Enhancing Recommendation Explanations through User-Centric Refinement

Jingsen Zhang[1]*, Zihang Tian[1]*, Xueyang Feng[1], Xu Chen[1]†, Chong Chen[2]
[1]Renmin University of China, Beijing, China
[2]Huawei Cloud BU, Beijing, China
{zhangjingsen, zihangtian, xu.chen}@ruc.edu.cn

Abstract

Generating natural language explanations for recommendations has become increasingly important in recommender systems. Traditional approaches typically treat user reviews as ground truth for explanations and focus on improving review prediction accuracy by designing various model architectures. However, due to limitations in data scale and model capability, these explanations often fail to meet key user-centric aspects such as factuality, personalization, and sentiment coherence, significantly reducing their overall helpfulness to users. In this paper, we propose a novel paradigm that refines initial explanations generated by existing explainable recommender models during the inference stage to enhance their quality across multiple aspects. Specifically, we introduce a multi-agent collaborative refinement framework based on large language models. To ensure alignment between the refinement process and user demands, we employ a plan-then-refine pattern to perform targeted modifications. To enable continuous improvements, we design a hierarchical reflection mechanism that provides feedback from both strategic and content perspectives. Extensive experiments on three public datasets demonstrate the effectiveness of our framework.

1 Introduction

Natural language explainable recommendation (NLER) aims to generate textual explanations that clarify why an item is recommended (or not), offering high flexibility and user interpretability (Zhang et al., 2020). In this field, researchers typically treat user reviews as ground-truth explanations and focus on developing advanced architectures to improve review prediction accuracy. For instance, NRT (Li et al., 2017) incorporates predicted ratings into the explanation generation process. PETER (Li et al., 2021) bridges IDs and texts using Transformer to produce explanations. PEPLER (Li et al., 2023) leverages prompt learning to further enhance the explanation quality.

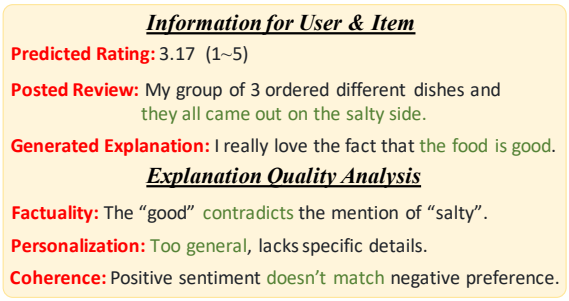

Figure 1: Illustration of inadequate user-centric quality in explanations produced by the PETER model.

While leveraging reviews can help explanations partially capture user preference, limitations in data scale and model capability often hinder the overall helpfulness of these explanations, resulting in deficiencies in key user-centric aspects. To effectively support user decision-making, it is essential for explanations to align with user demands on various aspects. For example, factuality ensures that the content is correct and verifiable, personalization requires explanations to highlight specific item features and user characteristics, and coherence demands alignment between the explanation's sentiment and the system-predicted user preference. Figure 1 illustrates these shortcomings, underscoring the importance of generating explanations that are both accurate and genuinely helpful.

Inspired by Large Reasoning Models, which enhance reasoning ability through stepwise thinking and strategic modification during inference, we propose a novel paradigm that refines initial explanations produced by existing explainable recommendation models, yielding targeted improvements across multiple user-centric aspects. Analogous to the Reranking process in recommender systems (Pei et al., 2019; Gao et al., 2024), our paradigm enhances performance in a post-hoc man-

*Equal Contribution.
†Corresponding Author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 8177–8191
November 4-9, 2025

ner, which we refer to as Refinement. This represents the first attempt in the field of explainable recommendation to modify explanations before presenting them to users. We perform this refinement by employing large language models (LLMs) with carefully designed instructions.

While this idea is promising, it poses several challenges. First, user demands for explanations are often multifaceted, with varying priorities across different aspects (Zhang et al., 2024e; Rahdari et al., 2024). Thus, the refinement process must effectively align with these diverse preferences and enable targeted improvements. Second, most existing explainable recommender models (Cao and Wang, 2021; Cheng et al., 2023; Raczyński et al., 2023) generate explanations in a single attempt, without assessing whether they satisfy user needs, often resulting in suboptimal outcomes. Therefore, it is crucial to incorporate feedback into the refinement process, allowing explanations to evolve iteratively from weak to strong and achieve continuous improvements.

To address these challenges, we propose an LLM-based multi-agent collaborative framework for explanation refinement. Specifically, it adopts a plan-then-refine pattern for targeted modifications guided by user demands, where the Planner first identifies which aspect should be refined at each round, and then the Refiner modifies the explanations according to the corresponding instructions. Additionally, to ensure continuous improvement, we design a hierarchical reflection mechanism, where the Reflector provides timely feedback and suggestions by analyzing the refinement process from both strategic and content perspectives. Furthermore, to support this process, we maintain an aspect library containing essential information about various aspects. By combining forward refinement and backward reflection phases, our framework achieves self-evolving and iterative enhancement until user demands are satisfied.

Our key contribution is to enhance recommendation explanation quality across multiple user-centric aspects, which is the first to achieve this. Specifically, (1) We identify key limitations of existing explainable recommender models in user-centric aspects and propose a novel paradigm to perform targeted Refinement to eXplanations (called **RefineX**). (2) We design an LLM-based multi-agent refinement framework, which employs a plan-then-refine pattern to align with user demands and incorporates a hierarchical reflection mechanism for continuous improvement. (3) Extensive experiments demonstrate our framework's effectiveness in enhancing explanation quality and high adaptability to diverse user demands.

2 Preliminary

Natural Language Explainable Recommendation (NLER) aims to provide textual explanations that clarify why an item is recommended (or not). Formally, given a user set $\mathcal{U}$ and an item set $\mathcal{I}$, their interactions are recorded in $\mathcal{D} = \{u, i, r_{u,i}, \boldsymbol{s}_{u,i} | u \in \mathcal{U}, i \in \mathcal{I}\}$, where $r_{u,i} \in [1, 5]$ denotes the user's rating, and $\boldsymbol{s}_{u,i}$ is the corresponding review. Given a user-item pair (u, i), NLER predicts the rating $\hat{r}_{ui}$, indicating user preference, and generates a textual explanation $\boldsymbol{e}_{ui}$ to justify the recommendation. Since rating prediction has been well studied, this work focuses on explanation generation, where user reviews are commonly treated as ground-truth explanations, and prior work mainly develops advanced architectures to improve review prediction accuracy (Li et al., 2021; Cheng et al., 2023; Zhou et al., 2024a). Despite these advances, such approaches often fall short in capturing key user-centric aspects, limiting the practical utility of the explanations.

Task Formulation. In real-world scenarios, user goals (or demands) for recommendation explanations are multifaceted and centered on practical helpfulness (Zhang et al., 2024e; Rahdari et al., 2024). Given a user-item pair (u, i), let $G_{ui} = \{a_1, a_2, ..., a_n\}$ denote the user goal for explanations, where each $a \in G_{ui}$ represents a specific aspect of user concern. Guided by G_{ui}, our task is to iteratively refine the initial explanation $\boldsymbol{e}^0_{ui}$ generated by existing models over t rounds to obtain the final explanation $\boldsymbol{e}^t_{ui}$, with the objective:

$$Q_a(\boldsymbol{e}^t_{ui}) > Q_a(\boldsymbol{e}^0_{ui}), \forall a \in G_{ui}, \quad (1)$$

where Q_a is an evaluation function measuring explanation quality on the aspect a. The key challenge lies in aligning the refinement process with the user goal and incorporating effective feedback to achieve continuous improvement.

3 Approach

3.1 Framework Overview

The overall framework is shown in Figure 2. Traditional NLER models directly provide users with explanations without feedback on whether they

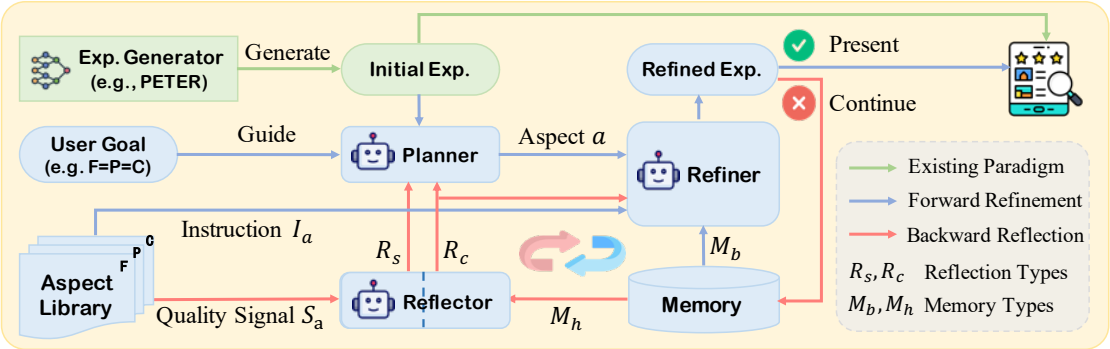

Figure 2: The overview of our RefineX framework, where "Exp" denotes "Explanation".

meet user goals. Our framework addresses this gap by refining generated explanations based on user goals before presenting them to users. Specifically, the entire process consists of a forward refinement phase and a backward reflection phase. In the refinement phase, we adopt a plan-then-refine pattern, where the Planner first identifies the aspects needing refinement, then the Refiner performs targeted modifications. In the reflection phase, the Reflector analyzes previous behaviors from both strategic and content perspectives and provides feedback for improvement. To support this process, we construct the Memory module and Aspect Library to maintain essential information relevant to refinement. We detail these components below.

3.2 The Refinement Phase

LLMs have demonstrated powerful planning capabilities (Huang et al., 2022a; Wang et al., 2023d; Huang et al., 2022b) in handling complex tasks. To align with users' multifaceted goals and perform targeted refinements, we adopt a plan-then-refine pattern for each round of the refinement. This phase involves the following components:

Planner plays a crucial role in controlling the refinement process, determining which aspect to refine in each round or terminating the process when the explanation meets user goals. We implement the Planner's behavior using the ReAct (Yao et al., 2022) framework, which employs a thought-action-observation pattern for task solving. Specifically, for round t, the Planner first determines which aspect a^t of the previous explanation $\boldsymbol{e}^{t-1}$ to refine by analyzing the user goal G, refinement trajectory $\boldsymbol{T}^{1:t}$, and reflections from the previous round $t-1$ at the strategic level R_s^{t-1} and content level R_c^{t-1} (details in Section 3.3). The Planner's thought function is formulated as:

$$a^t = \text{Planner}(\boldsymbol{e}^{t-1}, G, \boldsymbol{T}^{1:t-1}, R_s^{t-1}, R_c^{t-1}), \quad (2)$$

where $\boldsymbol{T}^{1:t} = [a^1, a^2, ..., a^t]$ represents the trajectory of refined aspects, and the subscript "ui" for the specific user-item pair (u, i) is omitted for clarity. Once the refined aspect is determined, the Planner takes action to call the corresponding information acquisition functions organized in the aspect library $\mathcal{A}$ (details in Appendix A.5), and the obtained aspect instructions for the sample (u, i) are conveyed to the Refiner for targeted refinement.

Refiner is the dedicated agent responsible for refining the previous explanation $\boldsymbol{e}^{t-1}$ on the aspect a^t selected by the Planner. This process is guided by the refinement instructions I_{a^t} for aspect a^t, including standards and auxiliary information (*e.g.*, item characteristics). Additionally, reflections on refined explanation content $R_c^{1:t-1}$ from the past rounds are also used to guide the Refiner's refinement performance. To ensure relevance, we summarize the reflections about aspect a^t, following prior works (Zhang et al., 2024f; Wang et al., 2023b), denoted as:

$$\tilde{\boldsymbol{R}}_c^{1:t-1} = \text{Summarize}(a^t, \boldsymbol{R}_c^{1:t-1}) \quad (3)$$

The Refiner's refinement function is formulated as:

$$\boldsymbol{e}^t = \text{Refiner}(\boldsymbol{e}^{t-1}, a^t, I_{a^t}, \tilde{\boldsymbol{R}}_c^{1:t-1}) \quad (4)$$

Memory module is designed to store key information about the target sample and record the refinement process. It consists of two components: Background Memory and Refinement Memory. Background memory M_b includes profiles of the target user-item pair (u, i), such as item attributes (*e.g.*, title and category) and user interactions (*e.g.*, ratings and reviews). This component serves as the resource for acquiring auxiliary information using acquisition functions in the aspect library. Refinement memory M_h records the refinement history, serving as the reference for producing reflections. The refinement record h^t in round t includes the refined explanation $\boldsymbol{e}^t$, refined aspect a^t, refinement

instructions I_{a^t}, and two levels of reflection, R_s^t and R_c^t, denoted as:

$$h^t = \{\boldsymbol{e}^t, a^t, I_{a^t}, R_s^t, R_c^t\} \tag{5}$$

The refinement memory is updated at each round as $\boldsymbol{M}_h^{1:t} = \boldsymbol{M}_h^{1:t-1} \cup \{h^t\}$. This structured memory effectively supports information acquisition and reflection generation.

3.3 The Reflection Phase

To provide feedback and suggestions to the Planner and Refiner, we introduce a hierarchical reflection mechanism to assess the refinement process from both strategic and content perspectives. This mechanism consists of two components:

Strategic Reflector aims to examine the alignment of the refinement trajectory with user goals through rule-based reflection, focusing on three key criteria: accuracy, completeness, and priority. Specifically, with the guidance of user goals, it mainly evaluates whether the refinement process involves irrelevant aspects, omits key aspects, or overlooks the relative priority of aspects. Besides providing feedback on planning, this Reflector is also expected to offer constructive suggestions to enhance the Planner. The generation of strategic reflection is formulated as:

$$R_s^t = \text{S\_Reflector}(G, \boldsymbol{M}_h^{1:t}, C_s), \tag{6}$$

where G is the user goal, $\boldsymbol{M}_h^{1:t}$ represents the refinement history, and C_s denotes the evaluation criteria at the strategic level.

Content Reflector evaluates the refinement performance from the perspective of explanation content. It provides insights about whether the explanation conforms to several content criteria, such as following the refinement instructions, covering necessary details, and excluding irrelevant content. Notably, it achieves tool-augmented reflection by calling external aspect metrics in the aspect library at each round and these values serve as quality reference signals for more comprehensive evaluation. The Content Reflector provides a timely updated view of explanation quality to the Planner and the Refiner for targeted planning and refinement. Its generation function is:

$$R_c^t = \text{C\_Reflector}(\boldsymbol{e}^t, a^t, I_{a^t}, S_{a^t}, C_c), \tag{7}$$

where I_{a^t} and S_{a^t} are the refinement instructions and external quality signal for aspect a^t, respectively. C_c denotes the content criteria.

In summary, these two types of Reflectors complement each other in evaluating the refinement process. The Strategic Reflector provides high-level feedback, ensuring overall alignment between planning and user goals, while the Content Reflector offers fine-grained feedback, ensuring the precision of refined content with respect to specific aspects. To support the entire process, we construct an aspect library containing key information for several user-centric aspects. More details and examples are provided in Appendix A.5.

3.4 Discussion

Advantages of Refinement. Due to the complexity and diversity of user behaviors, directly generating explanations with LLMs often struggles to accurately identify user preferences (Lei et al., 2024; Ma et al., 2024), especially since LLMs are less effective at processing domain-specific data. In contrast, we refine explanations generated by existing explainable recommender models, which already capture predicted user preferences. This approach effectively combines the strengths of recommender models with LLMs, resulting in explanations that are both accurate and helpful.

Relation to Agent-Based Frameworks. Our framework is general enough to subsume existing agent-based paradigms. As shown in the ablation study (Section 4.3), removing both reflection modules reduces the system to a plan-then-refine structure, similar to ReAct (Yao et al., 2022), while limiting refinement to a single round approximates Reflexion (Shinn et al., 2024). However, our framework introduces key distinctions tailored to explainable recommendation: (1) Personalization is explicitly addressed, a core requirement often overlooked in general NLP agents. (2) Multifaceted improvements is supported, unlike the single-objective common in prior works (3) Multi-agent collaborative architecture enables iterative, targeted optimization. To our knowledge, this is the first agent-based framework that achieves explanation quality improvements across multiple aspects, marking a significant contribution to the field.

4 Experiments

4.1 Experiment Setup

Datasets. We conduct experiments on three real-world datasets from distinct domains: **Yelp**, **Amazon-Beauty** (Beauty), and **Amazon-VideoGames** (Games).

Baselines. We use three common models in explainable recommendation, **PETER** (Li et al., 2021), **PEPLER** (Li et al., 2023) and **NRT** (Li et al., 2017), as base models to generate initial explanations. We compare their performance with two types of enhanced methods: (1) Model-Oriented methods, which enhance specific aspects of PETER by modifying its architecture or training process: **CLIFF** (Cao and Wang, 2021) for factuality, **ERRA** (Cheng et al., 2023) for personalization, and **CER** (Raczyński et al., 2023) for coherence. (2) Model-Agnostic methods, which refine initial explanations in a post-hoc manner: the LLM-based approach **LLMX** (Luo et al., 2023) and our proposed approach **RefineX**.

Evaluation Metrics. To evaluate the factuality of explanations, we employ **Entailment Ratio (Entail)** (Xie et al., 2023; Zhuang et al., 2024), which assesses the proportion of explanations that can be entailed or supported by existing reviews. For personalization, we first use **Feature Coverage Ratio (FCR)** (Li et al., 2020, 2021) to evaluate at the feature level by measuring the fraction of features present in the explanations. Additionally, we utilize **ENTR** (Jhamtani et al., 2018; Xie et al., 2023) to assess personalization from the diversity perspective by calculating the entropy of the n-grams distribution in the generated text. To evaluate coherence, we apply **Coherence Ratio (CoR)** (Yang et al., 2021; Zhao et al., 2024a), which measures the proportion of samples achieving coherence between the sentiment of generated explanations and the predicted user preference.

Implementation. Following recent LLM-based recommendation studies (Zhang et al., 2024d; Zhao et al., 2024b; Huang et al., 2023; Zhou et al., 2024b), we organize each user's interactions chronologically for all datasets and divide training and testing sets using the *leave-one-out* strategy. We randomly sample 200 users from the testing set of each dataset for evaluation. Two post-hoc methods, LLMX and RefineX, are implemented based on GPT-3.5 [1], with the user goal defaulted as "Assign equal importance to three aspects: factuality, personalization and sentiment coherence."

Additional implementation details are provided in Appendix A. Further analysis is included in Appendix B, the prompt templates used in our framework are listed in Appendix C, and the overall algorithm is presented in Appendix D.

[1] gpt-3.5-turbo-0125

4.2 Overall Performance

The overall comparison results between RefineX and baselines are presented in Table 1. We can see:

(1) Compared to the base model PETER, model-oriented methods modify the architecture to target specific aspects, leading to improvements in the corresponding metrics. However, these improvements are limited and unstable.

(2) In contrast, model-agnostic methods refine explanations generated by base models in a post-hoc manner and achieve more comprehensive improvements across multiple aspects. We speculate that user-centric explanations place high demands on both user preference and textual expression. Based on the predicted user preferences from base models, these methods leverage LLMs to produce more natural and fluent expressions, thereby contributing to their superior performance.

(3) Notably, our RefineX framework achieves the best performance for each base model, and the superiority is consistent across all datasets and aspects. These observations verify the effectiveness of our framework in enhancing explanations in user-centric aspects. RefineX achieves this through a multi-agent collaborative mechanism that adopts a plan-then-refine pattern for targeted refinement and incorporates hierarchical reflection for continuous refinement. For completeness, we also report the evaluation of textual similarity between generated explanations and user reviews in Appendix B.1, although this is not the focus of our work.

4.3 Ablation Study of the Reflection

Reflections provide analysis and guidance during the refinement process, playing a key role in system self-evolution. We investigate the impact of different reflection components by sequentially removing strategic and content reflections. Table 2 shows results on the Yelp and Beauty datasets, revealing the following insights:

Removing either component degrades explanation quality, with the worst performance observed when both are removed. This highlights the importance of both planning- and content-level feedback. Notably, removing content reflection leads to a larger performance drop than removing strategic reflection. We speculate that strategic reflection provides macro-level guidance by improving aspect-level accuracy and overall process efficiency, thereby indirectly influencing the final explanation quality. In contrast, content reflection directly eval-

Table 1: Performance comparison of different approaches. For each base model, the best and second-best results are highlighted in bold and underline, respectively. Higher values indicate better performance on all metrics.

Method	Yelp				Beauty				Games			
	Entail	FCR	ENTR	CoR	Entail	FCR	ENTR	CoR	Entail	FCR	ENTR	CoR
PETER	0.295	0.0166	7.089	0.455	0.265	0.0286	5.637	0.605	0.380	0.0078	6.280	0.555
CLIFF	0.450	0.0091	6.062	0.445	0.430	0.0233	6.254	0.745	0.395	0.0132	6.341	0.665
CER	0.195	0.0155	6.529	0.530	0.450	0.0318	6.444	0.840	0.300	0.0102	5.918	0.625
ERRA	0.185	0.0178	6.922	0.470	0.220	0.0424	6.709	0.790	0.325	0.0112	6.202	0.520
+LLMX	0.555	0.0299	9.721	0.530	0.575	0.1081	9.459	0.765	0.670	0.0351	9.527	0.635
+RefineX	**0.835**	**0.0424**	**10.082**	**0.770**	**0.800**	**0.1441**	**10.160**	**0.885**	**0.835**	**0.0473**	**10.099**	**0.750**
PEPLER	0.420	0.0121	6.789	0.525	0.140	0.0297	6.909	0.550	0.385	0.0093	6.029	0.570
+LLMX	0.600	0.0242	9.420	0.520	0.575	0.1197	9.759	0.760	0.735	0.0390	9.494	0.605
+RefineX	**0.845**	**0.0390**	**10.007**	**0.680**	**0.775**	**0.1472**	**10.335**	**0.870**	**0.805**	**0.0527**	**10.098**	**0.760**
NRT	0.660	0.0012	3.328	0.000	0.215	0.0191	5.344	0.275	0.605	0.0039	4.270	0.320
+LLMX	0.810	0.0238	9.451	0.090	0.560	0.1186	9.645	0.905	0.700	0.0400	9.331	0.355
+RefineX	**0.845**	**0.0327**	**9.935**	**0.590**	**0.785**	**0.1494**	**10.300**	**0.935**	**0.860**	**0.0478**	**10.108**	**0.655**

Table 2: Ablation study of the reflection mechanism. "SR" and "CR" denote "strategic reflector" and "content reflector", respectively.

Dataset	Method	Entail	FCR	ENTR	CoR
Yelp	RefineX	0.835	**0.0424**	**10.082**	**0.770**
	-*w/o* SR	**0.845**	0.0393	10.080	0.715
	-*w/o* CR	0.820	0.0392	10.041	0.755
	-*w/o* SR&CR	0.760	0.0363	10.011	0.700
Beauty	RefineX	**0.800**	**0.1441**	**10.160**	**0.885**
	-*w/o* SR	0.755	0.1292	10.111	0.855
	-*w/o* CR	0.710	0.1250	10.041	0.830
	-*w/o* SR&CR	0.760	0.1239	10.093	0.820

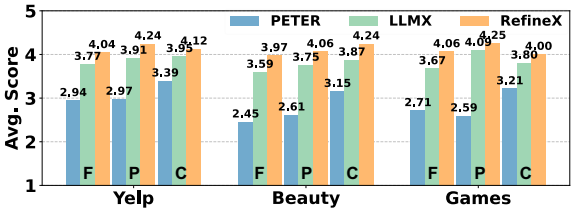

Figure 3: Human evaluation results of PETER, LLM and RefineX on three datasets across three aspects.

uates and refines explanation content, offering fine-grained, micro-level feedback. Their complementary roles together enhance the overall effectiveness of our framework.

4.4 Adaptability Analysis

Compared to existing approaches, a significant advantage of our framework is its high adaptability to various user goals. To verify this, following the setting of prior studies (Gao et al., 2024), we define three distinct user goals by assigning different weights to aspects and analyze the performance on explanation quality, refined aspect ratio, and refinement trajectory.

• F=P=C: Assign equal importance to three aspects: factuality, personalization and coherence.

• F>P>C: Assign primary importance to factuality, followed by personalization, and then coherence.

• P>F>C: Assign primary importance to personalization, followed by factuality, and then coherence.

The results on the Yelp and Beauty datasets are presented in Table 3, revealing the following findings: Our framework can flexibly tailor its refinement strategy to align with different user goals, which places greater emphasis on high-priority aspects and improves performance on corresponding metrics. From the perspective of refinement trajectory, our framework prioritizes refining the most important aspects and may refine them multiple times. This is further supported by the observed ratio of refined aspects. Notably, when emphasizing personalization in the Yelp dataset, this aspect is often refined multiple times. We speculate that the larger size and richer reviews in Yelp enable the refinement process to incorporate more item- and user-specific features, thereby continuously enhancing personalization quality.

Additionally, although some refinement processes reach the maximum number of rounds (defaulted to 6), the average trajectory length remains around 4. This demonstrates the effectiveness of our planning mechanism, which achieves precise aspect selection and efficient refinement, thereby supporting adaptability to diverse user goals.

Table 3: Adaptability analysis of RefineX on various user goals. The comparison symbols in goals indicate the priority of different aspects: Factuality (F), Personalization (P) and Coherence (C). "Aspect Ratio" represents the proportion of refined aspects in all testing samples. "Representative" and "Ratio" denote the most representative refinement trajectory under each goal and its corresponding ratio. "Length" and "Max Stop" refer to the average length of trajectory and the proportion of samples reaching the maximum number of rounds, respectively.

Dataset	Goal	Explanation Quality				Aspect Ratio	Trajectory			
		Entail	FCR	ENTR	CoR	F : P : C	Representative	Ratio	Length	Max Stop
Yelp	F=P=C	0.835	**0.0424**	10.082	**0.770**	32 : 40 : 28	[F, P, C]	30.4%	4.00	34.5%
	F>P>C	**0.855**	0.0413	10.085	0.605	46 : 38 : 16	[F, P, C, F]	10.9%	3.87	38.0%
	P>F>C	0.790	0.0422	**10.119**	0.545	16 : 73 : 11	[P, P, P, F, P, C]	12.5%	4.47	50.5%
Beauty	F=P=C	**0.800**	**0.1441**	10.160	**0.885**	34 : 41 : 25	[F, P, C]	35.4%	4.06	31.5%
	F>P>C	0.790	0.1314	10.139	0.805	48 : 35 : 17	[F, P, C, F]	16.4%	3.96	34.5%
	P>F>C	0.705	0.1356	**10.206**	0.825	18 : 68 : 14	[P, C, F]	10.3%	4.01	39.0%

4.5 Human Evaluation

To further investigate whether the generated explanations truly assist users, we conduct a human evaluation with five experts in recommender systems. They rate each explanation along three aspects: factuality, personalization, and coherence, using a 5-point Likert scale (1-strongly disagree to 5-strongly agree). To control cost, we randomly select 30 user-item pairs from each dataset and present explanations produced by PETER, LLMX and RefineX in random order. Each expert provides a total of 810 scores. The average Cohen's kappa coefficient is approximately 0.6, indicating moderate agreement among annotators and supporting the reliability of the evaluation.

As shown in Figure 3, RefineX consistently receives higher scores than PETER across all datasets and aspects. On average, it outperforms PETER by 49.0%, 53.6%, and 26.8% across three aspects, respectively. These results further reveal the limitations of traditional explainable models in capturing user-centric qualities and demonstrate the effectiveness of our framework. Notably, human evaluation results align well with the automatic metrics in Table 1, offering a more comprehensive validation.

4.6 Efficiency Analysis

Refinement Efficiency. Benefiting from the Planner module, we observe that most examples require only 2–4 iterations to achieve significant quality improvements. In practice, each refinement process takes an average of 14.27 seconds and 1,020 tokens using GPT-3.5. The slight increase in inference time compared to traditional models is primarily due to the overhead of LLM calls.

Effectiveness of Model Distillation. To further improve inference efficiency, we explore distilling LLM-generated knowledge into the base model. Specifically, we retrain each base model using 1,000 refined explanations generated by our framework for that model, resulting in a distilled version used for final explanation generation. Table 4 presents the performance comparison. With only a small amount of refined data, the distilled model achieves notable improvements over the base model. While it does not fully match the performance of our framework, it provides a practical trade-off between effectiveness and efficiency.

Table 4: Performance comparison between base models, distilled models, and RefineX on the Beauty dataset

Method	Entail	FCR	ENTR	CoR
PETER	0.265	0.0286	5.637	0.605
+Distillation	0.370	0.0413	7.024	**0.925**
+RefineX	**0.800**	**0.1441**	**10.160**	0.885
PEPLER	0.140	0.0297	6.909	0.550
+Distillation	0.200	0.0339	6.657	0.470
+RefineX	**0.775**	**0.1472**	**10.335**	**0.870**
NRT	0.215	0.0191	5.344	0.275
+Distillation	0.255	0.0413	6.397	0.530
+RefineX	**0.785**	**0.1494**	**10.300**	**0.935**

4.7 Case Study

To provide a more intuitive understanding of our framework, we present a refinement example in Figure 4. The following insights can be drawn: On one hand, our framework effectively plans according to the user goal. The refinement trajectory closely follows the specified priorities, refining the most important aspect first and repeating it as needed. On the other hand, it demonstrates effective refinement capabilities. For example, the phrase "the broth was good" in the initial explanation conflicts with the descriptions in existing

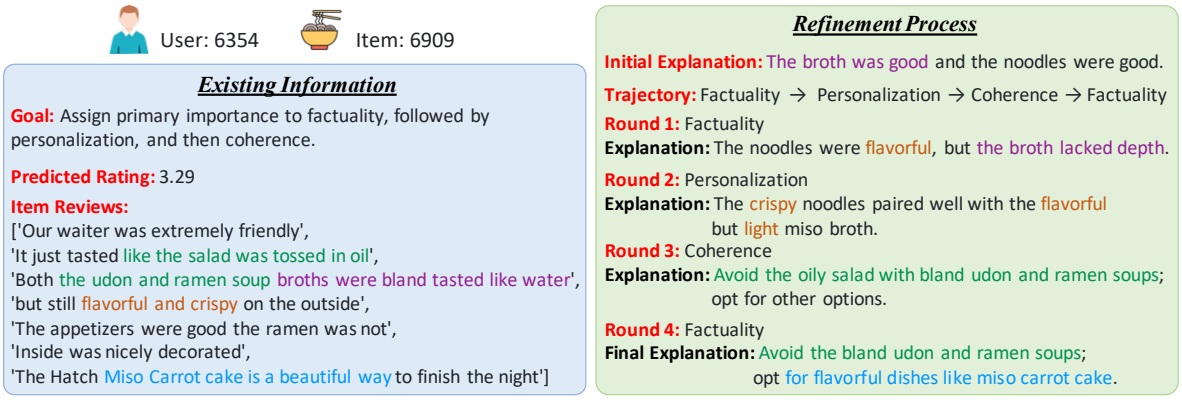

Figure 4: A refinement example. Colored text in explanations highlights content aligned with the original reviews.

reviews. Our framework identifies and corrects this inconsistency in the first round. In the second round, RefineX enhances personalization by extracting distinctive item features. To ensure coherence with the predicted neutral preference (a rating of 3.29), RefineX balances the sentiment by adding some disadvantages. In the final round, it further improves factuality by incorporating additional review details.

Furthermore, we observe that the initial explanation from PETER provides a general summary, mainly focusing on food rather than service or decor. This may reflect PETER's preference modeling, which likely infers that the user is more interested in taste than other dimensions. Our framework builds on this and enhances the explanation accordingly. By integrating explainable models with LLM-based agents, RefineX generates explanations that are both accurate and user-centric.

5 Related Work

Explainable Recommendation. Recommendation explanations are pivotal for improving user satisfaction and system transparency (Zhang et al., 2020; Chen et al., 2023). Recently, natural language explanations have gained more attention, which are generated using different language models (Li et al., 2020, 2021, 2023). To enhance explanation quality, some studies focus on modifying model architecture and training process. For example, integrating additional components to capture auxiliary information (Cheng et al., 2023; Zhang et al., 2024b), and employing techniques such as unbiased learning (Zhang et al., 2023) and adversarial learning (Zhang et al., 2024c) for targeted training. Recent studies also explore directly generating explanations by prompting LLMs (Luo et al., 2023; Lei et al., 2024; Tang et al., 2025). Unlike these methods, our work focuses on refining explanations produced by existing models in a post-hoc manner. While a recent study (Qin et al., 2024) investigates response refinement, our framework differs in task, scenario, and methodology.

LLM-based Autonomous Agents. LLM-based agents have showcased remarkable abilities in reasoning (Yao et al., 2022), planning (Shinn et al., 2024), and tool use (Schick et al., 2024). Applications in this field can be divided into two categories (Wang et al., 2024): The first focuses on assisting humans with complex tasks, such as software development (Qian et al., 2023) and role-playing in games (Wang et al., 2023a), while the second aims to simulate human behaviors in diverse scenarios (Park et al., 2023; Gao et al., 2023; Liu et al., 2023). Several studies apply LLM-based agents to recommender systems. Some improve recommendation performance by equipping agents with recommendation tools (Huang et al., 2023; Zhang et al., 2024d; Wang et al., 2023c), while others simulate user behaviors in recommendation scenarios (Zhang et al., 2024a; Wang et al., 2023b). In contrast to these studies, our study is the first to design an LLM-based agent framework specifically for generating recommendation explanations.

6 Conclusion

In this paper, we highlight the limitations of existing explainable recommender models in meeting user-centric demands and propose a novel paradigm for targeted explanation refinement. To this end, we design an LLM-based multi-agent collaborative framework that adopts a plan-then-refine strategy and incorporates a hierarchical reflection mechanism. Extensive experiments demonstrate the effectiveness of our framework in improving user-centric explanation quality and its adaptability to diverse user demands, ultimately enhancing the helpfulness of explanations.

Limitations

This study presents the first exploration of using LLM-based agents to enhance recommendation explanations. Despite its effectiveness, some limitations remain: First, the refinements in our experiments are achieved based on GPT-3.5. With the rapid development of LLMs, more advanced models could be adopted to provide more accurate refinements. Second, we demonstrate the effectiveness of our framework in three common user-centric aspects. Other aspects, such as informativeness and comparability, could be integrated for more comprehensive refinement. Finally, since users often have similar demands across various scenarios, our framework has potential to adapt to other generative tasks, such as dialogue generation.

Ethical Considerations

All the datasets used in our experiments are publicly available and have been widely employed in previous studies. They do not contain any personal privacy information. Additionally, due to the training mechanisms of large language models, the generated text may contain potential biases.

Acknowledgments

This work is supported in part by National Natural Science Foundation of China (No. 62422215 and No. 62472427), Major Innovation & Planning Interdisciplinary Platform for the "DoubleFirst Class" Initiative, Renmin University of China, Public Computing Cloud, Renmin University of China, fund for building world-class universities (disciplines) of Renmin University of China, and the Outstanding Innovative Talents Cultivation Funded Programs 2024 of Renmin University of China.

References

Shuyang Cao and Lu Wang. 2021. Cliff: Contrastive learning for improving faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2109.09209*.

Xu Chen, Jingsen Zhang, Lei Wang, Quanyu Dai, Zhenhua Dong, Ruiming Tang, Rui Zhang, Li Chen, Xin Zhao, and Ji-Rong Wen. 2023. Reasoner: an explainable recommendation dataset with comprehensive labeling ground truths. *Advances in Neural Information Processing Systems*, 36:14497–14515.

Hao Cheng, Shuo Wang, Wensheng Lu, Wei Zhang, Mingyang Zhou, Kezhong Lu, and Hao Liao. 2023. Explainable recommendation with personalized review retrieval and aspect learning. *arXiv preprint arXiv:2306.12657*.

Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2023. S3: Social-network simulation system with large language model-empowered agents. *arXiv preprint arXiv:2307.14984*.

Jingtong Gao, Bo Chen, Xiangyu Zhao, Weiwen Liu, Xiangyang Li, Yichao Wang, Zijian Zhang, Wanyu Wang, Yuyang Ye, Shanru Lin, et al. 2024. Llm-enhanced reranking in recommender systems. *arXiv preprint arXiv:2406.12433*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022a. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. 2022b. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*.

Xu Huang, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. 2023. Recommender ai agent: Integrating large language models for interactive recommendations. *arXiv preprint arXiv:2308.16505*.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, Graham Neubig, and Taylor Berg-Kirkpatrick. 2018. Learning to generate move-by-move commentary for chess games from large-scale social forum data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1661–1671.

Yuxuan Lei, Jianxun Lian, Jing Yao, Xu Huang, Defu Lian, and Xing Xie. 2024. Recexplainer: Aligning large language models for explaining recommendation models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1530–1541.

Lei Li, Yongfeng Zhang, and Li Chen. 2020. Generate neural template explanations for recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 755–764.

Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized transformer for explainable recommendation. *arXiv preprint arXiv:2105.11601*.

Lei Li, Yongfeng Zhang, and Li Chen. 2023. Personalized prompt learning for explainable recommendation. *ACM Transactions on Information Systems*, 41(4):1–26.

Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 345–354.

Ruibo Liu, Ruixin Yang, Chenyan Jia, Ge Zhang, Denny Zhou, Andrew M Dai, Diyi Yang, and Soroush Vosoughi. 2023. Training socially aligned language models in simulated human society. *arXiv preprint arXiv:2305.16960*.

Yucong Luo, Mingyue Cheng, Hao Zhang, Junyu Lu, and Enhong Chen. 2023. Unlocking the potential of large language models for explainable recommendations. *arXiv preprint arXiv:2312.15661*.

Qiyao Ma, Xubin Ren, and Chao Huang. 2024. Xrec: Large language models for explainable recommendation. *arXiv preprint arXiv:2406.02377*.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, Wenwu Ou, et al. 2019. Personalized re-ranking for recommendation. In *Proceedings of the 13th ACM conference on recommender systems*, pages 3–11.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 6.

Peixin Qin, Chen Huang, Yang Deng, Wenqiang Lei, and Tat-Seng Chua. 2024. Beyond persuasion: Towards conversational recommender system with credible explanations. *arXiv preprint arXiv:2409.14399*.

Jakub Raczyński, Mateusz Lango, and Jerzy Stefanowski. 2023. The problem of coherence in natural language explanations of recommendations. In *ECAI 2023*, pages 1922–1929. IOS Press.

Behnam Rahdari, Hao Ding, Ziwei Fan, Yifei Ma, Zhuotong Chen, Anoop Deoras, and Branislav Kveton. 2024. Logic-scaffolding: Personalized aspect-instructed recommendation explanation generation using llms. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 1078–1081.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Zhongxiang Sun, Zihua Si, Xiaoxue Zang, Kai Zheng, Yang Song, Xiao Zhang, and Jun Xu. 2024. Large language models enhanced collaborative filtering. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2178–2188.

Jiakai Tang, Jingsen Zhang, Zihang Tian, Xueyang Feng, Lei Wang, and Xu Chen. 2025. Hf4rec: Human-like feedback-driven optimization framework for explainable recommendation. *ACM Transactions on Information Systems*.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023a. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Lei Wang, Jingsen Zhang, Hao Yang, Zhiyuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Ruihua Song, Wayne Xin Zhao, et al. 2023b. User behavior simulation with large language model based agents. *arXiv preprint arXiv:2306.02552*.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. 2023c. Recmind: Large language model powered agent for recommendation. *arXiv preprint arXiv:2308.14296*.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. 2023d. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv preprint arXiv:2302.01560*.

Zhouhang Xie, Sameer Singh, Julian McAuley, and Bodhisattwa Prasad Majumder. 2023. Factual and informative review generation for explainable recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13816–13824.

Aobo Yang, Nan Wang, Hongbo Deng, and Hongning Wang. 2021. Explanation as a defense of recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1029–1037.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024a. On generative agents in recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in Information Retrieval*, pages 1807–1817.

Jingsen Zhang, Xiaohe Bo, Chenxi Wang, Quanyu Dai, Zhenhua Dong, Ruiming Tang, and Xu Chen. 2024b. Active explainable recommendation with limited labeling budgets. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5375–5379. IEEE.

Jingsen Zhang, Xu Chen, Jiakai Tang, Weiqi Shao, Quanyu Dai, Zhenhua Dong, and Rui Zhang. 2023. Recommendation with causality enhanced natural language explanations. In *Proceedings of the ACM web conference 2023*, pages 876–886.

Jingsen Zhang, Jiakai Tang, Xu Chen, Wenhui Yu, Lantao Hu, Peng Jiang, and Han Li. 2024c. Natural language explainable recommendation with robustness enhancement. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4203–4212.

Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024d. Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the ACM on Web Conference 2024*, pages 3679–3689.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Xiaoyu Zhang, Yishan Li, Jiayin Wang, Bowen Sun, Weizhi Ma, Peijie Sun, and Min Zhang. 2024e. Large language models as evaluators for recommendation explanations. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 33–42.

Yongfeng Zhang, Xu Chen, et al. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024f. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*.

Yurou Zhao, Yiding Sun, Ruidong Han, Fei Jiang, Lu Guan, Xiang Li, Wei Lin, Weizhi Ma, and Jiaxin Mao. 2024a. Aligning explanations for recommendation with rating and feature via maximizing mutual information. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 3374–3383.

Yuyue Zhao, Jiancan Wu, Xiang Wang, Wei Tang, Dingxian Wang, and Maarten de Rijke. 2024b. Let me do it for you: Towards llm empowered recommendation via tool learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1796–1806.

Huachi Zhou, Shuang Zhou, Hao Chen, Ninghao Liu, Fan Yang, and Xiao Huang. 2024a. Enhancing explainable rating prediction through annotated macro concepts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11736–11748.

Yujia Zhou, Qiannan Zhu, Jiajie Jin, and Zhicheng Dou. 2024b. Cognitive personalized search integrating large language models with an efficient memory mechanism. In *Proceedings of the ACM on Web Conference 2024*, pages 1464–1473.

Haojie Zhuang, Wei Zhang, Weitong Chen, Jian Yang, and Quan Z Sheng. 2024. Improving faithfulness and factuality with contrastive learning in explainable recommendation. *ACM Transactions on Intelligent Systems and Technology*.

Table 5: Statistics of the datasets.

Dataset	#User	#Item	#Inter.	Sparsity	Domain
Yelp	15,025	12,445	698,084	99.63%	Restaurant
Beauty	5,396	3,178	54,805	99.68%	Cosmetic
Games	13,957	7,378	140,353	99.86%	Game

A Details of Experiment Setup

A.1 Datasets

We conduct experiments using three real-world datasets from distinct domains. **Yelp** [2] includes user ratings and reviews of various restaurants. **Amazon-Beauty** [3] **(Beauty)** and **Amazon-Video Games (Games)** contain user interactions regarding cosmetics and video games, respectively, on the Amazon e-commerce platform. Detailed statistics of the datasets are presented in Table 5.

A.2 Baselines

We provide a detailed description of each approach compared in our experiments:

• **PETER** (Li et al., 2021) is a state-of-the-art method for explainable recommendation, which personalizes the Transformer by integrating IDs with texts. We utilize it as the base model to generate initial explanations.

• **PEPLER** (Li et al., 2023) leverages prompt learning with pre-trained language models to further enhance the explanation quality.

• **NRT** (Li et al., 2017) incorporates user preference signals by integrating predicted ratings into the explanation generation process.

• **CLIFF** (Cao and Wang, 2021) enhances factuality in the abstractive summarization task by introducing a contrastive learning framework to distinguish positive and negative samples, subsequently extending it to the explanation generation task (Zhuang et al., 2024).

•**ERRA** (Cheng et al., 2023) improves personalization based on PETER by incorporating an aspect enhancement component, selecting aspects most relevant to users to better capture user preference.

• **CER** (Raczyński et al., 2023) aims to generate more coherent explanations based on PETER by introducing an auxiliary task of explanation-based rating estimation as a regularizer.

• **LLMX** employs LLMs directly to refine explanations. Since no existing study focuses on refining explanations, we implement this method following common prompt templates from studies that generate explanations using LLMs (Luo et al., 2023; Lei et al., 2024; Rahdari et al., 2024).

• **RefineX**, our proposed approach, which designs an LLM-based multi-agent collaborative refinement framework to improve explanation quality focusing on user-centric aspects.

A.3 Evaluation Metrics

We utilize common metrics in the field of explainable recommendation to evaluate the quality of explanations across various aspects. For more precise evaluation, we use GPT-4 [4] to implement two LLM-based metrics, Entail and CoR. Details of each metric are as follows:

• **Entailment Ratio (Entail)** (Xie et al., 2023; Zhuang et al., 2024) measures the proportion of explanations that can be entailed or supported by existing reviews, which uses the following prompt:

Prompt for Judging Entailment Relation.

You will be given a {Recommendation_Explanation} and a list of existing {Item_Reviews}.
Your task is to evaluate whether all information in the explanation is explicitly described or implied by the reviews.
- Return 1 if all information is entailed or supported by the reviews.
- Return 0 if any information is not.

• **Coherence Ratio (CoR)** (Yang et al., 2021; Zhao et al., 2024a) evaluates the proportion of samples achieving coherence between explanation sentiment and predicted user preference. Sentiment is identified using the following prompt:

Prompt for Sentiment Identification.

You will be given a {Text}, which serves as a recommendation explanation aimed to inform the user about why an item is recommended or not.
Your task is to analyze the sentiment of the explanation and classify it as either positive or negative:
- Positive (1): The explanation suggests recommending the item to the user.
- Negative (-1): The explanation suggests not recommending the item to the user.

[2] https://www.yelp.com/dataset
[3] https://jmcauley.ucsd.edu/data/amazon/index_2014.html
[4] gpt-4o-2024-08-06

Table 6: Examples of aspect materials in the aspect library.

Aspects	Materials
Factuality	Standard: The aspect to refine is Factuality, and its standard is to Ensure the explanation is factually correct and can be supported by provided information. Instruction: Refine the recommendation explanation using the information in {Item_Characteristics}, ensuring the explanation is factually correct. Equipped Functions: get_item_characteristics() Quality Signal: Entailment Ratio (Entail)
Personalization	Standard: The aspect to refine is Personalization, and its standard is to Customize the explanation to reflect specific item characteristics and user personalities. Instruction: Refine the recommendation explanation using the information in {Item_Characteristics} and {User_Personalities}, making the explanation content personalized and reflecting user's key concerns. Equipped Functions: get_item_characteristics(), get_user_personalities() Quality Signal: Feature Coverage Ratio (FCR)
Sentiment Coherence	Standard: The aspect to refine is Sentiment Coherence, and its standard is to Ensure the explanation's sentiment (positive/negative) aligns with the predicted user preference (like/dislike). Instruction: Refine the recommendation explanation using the information in {Item_Pros} and {Item_Cons}. To match the explanation's sentiment with {User_Preference}, emphasize advantages for positive preferences and highlight disadvantages for negative preferences. Equipped Functions: get_item_pros(), get_item_cons(), predict_user_preference() Quality Signal: Coherence Ratio (CoR)

• **Feature Coverage Ratio (FCR)** (Li et al., 2020, 2021) evaluates personalization at the feature level by measuring the fraction of features present in generated explanations. It is denoted as:

$$\text{FCR} = N_e/|\mathcal{F}|,$$

where N_e is the number of features included in the generated explanations, and $\mathcal{F}$ denotes the feature set collected in the dataset.

• **ENTR** (Jhamtani et al., 2018; Xie et al., 2023) assesses personalization from the diversity perspective by measuring the entropy of n-grams distribution in the generated text, formulated as:

$$\text{ENTR} = \left(\prod_{n=1}^{3} -\sum_{x\in X_n} p(x)\log p(x)\right)^{\frac{1}{3}},$$

where each term $-\sum_{x\in X_n} p(x)\log p(x)$ represents the entropy of the unigrams, bigrams, and trigrams distribution, respectively.

A.4 Implementation Details

We organize each user's interactions chronologically for all datasets and divide them for training and testing using the *leave-one-out* strategy (Zhang et al., 2024d; Luo et al., 2023; Sun et al., 2024), where the last interaction of each user is used for testing and the others are used for training. We implement the baselines based on the code released by their authors. For training-oriented methods, the batch size and embedding size are set to 128 and 512, respectively, and other parameters are set to their optimal values as reported in the original papers. For RefineX, the maximum number of refinement rounds per sample is set to 6. To ensure fair comparisons, we follow the common setting in prior studies (Li et al., 2023; Ma et al., 2024) and set the maximum length of generated explanations for all methods to 20.

A.5 Aspect Library

To facilitate explanation refinement, we construct an aspect library containing essential materials of user-centric aspects, including aspect standards, refinement instructions, information acquisition functions, and external quality signals. These materials support both the refinement and reflection phases. Additionally, this structured library is designed for flexibility, enabling the dynamic combination and seamless integration of diverse aspects to accommodate personalized user goals. Table 6 shows details on three aspects used in our experiments.

Table 7: Textual similarity between generated explanations and user reviews on the Beauty dataset, measured by BLEU (B), ROUGE (R), and BERTScore (BS).

Method	B-1	B-2	R-1	R-2	BS
PETER	**11.206**	**3.706**	**13.835**	**1.474**	**88.311**
+LLMX	9.474	1.954	13.452	0.612	86.583
+RefineX	7.716	1.968	10.562	0.839	85.310
PEPLER	**12.055**	**3.784**	**16.733**	**2.031**	85.192
+LLMX	8.093	1.826	13.234	0.684	**85.896**
+RefineX	6.980	1.274	11.037	0.340	85.221
NRT	7.587	**2.652**	10.699	**1.142**	**88.355**
+LLMX	**7.650**	1.618	**12.847**	0.409	85.873
+RefineX	7.236	1.381	11.618	0.473	85.159

Table 8: Evaluation of additional aspects (Novelty and Conciseness) on the Beauty dataset.

Method	Novelty	Conciseness
PETER	0.005	0.920
+LLMX	0.205	0.990
+RefineX	**0.380**	**0.995**
PEPLER	0.105	0.320
+LLMX	0.460	0.985
+RefineX	**0.610**	**0.995**
NRT	0.030	0.815
+LLMX	0.345	0.985
+RefineX	**0.535**	**0.995**

B Further Analysis

B.1 Evaluation of Textual Similarity

In this section, we measure the textual similarity between generated explanations and user reviews using common metrics, including BLEU, ROUGE, and BERTScore (Zhang et al., 2019), although this is not the primary focus of our paper. As shown in Table 7, base models tend to achieve higher similarity scores, which aligns with their training objective of directly optimizing toward reference reviews. In contrast, our approach focuses on improving explanation quality on user-centric aspects, which are not effectively captured by these metrics, especially those based on n-gram overlap. Moreover, user reviews often contain noise and inconsistencies, further limiting the reliability of such metrics in evaluating explanation quality.

B.2 Evaluation of More Aspects

As illustrated in the examples in Figure 4, improvements in user-centric aspects naturally lead to gains in other dimensions, such as novelty and conciseness. To further validate this observation, we evaluate these additional aspects using GPT-4. As shown in Table 8, our framework achieves clear improvements in both aspects. These results underscore the broader impact of our method beyond the explicitly targeted aspects.

C Prompt Design

This section introduces the prompts used by the agents within our RefineX framework, which are designed with several key components such as background clarification, system instruction, required information, and output format. These structured prompts enable agents to execute their tasks accurately and facilitate effective collaboration, enhancing the quality of explanations on user-concerned aspects. Detailed prompt templates are presented in Table 9.

D Overall Algorithm of RefineX

The complete pipeline of RefineX is shown in Algorithm 1. It comprises two main phases: a forward refinement phase for planning and generation, and a backward reflection phase for feedback-driven improvement. This process is analogous to model optimization in deep learning, where forward inference generates task outputs, while backpropagated gradients guide model update.

Algorithm 1: The Pipeline of RefineX.

Specify the user goal G and the maximum number of refinement rounds N.
Prepare the aspect library $\mathcal{A}$.
Initialize the background memory M_b.
Generate the initial explanation $\boldsymbol{e}^0$ using the pre-trained explainable recommender model.
for *round t in [1, N]* **do**
    `// The Refinement Phase:`
    Obtain a plan from the Planner using Eq. (2).
    if *fully refined* **then**
        Terminate the process.
    else if *an aspect a^t is selected to refine* **then**
        Retrieve sample information from M_b by calling functions in $\mathcal{A}$.
        Summarize content reflections for aspect a^t using Eq. (3).
        Generate the refined explanation e^t by the Refiner using Eq. (4).
    `// The Reflection Phase:`
    Obtain the strategic reflection R_s^t using Eq. (6).
    Derive the external quality signals S_{a^t} from $\mathcal{A}$.
    Obtain the content reflection R_c^t using Eq. (7).
    Update refinement memory M_h using Eq. (5).
Output the final explanation to the user.

Table 9: Prompt templates used in various agents.

Agents	Prompts
Planner	# Background Clarification This framework refines recommendation explanations to better meet users' goals, such as Factuality, Personalization, and Sentiment coherence. It includes the following agents: - Planner: Identifies which aspect of the explanation to refine next or decides whether to terminate the process. - Refiner: Modifies the explanation on the selected aspect following the instructions. - Reflector: Evaluates the Planner's and Refiner's actions to provide feedback for improvements. Together, these agents enhance the recommendation explanation to align with user's goal. # System Instruction You are the Planner. Your role is to identify which aspect of the current explanation requires refinement in the next step, guided by the user's overall goals, refinement trajectory, and the Reflector's feedback. The framework permits up to {Max_Count} modifications per explanation and will terminate when this limit is reached, necessitating careful planning of the refinement process. # Required Information The current explanation is: {Current_Explanation} The user's overall goal for explanation is: {User_Goal} The refinement trajectory is: {Refinement_Trajectory} Reflector's feedback on the Planner's strategies: {Strategic_Reflection} Reflector's feedback on the content of explanation: {Content_Reflection} # Output Format { "aspect": <int> // Choose one: 0 (Finish), 1 (Factuality), 2 (Personalization), 3 (Sentiment Coherence) }
Refiner	# Background Clarification {Background_Clarification} # System Instruction You are the Refiner. Your role is to improve the current explanation on a specific aspect, based on the provided refinement instructions and the summarized reflections from the Reflector. Please ensure the explanation is no longer than {Max_Length} words! # Required Information The current explanation is: {Current_Explanation} The aspect to be refined is: {Refined_Aspect} Refinement instructions and information for the refined aspect: {Refinement_Instruction} Summarized Content Reflections on the refined aspect: {Summarize_Reflection} # Output Format { "explanation": <string> // The refined explanation. }
Strategic Reflector	# Background Clarification {Background_Clarification} # System Instruction You are the Strategic Reflector. Your role is to evaluate the Planner's aspect-selection decisions at each round of the refinement based on the user's overall goal, refinement history and evaluation criteria. Assess whether these decisions align with the user's overall goal and provide constructive feedback to help the Planner improve. # Required Information The user's overall goal for explanation is: {User_Goal} The refinement history is: {Refinement_Memory} At the round {Time_Step}, the aspect being refined is {Refined_Aspect} Evaluate the Planner's selection, focusing on the following criteria: {Strategy_Criteria} # Output Format { "strategic reflection": <string> // The generated strategic reflection. }
Content Reflector	# Background Clarification {Background_Clarification} # System Instruction You are the Content Reflector. Your role is to evaluate the Refiner's modifications to the content of the explanation based on the current explanation, refined aspect name, aspect instruction, quality signal and evaluation criteria. Assess whether these refinements meet the aspect standard and provide constructive suggestions for improvement. # Required Information The current explanation is: {Current_Explanation} The aspect to be refined is: {Refined_Aspect} Refinement instructions and information for the refined aspect: {Refinement_Instruction} The external reference signal for the quality on the refined aspect is: {Quality_Signal} Evaluate the Refiner's modifications to the content of explanation, focusing on the following criteria: {Content_Criteria} # Output Format { "content reflection": <string> // The generated content reflection. }