CoExRec: Collaborative Filtering-grounded Large Language Model for Sequential Recommendation and Explanation Generation

Anonymous ACL submission

Abstract

Large language models (LLMs) generate fluent explanations for recommender systems, but these explanations often lack grounding in the collaborative signals that drive actual predictions. To this end, we propose CoExRec, a **co**llaborative filtering-grounded framework for joint sequential recommendation and **ex**planation generation. CoExRec extracts prediction-relevant evidence from an external sequential recommender using removal-based contribution scores, and uses this evidence to construct schematic profiles and target-specific explanation supervision. It also injects collaborative filtering (CF) embeddings into a frozen LLM through cross-attention adapters, allowing the model to use CF knowledge without updating the LLM backbone. Extensive experiments show that CoExRec improves explanation generation over existing baselines and achieves competitive recommendation performance. Our analyses further show that CF-grounded evidence leads to reliable profile construction and stable explanation.

1 Introduction

Recommender systems are increasingly expected not only to predict relevant items, but also to explain why those items are recommended (Ma et al., 2024). Natural-language explanations can help users understand recommendation results and make recommendation behavior more transparent. Recent progress in large language models (LLMs) has therefore motivated LLM-based recommender systems that generate fluent explanations from user histories, item information, and reviews (Wu et al., 2024; Liu et al., 2025).

However, fluency alone does not guarantee that an explanation is grounded in the recommendation decision. Many existing explanation models are trained to generate review-like text, or use textual histories as post-hoc context, without explicitly connecting the generated explanation to the collaborative signals that drive the recommender's prediction (Geng et al., 2022a; Lyu et al., 2023; Zhang et al., 2025). This limitation is especially important in sequential recommendation (Yang et al., 2024). When a model recommends a target item from a user's interaction prefix, a useful explanation should identify which preceding interactions support that target recommendation, rather than merely describe the target item or summarize the user's general preferences.

We argue that explanation generation for sequential recommendation should be grounded in the recommender's collaborative filtering (CF) signal. The key challenge is to use this signal consistently for both textual data construction and model conditioning. For data construction, the model needs compact and faithful user/item profiles and target-specific explanation supervision that reflect prediction-relevant evidence (Ni et al., 2019; Ma et al., 2024). For model conditioning, the LLM needs access to collaborative knowledge beyond symbolic user and item tokens (Zheng et al., 2024; Kim et al., 2024). These requirements should be handled consistently, so that the textual data used for training and the model computation are grounded in the same recommendation signal.

We propose CoExRec, a CF-grounded framework for joint sequential recommendation and explanation generation. CoExRec uses a single external sequential recommender in two ways. First, it extracts removal-based contribution scores (RCS), which measure how much the target item score decreases when each prefix item is removed. RCS is used as a prediction-grounded evidence selector for schematic user/item profile construction and target-specific explanation supervision. Second, CoExRec injects CF embeddings from the same recommender into a frozen LLM through cross-attention adapters. This design allows CoExRec to perform next-item prediction and target-specific ex-

planation generation in a single framework, while preserving the pretrained language knowledge of the LLM.

Experiments on public datasets show that CoExRec improves explanation generation over existing explanation baselines and achieves competitive recommendation performance. Ablation studies show that CF-aware cross-attention, explanation supervision, sequence-level CF context, and CF embeddings contribute to the joint objective. Further analyses show that RCS provides prediction-grounded evidence for reliable profile construction and stable target-specific explanation supervision.

Overall, this paper contributes CoExRec, a CF-grounded LLM-based framework for joint full-catalog sequential recommendation and target-specific explanation generation. CoExRec combines an RCS-based data construction pipeline, which converts prediction-relevant collaborative evidence into schematic user/item profiles and target-specific explanation supervision, with CF-conditioned cross-attention adapters that inject collaborative knowledge into a frozen LLM. We also provide detailed analyses of CF-grounded data construction, including evidence quality, counterfactual stability, prompt robustness, atom-level structure, and judge-based preference.

2 Related Works

Sequential recommendation. Sequential recommendation predicts the next item from a user's chronologically ordered interaction history. Early neural approaches model user behavior with recurrent or convolutional architectures (Jannach and Ludewig, 2017), while self-attentive models capture long-range dependencies among interacted items (Kang and McAuley, 2018). Recent studies further improve sequence modeling through contrastive learning, pretraining objectives, and generative formulations (Xie et al., 2022). These methods are effective because they learn collaborative patterns from user-item interactions, which remain difficult to replace with text alone. However, standard sequential recommenders are mainly optimized for ranking and do not directly provide natural-language explanations for individual target predictions.

LLM-based recommendation and explanation. Large language models have been increasingly adapted to recommendation because they can encode textual semantics, reason over item descriptions and reviews, and generate natural-language outputs (Wu et al., 2024). Existing approaches can be broadly viewed along several directions: using LLMs directly through prompting (Lyu et al., 2023), fine-tuning LLMs as recommenders (Geng et al., 2022a; Bao et al., 2023), enriching user or item representations, or building modular pipelines that combine LLMs with external recommendation components. However, prompt-only methods can be unstable, full fine-tuning is computationally expensive, and text-only modeling may not fully capture collaborative structure. Representation- or adapter-based methods connect LLMs with collaborative signals (Zheng et al., 2024; Kim et al., 2024; Huber et al., 2025), but aligning these signals with explanation supervision remains underexplored. In parallel, explainable recommendation methods generate textual justifications from reviews, user-item interactions, item features, or pretrained language models (Ni et al., 2019; Li et al., 2023b, 2017). Recent LLM-based approaches improve fluency and can construct synthetic supervision, but fluent explanations can still be weakly tied to the recommender's prediction process (Ma et al., 2024; Li et al., 2025). CoExRec differs by jointly performing next-item prediction and target-specific explanation generation, using RCS-filtered collaborative evidence to construct schematic profiles and explanation supervision.

3 Method

We describe CoExRec, a CF-grounded framework for joint sequential recommendation and explanation generation. Figure 1 gives an overview. CoExRec uses a single external sequential recommender in two ways. RCS from this recommender selects prediction-grounded evidence for profile and explanation construction, while its CF embeddings condition a frozen LLM through cross-attention adapters. The resulting model jointly performs full-catalog next-item prediction and explanation generation.

3.1 Problem Setup

Let $\mathcal{U}$ and $\mathcal{I}$ denote the sets of users and items. For each user $u \in \mathcal{U}$, we observe a chronological interaction history $H_u = (i^u_1, \ldots, i^u_{n_u})$, where each interaction is associated with a textual review. Following standard sequential recommendation practice, we reserve the last item $i^u_{n_u}$ for testing, the second-to-last item $i^u_{n_u-1}$ for validation, and use

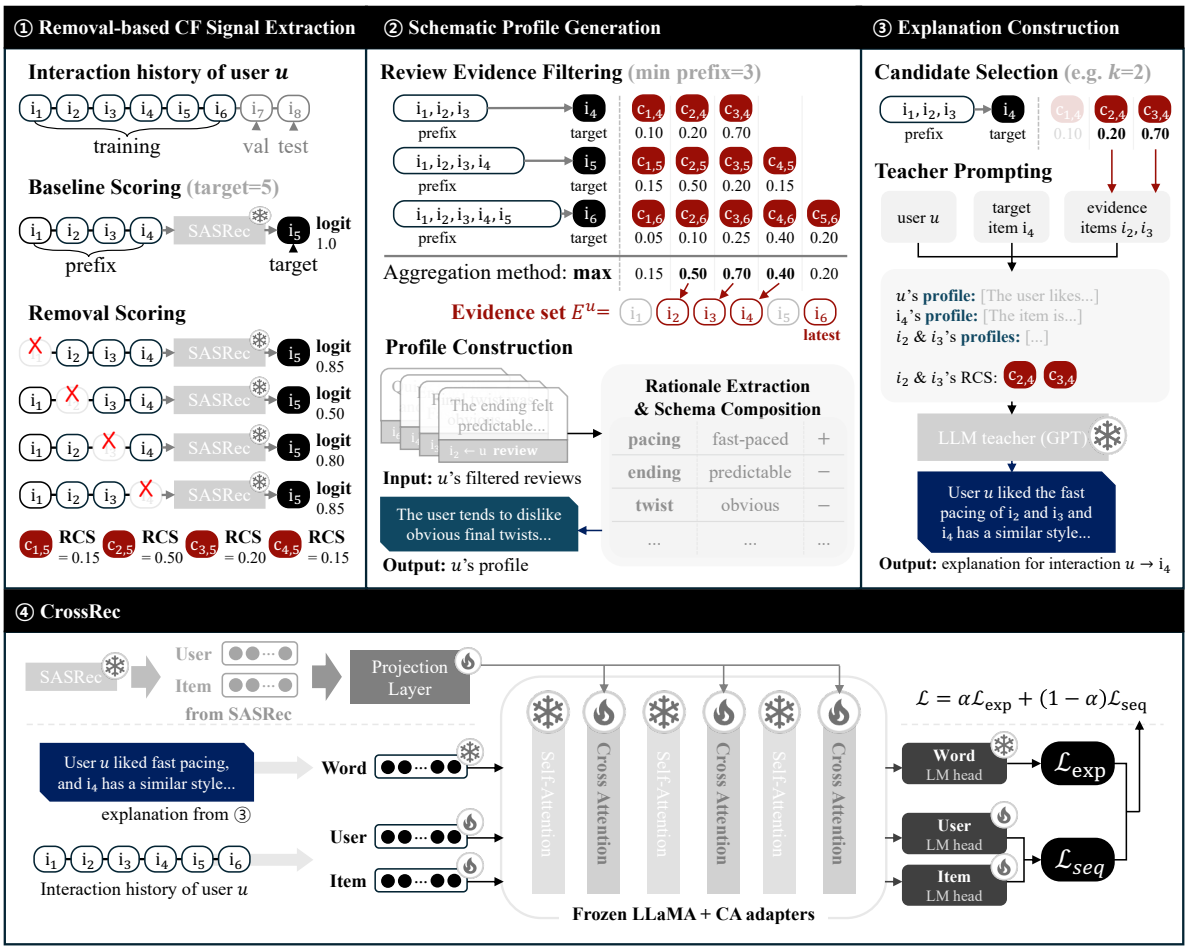

Figure 1: Overall architecture of CoExRec. CoExRec first extracts CF signals from a sequential recommender, uses RCS to construct profiles and explanation supervision, and injects CF embeddings into a frozen LLM through cross-attention adapters.

the remaining prefix $H_u^{\text{train}} = (i_1^u, \ldots, i_{n_u-2}^u)$ for training. All RCS computations, profiles, and explanation supervision are constructed from training-side interactions unless explicitly stated otherwise.

CoExRec jointly addresses full-catalog next-item prediction and target-specific explanation generation. During training, a target item is not restricted to the last item of the sequence. Instead, for a training-side position $t \leq n_u - 2$, the item i_t^u can serve as a target interaction, and only its preceding prefix $H_{<t}^u = (i_1^u, \ldots, i_{t-1}^u)$ is available as evidence. This prefix-only rule is used when computing RCS, constructing explanation supervision, and evaluating target-specific explanations, preventing future interactions from being used.

3.2 Removal-based Contribution Score

CoExRec uses an independent sequential recommender as a CF signal extractor. From this recommender, we obtain two complementary signals. The first is a removal-based contribution score (**RCS**), which selects collaborative evidence for profile and explanation construction. The second is a set of CF embeddings, which are later injected into the frozen LLM through cross-attention adapters. Thus, the same CF substrate is used both for data construction and for model conditioning.

As illustrated in the first block of Figure 1, RCS measures how much the target score decreases when each prefix item is removed. For a user u and a target position t, let $H_{<t}^u = (i_1^u, \ldots, i_{t-1}^u)$ denote the historical prefix. The recommender evaluates the relevance of the target item i_t^u with a scoring function $s_\theta(i_t^u \mid H_{<t}^u)$. To measure the contribution of a past item i_k^u with $k < t$, we remove it from the prefix and compute the target-score decrease. Let $H_{<t}^{u,-k}$ denote the prefix after removing i_k^u. We define the score change and the corresponding RCS as

$$\begin{aligned} \Delta_{k,t}^u &= s_\theta(i_t^u \mid H_{<t}^u) - s_\theta(i_t^u \mid H_{<t}^{u,-k}), \\ c_{k,t}^u &= \max(0, \Delta_{k,t}^u). \end{aligned} \tag{1}$$

Step 1: Evidence Filtering

Review	Item	Text
r_1^u	i_2	The mystery was **fast–paced** and kept me engaged, but the **ending felt predictable**.
r_2^u	i_3	I liked the **quick storyline**, although the **final twist was too obvious.**

Step 2: Structured Rationale Extraction

Review	Attribute	Value	Pref	Evidence
r_1^u	pacing	fast-paced	+	"fast-paced", "kept me engaged"
r_1^u	ending	predictable	–	"ending felt predictable."
r_2^u	speed	quick	+	"quick storyline"
r_2^u	final twist	obvious	–	"final twist was too obvious."

Step 3: Schema Composition

Canonical attribute	Values	Pref	Evidence
narrative pace	fast-paced, quick	+	"fast-paced", "kept me engaged", "quick storyline"
plot resolution	predictable, obvious	–	"ending felt predictable.", "final twist was too obvious."

Step 4: Profile Construction

The user prefers mystery books with a fast narrative pace and engaging detective elements, but tends to dislike predictable endings or obvious final twists.

Figure 2: Example of Schema Profile Generation

If removing an item increases the target score, the item does not provide positive evidence for the target, and its contribution is set to zero. We normalize the contribution scores over the prefix:

$$\tilde{c}_{k,t}^u = \frac{c_{k,t}^u}{\sum_{j<t} c_{j,t}^u + \epsilon}, \tag{2}$$

where ϵ is a small constant for numerical stability. The normalized score $\tilde{c}_{k,t}^u$ represents the relative contribution of item i_k^u to the prediction of i_t^u. We aggregate these scores for user-level profile construction and use them directly for target-specific explanation construction. CF embeddings from the same recommender are used for model conditioning in Section 3.5.

3.3 Schematic Profile Generation

Schematic profile generation turns CF-grounded evidence into compact textual profiles. Directly summarizing all reviews can introduce noise and hallucinated preferences, and it may obscure which reviews are relevant to the recommender's prediction process. We therefore construct user and item profiles through a four-stage pipeline: evidence filtering, rationale extraction, schema composition, and profile verbalization.

Review evidence filtering. Rather than relying on recency or random sampling, we use RCS to identify items that strongly contribute to the user's later interactions. As shown in the second block of Figure 1, this step aggregates target-conditioned RCS values into user-level evidence weights. For a user u and target position t with prefix $H_{<t}^u = (i_1^u, \ldots, i_{t-1}^u)$, we aggregate the normalized contribution scores $\tilde{c}_{k,t}^u$ of each prefix item i_k^u across admissible targets:

$$a_k^u = \mathrm{Agg}_{t:\, k<t}\, \tilde{c}_{k,t}^u. \tag{3}$$

The resulting value a_k^u represents how strongly item i_k^u contributed to later item predictions in the training sequence. To avoid underrepresenting the most recent training item $i_{n_u-2}^u$, which appears in the fewest prefix-target pairs, we explicitly include it in the evidence candidate set when it is missing. Its weight is set to the mean weight of the selected evidence items. We then select the top-$K(u)$ items ranked by a_k^u to form the user evidence set $\mathcal{E}_u$.

For item profiles, we also select a compact set of informative reviews written for the item. Since items do not have a sequential prefix in the same way as users, item-side evidence is selected using CF similarity between item embeddings and user representations from the same recommender.

Structured rationale extraction. For each selected review, an LLM teacher extracts preference statements as structured rationale units. For each selected review, an LLM teacher extracts structured rationale units with six fields: target, attribute, value, preference, context, and verbatim evidence, as illustrated in Figure 2. The evidence span must be copied from the source review, which reduces unsupported claims.

Schema composition and profile construction. The extracted rationale units may describe similar aspects with different surface forms, so the LLM canonicalizes attributes into shared names and groups rationales by target and canonical attribute. Each group stores supporting counts, preference distributions, and representative evidence spans. The grouped schema is then verbalized into a concise profile, making the output a controlled abstraction of RCS-filtered evidence rather than an open-ended review summary.

3.4 Explanation Construction

CoExRec learns explanation generation together with next-item prediction, but standard recommen-

dation datasets do not provide target-specific explanations. We therefore construct reference explanations with an LLM teacher using RCS-filtered evidence. For a target interaction i_t^u at position t, the available evidence is restricted to the historical prefix $H_{<t}^u$. This prefix-only rule prevents the teacher from using interactions that occur after the target item. Using the normalized RCS scores from Section 3.2, we select the top-K prefix items as the target-specific evidence set $\mathcal{E}^{u,t}$.

For each candidate (u, t), the teacher prompt includes the user profile, the target item profile, and the profiles of the selected evidence items. For each evidence item, the prompt also includes its RCS importance weight, so that the teacher can focus on the interactions most relevant to the target prediction. The teacher then generates a reference explanation e_t^u that describes why the target item is appropriate for the user based on the selected evidence. The generated explanations are used as labels for the explanation generation loss $\mathcal{L}_{\text{exp}}$.

3.5 CoExRec

CoExRec is a frozen LLM-based framework for joint sequential recommendation and explanation generation. It represents users and items with special tokens and injects CF signals through cross-attention adapters. As shown in the fourth block of Figure 1, this design follows the idea of multimodal LLMs, where a frozen language model attends to external representations. In CoExRec, the external representations are CF embeddings extracted from the sequential recommender.

Architecture. We extend the vocabulary with user and item tokens. Text tokens use the pretrained token embeddings, while user and item tokens are mapped to newly introduced trainable embedding tables. With this interface, next-item prediction is formulated as item-token prediction, and explanation generation is handled as natural-language generation. However, user and item tokens provide only a symbolic interface and do not directly encode the collaborative structure learned by the external recommender. To condition the LLM on this structure, CoExRec constructs a CF context for each training instance. Let $\mathbf{z}_u$, $\mathbf{z}_{i_t^u}$, and $\mathbf{z}_{i_k^u}$ denote the CF embeddings of user u, target item i_t^u, and history item i_k^u, respectively. For a history $H_{<t}^u$, we define

$$\mathbf{C}_{\text{CF}} = \left[W_u \mathbf{z}_u ; W_i \mathbf{z}_{i_t^u} ; \frac{1}{t-1} \sum_{k<t} W_i \mathbf{z}_{i_k^u} \right], \quad (4)$$

where W_u and W_i project CF embeddings into the LLM hidden dimension, and $[\cdot;\cdot]$ denotes concatenation.

The CF context is injected through cross-attention adapters inserted into selected LLM layers. At each adapter, the LLM hidden states attend to $\mathbf{C}_{\text{CF}}$ as external keys and values. The adapter produces CF-aware hidden states through gated cross-attention and a feed-forward sub-block, each with a residual connection. To prevent degradation of the pretrained language model, the LLM backbone remains frozen. The trainable parameters are limited to the user and item embedding tables, their output projections, the CF projection layers, and the cross-attention adapters. The adapters are activated only for recommendation instances with user/item tokens and CF context. For ordinary text-only inputs, CoExRec follows the original frozen LLM computation. Thus, CoExRec adds recommendation-specific functionality without overwriting the LLM's language knowledge.

Training objective. CoExRec is trained with a multi-task objective. The sequential recommendation loss $\mathcal{L}_{\text{seq}}$ is computed over next-item token predictions, and the explanation generation loss $\mathcal{L}_{\text{exp}}$ is computed over the reference explanation tokens. The final objective is $\mathcal{L} = \alpha \mathcal{L}_{\text{seq}} + (1-\alpha)\mathcal{L}_{\text{exp}}$, where $\alpha \in [0, 1]$ controls the balance between recommendation and explanation generation.

4 Experiments

We evaluate CoExRec from three perspectives. First, we compare its recommendation and explanation performance against conventional recommenders, LLM-based recommenders, and explanation generation baselines. Second, we conduct ablation studies to examine the contribution of the main components, including CF-aware cross-attention, explanation supervision, and pretrained CF initialization. Third, we analyze whether the CF signals used for data construction provide meaningful evidence for profiles and explanations.

4.1 Experimental Setup

We evaluate CoExRec on Amazon Books (McAuley et al., 2015; Ni et al., 2019), Yelp (Asghar, 2016), and Google Reviews (Li et al., 2022; Yan et al., 2023), which contain chronological interactions and review text. Following standard sequential recommendation protocols (Kang and McAuley, 2018; Sun et al.,

Table 1: Overall explanation generation results. We report BLEU-1 (B-1), BLEU-4 (B-4), ROUGE-1 F1 (R-1), USR, and FCR. Best and second-best results are marked in **bold** and *underline*. Full results are provided in Appendix **??**.

Models	Amazon Book					Yelp					Google Review				
	B-1	**B-4**	**R-1**	**USR**	**FCR**	**B-1**	**B-4**	**R-1**	**USR**	**FCR**	**B-1**	**B-4**	**R-1**	**USR**	**FCR**
Att2Seq	12.67	0.95	14.93	0.014	0.014	9.88	0.72	12.25	0.049	0.028	12.61	2.42	16.37	0.081	0.037
PETER	19.28	2.83	25.34	0.178	0.138	19.37	3.68	27.67	0.268	0.327	22.97	5.79	33.00	0.303	0.261
PEPLER	10.38	1.05	13.72	0.098	0.074	8.54	0.68	12.13	0.149	0.114	12.90	3.68	16.81	0.204	0.092
CER	19.37	2.63	25.29	0.185	0.151	19.31	3.73	28.35	0.230	0.319	23.28	**5.89**	33.19	0.342	0.272
XRec	24.10	2.09	18.81	0.822	0.261	10.18	0.82	8.65	0.571	0.264	23.14	2.93	19.92	0.851	0.245
Ours	**29.39**	**4.88**	**39.23**	**1.000**	**0.635**	**35.59**	**4.31**	**33.90**	**1.000**	**0.473**	**28.68**	4.99	**38.12**	**1.000**	**0.705**

Table 2: Overall sequential recommendation results on three datasets. We report Recall@k (R@k) and NDCG@k (N@k) for $k \in \{5, 10\}$. Best and second-best results are marked in **bold** and *underline*.

Models	Amazon Book				Yelp				Google Review			
	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
SASRec	0.0093	0.0129	0.0070	0.0082	0.0065	0.0116	0.0037	0.0054	0.0252	0.0446	0.0160	0.0222
BERT4Rec	0.0065	0.0114	0.0044	0.0060	0.0066	0.0132	0.0040	0.0061	0.0250	0.0427	0.0153	0.0209
ICLRec	0.0042	0.0073	0.0025	0.0034	0.0211	0.0382	0.0131	0.0185	0.0324	0.0358	0.0202	0.0271
TIGER	0.0336	0.0552	0.0222	0.0291	0.0264	0.0483	0.0161	0.0231	0.0242	0.0444	0.0145	0.0210
P5	0.0454	0.0585	0.0360	0.0402	0.0075	0.0119	0.0045	0.0059	0.0136	0.0236	0.0083	0.0115
LLaRA	0.0067	0.0101	0.0041	0.0053	0.004	0.0120	0.0031	0.0052	0.0031	0.0033	0.0014	0.0015
ReLRec	0.0343	0.0485	0.0231	0.0276	0.0122	0.0074	0.0250	0.0116	0.0225	0.0141	0.0392	0.0194
LLM-only	0.0200	0.0285	0.0153	0.0181	0.0176	0.0267	0.0112	0.0141	0.0062	0.0103	0.0039	0.0052
Ours	**0.0556**	**0.0883**	**0.0366**	**0.0471**	**0.0299**	**0.0514**	**0.0185**	**0.0252**	**0.0355**	**0.0656**	**0.0216**	**0.0312**

2019), we treat ratings of 3 or higher as implicit feedback, apply a 5-core filter, sort interactions chronologically, and use the last item for testing and the second-to-last item for validation. Recommendation performance is evaluated by ranking the ground-truth target against the full item catalog. All profiles, RCS computations, and explanation supervision are constructed only from training-side interactions to prevent temporal leakage (Luo et al., 2023). We report full-catalog Recall@K and NDCG@K for recommendation with $K \in \{5, 10\}$, and BLEU-1, BLEU-4, ROUGE-1, USR, and FCR for explanation generation. We use SASRec (Kang and McAuley, 2018) as the external sequential recommender and LLaMA-3.2-1B-Instruct (Grattafiori et al., 2024) as the frozen LLM backbone; only the user/item embeddings and heads, CF projection layers, and cross-attention adapters are trained. Profiles and reference explanations are generated with GPT-4o-mini, and pairwise judge evaluation uses GPT-4o (Kim et al., 2025). Further dataset statistics, implementation details, baseline settings, and full metric definitions are provided in Appendix **??**.

4.2 Main Experiments

Baselines. We compare CoExRec with baselines for both explanation generation and sequential recommendation. For explanation generation, we compare CoExRec with Att2Seq (Dong et al., 2017), PETER (Li et al., 2021), PEPLER (Li et al., 2023a), CER (Raczyński et al., 2023), and XRec (Ma et al., 2024). For XRec, we construct XRec-style supervision from our datasets using the data-generation procedure released with its original code, and evaluate its outputs against the same target-specific references as other explanation models. For sequential recommendation, we include conventional recommenders, including SASRec (Kang and McAuley, 2018), BERT4Rec (Sun et al., 2019), and ICLRec (Chen et al., 2022), and LLM-based or generative recommenders, including P5 (Geng et al., 2022b), TIGER (Rajput et al., 2023), LLaRA (Liao et al., 2024), and ReLRec (Na et al., 2024). We also include an *LLM-only* variant to test whether the gains come from the proposed CF-grounded recommendation components rather than from the LLM backbone alone. All methods use the same train, validation, and test splits; full baseline details are provided in Appendix **??**.

Table 3: Ablation study of CoExRec on recommendation and explanation generation. We report Recall@10 (R@10), NDCG@10 (N@10), BLEU-4 (B-4), and ROUGE-1 F1 (R-1).

Model Variants	Amazon Book				Yelp				Google Review			
	R@10	N@10	B-4	R-1	R@10	N@10	B-4	R-1	R@10	N@10	B-4	R-1
Ours	0.0883	0.0471	4.88	39.23	0.0514	0.0252	4.31	33.90	0.0656	0.0312	4.99	38.12
w/o CF adapter	0.0294	0.0159	4.93	35.44	0.0369	0.0187	3.23	30.13	0.0134	0.0062	4.97	35.00
w/o Explanation	0.0286	0.0151	–	–	0.0396	0.0204	–	–	0.0177	0.0086	–	–
w/o Sequence Context	0.0730	0.0385	4.99	35.48	0.0472	0.0240	3.22	30.12	0.0367	0.0172	4.96	34.97
Random CF Init.	0.0708	0.0435	4.89	35.45	0.0356	0.0178	3.21	30.09	0.0356	0.0177	4.94	35.00

Explanation performance. Table 1 reports the explanation generation results. CoExRec achieves the strongest performance on most reported overlap metrics across the three datasets, including BLEU-1 and ROUGE-1, and also obtains high USR and FCR scores. These results suggest that CF-grounded target-specific supervision helps CoExRec generate explanations that are more tied to the recommendation decision than short review-like descriptions. The reference explanations are constructed from RCS-filtered evidence, schematic user and item profiles, and the target item.

Recommendation performance. Table 2 reports the full-catalog recommendation results. CoExRec achieves strong performance against both conventional sequential recommenders and LLM-based recommendation models, while also supporting target-specific explanation generation. This shows that CoExRec is not an explanation-only model, but a joint framework for full-catalog next-item prediction and explanation generation. The gains over LLM-only and other LLM-based baselines support the use of CF embeddings as model conditioning beyond symbolic user and item tokens.

4.3 Ablation Study

Table 3 evaluates four variants: removing the CF adapter, removing explanation supervision, removing the sequence-level CF context, and replacing pretrained CF embeddings with random initialization. Removing the CF adapter causes the largest recommendation drop across datasets, showing that symbolic user/item tokens alone do not sufficiently transfer collaborative knowledge into the frozen LLM. Removing explanation supervision also lowers recommendation performance, suggesting that target-specific explanation learning provides useful auxiliary supervision. The sequence-context and random-initialization variants further show the benefit of history-level CF context and pretrained collaborative structure. Overall, these results show that CF-aware conditioning and explanation supervision jointly support CoExRec's two objectives.

4.4 CF-Grounded Data Construction

We analyze whether the CF signals used for data construction provide meaningful and stable evidence. CoExRec does not build profiles or explanations from unfiltered review histories Instead, it selects compact CF-grounded evidence and converts it into structured profiles and target-specific explanation supervision. We compare RCS with MF, Naive, and Attention on a fixed stratified sample of 400 users from each dataset, while keeping the downstream rationale extraction, schema composition, profile construction, and explanation construction procedures fixed. MF selects evidence by embedding similarity, Naive samples uniformly from the history, and Attention uses attention weights from the sequential recommender. Full protocols are provided in Appendix D.

Profile construction. Table 4 evaluates whether the selected evidence supports reliable profile construction. We consider two aspects: quality and stability. For quality, centroid quality measures how close the selected evidence centroid is to the full-history CF centroid, which serves as an upper-bound representation of the user's training history. We also use pairwise LLM-as-judge evaluation to compare profiles generated from RCS evidence against profiles generated from each alternative selector. For stability, CF shift measures how much the selected evidence representation changes after counterfactual replacement, and atom drop measures how many structured rationale units disappear after perturbing the evidence.

RCS achieves the highest centroid quality on all three datasets, showing that it selects evidence closest to the full-history CF signal. This supports the use of RCS as a prediction-grounded evidence

Table 4: Profile analysis of CF-grounded evidence selection. Judge win compares RCS against each baseline. ↑ / ↓ indicate higher / lower is better.

Type	Metric	Amazon Book				Yelp				Google Review			
		RCS	**MF**	**Naive**	**Attn.**	**RCS**	**MF**	**Naive**	**Attn.**	**RCS**	**MF**	**Naive**	**Attn.**
Quality	Centroid ↑	**0.718**	0.679	0.698	0.682	**0.673**	0.627	0.654	0.645	**0.834**	0.760	0.767	0.745
	Judge win ↑	–	0.543	0.550	0.646	–	0.570	0.508	0.533	–	0.632	0.640	0.502
Stability	CF shift ↓	**0.158**	0.164	0.175	0.171	**0.175**	0.177	0.186	0.186	**0.090**	0.125	0.138	0.142
	Atom drop ↓	**0.179**	0.198	0.198	0.196	0.722	0.731	0.725	**0.604**	**0.147**	0.190	0.739	0.187

Table 5: Analysis of CF-grounded explanation construction. Judge win compares RCS against each variant. ↑ / ↓ indicate higher / lower is better.

Variant	Judge Win ↑			CF Drop ↓			ESS ↑		
	Amazon	**Yelp**	**Google**	**Amazon**	**Yelp**	**Google**	**Amazon**	**Yelp**	**Google**
RCS	–	–	–	**0.004**	**0.006**	**0.004**	0.841	**0.859**	**0.855**
↪ w/o Score	0.648	0.618	0.738	0.154	0.026	0.084	0.802	0.814	0.808
MF	0.643	0.550	0.666	0.009	**0.006**	0.009	**0.852**	0.856	**0.855**
Attention	0.638	0.565	0.658	0.019	**0.006**	0.008	**0.852**	0.858	0.852
Naive	0.995	1.000	1.000	0.557	0.563	0.538	0.628	0.628	0.629

selector, rather than a text relevance or recency-based selector. The judge results also show that RCS-based profiles are preferred over profiles constructed with MF, Naive, or Attention evidence. The comparison with Attention is close on Google Review, suggesting that different selectors can produce similarly acceptable profiles in some domains. For stability, RCS gives the lowest CF shift under random-global replacement and the lowest atom drop rate among the available methods. Overall, RCS provides strong CF-level evidence quality while maintaining stable profile construction.

Explanation construction. Table 5 analyzes the construction of target-specific explanation supervision. Unlike the profile analysis, which evaluates user-level evidence aggregated across training targets, this analysis focuses on target-conditioned evidence for each explanation instance. We compare RCS with RCS w/o Score, MF, Attention, and Naive. RCS w/o Score uses the same evidence items as RCS but removes the importance scores from the teacher prompt. This variant isolates whether the teacher benefits from the RCS scores themselves, beyond the selected evidence items.

RCS clearly outperforms Naive in judge preference and shows much lower CF drop, confirming that randomly selected evidence is not sufficient for target-specific explanation construction. Compared with RCS w/o Score, RCS gives lower CF drop on all datasets, showing that the teacher benefits from the RCS importance scores in addition to the selected evidence. MF and Attention are closer to RCS in judge preference, indicating that alternative CF-based evidence can also provide useful context. However, RCS remains the most stable under counterfactual evidence perturbation. Overall, RCS provides both target-conditioned evidence and salience information for constructing stable explanation supervision.

5 Conclusion

We presented CoExRec, a CF-grounded framework for joint sequential recommendation and explanation generation. CoExRec uses removal-based contribution scores to construct schematic profiles and target-specific explanation supervision, and conditions a frozen LLM on CF embeddings through cross-attention adapters. Experiments show that CoExRec improves explanation generation over existing baselines while achieving strong recommendation performance. Ablations and analyses further show that CF-aware conditioning and RCS-based evidence selection contribute to reliable recommendation-grounded explanations. These results suggest that collaborative signals can serve as a useful bridge between LLM-based text generation and sequential recommendation.

Limitations

The limitations of the proposed model, CoExRec, can be summarized into three fold: dependence on the quality of the external sequential recommender, additional offline computation for removal-based scoring, and reliance on textual reviews and LLM-generated supervision. Although the limitations may affect scalability and generalizability, they are not fundamental restrictions of the framework itself. In future work, we plan to improve the aforementioned aspects through more efficient contribution estimation and richer multimodal evidence integration.

References

Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM conference on recommender systems*, pages 1007–1014.

Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM web conference 2022*, pages 2172–2182.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 623–632.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022a. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022b. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM conference on recommender systems*, pages 299–315.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Bernd Huber, Ghazal Fazelnia, Andreas Damianou, Sebastian Peleato, Max Lefarov, Praveen Ravichandran, Marco De Nadai, Mounia Lalmas-Roellke, and Paul N Bennett. 2025. Embedding-to-prefix: Parameter-efficient personalization for pre-trained large language models. *arXiv preprint arXiv:2505.17051*.

Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 306–310.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.

Sein Kim, Hongseok Kang, Seungyoon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. 2024. Large language models meet collaborative filtering: An efficient all-round llm-based recommender system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1395–1406.

Sein Kim, Hongseok Kang, Kibum Kim, Jiwan Kim, Donghyun Kim, Minchul Yang, Kwangjin Oh, Julian McAuley, and Chanyoung Park. 2025. Lost in sequence: Do large language models understand sequential recommendation? In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 1160–1171.

Guanrong Li, Haolin Yang, Xinyu Liu, Zhen Wu, and Xinyu Dai. 2025. Counterfactual language reasoning for explainable recommendation systems. *arXiv preprint arXiv:2503.08051*.

Jiacheng Li, Jingbo Shang, and Julian McAuley. 2022. Uctopic: Unsupervised contrastive learning for phrase representations and topic mining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6159–6169.

Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1419–1428.

Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized transformer for explainable recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4947–4957.

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

Lei Li, Yongfeng Zhang, and Li Chen. 2023b. Prompt distillation for efficient llm-based recommendation.

In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 1348–1357.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-recommendation assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1785–1795.

Yibin Liu, Ang Li, and Shijian Li. 2025. Hierarchical interaction summarization and contrastive prompting for explainable recommendations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 485–501. Springer.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Christopher Leung, Jiajie Tang, and Jiebo Luo. 2023. Llm-rec: Personalized recommendation via prompting large language models. *arXiv preprint arXiv:2307.15780*.

Qiyao Ma, Xubin Ren, and Chao Huang. 2024. Xrec: Large language models for explainable recommendation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 391–402.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52.

Hyunsoo Na, Minseok Gang, Youngrok Ko, Jinseok Seol, and Sang-goo Lee. 2024. Enhancing large language model based sequential recommender systems with pseudo labels reconstruction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7213–7222.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197.

Jakub Raczyński, Mateusz Lango, and Jerzy Stefanowski. 2023. The problem of coherence in natural language explanations of recommendations. In *ECAI*.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, and 1 others. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, and 1 others. 2024. A survey on large language models for recommendation. *World Wide Web*, 27(5):60.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 1259–1273. IEEE.

An Yan, Zhankui He, Jiacheng Li, Tianyang Zhang, and Julian McAuley. 2023. Personalized showcases: Generating multi-modal explanations for recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2251–2255.

Shenghao Yang, Weizhi Ma, Peijie Sun, Qingyao Ai, Yiqun Liu, Mingchen Cai, and Min Zhang. 2024. Sequential recommendation with latent relations based on large language model. *arXiv preprint arXiv:2403.18348*.

Jingsen Zhang, Zihang Tian, Xueyang Feng, Xu Chen, and Chong Chen. 2025. Enhancing recommendation explanations through user-centric refinement. *arXiv preprint arXiv:2502.11721*.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1435–1448. IEEE.

A Example Appendix

Example of evidence filtering. Consider a user training sequence $(i_1, \ldots, i_7)$. For profile construction, we compute target-conditioned RCS scores for multiple training-side targets. For each target item, only its prefix is used as candidate evidence. Table **??** shows an example with three target items. After aggregating the scores across targets, the top-ranked items are selected as evidence reviews for schematic profile generation.

B Profile Generation

C Explanation Generation

D Data Analysis

,

You are an information extraction system. Your task is to extract structured atomic facts ("atoms") from item reviews. You must strictly follow the provided schema and rules. You must not add explanations, commentary, or extra text. You must not infer or invent information beyond the review text. You must return ONLY valid JSON.

IMPORTANT CONCEPTS: The review describes ONE reviewed item. The target is the TYPE of the reviewed item (review-level), such as: restaurant, guided_tour, book, hotel, attraction, business. ALL atoms in the same review MUST share the SAME target. The goal is to extract factual, explainable preference signals that can later be aggregated into user/item pseudo-profiles. Prefer precision and faithfulness over completeness.

WHAT TO EXTRACT: Each atom represents ONE clear, self-contained preference statement explicitly supported by the review text. Return ONLY valid JSON with this schema:

```
{
  "atoms": [
    {
      "target": "string (e.g., guided_tour)",
      "attribute": "string",
      "value": "string",
      "preference": "positive | negative | neutral",
      "context": "string",
      "evidence": "verbatim short quote from the review text"
    }
  ]
}
```

DETAILED RULES

TARGET: First infer the reviewed item's type, use this SAME target value for ALL atoms, and do NOT use staff, guide, environment, location, or price as target.
ATTRIBUTE: Attribute describes WHAT aspect of the item is evaluated. Use abstract, reusable aspect names (e.g., overall, price, content, convenience, facilities, organization, duration, route), prefer a small consistent vocabulary, and choose the closest existing abstract aspect rather than inventing a new one.
VALUE: Value describes HOW the aspect is judged. Use concise qualitative descriptors (e.g., enjoyable, great, rich, high, acceptable), do NOT copy full sentences, and keep value descriptive, not strongly sentiment-laden.
PREFERENCE: positive for explicit enjoyment, praise, or recommendation; negative for complaint, dissatisfaction, or warning; neutral for factual mention without clear sentiment.
CONTEXT: Context explains UNDER WHAT SITUATION the judgment applies. Use short situational phrases (e.g., first_time_visit, groupon, history_learning, pre_tour_meeting), and do NOT restate the attribute or value.
EVIDENCE: Evidence MUST be a short, verbatim quote from the review text. Do NOT paraphrase or summarize.
ATOM SELECTION: Extract ALL clearly supported atoms (typically 3–8), do NOT invent information, do NOT merge unrelated aspects into one atom, and prefer precision over completeness.

Now extract atoms from the following review. Return ONLY the JSON object.

{"uid": 27, "iid": 27, "user_id": "…", "item_id": "…", "rating": 5.0, "time": "2015-12-05 05:09:27", "atoms": [{"target": "restaurant", "attribute": "overall", "value": "amazing", "preference": "positive", "context": "family_visit", "evidence": "the food is amazing"}, {"target": "restaurant", "attribute": "facilities", "value": "friendly", "preference": "positive", "context": "family_visit", "evidence": "Super friendly staff"}, {"target": "restaurant", "attribute": "overall", "value": "great", "preference": "positive", "context": "family_visit", "evidence": "Great with kids"}, {"target": "restaurant", "attribute": "price", "value": "acceptable", "preference": "positive", "context": "general", "evidence": "it's not hard on the wallet"}]}

Figure 3: Prompt for User Profile Step 1.

You are a precise information processing assistant. You must follow these rules: Output must be valid JSON only. No extra text. Never invent new facts or atoms. Never modify target, preference, context, or evidence. Your job is only to canonicalize attribute names (and optionally values) and provide mappings by atom_id. If unsure, choose the closest existing canonical attribute rather than creating a new one. You are given a list of extracted review atoms in JSON. Each atom already contains: target, attribute, value, preference, context, evidence Your task is NOT to extract new atoms. Your task is to STANDARDIZE (canonicalize) the attribute names (and optionally values) so that downstream grouping/aggregation can be done deterministically.

IMPORTANT CONSTRAINTS: Do NOT change target, preference, context, or evidence. Do NOT invent new information. Only output valid JSON (no markdown, no extra text). Canonical attributes should be a SMALL, reusable vocabulary. Prefer mapping to an existing canonical attribute if possible. If you introduce a new canonical attribute, it must be strongly justified by multiple atoms.

OUTPUT JSON SCHEMA

```
{
  "canonical_attributes": [
    {
      "name": "string_snake_case",
      "description": "short description of what this attribute captures"
    }
  ],
  "canonical_values": [
    {
      "name": "string_snake_case",
      "description": "optional: value vocabulary description"
    }
  ],
  "mappings": [
    {
      "atom_id": int,
      "canonical_attribute": "string_snake_case",
      "canonical_value": "string_snake_case (optional)"
    }]
}
```

CANONICALIZATION GUIDELINES:
Canonical Attribute: Merge synonyms and surface variants into the same canonical attribute. Example: "price", "cost", "expensive", "value for money" -> price_value. Keep the vocabulary abstract and reusable across items. Use snake_case. Avoid overly specific attributes unless truly necessary.
Canonical Value (optional but recommended when feasible): Normalize value variants into a small set if it is natural. Example: "too high", "overpriced", "expensive" -> high; "cheap", "low cost", "affordable" -> low; "okay", "fine", "acceptable" -> okay. If values are too diverse or domain-specific, omit canonical_value for that atom.
Coverage: Every atom MUST receive a canonical_attribute in mappings.

INPUT ATOMS JSON: {atoms_json}

{"uid": 27, "atom_count": 31, "canonicalization": {"canonical_attributes": [{"name": "overall", "description": "general satisfaction with the restaurant"}, {"name": "facilities", "description": "quality of the restaurant's facilities"}, {"name": "price", "description": "affordability of the restaurant"}, {"name": "organization", "description": "efficiency and cleanliness of the restaurant"}, {"name": "food", "description": "quality of the food served"}, {"name": "menu", "description": "variety and quality of the menu options"}], "mappings": [{"atom_id": 858, "canonical_attribute": "overall", "canonical_value": "positive_generic"}, {"atom_id": 859, "canonical_attribute": "facilities", "canonical_value": "positive_generic"}, ⋯, {"atom_id": 888, "canonical_attribute": "food", "canonical_value": "positive_generic"}]}, "atoms": [{"atom_id": 858, "target": "restaurant", "attribute": "overall", "value": "amazing", "preference": "positive", "context": "family_visit", "evidence": "the food is amazing", "item_id": "…", "iid": 27, "rating": 5.0, "time": "2015-12-05 05:09:27", "user_id": "…"}, ⋯, {"atom_id": 888, "target": "restaurant", "attribute": "food", "value": "tasty", "preference": "positive", "context": "peking_duck_experience", "evidence": "the Peking duck was tasty", "item_id": "…", "iid": 13104, "rating": 4.0, "time": "2019-06-16 00:38:10", "user_id": "…"}]}

Figure 4: Prompt for User Profile Step 2.

You are a profile rendering system for a recommendation model. Your task is to generate a concise natural-language profile based ONLY on the provided evidence and metadata.

Hard constraints: Do NOT invent new facts. Do NOT add attributes, values, or contexts not supported by evidence. Do NOT exaggerate strength beyond what the evidence supports. Do NOT mention atom IDs, group IDs, counts, or internal data structures. Do NOT mention "reviews" explicitly.

Writing style: 2–5 sentences. Neutral, factual tone. No marketing language. No first-person narration.
Output format: Return ONLY valid JSON: {"profile": "..."} (no extra keys, no extra text).

You will be given a JSON object containing:
- profile_type: "user" or "item"
- evidence_type: "groups", "atoms", or "reviews"
- optional metadata (item-side fields may exist)
- evidence:
  - if evidence_type="groups": aspect-level grouped evidence with distributions and a few evidence_samples
  - if evidence_type="atoms": raw structured atoms (target/attribute/value/preference/context/evidence)
  - if evidence_type="reviews": raw user reviews (item_id, rating, free-form text)

How to write the profile:
- If profile_type="user":
  - Summarize stable preferences/tendencies reflected by evidence.
- If profile_type="item":
  - Summarize commonly perceived characteristics reflected by evidence.

How to use evidence:
- If evidence_type="groups":
  - Focus on attributes with stronger support (larger group size or repeated patterns).
  - Use dominant contexts/values, but do NOT overclaim.
  - You may be moderately confident when multiple groups support the same theme.
- If evidence_type="atoms":
  - Be conservative when evidence is sparse or conflicting.
  - When multiple atoms support the same theme, state it directly rather than hedging with vague qualifiers ("mixed", "may vary").
  - Use hedging ("tends to", "often") only when evidence genuinely conflicts or is thin.
- If evidence_type="reviews":
  - Infer preferences ONLY from what the reviews literally say.
  - Pay attention to repeated themes across multiple reviews; state them directly when supported.
  - Be conservative only when reviews genuinely conflict; do NOT add empty hedging when reviews agree.
  - Do NOT mention the word "reviews" in the output, and do NOT quote review text directly.

Strict: Use ONLY what appears in the input JSON. Now generate the profile.

INPUT JSON: <<INPUT_JSON>>

{"uid": 27, "profile_type": "user", "evidence_type": "groups", "mode": "ta", "profile": "The user tends to prefer restaurants with friendly and welcoming staff, often described as family-friendly. They appreciate homemade and tasty food, with positive mentions of specific dishes like lemon dessert and Peking duck. While they find the overall experience satisfactory and enjoyable, there are concerns regarding organization, particularly related to wait times and cleanliness. The user also values a decent kids menu and considers the pricing to be acceptable."}

Figure 5: Prompt for User Profile Step 3.

{"uid": 27, "iid": 7193, "system": "You are a recommendation explanation model. Given a user's history items (with profiles) and importance weights derived from a recommender, explain why the user would like the target item in a way that is faithful to these importance weights.", "user": "User: uid_27 - The user tends to prefer restaurants with friendly and welcoming staff, often described as family-friendly. They appreciate homemade and tasty food, with positive mentions of specific dishes like lemon dessert and Peking duck. While they find the overall experience satisfactory and enjoyable, there are concerns regarding organization, particularly related to wait times and cleanliness. The user also values a decent kids menu and considers the pricing to be acceptable.

History:
iid_11809: The restaurant is often described as having a great atmosphere, with a cute and artistic touch. The food is generally perceived as delicious and fresh, with positive remarks about its healthiness, particularly for breakfast and lunch options. However, the facilities have mixed feedback, with some noting a need for upgrades and limited parking. Service has been criticized for being inconsistent, while the portions and prices are viewed positively. [importance=0.0282],
iid_10464: This restaurant is often described as dependable and delicious, with a menu that includes fantastic and tasty dishes. The owner is noted to be charming, and the service is characterized as friendly. Customers find the prices to be fair, contributing to a positive overall dining experience. [importance=0.7438],
iid_3903: This restaurant is often described as having a comfy ambience and great service. The food is perceived as authentic and fresh, with some positive remarks about its tenderness, though there are mentions of muted flavors and under-spicing. Overall, the experience is generally enjoyable and impressive, with staff noted for being kind, friendly, and attentive. [importance=0.1173],
iid_2528: This restaurant is often described as having great food, with specific dishes like dry cauliflower manchurian and chicken tikka masala receiving positive remarks. Overall, it is perceived as delicious and a good option for takeout, with a strong local presence. [importance=0.0845],
iid_6623: The restaurant is often described as having outstanding and modern decor. The food quality is generally perceived positively, with fresh ingredients noted, although some find certain dishes too sweet. Service is consistently rated as excellent and attentive. However, the dining experience can be confusing at times, particularly regarding the organization and the presentation of food. Additionally, the music is reported to be too loud, which may detract from the overall experience. [importance=0.0148]

Target item: iid_7193: The restaurant is often described as cozy, particularly appealing during winter months. It provides a nostalgic experience reminiscent of a roadside store, with a focus on breakfast and specific dishes. The food is generally viewed positively, with descriptors like 'tasty' and 'great', although there are some negative mentions regarding temperature and quality. Overall, the restaurant is considered superb and affordable, with variable service quality noted.

Write a concise explanation (3-5 sentences) for why the user would like the target item, explicitly grounding your reasoning in the most important history items according to their importance weights.Do not write importance weights in the explanation.", "gpt_explanation": "You would likely enjoy this restaurant because it offers a cozy and welcoming atmosphere, reminiscent of your preference for family-friendly environments with friendly staff. The focus on tasty food aligns with your appreciation for homemade meals, similar to the dependable and delicious dishes you've liked in your history. Additionally, the affordability of the restaurant echoes your value for reasonable pricing, enhancing your overall dining experience. While there are some concerns about service quality, the positive aspects of the food and ambiance may outweigh these, making it a potential favorite for you."}

Figure 6: Explanation Construction Prompt

You are an expert evaluator for personalized user profiles in recommendation systems.
Your task: given a user's review history AND a gold reference profile (generated from the user's complete review history), decide which of two candidate profiles better matches the gold reference.
Evaluate the profiles using the following criteria:

1. Faithfulness: The profile must accurately reflect the user's actual opinions, preferences, and experiences in the review history. Penalize hallucinated facts, unsupported preferences, exaggerated claims, or contradictions.

2. Specificity: The profile should include concrete and informative preference details, such as item types, genres, attributes, use cases, quality concerns, disliked features, or repeated behavioral patterns. Penalize vague statements such as "the user likes high-quality products" unless they are clearly supported and made more specific.

3. Coverage: The profile should cover the main preference patterns across the review history, not just one or two isolated reviews. Penalize profiles that ignore important recurring interests, dislikes, constraints, or item categories.

4. Personalization: The profile should describe this particular user, not a generic user. Penalize profiles that could apply to almost anyone or simply summarize common preferences.

5. Usefulness for recommendation: The profile should help a recommender infer what kinds of items the user may like or dislike in the future. Prefer profiles that identify actionable preferences and constraints.

6. Clarity and conciseness: The profile should be easy to understand and not unnecessarily verbose. Do not favor a longer profile if it adds unsupported, redundant, or generic content.

Rules: Use the gold reference as the upper-bound standard of quality. Base your judgement on both the review history AND how well each candidate matches the gold reference. Do NOT favour longer profiles — quality and alignment with the gold reference matter more than length. If the candidates are truly indistinguishable relative to the gold reference, output "tie". Output ONLY a valid JSON object — no extra text, no markdown.

Output ONLY a valid JSON object with the following schema:
{
"winner": "A" | "B" | "tie",
"faithfulness": "A" | "B" | "tie",
"specificity": "A" | "B" | "tie",
"coverage": "A" | "B" | "tie",
"personalization": "A" | "B" | "tie",
"usefulness": "A" | "B" | "tie",
"clarity_conciseness": "A" | "B" | "tie",
"reasoning": "<1-2 sentence explanation grounded in the review history>"
}

Figure 7: LLM Judge Prompt