperparameters as follows: p, q, t, and s are 1.5, 0.75, 50, and 5, respectively. r is configured to 100 for ML-100K, 250 for ML-1M, and 300 for Netflix-1M. Figure 3 presents the results of performance variations with different r values. As the number of items in the dataset increases, the optimal r also rises due to more items similar to confirmed negatives.

Table 1: Statistics of the datasets. Ratio denotes negative feedback ratio per positive feedback (pos:neg).

Dataset	# Users	# Items	# Ratings	Density	Ratio
ML-100K	938	1,008	95,215	0.1007	1:0.75
ML-1M	6,034	3,125	994,338	0.0527	1:0.73
Netflix-1M	4,803	5,575	929,436	0.0347	1:0.77

3 Experiments

3.1 Experimental Setup

Datasets. We experiment with three real-world datasets: ML-100K, ML-1M, and Netflix-1M. Each dataset contains 1–5 ratings, movie titles, and genres. Netflix-1M is a subset of the Netflix dataset, randomly sampled 5,000 users with a fixed seed. These datasets are preprocessed with a 5-core filter, where users must have rated at least 5 items positively, and items must have been rated positively by at least 5 users. For data splitting, positive feedback is divided into train:validation:test in a 7:1:2 ratio, while all negative feedback that is unable to serve as ground truth is included in the train set. Table 1 shows the statistics of each dataset.

Baselines and Metrics. We compare the performance of our model against the backbone model, LightGCN [4], as well as existing models designed to leverage negative feedback: SiReN [9], PANE-GNN [7], SiGRec [5], LSGRec [6], and SIGformer [1]. LightGCN serves as a standard framework in the category of GCN-based methods, known for its simplicity and strong performance. SiReN is the first graph-based recommendation model to utilize negative feedback. It trains negative feedback with a Multi-Layer Perceptron (MLP) and positive feedback with GCN, integrating the two through an attention mechanism. PANE-GNN leverages contrastive learning to train negative feedback, aiming to exclude items the user is likely to dislike from the recommendation list. SiGRec argues that negative feedback can be interpreted as positive feedback. This model proposes mapping the negative embeddings from positive embeddings using an MLP. LSGRec points out limitations in prior methods that train positive and negative feedback independently and proposes a method to model them jointly. SIGformer employs a transformer architecture to model negative feedback. We adopt two widely adopted evaluation metrics in recommender systems: Recall@K and NDCG@K.

Settings. We implement the model with PyTorch Geometric [3]. The batch size is set to 1024, embedding dimension to 64, the number of layers is 4, learning rate to 1e-3, and the training runs for 1000 epochs. The model is evaluated on the validation set at each epoch,

Table 2: Overall performance comparison with baseline models at K=20. RW refers to Re-Weighted BPR. All results are reproduced using the released codes from the authors.

Model	ML-100K		ML-1M		Netflix-1M	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
LightGCN SIGIR'20 [4]	0.3447	0.2775	0.2561	0.2584	0.1984	0.2128
SiReN IEEE TNNLS'22 [9]	0.4032	0.3397	0.2877	0.3019	0.2383	0.2674
PANE-GNN arXiv'23 [7]	0.3930	0.3294	0.2905	0.3058	0.2383	0.2671
SiGRec IP&M'23 [5]	0.4187	0.3633	0.2844	0.3153	0.2081	0.2590
LSGRec arXiv'24 [6]	0.3755	0.3054	0.2675	0.2758	0.2218	0.2440
SIGformer SIGIR'24 [1]	0.4090	0.3410	0.2719	0.2784	0.2161	0.2470
ReFINe w/o RW	0.4194	0.3506	0.2937	0.3013	0.2285	0.2541
ReFINe	**0.4296**	**0.3661**	**0.3053**	**0.3196**	**0.2404**	**0.2697**

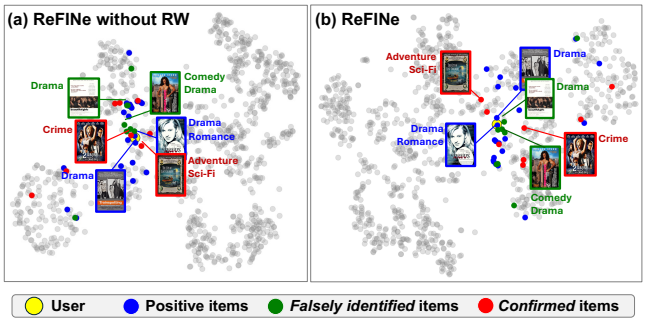

Figure 4: Visualization of *user_467* in the ML-100K dataset, comparing ReFINe without RW and with RW.

and early stopping is applied if no improvement in recall@20 is observed over 50 consecutive epochs. The experiments are conducted using a single NVIDIA RTX A6000 GPU.

3.2 Results

Table 2 shows the efficacy of our method, where ReFINe consistently outperforms the performances of baseline models across three benchmark datasets; for the baselines, we use the official codes to reproduce the results. Specifically, our method shows relatively higher performance improvements in the MovieLens (ML) datasets compared to Netflix. As reviewers in ML left their collected ratings once in a while not directly after each watching [2], it potentially contains noise in ratings. This noise exacerbates the limitation of using a fixed threshold for negative feedback, which inherently overlaps with the positive feedback regime. By effectively distinguishing samples that contain positive signals within the negative feedback, our approach reduces the impact of this noise. Thus, our method exhibits robustness even in challenging datasets with noise.

We further investigate the impact of *Re-Weighted* BPR (RW); see the last two rows in Table 2. RW adjusts the negative sampling probability based on whether an item is *falsely identified* negative, *confirmed* negative, or positive feedback. When RW is excluded from ReFINe, samples have uniform sampling probabilities regardless of negative intensity, leading to a consistent performance decline. This demonstrates that our redesigned objective effectively

Table 3: Comparison of ReFINe with other variants using fixed positive/negative split strategies. ReFINe always exhibits higher performance than the strategies with fixed thresholds. These results indicate the effectiveness of generated splits from LLM.

Evaluation metric	Varying Thresholds for Positives				ReFINe
	≥1	≥2	≥3	≥4	
Recall@20	0.4079	0.4194	0.4255	0.4096	**0.4296**
NDCG@20	0.3387	0.3473	0.3561	0.3446	**0.3661**

distances the confirmed negative samples from the positive regime, thereby enabling a more accurate representation of user embedding. We qualitatively analyze the effects of RW in the following section.

3.3 Representation Visualization

We use t-SNE and visualize the learned representations, showing a selected user and the items the user has rated. From Section 2.1, we remind the readers that the *falsely identified* negative items are resurrected and treated as positive feedback. In Figure 4 (a), falsely identified negative items are mapped near the user, sharing similar genres with positive items. However, *confirmed* negative items are embedded alongside the falsely identified and positive items. To mitigate this problem, we leverage our *Re-Weighted* BPR (RW), which adjusts probabilities in negative sampling to redefined feedback, proposed in Section 2.2. As a result, Figure 4 (b) shows that confirmed negative items are distinctly located further from the user. The results demonstrate the need to adjust negative sampling probabilities for negative feedback—*falsely identified* and *confirmed*—that exhibit different signals.

3.4 Comparison with Data Refinement Variants

Our approach leverages an LLM to identify *falsely identified* negatives based on users' preferences. However, one might question the LLM's effectiveness in discriminating items. To address this concern, we compare the performance of the LLM against rule-based scenarios where items satisfying specific thresholds are defined as

positive feedback. In Table 3, we observe that feedback classification based solely on rating values consistently results in inferior results compared to LLM-based refinement (ReFINe). Given the low likelihood of positive signals in the lowest-rated (rating of 1) items, using all feedback as positive feedback leads to the worst performance. In recommender systems, a threshold of 4 has traditionally been used to define the range of positive feedback. However, this conventional criterion leads to performance degradation, while setting the threshold at 3 and above yields the highest performance in rule-based scenarios. This indicates that potential positive signals may exist in the intermediate rating range. Thus, we implement the flexible threshold through LLM that predicts item preferences based on user history, offering a more fine-grained understanding than a fixed threshold. Consequently, our ReFINe method demonstrates superior performance, particularly excelling on the NDCG@20 metric, which represents the quality of ranking.

4 Conclusion

In this paper, we address the challenge posed by existing recommender systems, which simplify user ratings and thereby neglect the diversity inherent in negative feedback. To address this issue, we introduce **ReFINe**, which employs LLMs with user-specific prompts to accurately distinguish negative feedback: *Falsely identified* and *Confirmed* negative. From our new perspective, we redefine *falsely identified* negative items as positive feedback. To maximize learning effectiveness based on the refined data, we introduce *Re-Weighted* BPR loss, which adjusts the probabilities for negative sampling. Our method achieves superior performance across three real-world datasets.

Acknowledgments

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2025-RS-2024-00438056) and the Artificial Intelligence Graduate School Program of Chung-Ang Univ.(No. 2021-0-01341) both supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation).

References

[1] 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*. 1274–1284.

[2] Yu-chen Fan, Yitong Ji, Jie Zhang, and Aixin Sun. 2024. Our Model Achieves Excellent Performance on MovieLens: What Does it Mean? *ACM Transactions on Information Systems* 42, 6 (2024), 1–25.

[3] Matthias Fey and Jan E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

[4] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[5] 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 (2023), 103403.

[6] Yuting Liu, Yizhou Dang, Yuliang Liang, Qiang Liu, Guibing Guo, Jianzhe Zhao, and Xingwei Wang. 2024. Towards Unified Modeling for Positive and Negative Preferences in Sign-Aware Recommendation. *arXiv preprint arXiv:2403.08246* (2024).

[7] Ziyang Liu, Chaokun Wang, Jingcao Xu, Cheng Wu, Kai Zheng, Yang Song, Na Mou, and Kun Gai. 2023. PANE-GNN: Unifying Positive and Negative Edges in Graph Neural Networks for Recommendation. *arXiv preprint arXiv:2306.04095* (2023).

[8] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[9] 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 (2022), 4729–4743.

[10] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).