Retrieval-Augmented Recommendation Explanation Generation with Hierarchical Aggregation

Bangcheng Sun, Yazhe Chen, Jilin Yang, Xiaodong Li, Hui Li*
Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, Xiamen University, China
Email: {36920231153231, 23020241154376, 22920212204262}@stu.xmu.edu.cn, {xdli, hui}@xmu.edu.cn
*Corresponding author: hui@xmu.edu.cn

***Abstract*—Explainable Recommender System (ExRec) provides transparency to the recommendation process, increasing users' trust and boosting the operation of online services. With the rise of large language models (LLMs), whose extensive world knowledge and nuanced language understanding enable the generation of human-like, contextually grounded explanations, LLM-powered ExRec has gained great momentum. However, existing LLM-based ExRec models suffer from profile deviation and high retrieval overhead, hindering their deployment. To address these issues, we propose Retrieval-Augmented Recommendation Explanation Generation with Hierarchical Aggregation (REXHA). Specifically, we design a hierarchical aggregation based profiling module that comprehensively considers user and item review information, hierarchically summarizing and constructing holistic profiles. Furthermore, we introduce an efficient retrieval module using two types of pseudo-document queries to retrieve relevant reviews to enhance the generation of recommendation explanations, effectively reducing retrieval latency and improving the recall of relevant reviews. Extensive experiments demonstrate that our method outperforms existing approaches by up to 12.6% w.r.t. the explanation quality while achieving high retrieval efficiency.**

***Index Terms*—large language model, explainable recommendation, information retrieval**

I. Introduction

The exponential growth of digital content and products across online platforms (e.g., Yahoo News, Amazon and TikTok) has intensified the information overload problem [1], where users face overwhelming challenges in identifying relevant items amid vast information. To mitigate cognitive burdens in decision-making, Recommender System (RecSys) has emerged as an essential tool and is prevalently incorporated into many online services. RecSys effectively filters irrelevant information and delivers tailored recommendations. This way, it not only helps users better find their desired targets but also facilitates the operation of platforms. Hence, RecSys has attracted much attention and progressed actively over the past few decades [2], [3].

As users increasingly rely on recommendations for high-stakes decisions (e.g., financial investments and healthcare choices), mere predictive accuracy proves insufficient without explainable justification for recommendations This challenge has catalyzed the emergence of Explainable Recommender System (ExRec) [4]. ExRec explains the recommendation results to users and enhances the transparency of the decision-making process. Therefore, it increases users' trust in RecSys, further boosting the operation of online services.

Prior works on ExRec mainly study how to generate interpretable explanations for user-item interactions [5]. For instance, some works adopt deep learning techniques like RNNs [6], GNNs [7] and Transformer [8] for capturing intricate patterns in user behaviors and item attributes, enabling the generation of persuasive explanations. Although these methods are effective in some cases, they naturally suffer from limited language competence as they are trained over the restricted recommendation data. Hence, a branch of work on ExRec opts to use language models (e.g., BERT [9]) pre-trained on vast, diverse textual corpora to generate human-like explanations. This paradigm shift has gained momentum with the rise of large language models (LLMs), whose extensive world knowledge and nuanced language understanding enable the generation of human-like, contextually grounded explanations [10]–[12].

XRec is one representative LLM-based ExRec [11]. It enables LLMs to better understand complex user-item interaction patterns and offer explanations via integrating collaborative filtering (CF) signals. Building on this foundation, G-Refer [12] introduces a hybrid graph retrieval method to enhance ExRec by extracting both structural and semantic CF information from the user-item interaction graph. Despite these notable advancements, two critical limitations persist in LLM-powered ExRec: **(C1) Profile Deviation.** To assist with explanation generation, XRec and G-Refer construct user/item profiles by randomly sampling a small subset of user/item reviews, neglecting the broader contextual information embedded in the remaining data. This selective sampling introduces information bias, causing LLMs to generate explanations misaligned with holistic user preferences or item characteristics. **(C2) High Retrieval Overhead.** G-Refer employs the Dijkstra algorithm in path-level retrieval and can only be calculated on the CPU. It is not friendly to parallelism and has a high time complexity of $\mathcal{O}(N^2)$, resulting in a prohibitive retrieval cost.

To alleviate the limitations of existing ExRec works, we present Retrieval-Augmented Recommendation Explanation Generation with Hierarchical Aggregation (REXHA), a framework designed to improve the credibility and efficiency of recommendation explanation generation. The contributions of this work are summarized as follows:

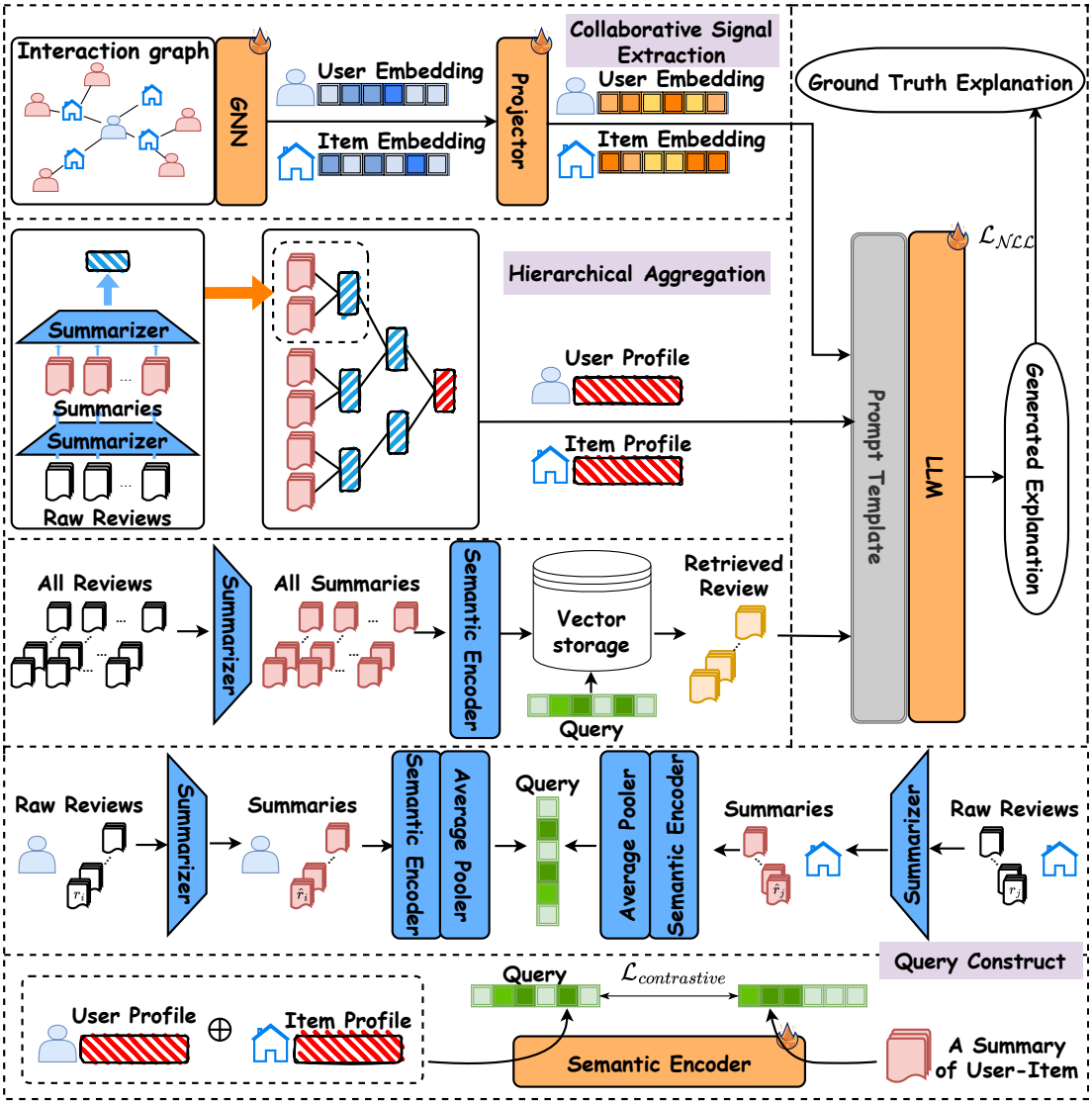

Fig. 1. Overview of REXHA. It contains three key components:(1) **Collaborative Signal Extraction** provides collaborative filtering information to LLMs. (2) **Hierarchical Aggregation** compresses and summarizes reviews layer by layer, finally construct textual profiles for user/item. (3) **Review Retrieval** module retrieves relative reviews to enhance LLM generating explanations.

- **Holistic User/Item Profiling:** Unlike prior methods (e.g., XRec and G-Refer) that construct incomplete profiles via random review sampling, we propose a hierarchical review aggregation module to systematically encode all user/item review data. This approach mitigates profile deviation (**C1**) by capturing nuanced user preferences and item characteristics through multi-layered summarization, ensuring LLMs generate explanations grounded in comprehensive contextual signals.
- **Efficient Retrieval via Pseudo-Document Queries:** To overcome the computational bottlenecks of retrieval (**C2**), we introduce aggregated pseudo-document query generation. By synthesizing latent queries from user/item reviews, REXHA efficiently retrieves semantically rich opinions from vectorized historical reviews, drastically reducing latency compared to prior LLM-powered ExRec.
- **Extensive and Rigorous Evaluation:** We have conducted extensive experiments on public datasets. The results demonstrate that REXHA achieves significant performance gains, outperforming a range of state-of-the-art baselines by up to 12.6% on recommendation explanation generation.

Our implementation is available at https://github.com/d8e-lab/REXHA.

II. OUR METHOD REXHA

Fig. 1 provides an overview of REXHA. The primary idea of REXHA is to efficiently retrieve and harness reviews relevant to the target user-item interaction to enhance the credibility and personalization of the generated explanation. To achieve this, REXHA uses three main modules, collaborative signal extraction (Sec. II-A), hierarchical aggregation based profil-

ing (Sec. II-B) and review retrieval for data augmentation (Sec. II-C), to prepare rich and beneficial background data for the target user-item interaction that is further fed into LLM for retrieval-augmented explanation generation (Sec. II-D).

A. Collaborative Signal Extraction

Collaborative information is crucial to model recommendation process [2]. Constructing a user-item interaction graph and then capturing interaction patterns from the perspective of graph semantics and structure is a prevalent method, providing insights into understanding user preferences and item characteristics [13]. Hence, in REXHA, we leverage the capability of Graph Convolutional Networks (GCNs) to encode collaborative information in the user-item interaction graph into latent embeddings, complementing later LLM-based explanation generation by incorporating collaborative signals in recommendation scenarios.

To be specific, we adopt LightGCN [14] to encode the collaborative information:

$$\mathbf{e}_u^{(l+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{(l)}, \tag{1}$$

$$\mathbf{e}_i^{(l+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(l)} \tag{2}$$

where $\mathbf{e}_u^{(l)}$ and $\mathbf{e}_i^{(l)}$ denote the embedding of user u and item i after l-layer propagation, respectively. $\mathcal{N}_u$ denotes the set of users who have interacted with item i, and $\mathcal{N}_i$ indicates the set of items interacted by user u.

The user and item embeddings are extracted by averaging their embeddings from all GCN layers:

$$\mathbf{e}_u = \sum_{l=0}^{L} \frac{1}{L+1} \mathbf{e}_u^{(l)}, \quad \mathbf{e}_i = \sum_{l=0}^{K} \frac{1}{L+1} \mathbf{e}_i^{(l)} \tag{3}$$

where L denotes the number of layers. We employ a three-layer feedforward neural network to user and item project embeddings to hidden embedding sapce of LLMs, which is trained with negative log-likelihood loss.

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C_i} y_{i,c} \log(\hat{y}_{i,c}). \tag{4}$$

Here N denotes the total number of user-item pairs to be explained, and C denotes the token count in each explanations. $y_{i,c}$ and $\hat{y}_{i,c}$ correspond to the ground-truth and predicted tokens at position c of i-th sample, respectively. Reviews written by users are regarded as ground-truth explanation.

B. Hierarchical Aggregation Based Profiling

Item textual attributes (e.g., item title, item category, etc) contain important features for capturing item characteristics. Nevertheless, as many items share similar textual attributes, there is insufficient information for constructing distinguishing profiles, making later explanation generation lack useful supporting inputs.

Instead, reviews written by humans reflect user preferences and item characteristics. Therefore, incorporating and summarizing fragmented human-written reviews using natural language as user and item profiles can enhance the richness and usefulness of the constructed profiles, providing better supporting inputs for LLM to generate persuasive and comprehensive recommendation explanations.

Existing LLM-based ExRec methods XRec and G-Refer utilize LLM as the summarizer. Since processing all the reviews of a popular item may exceed the context length of LLM, they randomly sample a few reviews for constructing profiles, losing quite a lot of information beneficial for generating explanations. Therefore, a flawed task profile was pieced together. Despite that LLMs with a longer context window can be applied for profile summarization, the lost-in-middle issue [15] still affects profile construction.

To conquer the above problems, we opt to maintain a relative fixed context length via hierarchical aggregating the reviews when merging the reviews. REXHA compresses and summarizes reviews layer by layer in parallel, and finally constructs a refined and informative profile.

1) Raw Review Summarization: Given a user-item interaction, we conduct hierarchical aggregation twice: one for user profile and the other for item profile. At the bottom of the hierarchical aggregation tree, each leaf contains one raw item review and each review is randomly placed in a leaf node. When constructing a user profile, the raw review comes from one item interacted by that user. When constructing an item profile, we use this item's reviews as raw reviews. As shown in Fig. 1, the first step is to aggregate reviews of two adjacent sibling nodes and use LLM to summarize the two raw reviews.

The primary motivation for employing LLM to summarize raw item reviews lies in addressing two critical challenges: excessive information redundancy and semantic ambiguity. By condensing user feedback into summaries, this approach establishes a robust semantic foundation that enhances the later hierarchical aggregation process.

Raw review data often suffers from verbosity, noise, and conflicting signals, such as redundant feature mentions (e.g., "camera quality"), irrelevant details, or contradictory sentiments ("loves the design but hates the price"). Directly aggregating unprocessed text risks diluting actionable insights or introducing logical inconsistencies, which undermines both analytical precision and contextual coherence. LLM excels at distilling these noisy inputs into concise, semantically dense representations. This process not only overcomes the limitation of context length but also extracts nuanced user preferences (e.g., "prioritizes camera performance over battery life") and item characteristics (e.g., "high-resolution sensor with limited battery capacity"). The resulting summaries act as high-fidelity intermediaries, filtering out extraneous details while preserving critical semantic relationships.

Based on the above motivation, we design the instruction prompt to order the LLM to preliminarily summarize each raw reviews. The prompt can be found in Appendix D. By combining the instruction prompt with a raw review r_i

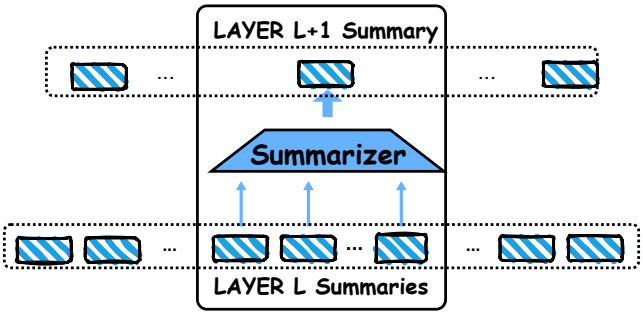

Fig. 2. Hierarchical aggregation unit. Multiple summaries from layer L are grouped and fed into a summarizer, which outputs a single summary at layer $L+1$. This unit is applied recursively.

as the input to LLM, we can leverage LLM to generate review summarization and similar processes are conducted in parallel for raw reviews. This is similar to a k-ary-tree merging operation, where one raw review takes part in only one summarization. Assume there are m raw reviews at the bottom level. Then, after raw review summarization, there are m/k brief but more information-dense summaries and they will be engaged in the next step of hierarchical aggregation.

2) Hierarchical Aggregation: When constructing an item profile for an item, after generating summaries from raw reviews, we concatenate every p summaries into a group and conduct summarization similar to the summarization operation at the bottom level. This process is conducted recursively until reaching the root level where we only have one summarization (i.e., the constructed item profile), forming a hierarchical aggregation tree as shown in Fig. 2. When integrated into the hierarchical aggregation process, the refined summaries from each level enable progressive fusion of granular features into the holistic profile. Similarly, a user profile can be constructed in a hierarchical manner.

C. Review Retrieval for Data Augmentation

1) Review Retrieval Augmentation: Hierarchical aggregation only utilizes reviews directly related to the target user/item to construct the user/item profile. In practice, users often refer to the reviews of other similar items or consider reviews made by people with similar preferences before making decisions. Such reviews may contain auxiliary and use information that may support their decisions.

Based on the above observation, REXHA leverages an embedding model f as the semantic encoder to transfer raw review summaries obtained in Sec. II-B1 into representation vectors. Then, for a target user-item interaction that requires explanation generation, REXHA uses a retrieval query to find the most relevant raw review summaries based on cosine similarity. The query construction method will be introduced in the next section. Finally, top-q relevant raw review summaries to the target user-item interaction are listed as data augmentation for generating explanations.

2) Retrieval Query Construction: Unlike general information retrieval scenarios, the recommendation system domain inherently lacks readily available natural language queries that can systematically retrieve review-based evidence for specific user-item interactions to enhance later explanation generation. In conventional recommendation frameworks, there exists no straightforward mechanism to translate implicit user preferences and item characteristics into search-compatible queries that directly enable contextual raw review summaries retrieval.

To address this problem, we propose two types of pseudo-document query construction strategies: the latent representation query and the profile query. The former leverages the global information of the user-item pair, aiming to retrieve diverse and informative reviews. The latter uses user and item profiles—constructed to emphasize high-frequency, long-term preference attributes—as queries. By retrieving based on these specialized profiles, the resulting reviews exhibit a more concentrated semantic distribution and are more closely aligned with the underlying preferences of the user or item.

a) Latent Representation Query: A latent representation query encodes the information of the target user-item interaction. Concretely, REXHA encodes all raw reviews of the target user and item using the embedding model f (semantic encoder) and then aggregates all the encoded representations as the latent query for the target user-item interaction:

$$\hat{r}_i^u = f(r_i^u), \hat{r}_j^v = f(r_j^v), \tag{5}$$

$$\hat{R}^u = \{\hat{r}_1^u, \hat{r}_2^u, \ldots, \hat{r}_n^u\}, \hat{R}^v = \{\hat{r}_1^v, \hat{r}_2^v, \ldots, \hat{r}_n^v\}, \tag{6}$$

$$\mathbf{Q}^u = f(\hat{R}^u), \mathbf{Q}^v = f(\hat{R}^v), \tag{7}$$

$$\mathbf{q}^u = \frac{1}{N} \sum_{q_i^u \in \mathbf{Q}^u} \mathbf{q}_i^u, \mathbf{q}^v = \frac{1}{N} \sum_{q_i^v \in \mathbf{Q}^v} \mathbf{q}_i^v, \tag{8}$$

$$\mathbf{q}_{latent} = \frac{\mathbf{q}^u + \mathbf{q}^v}{2}. \tag{9}$$

where r_i^u is the raw review text written by the user, and $\hat{r}_i^u$ denotes its corresponding summarized opinion. Let $\hat{R}^u$ be the set of all user opinions $\hat{r}^u$. After mapping $\hat{R}^u$ to higher-dimension space by embedding model f, we obtain $\mathbf{Q}^u$, the set of embedding $\mathbf{q}^u$. N denotes the number of opinions. The same applies to the item side.

b) Profile Query: Since user and item profiles constructed in Sec. II-B contain precise and comprehensive information of user preferences and item characteristics, we design the profile query that directly uses the textual descriptions in profiles of the target user/item as the retrieval query. Because user/item profiles misalign with raw review summaries due to filtering information during the hierarchical aggregation step. In such scenarios, fine-tuning the embedding model is required — a process distinct from the GCN training described in Sec. II-A.

$$\mathcal{L}_{\text{contrastive}} = \frac{e^{s_{i,j}/\tau}}{e^{s_{i,j}/\tau} + D}, \tag{10}$$

$$D = \sum_{(i',j') \neq (i,j)} e^{s_{i',j'}/\tau}, \tag{11}$$

$$s_{i,j} = \text{sim}(\mathbf{p}_{u_i,v_j}, \hat{\mathbf{r}}_{u_i,v_j}).$$

where $S(\cdot,\cdot) = \mathrm{sim}(\cdot,\cdot)$ indicates cosine similarity, $\mathbf{p}_{u,v}$ and $\mathbf{r}_{u,v}$ denote embeddings of profiles and summarized reviews of the user-item pair, respectively. The embedding $\mathbf{p}_{u,v}$ and $\mathbf{r}_{u,v}$ are computed as follows:

$$\mathbf{p}_{u,v} = f'(p_{u,v}), \mathbf{r}_{u,v} = f'(\hat{r}_{u,v}). \tag{12}$$

where f' denotes fine-tuned embedding model. $p_{u,v}$ and $\hat{r}_{u,v}$ denote the profile and summarized reviews of user-item pair (u, v).

D. Recommendation Explanation Generation

For a target user-item interaction, we concatenate the corresponding constructed user and item profiles (Sec. II-B), the extracted user and item embeddings (Sec. II-A) and the relevant reviews for data augmentation (Sec. II-C) and use the prompt template shown in Tab. V to instruct LLM to generate the recommendation explanation.

Training procedure. Our system is trained in a modular manner. The GCN is trained first using the NLL loss in (4) to learn collaborative user–item embeddings, with LLM parameters frozen. The retrieval encoder is trained separately using the contrastive loss in (10) to align hierarchical user/item profiles with summarized reviews. During explanation generation, pretrained embeddings, profiles, and retrieved reviews are provided to the LLM without further training.

III. Experiments

A. Experimental Settings

1) Datasets: We conduct experiments on three public datasets from different domains: **Yelp**[1], **Amazon-books**[2], and **Google-reviews** [16], [17]. More details of datasets can be found in Tab. IV in Appendix A.

2) Evaluation Metrics: For evaluating the semantic explainability and stability of the generated explanation, we follow XRec [11] and G-Refer [12] to employ a suite of metrics that assessing the semantic explainability of generated explanations. Concretely, we use $\mathrm{GPT}_{\mathrm{score}}$ [18], $\mathrm{BERT}_{\mathrm{score}}$ [19] to measure the explainability, and reviews written by the users are regarded as ground-truth explanations. The details of metric can be found in Appendix B.

3) Baselines: We compare our method with the following state-of-the-art methods, including NRT [6], Att2Seq [20], PETER [21], PEPLER [22], XRec [11] and G-Refer [12]. The details of baseline can be found in Appendix C.

4) Implementation Details: For the collaborative signal extractor module, we train LigntGCN with a learning rate of 1e-3 and a batch size of 1024. For hierarchical aggregation (HA) module, we set 4 reviews as a set to be summarized. For the retrieval module, we use llm-embedder model [3] to generate the embeddings of the reviews. For the generation module, we use the LLaMA-2-7B model as the base model. We set the learning rate, epochs, and batch size as 8e-4, 2 and 12, respectively. For inference, we set the temperature to 0 and the max output tokens as 128.

[1]https://www.yelp.com/dataset/challenge

[2]https://jmcauley.ucsd.edu/data/amazon/

[3]https://huggingface.co/BAAI/llm-embedder

B. Overall Performance

Tab. I provides the overall results. It can be observed that REXHA achieves outstanding performance in explanation quality across evaluation metrics based on GPT and BERT. XRec and G-Refer take different approaches: the former leverages graph neural networks to capture collaborative filtering information and integrates it into large models, while the latter further exploits graph-structured data using a hybrid retrieval method that combines node-level and path-level strategies to more precisely utilize user-item interaction graphs. In contrast, REXHA-L, which uses latent representation query, focuses on mining the semantic information in reviews, and achieves notable improvements on the $\mathrm{BERT}^{\mathrm{Precision}}_{\mathrm{score}}$ metrics across three datasets with increases of 12.6%, 26.8% and 7.46%. Meanwhile, comparing with G-Refer on $\mathrm{BERT}^{\mathrm{F1}}_{\mathrm{score}}$, REXHA achieves increases of 2.76%, 9.59% and 0.15%. We observe that REXHA-P with profile query achieves higher GPT scores compared to REXHA-L across all three datasets, with improvements of 2.26%, 2.60%, and 0.64% on Amazon-books, Yelp, and Google-reviews, respectively. However, it performs slightly worse in terms of BERT scores.

TABLE I

OVERALL PERFORMANCE. SUPERSCRIPS "P", "R" AND "F1" RESPECTIVELY DENOTE PRECISION, RECALL, F1-SCORE. NUMBERS IN BOLD INDICATE THE BEST RESULTS, WHILE UNDERLINED NUMBERS MEAN THE SECOND-BEST RESULTS. "REXHA-P" AND "REXHA-L" CORRESPOND TO REXHA WITH LATENT REPRESENTATION QUERIES AND PROFILE QUERIES, RESPECTIVELY.

Models	Explainability↑			
	$\mathrm{GPT}_{\mathrm{score}}$	$\mathrm{BERT}^{\mathrm{P}}_{\mathrm{score}}$	$\mathrm{BERT}^{\mathrm{R}}_{\mathrm{score}}$	$\mathrm{BERT}^{\mathrm{F1}}_{\mathrm{score}}$
Amazon-books				
NRT	75.63	0.3444	0.3440	0.3443
Att2Seq	76.08	0.3746	0.3624	0.3687
PETER	77.65	0.4279	0.3799	0.4043
PEPLER	78.77	0.3506	0.3569	0.3543
XRec	82.57	0.4193	0.4038	0.4122
G-Refer (7B)	82.70	0.4076	**0.4476**	0.4282
REXHA-P	**83.28**	0.3995	0.3752	0.3881
REXHA-L	81.44	**0.4722**	0.4070	**0.4400**
Yelp				
NRT	61.94	0.0795	0.2225	0.1495
Att2Seq	63.91	0.2099	0.2658	0.2379
PETER	67.00	0.2102	0.2983	0.2513
PEPLER	67.54	0.2920	0.3183	0.3052
XRec	74.53	0.3946	0.3506	0.3730
G-Refer (7B)	74.91	0.3573	**0.4264**	0.3922
REXHA-P	**76.25**	0.4879	0.3604	0.4237
REXHA-L	74.32	**0.5005**	0.3603	**0.4298**
Google-reviews				
NRT	58.27	0.3509	0.3495	0.3496
Att2Seq	61.31	0.3619	0.3653	0.3636
PETER	65.16	0.3892	0.3905	0.3881
PEPLER	61.58	0.3373	0.3711	0.3546
XRec	69.12	0.4546	0.4069	0.4311
G-Refer (7B)	**71.47**	0.4253	**0.4873**	0.4566
REXHA-P	70.35	0.4565	0.4200	0.4385
REXHA-L	69.91	**0.4884**	0.4259	**0.4573**

TABLE II

ABLATION STUDY. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD AND THE WORST IN RED. "RR" REPRESENTS THE RETRIEVAL MODULE, AND "HA" IS THE HIERARCHICAL AGGREGATION MODULE. "RANDOM" INDICATES THAT USER/ITEM PROFILES ARE GENERATED BY RANDOMLY SAMPLING THE USER/ITEM REVIEWS.

Datasets		Amazon-books↑		Yelp↑	
RR	HA	$\text{BERT}^{\text{Precision}}_{\text{score}}$	$\text{BERT}^{\text{F1}}_{\text{score}}$	$\text{BERT}^{\text{Precision}}_{\text{score}}$	$\text{BERT}^{\text{F1}}_{\text{score}}$
w/o RR	Random	0.4570	0.4319	0.4860	0.4192
w/o RR	w/ HA	0.4563	0.4347	0.4930	0.4250
w/ RR	Random	0.4417	0.4085	0.4791	0.4191
REXHA		**0.4726**	**0.4403**	**0.5031**	**0.4310**

TABLE III

CONTRAST STUDY FOR HIERARCHICAL AGGREGATION MODULE. WE EVALUATE THE PERFORMANCE WITHOUT REVIEW RETRIEVAL MODULE. IN THIS TABLE, "DIRECTLY" DENOTES THAT FOR ALL REVIEWS OF USER/ITEM, LLMS ARE INVOKED ONCE TO GENERATE PROFILES, AND "SECOND LAYER" DENOTES THAT SUB-NODES OF THE FINAL NODE ARE USED AS PROFILES.

Models	Type	$\text{BERT}^{\text{P}}_{\text{score}}$	$\text{BERT}^{\text{R}}_{\text{score}}$	$\text{BERT}^{\text{F1}}_{\text{score}}$
Qwen2.5-7B-1M	HA	0.4712	0.3576	0.4142
Qwen2.5-7B-1M	Directly	0.4639	0.3570	0.4100
Qwen2.5-7B-1M	Second Layer	0.4716	0.3557	0.4134
Qwen2.5-7B	Directly	—	—	—
Qwen2.5-7B	HA	**0.4930**	**0.3580**	**0.4250**

C. Ablation Study

We provide the ablation study in Tab. II. It is observed that the hierarchical aggregation module improves the performance, which proves that the profile generated by this module exactly extracts the important information from user/item reviews, benefiting the generation step. Rarely using review retrieval module without the hierarchical aggregation module get the worst results. However, when two modules are combined, the best results are obtained. This observation indicates that the deviation of profiles introduces noise and makes LLM confused when faced with the contradiction between the profile and the retrieved reviews. In contrast, with the combination of the two modules, LLM can better utilize the retrieved reviews according to correct profiles. By integrating these two modules, we achieve a compounded benefit that exceeds what each could offer alone.

D. Analysis of Retrieval Efficiency

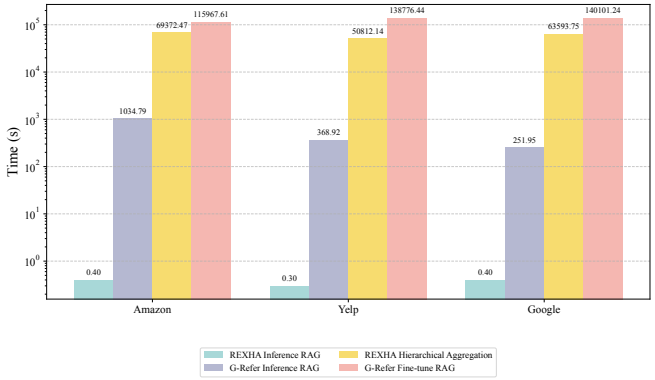

Fig. 3. Comparisons of efficiency between REXHA and G-Refer.

We evaluate the preprocessing time required by REXHA and G-Refer on three datasets. Across all three datasets, the retrieval time during the inference stage for REXHA is consistently under 1 second, significantly faster than G-Refer, which requires over 4 minutes. This highlights the superior inference efficiency of REXHA. Unlike G-Refer, REXHA does not perform any retrieval during the training phase. However, the profile preprocessing step in REXHA involves the Hierarchical Aggregation (HA) module, which incurs a relatively high computational cost—taking up to 20 hours. Despite this, it is still considerably more efficient compared to G-Refer, whose total retrieval time during training exceeds 40 hours across all three datasets. These results demonstrate that the HA module, while computationally intensive, remains more efficient than G-Refer's training-time retrieval operations.

E. Analysis of Hierarchical Aggregation

To validate the effectiveness of the Hierarchical Aggregation (HA) method, we compare it with the "Direct" method, where the LLM is invoked only once to generate the profile based on all reviews at once. We also compare the performance of Qwen2.5-7B with Qwen2.5-7B-1M, which supports a 1M-token context length.

Tab. III reports the results. We observe that Qwen2.5-7B fails to generate profiles using the Direct method since the input exceeds its context length. In contrast, the HA method outperforms the Direct method, achieving improvements of 1.57%, 0.17%, and 1.02% on $\text{BERT}^{\text{P}}_{\text{score}}$, $\text{BERT}^{\text{R}}_{\text{score}}$, and $\text{BERT}^{\text{F1}}_{\text{score}}$, respectively. Additionally, we evaluate an alternative approach in which second-layer summaries are directly concatenated to form the profiles. This method achieves slightly higher BERT precision than HA (an increase of 0.08%), but results in a slight drop in recall (a decrease of 0.19%).

IV. RELATED WORK

A. Explainable Recommender Systems (ExRec)

ExRec aims to enhance user trust by providing recommendation rationales [4], [5]. Prior works categorize into generation-based [6], [8], [23], extraction-based [7], [9], and hybrid approaches [24], [25]. Recently, LLM-based methods have dominated the field due to their superior language capabilities. Approaches like PEPLER [22] and POD [26] employ prompt engineering, while recent models like XRec [11] and G-Refer [12] further improve performance by integrating collaborative filtering signals and structured retrieval into the generation process.

B. Retrieval-Augmented Generation (RAG)

RAG enhances generative models by integrating retrieved external knowledge. Standard approaches like In-Context RALM [27] directly concatenate retrieved context with prompts. Recent research optimizes this interaction: SKR [28] balances internal versus external knowledge usage, HyDE [29]

utilizes hypothetical document embeddings for better retrieval, and RichRAG [30] synthesizes multi-aspect information to construct richer inputs for the LLM.

V. Conclusion and Limitations

We analyze the limitations of existing explainable recommendation methods in terms of profile construction and review retrieval, and propose REXHA to address these challenges. REXHA introduces a hierarchical aggregation module to generate concise yet informative user and item profiles, together with two query construction strategies that capture both global and fine-grained characteristics for effective review retrieval. Extensive experimental results demonstrate that REXHA significantly improves explanation quality and retrieval performance, indicating its strong potential for real-world applications.

However, this work has several limitations. First, human evaluation of explanation interpretability is not included due to the high cost of recruiting qualified annotators and the lack of standardized evaluation protocols. We plan to conduct systematic human evaluation in future work. Second, while effective, the current retrieval module does not fully exploit the retrieved reviews, as deeper post-processing such as fine-grained information extraction is not yet explored. Future work will focus on enhancing the utilization of retrieved reviews to further improve explanation quality.

Acknowledgment

This work is supported by the National Key Research and Development Program of China (No. 2025YFE0113500), the National Science Fund for Distinguished Young Scholars (No. 62525605), and the National Natural Science Foundation of China (No. 62572410, No. 62272401, and No. U22B2051).

References

[1] A. Borchers, J. L. Herlocker, and J. Riedl, "Ganging up on information overload," *Computer*, vol. 31, no. 4, pp. 106–108, 1998.

[2] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang, "A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4425–4445, 2023.

[3] F. Ricci, L. Rokach, and B. Shapira, Eds., *Recommender Systems Handbook*. Springer US, 2022.

[4] Y. Zhang and X. Chen, "Explainable recommendation: A survey and new perspectives," *Found. Trends Inf. Retr.*, vol. 14, no. 1, pp. 1–101, 2020.

[5] A. Ariza-Casabona, L. Boratto, and M. Salamó, "A comparative analysis of text-based explainable recommender systems," in *RecSys*, 2024, pp. 105–115.

[6] P. Li, Z. Wang, Z. Ren, L. Bing, and W. Lam, "Neural rating regression with abstractive tips generation for recommendation," in *SIGIR*, 2017, pp. 345–354.

[7] P. Wang, R. Cai, and H. Wang, "Graph-based extractive explainer for recommendations," in *WWW*, 2022, pp. 2163–2171.

[8] L. Li, Y. Zhang, and L. Chen, "Personalized transformer for explainable recommendation," in *ACL/IJCNLP*, 2021, pp. 4947–4957.

[9] R. A. Pugoy and H. Kao, "Unsupervised extractive summarization-based representations for accurate and explainable collaborative filtering," in *ACL/IJCNLP*, 2021, pp. 2981–2990.

[10] Y. Lei, J. Lian, J. Yao, X. Huang, D. Lian, and X. Xie, "Recexplainer: Aligning large language models for explaining recommendation models," in *KDD*, 2024, pp. 1530–1541.

[11] Q. Ma, X. Ren, and C. Huang, "Xrec: Large language models for explainable recommendation," in *EMNLP (Findings)*, 2024, pp. 391–402.

[12] Y. Li, X. Zhang, L. Luo, H. Chang, Y. Ren, I. King, and J. Li, "G-refer: Graph retrieval-augmented large language model for explainable recommendation," in *WWW*, 2025, pp. 240–251.

[13] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: A survey," *ACM Comput. Surv.*, vol. 55, no. 5, pp. 97:1–97:37, 2023.

[14] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *SIGIR*, 2020, pp. 639–648.

[15] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *Trans. Assoc. Comput. Linguistics*, vol. 12, pp. 157–173, 2024.

[16] J. Li, J. Shang, and J. J. McAuley, "Uctopic: Unsupervised contrastive learning for phrase representations and topic mining," in *ACL*, 2022, pp. 6159–6169.

[17] A. Yan, Z. He, J. Li, T. Zhang, and J. J. McAuley, "Personalized showcases: Generating multi-modal explanations for recommendations," in *SIGIR*, 2023, pp. 2251–2255.

[18] J. Wang, Y. Liang, F. Meng, H. Shi, Z. Li, J. Xu, J. Qu, and J. Zhou, "Is chatgpt a good NLG evaluator? A preliminary study," *arXiv Preprint*, 2023. [Online]. Available: https://arxiv.org/abs/2303.04048

[19] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with BERT," in *ICLR*, 2020.

[20] L. Dong, S. Huang, F. Wei, M. Lapata, M. Zhou, and K. Xu, "Learning to generate product reviews from attributes," in *EACL*, 2017, pp. 623–632.

[21] L. Li, Y. Zhang, and L. Chen, "Personalized transformer for explainable recommendation," in *ACL/IJCNLP*, 2021, pp. 4947–4957.

[22] ——, "Personalized prompt learning for explainable recommendation," *ACM Trans. Inf. Syst.*, vol. 41, no. 4, pp. 103:1–103:26, 2023.

[23] A. Ariza-Casabona, M. Salamó, L. Boratto, and G. Fenu, "Towards self-explaining sequence-aware recommendation," in *RecSys*, 2023, pp. 904–911.

[24] H. Cheng, S. Wang, W. Lu, W. Zhang, M. Zhou, K. Lu, and H. Liao, "Explainable recommendation with personalized review retrieval and aspect learning," in *ACL*, 2023, pp. 51–64.

[25] H. Zhan, L. Li, S. Li, W. Liu, M. Gupta, and A. C. Kot, "Towards explainable recommendation via bert-guided explanation generator," in *ICASSP*, 2023, pp. 1–5.

[26] L. Li, Y. Zhang, and L. Chen, "Prompt distillation for efficient llm-based recommendation," in *CIKM*, 2023, pp. 1348–1357.

[27] O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham, "In-context retrieval-augmented language models," *Trans. Assoc. Comput. Linguistics*, vol. 11, pp. 1316–1331, 2023.

[28] Y. Wang, P. Li, M. Sun, and Y. Liu, "Self-knowledge guided retrieval augmentation for large language models," in *EMNLP (Findings)*, 2023, pp. 10 303–10 315.

[29] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," in *ACL*, 2023, pp. 1762–1777.

[30] S. Wang, X. Yu, M. Wang, W. Chen, Y. Zhu, and Z. Dou, "Richrag: Crafting rich responses for multi-faceted queries in retrieval-augmented generation," in *COLING*, 2025, pp. 11 317–11 333.

APPENDIX

A. Datasets

We use three public datasets in the experiments and Tab. IV provides the data statistics.

- **Amazon-books** comes from the Amazon book platform and contains user reviews, ratings, metadata, and other information on books. It can reflect user reading preferences and product popularity.
- **Yelp** comes from the American review website Yelp, where local businesses (such as restaurants and bars) are considered as items. The dataset provides rich information about local businesses, including user review text and ratings under multiple categories.
- **Google-reviews** comes from user reviews of businesses (such as restaurants, stores, attractions, etc.) on Google, including ratings, review text and timestamps, as well as business metadata.

B. Metrics

- $\mathbf{GPT_{score}}$ [18] utilizes LLMs to evaluate text quality. We regard GPT-3.5-Turbo ChatGPT as a human evaluator and give task-specific (e.g., summarization) and aspect-specific (e.g., relevance) instruction to prompt ChatGPT to evaluate the generated results of NLG models.
- $\mathbf{BART_{score}}$ [19] assesses the similarity between each token in the text by using the contextual embeddings generated by the pre-trained BERT model and calculating the sum of the cosine similarities of the token embeddings between two sentences.
  Given a reference sentence $x = \langle x_1, x_2, \ldots, x_n \rangle$ and a candidate sentence $\hat{x} = \langle \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_m \rangle$. Bert computes word embeddings sequence for the reference sentence and the candidate sentence as follows:

$$\begin{aligned} \mathbf{x} &= \langle \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \rangle \\ &= \mathrm{BERT}(x = \langle x_1, x_2, \ldots, x_n \rangle) \end{aligned} \tag{13}$$

$$\begin{aligned} \hat{\mathbf{x}} &= \langle \hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_n \rangle \\ &= \mathrm{BERT}(x = \langle \hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n \rangle) \end{aligned} \tag{14}$$

  The cosine similarity of a reference token x_i and a candidate token $\hat{x}_j$ is $\frac{\mathbf{x}_i^\top \hat{\mathbf{x}}_j}{\|\mathbf{x}_i\| \|\hat{\mathbf{x}}_j\|}$. As both embeddings are pre-normalized, the formula is reduced to the inner product $\mathbf{x}_i^\top \hat{\mathbf{x}}_j$. Then the $\mathrm{BERT}_{\mathrm{score}}^{\mathrm{Precision}}$, $\mathrm{BERT}_{\mathrm{score}}^{\mathrm{Recall}}$ and $\mathrm{BERT}_{\mathrm{score}}^{\mathrm{F1}}$ are computed as follows:

$$\mathrm{BERT}_{\mathrm{score}}^{\mathrm{Precision}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j, \tag{15}$$

$$\mathrm{BERT}_{\mathrm{score}}^{\mathrm{Recall}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j, \tag{16}$$

$$\mathrm{BERT}_{\mathrm{score}}^{\mathrm{F1}} = 2 \frac{P_{\mathrm{BERT}} \cdot R_{\mathrm{BERT}}}{P_{\mathrm{BERT}} + R_{\mathrm{BERT}}}. \tag{17}$$

TABLE IV
STATISTICS OF THE EXPERIMENTAL DATASETS

Datasets	#Users	#Items	#Interactions
Amazon	15,349	15,247	360,839
Yelp	15,942	14,085	393,680
Google	22,582	16,557	411,840

C. Baselines

- **NRT** [6]: A multi-task framework utilizing a joint objective for rating prediction and a GRU for abstractive tip generation.
- **Att2Seq** [20]: An attribute-to-sequence model employing attention mechanisms and stacked LSTMs to generate reviews from input attributes.
- **PETER** [21]: A Transformer-based approach that personalizes generation by injecting user/item ID embeddings into the decoding process.
- **PEPLER** [22]: Applies continuous prompt learning to pretrained language models (e.g., GPT-2) for generating personalized explanations.
- **XRec** [11]: Enhances LLMs by injecting collaborative signals from GNN-based encoders into the language model layers via a lightweight adaptor.
- **G-Refer** [12]: A retrieval-augmented method that translates graph-based CF signals into text to guide LLMs in generating explanations.

D. Prompt Template

The generation process employs a structured prompt, as illustrated in Tab. V, combining multiple data components such as retrieved reviews, user/item embeddings, and user/item metadata. The prompt is tokenized and transformed into an embedding space representation. To distinguish special tokens within the prompt as distinct entities, we integrate them into the LLM's tokenizer. These tokens are subsequently replaced with their corresponding modified embeddings in the final token embedding representation.

TABLE V
PROMPT FOR GENERATING EXPLANATION.

System Prompt:
Explain why the user would buy with the book within 50 words.
User prompt:
Here are some comments from similar users and items,
you can use them to help you write the review.
1. The user would enjoy the business because of the ...
2. This business would be enjoyed by the user ...
...
user record: [USER_EMBED] item record: [ITEM_EMBED]
item name: ... user profile: ... item profile: ...