DUET: Joint Exploration of User–Item Profiles in Recommendation System

Yue Chen[1,*], Yifei Sun[1,*], Lu Wang[2,†], Fangkai Yang[2], Pu Zhao[2], Minjie Hong[3], Yifei Dong[4], Minghua He[2], Nan Hu[2], Jianjin Zhang[2], Zhiwei Dai[2], Yuefeng Zhan[2], Weihao Han[2], Hao Sun[2], Qingwei Lin[2], Weiwei Deng[2], Feng Sun[2], Qi Zhang[2], Saravan Rajmohan[2], Dongmei Zhang[2]

[1]Peking University [2]Microsoft [3]Zhejiang University [4]KTH Royal Institute of Technology
[*]Equal contribution [†]Corresponding author

Abstract

Traditional recommendation systems represent users and items as dense vectors and learn to align them in a shared latent space for relevance estimation. Recent LLM-based recommenders instead leverage natural-language representations that are easier to interpret and integrate with downstream reasoning modules. This paper studies how to construct effective *textual profiles* for users and items, and how to align them for recommendation. A central difficulty is that the best profile format is not known a priori: manually designed templates can be brittle and misaligned with task objectives. Moreover, generating user and item profiles independently may produce descriptions that are individually plausible yet semantically inconsistent for a specific user–item pair. We propose DUET, an interaction-aware profile generator that jointly produces user and item profiles conditioned on both user history and item evidence. DUET follows a three-stage procedure: it first turns raw histories and metadata into compact cues, then expands these cues into paired profile prompts and then generate profiles, and finally optimizes the generation policy with reinforcement learning using downstream recommendation performance as feedback. Experiments on three real-world datasets show that DUET consistently outperforms strong baselines, demonstrating the benefits of template-free profile exploration and joint user–item textual alignment. Project page: https://duet-rec.github.io/.

1 Introduction

Traditional recommendation systems represent users and items as dense vectors and learn to align them in a shared latent space for relevance estimation (Covington et al., 2016; Wu, 2023). While effective, such embeddings are opaque: they offer limited interpretability and make it difficult to analyze why an item is recommended. Recent work therefore leverages large language models (LLMs) to introduce semantically rich, human-readable

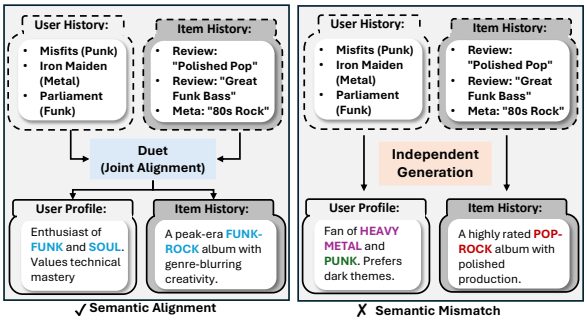

Figure 1: DUET aligns raw user and item data by transforming them into textual profiles within a shared semantic space.

representations for recommendation (Wang et al., 2025; Zhang, 2024; Bao et al., 2023; Hong et al., 2025a,b,c; Wang et al., 2024). A natural direction is to replace latent vectors with textual user and item profiles that can be inspected, edited, and reused by downstream components incorporated with LLMs. However, existing LLM-based approaches remain limited in two important ways. First, directly prompting an LLM with raw user and item histories to obtain recommendations often yields noisy and incomplete signals, especially when histories are long, sparse, or heterogeneous (Wang et al., 2025). Second, profile-based methods typically rely on manually designed templates or handcrafted attributes, which requires substantial human engineering and constrains the representation space. More fundamentally, many approaches generate user profiles and item profiles independently, without modeling how user preferences and item semantics interact at decision time (Yang et al., 2023; Xi et al., 2024).

To address these challenges, we propose DUET, a joint user–item profile generator that takes both user history and item history as input and produces a paired set of profiles for the interaction. Crucially, DUET does not require profile templates: it is trained with reinforcement learning using feedback from downstream recommendation perfor-

Findings of the Association for Computational Linguistics: ACL 2026, pages 21392–21406
July 2-7, 2026

mance, enabling it to explore and discover effective profile formats automatically. Figure 1 illustrates why joint profiling matters. The user has listened to Misfits (punk), Iron Maiden (metal), and Parliament (funk), while the candidate album is described by reviews such as "polished pop" and "great funk bass" and a meta tag "'80s rock." Considering both sides together, DUET can reconcile these signals into a compatible interpretation (e.g., highlighting the user's funk/soul affinity and the item's funk-rock character). In contrast, independently generated profiles may amplify different facets, summarizing the user as "heavy metal/punk" while summarizing the item as "pop-rock", which resulting in a semantically mismatched pair that obscures the true relevance signal.

DUET proceeds in three stages. First, raw histories and metadata are distilled into minimal *cues* that capture compact but informative signals. Second, the model expands cues into richer profile prompts to generate textual profiles, allowing exploration over alternative profile structures and emphases. Third, the resulting profiles are consumed by downstream recommenders, and their task feedback is used to optimize the profile generation policy. By coupling both sides in a shared semantic space, DUET learns user profiles that reflect what kinds of items a user prefers and item profiles that reflect what kinds of users an item appeals to.

Our contributions are as follows:

- We represent users and items as natural-language profiles and align them in a shared semantic space, extending the classic vector-based alignment principle to interpretable textual representations.
- We introduce an exploration-based framework that starts from cue-based initialization, expands cues into candidate profiles, and *jointly* optimizes user and item profiles with downstream recommendation feedback via reinforcement learning, avoiding rigid templates.
- Extensive experiments across multiple real-world datasets show that DUET consistently outperforms strong baselines, validating both joint profiling and feedback-driven profile optimization.

2 Related Work

2.1 Profiles in Recommendation

Early recommendation systems primarily relied on pre-defined profiles based on structured attributes, as seen in works like CRESDUP (Chen et al., 2007) and UPCSim (Widiyaningtyas et al., 2021). While foundational, these methods were limited by their rigid, hand-engineered features. More recently, the advent of Large Language Models (LLMs) has enabled a shift towards generating profiles in natural language. Studies such as KAR (Xi et al., 2024), GPG (Zhang, 2024), PALR (Yang et al., 2023), and LettinGo (Wang et al., 2025) leverage LLMs to create textual user profiles from behavioral data. A key limitation of these approaches is twofold: they rely on static or **pre-defined templates** for profile generation, and they often focus exclusively on user profiles, neglecting the rich, expressive information inherent in items and the complex dynamics of user-item interactions. In contrast, our work introduces a new paradigm where both user and item profiles are not fixed but are dynamically explored and optimized in a shared semantic space to directly align with recommendation performance.

2.2 Reinforcement Learning for LLM-Based Recommendation Systems

Reinforcement Learning (RL), particularly through techniques like RLHF, has become a core method for aligning LLMs with specific objectives. This approach has been adapted for recommendation tasks (Wang et al., 2025; Lin et al., 2025; Deng et al., 2025), but existing efforts often face key limitations. They frequently rely on offline reward models (Jeong et al., 2023; Chen et al., 2025; He et al., 2025; Liu et al., 2025) that do not adapt in real-time to system feedback, a setup that risks issues like reward hacking (Skalse et al., 2025). Other methods (Sun et al., 2024; Lu et al., 2024) restrict themselves to offline preference tuning (e.g., DPO), which can easily overfit on static datasets. Our framework, DUET, overcomes these challenges by integrating RL into a closed-loop system where downstream recommendation performance serves as the real-time reward signal, allowing for the dynamic and interactive refinement of textual profiles.

3 Method

DUET is a closed-loop framework that transforms raw user–item interaction histories into performance-aligned textual profiles through learned representation strategies. As shown in Figure 2, DUET consists of three stages. (1) **Cue-Based Initialization** distills interaction histories into concise evidence-based cues. (2) **Joint Exploration via Adaptive Profile Prompt Discovery**

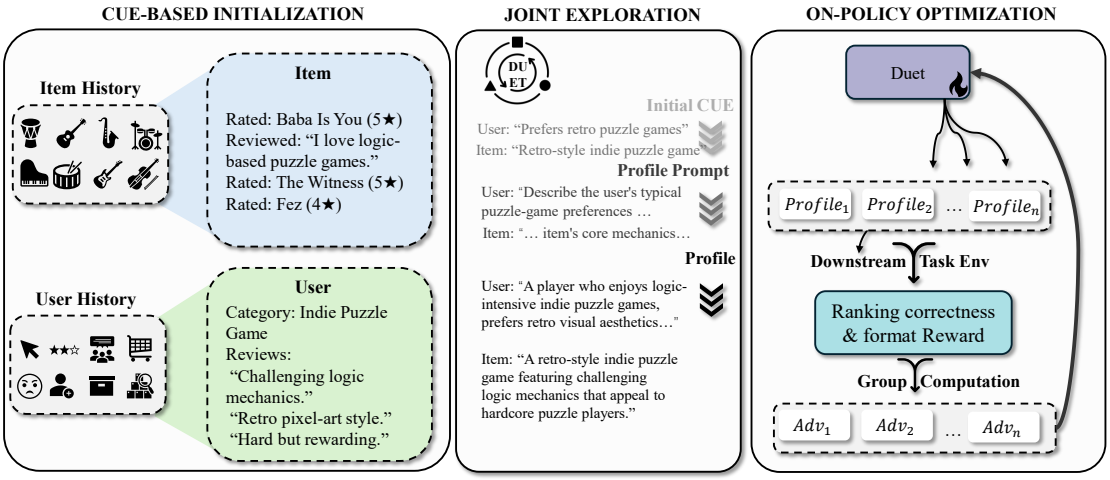

Figure 2: Overview of the DUET framework.

jointly explores user-item's profile prompts that define how user and item profiles should be written. (3) **Optimization via On-policy Exploration** jointly optimizes user and item profiles under downstream recommendation feedback.

All three stages are realized through a single pass input and output: cue extraction, self-prompt construction, and profile generation are produced in a single sequence-to-sequence generation pass at inference time, enabling efficient deployment.

3.1 Problem Formulation

We formulate profile generation in DUET as an **on-policy reinforcement learning** problem, motivated by the absence of any textual ground truth defining an optimal user or item profile. Profile quality is evaluated solely by its functional utility in a fixed recommendation environment.

DUET is modeled as a generative policy π_θ interacting with a frozen downstream recommender. For each user–item pair, the state is defined as $s = \{H_u, H_i\}$, where H_u and H_i denote the user and item interaction histories. An action corresponds to a single-pass joint generation

$$a = \{C_u, S_u, P_u, C_i, S_i, P_i\},$$

where (C_u, C_i) are cues distilled from history, (S_u, S_i) are constructed user-item's profile prompts, and (P_u, P_i) are the final textual profiles.

The policy $\pi_\theta(a \mid s)$ defines a joint distribution over the entire generation sequence. As an on-policy agent, DUET is optimized using rewards from its own sampled generations rather than by imitating fixed summaries.

3.2 Cue-Based Initialization

Raw user histories and item metadata are often noisy, redundant, and not directly suitable for profile construction. To address this, DUET introduces the concept of *cues*: concise hypotheses that summarize minimal but informative aspects of users and items. These cues act as lightweight seeds, which are deliberately underspecified so that the system can subsequently explore richer profile formats.

Definition 1 (Cue)**.** A *cue* is a minimal textual hypothesis derived from historical data that highlights one potential aspect of a user's preference or an item's characteristic. Rather than aiming for completeness, cues capture partial but salient signals that serve as starting points for profile exploration.

To extract cues automatically, the LLM is prompted to summarize minimal but informative aspects of user or item data. For example, given a user's interaction history, the model is guided with instructions such as:

Cue Extraction Prompt

"From the history below, analyze the user's historical interactions to understand preferences, rating behavior, review sentiment or any other dimension. Keep the description concise and avoid full sentences."

This lightweight guidance allows the LLM to map raw histories and metadata into compact textual cues. The detaild example of cue can be found in Appendix A.3.

3.3 Joint Exploration via Adaptive Profile Prompt Discovery

After cue extraction, DUET does not directly summarize user interests or item attributes. Instead, it explores the space of *profile construction strategies*—natural-language prompts that define the format, abstraction level, and attribute selection logic used to generate profiles. Exploration is therefore performed over how profiles should be constructed, rather than over superficial textual paraphrases.

For each user–item pair, DUET introduces an explicit intermediate variable, the `constructed_prompt`, which serves as a discrete and interpretable profile prompt S. This profile prompt is a short natural-language instruction (e.g., *"Describe the user's typical gaming preferences and engagement patterns"*) that specifies what aspects to describe and how to organize them, without containing the profile content itself. Conditioned on S, the model generates the final user and item profiles.

Exploration is driven by treating the profile prompt S as a stochastic action sampled from the policy $\pi_\theta(S \mid \text{Cue})$. During training, the agent samples different profile prompt instructions, executes them to generate profile pairs (P_u, P_i), and receives a reward based on downstream recommendation accuracy. The policy is optimized to reinforce profile prompts that consistently yield higher rewards, enabling the model to actively search for effective profile construction formats.

This process is realized as a unified generation pass,

$$O = [\underbrace{\text{Cue}}_{\text{Context}} \rightarrow \underbrace{\text{profile prompt } S}_{\text{Profile Construction Prompt}} \rightarrow \underbrace{\text{Profile}}_{\text{Execution}}], \tag{1}$$

where the profile prompt S parameterizes the profile generator. At inference time, DUET collapses the learned profile prompt distribution and greedily executes the optimal profile prompt in a single forward pass, introducing no additional latency.

Example of profile prompt Exploration. As illustrated in Figure 3, DUET resolves this by identifying a shared context in a single forward pass. The policy extracts a coordinated **Cue** (e.g., "prefers retro puzzle games") and constructs a **profile prompt** (the `constructed_prompt`) that instructs the model to highlight "nostalgic aesthetics and strategic logic" for the user, while simultaneously describing the item's "pixel-art graphics and intellectual difficulty" to match. RL reinforces this shared semantic direction, suppressing irrelevant signals and forcing the final profiles to converge into a **shared semantic space** of nostalgia and logic, significantly improving recommendation accuracy.

Single-Pass Input
Task: Generate structured profiles for the user and item from historical data.
Combined Context: `[User History` H_u`] + [Item History` H_i`] + [Avg Ratings]`
*"First, extract a concise **Cue** for both user and item. Second, based on the cues, construct a **Profile Prompt** that defines the description logic. Finally, generate the **Profile** guided by the constructed Profile Prompt."*

↓ **DUET Forward Pass** ↓

Single-Pass Output
User
Cue: *"prefers nostalgic logic-based challenges"*
Profile prompt: *"Focus on 1990s visual aesthetics and strategic depth."*
Profile: *"A player who seeks retro-style visual charm paired with deep strategic reasoning."*

Item
Cue: *"retro-style indie puzzle with high difficulty"*
Profile prompt: *"Describe pixel-art graphics and intellectual difficulty to match logic preference."*
Profile: *"A 2D experience featuring pixelated nostalgia and challenging mechanics that demand logical deduction."*

Figure 3: Single-pass generation in DUET: cue extraction, profile prompt (constructed prompt), and profile generation are produced in one pass for both user and item.

3.4 Optimization via On-policy Exploration

On-policy optimization. We train the profile generator π_θ in an on-policy manner against a frozen downstream model f, which serves as the environment critic. For each sampled user–item pair (u, i), the policy generates (P_u, P_i) and receives a scalar reward based on the prediction accuracy of $f(P_u, P_i)$. The policy parameters are updated to reinforce generations that lead to lower prediction error.

Continuous fractional reward. Using discrete integer ratings as rewards leads to sparse and unstable feedback, as near-miss predictions (e.g., predicting 4 for a ground truth 5) receive the same penalty as severe errors. To provide dense and informative feedback, we define a continuous fractional reward:

$$R_{\text{perf}}(u, i) = 1 - \frac{|y_{ui} - \hat{y}_{ui}|}{M}, \tag{2}$$

where $\hat{y}_{ui} = f(P_u, P_i)$ is the predicted relevance score and M is the maximum rating gap (e.g., $M = 4$ for a 1–5 scale). This reward provides fine-grained gradients that encourage incremental improvements in recommendation accuracy.

We adopt Group Relative Policy Optimization (GRPO) (DeepSeek-AI et al., 2025) to optimize the profile generator under the above reward. The downstream recommender f is frozen throughout training, which yields a stable policy optimization setting and prevents reward drift or feedback-induced representation collapse.

4 Experiment

4.1 Experimental Settings

Datasets. Experiments were conducted on three widely used real-world datasets. **Amazon Music (Music)** and **Amazon Book (Book)** are derived from the Amazon Product dataset[1], while **Yelp** is from the Yelp Open dataset[2]. All datasets include user reviews, ratings, and rich textual information. We used the full Amazon Music dataset, but only subsets (latest two months for Book and six for Yelp). Data was split by timestamp into training, validation, and test sets to prevent information leakage (Ji et al., 2023).

Evaluation Metrics

For each observed user–item interaction (i.e., a review record), we construct the evaluation instance based on the user's interaction history strictly prior to the corresponding timestamp. Specifically, the user's recent historical interactions before the current interaction are used to generate a *user profile*, while historical reviews from other users are used to generate an *item profile*. The downstream recommendation system then predicts the rating of the target item conditioned on two inputs: the generated user profile, and the generated item profile. An exception is the **10H** baseline, for which the downstream model directly consumes the raw recent interaction histories.

We evaluate performance using four widely adopted metrics (Wang et al., 2025; Fang et al., 2025): **Mean Absolute Error (MAE)**, **Root Mean Square Error (RMSE)**, **Accuracy**, and **F1 score**. In addition to rating prediction, we further evaluate the generated user and item profiles under a ranking setting based on downstream predicted rating scores. For each observed user–item interaction, we randomly sample nine items that the user has not interacted with to form a candidate set of ten items, which are ranked in descending order according to their predicted ratings. The ground-truth interacted item is treated as the only positive instance, and we adopt **NDCG@K** as the evaluation metric, with **K** set to **1**, **5**, and **10**, to assess how effectively the learned profiles support correct item ranking.

Baselines We compare our method with several representative baselines. **10H** directly uses the most recent interaction histories for prediction without constructing explicit profiles. **KAR** (Xi et al., 2024) augments recommendation models with external reasoning knowledge about user preferences and factual knowledge about items extracted from LLMs, which are transformed into task-compatible representations. **RLMRec** (Ren et al., 2024) leverages LLMs to learn semantic user and item representations from textual signals and aligns them with collaborative relational information through cross-view representation learning. **PALR** (Yang et al., 2023) fine-tunes a large language model as a ranking component that selects preferred items from retrieved candidates expressed in natural language. **LG (LettinGo)** (Wang et al., 2025) explores diverse user profile candidates with LLMs and aligns profile generation with downstream recommendation performance via preference optimization. **Reason4Rec** (Fang et al., 2025) introduces a deliberative recommendation framework that incorporates explicit step-wise reasoning over user preferences to guide rating prediction.

4.2 Main Results

Table 1 presents a comparison of our proposed method against five baselines on three datasets: Amazon Music, Amazon Books, and Yelp. We use Qwen3-8B (Team, 2025) and LLaMA3-8B (Dubey et al., 2024) as both the profile generator and the downstream recommendation model, with a prediction temperature of 0. In addition to rating prediction results, we also report ranking-based evaluation results in Table 2, which assess the effectiveness of the learned profiles from a downstream ranking perspective.

Overall superiority over strong baselines. As shown in Table 1, our method consistently outperforms the strongest baselines across all three datasets under both Qwen3-8B and LLaMA3-8B.

[1] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon/links.html

[2] https://business.yelp.com/data/resources/open-dataset/

Method	Yelp				Amazon Music				Amazon Books			
	MAE	RMSE	Acc (%)	F1 (%)	MAE	RMSE	Acc (%)	F1 (%)	MAE	RMSE	Acc (%)	F1 (%)
Qwen 3 (8B)												
10H	1.1235	1.9478	23.17	27.54	0.9102	1.4021	39.26	46.58	0.9314	1.4527	37.63	45.19
KAR (Xi et al., 2024)	0.7396	1.2184	55.34	48.67	0.7483	1.1380	58.65	60.29	0.7098	1.0923	56.17	58.78
RLMRec (Ren et al., 2024)	0.8197	1.3312	47.15	42.46	0.7438	1.1069	54.89	57.65	0.7812	1.1584	52.86	55.93
PALR (Yang et al., 2023)	0.7994	1.2876	48.53	43.19	0.6075	0.9531	57.35	56.77	0.7485	1.1187	54.24	56.38
LG (Wang et al., 2025)	0.6632	1.1047	56.18	48.95	0.4737	0.8834	62.37	57.09	0.5821	0.9416	59.35	60.57
R4Rec (Fang et al., 2025)	0.7028	1.1523	55.69	47.73	0.5654	0.9635	58.69	54.67	0.6397	1.0098	58.47	56.84
Ours	**0.5126**	**0.9485**	**61.23**	**55.18**	**0.3937**	**0.7564**	**67.96**	**63.89**	**0.4612**	**0.9089**	**64.38**	**59.27**
LLaMA 3 (8B)												
10H	1.0864	1.9532	22.09	27.30	0.7917	1.3346	38.13	46.87	0.8064	1.3866	37.15	45.27
KAR (Xi et al., 2024)	0.6427	1.1668	54.51	47.98	0.5726	0.9033	57.53	59.92	0.5892	0.9614	55.87	58.21
RLMRec (Ren et al., 2024)	0.7428	1.3572	46.74	42.11	0.6076	0.9886	53.78	57.42	0.6226	0.9477	52.12	55.79
PALR (Yang et al., 2023)	0.7238	1.3265	47.72	43.29	0.5823	0.9222	56.73	59.31	0.5977	0.8855	55.06	57.62
LG (Wang et al., 2025)	0.6196	1.1289	56.03	51.24	0.5204	0.9369	61.92	59.50	0.5543	0.7967	58.95	60.39
R4Rec (Fang et al., 2025)	0.7586	1.0418	55.80	53.00	0.5442	0.7722	60.86	54.88	0.6029	0.8345	59.70	56.35
Ours	**0.5367**	**0.9687**	**60.87**	**54.74**	**0.4680**	**0.8277**	**63.30**	**60.60**	**0.5092**	**0.9500**	**63.42**	**58.12**

Table 1: Performance on three datasets using Qwen 3 (8B) and LLaMA 3 (8B).

Method	Yelp			Amazon Music			Amazon Books		
	NDCG@1	NDCG@5	NDCG@10	NDCG@1	NDCG@5	NDCG@10	NDCG@1	NDCG@5	NDCG@10
10H	0.1823	0.2815	0.4928	0.1875	0.3796	0.5153	0.1841	0.3146	0.4263
KAR(Xi et al., 2024)	0.2156	0.3298	0.5412	0.3018	0.4896	0.6015	0.2965	0.4715	0.5834
RLMRec(Ren et al., 2024)	0.2419	0.3472	0.5587	0.3371	0.5434	0.6162	0.2748	0.4526	0.5719
PALR(Yang et al., 2023)	0.2494	0.3563	0.5691	0.3395	0.5247	0.6115	0.2627	0.4634	0.5538
LG(Wang et al., 2025)	0.3187	0.4685	0.5814	0.4012	0.5674	0.6489	0.3795	0.5189	0.6284
R4Rec(Fang et al., 2025)	0.2575	0.3792	0.5526	0.2928	0.5912	0.6343	0.3013	0.4928	0.5959
Ours	**0.3390**	**0.4873**	**0.6008**	**0.5123**	**0.6165**	**0.7025**	**0.4288**	**0.5638**	**0.6599**

Table 2: Ranking performance under EASE-based (Steck, 2019) hard negatives.

Under Qwen3-8B, our method achieves an accuracy of 61.23% on Yelp, 67.96% on Amazon Music and 64.38% on Amazon Books, surpassing **LG (LettinGo)** (Wang et al., 2025) by 5.05%, 5.59% and 5.03%, respectively. Similar improvements are observed under LLaMA3-8B, indicating that the gains are stable across backbone models.

Advantages over fixed-structure and deliberative baselines. Compared with strong fixed-structure or deliberative methods such as **KAR** (Xi et al., 2024) and **Reason4Rec** (Fang et al., 2025), our approach consistently achieves lower prediction error and higher accuracy. For example, on Amazon Books with Qwen3-8B, our method reduces MAE to 0.4612, compared to 0.7098 for KAR and 0.6397 for Reason4Rec, demonstrating more effective abstraction of user preferences without relying on predefined reasoning templates.

Consistent improvements in downstream ranking. Table **??** reports the ranking results under the Qwen3-8B backbone. Our method consistently achieves the best performance across all three datasets and evaluation cutoffs. On Yelp, our approach reaches an NDCG@1 of 0.5619 and an NDCG@10 of 0.7443, clearly outperforming the strongest baseline methods. Similar trends are observed on Amazon Music, where our method attains 0.5347 at NDCG@1 and 0.7331 at NDCG@10, as well as on Amazon Books with NDCG@1 of 0.4866 and NDCG@10 of 0.7107. These results indicate that the learned profiles enable more accurate identification and ordering of relevant items within candidate sets, leading to superior ranking quality in downstream recommendation. We further evaluate ranking performance under a more challenging setting using EASE-based (Steck, 2019) hard negatives, and observe consistent improvements; detailed results are provided in Appendix B.2.

4.3 Ablation Study

Effectiveness of profile generation, cue&strategy, and user–item joint optimization. Table 3 reports the ablation results under different design configurations. Starting from the history-only baseline (**10H**), introducing explicit **profile genera-**

Method	Yelp				Amazon Music				Amazon Books			
	MAE	RMSE	Acc	F1	MAE	RMSE	Acc	F1	MAE	RMSE	Acc	F1
10H (History Only)	1.1235	1.9478	23.17	27.54	0.9102	1.4021	39.26	46.58	0.9314	1.4527	37.63	45.19
Profile	0.7218	1.1863	55.48	48.09	0.6597	1.0218	58.67	57.48	0.6764	1.0469	57.14	57.68
Profile + Cue&Strategy	0.7085	1.1654	55.83	48.54	0.5708	0.9897	58.91	55.53	0.6389	1.0108	58.43	56.88
Profile + Joint Opt.(LG (Wang et al., 2025))	0.6632	1.1047	56.18	48.95	0.4737	0.8834	62.37	57.09	0.5821	0.9416	59.35	60.57
Profile + Cue&Strategy + Joint Opt. (**Ours**)	**0.5126**	**0.9485**	**61.23**	**55.18**	**0.3937**	**0.7564**	**67.96**	**63.89**	**0.4612**	**0.9089**	**64.38**	**59.27**

Table 3: Ablation study on different design configurations in DUET using Qwen 3 (8B).

Method	Yelp				Amazon Music				Amazon Books			
	MAE	**RMSE**	**Acc (%)**	**F1 (%)**	**MAE**	**RMSE**	**Acc (%)**	**F1 (%)**	**MAE**	**RMSE**	**Acc (%)**	**F1 (%)**
10H+30P	0.5126	0.9485	61.23	55.18	**0.3883**	**0.7494**	**67.96**	**63.89**	0.4612	0.9089	**65.13**	**59.97**
10H+50P	**0.4909**	**0.9207**	**62.43**	**56.24**	0.3924	0.7543	67.88	63.87	**0.4553**	**0.9023**	64.62	59.52
10H+70P	0.4987	0.9326	61.98	55.81	0.3937	0.7564	68.22	64.12	0.4608	0.9068	64.38	59.27

Table 4: Impact of historical interaction length on profile quality (using Qwen 3 (8B)).

tion yields substantial performance improvements across all datasets. For example, on Yelp, MAE is reduced from 1.1235 to 0.7218 and accuracy increases from 23.17% to 55.48%, confirming that textual profiles provide significantly more informative representations than raw interaction histories.

Adding the **cue&strategy layer** on top of profile generation leads to modest but consistent gains. On Amazon Music, MAE further decreases from 0.6597 to 0.5708, while accuracy slightly improves from 58.67% to 58.91%. Although the numerical improvements introduced by strategy alone are limited, these results suggest that strategy discovery primarily reshapes how preference information is abstracted and expressed, rather than directly optimizing prediction accuracy in isolation.

When enabling **joint optimization** without strategy (i.e., the LettinGo (Wang et al., 2025)-style configuration), performance improves more noticeably across datasets. On Amazon Music, accuracy increases to 62.37%, compared to 58.67% with profile generation alone. In our implementation, this setting corresponds to a reproduction of LettinGo, where profile generation and joint optimization are applied without the strategy layer. While the original method focuses primarily on user profiling, we extend the optimization to both user and item profiles to ensure a fair comparison.

The strongest and most consistent performance is achieved by combining **cue&strategy discovery** and **joint optimization**. Under this full configuration, accuracy reaches 61.23%, 67.96%, 64.38% on Yelp, Amazon Music, Amazon Books respectively, with corresponding MAE values of 0.5126, 0.3937, 0.4612, outperforming the LettinGo-style configuration across all datasets. Overall, the ablation results indicate that while **profile generation** and **joint optimization** contribute substantially to performance gains, integrating **cue&strategy discovery** further enhances the effectiveness of joint user–item optimization by providing a structured space for exploration and refinement.

Impact of historical interaction length on profile quality. Table 4 analyzes the impact of historical interaction length on profile quality. Across the three datasets, varying the number of historical interactions leads to only moderate performance differences, indicating that our method does not strongly depend on long histories. In particular, using a moderate history length (e.g., 30–50 interactions) already achieves competitive or best performance on most metrics, while further increasing the history length provides limited additional gains.

On Yelp and Amazon Books, extending the history length beyond this range does not consistently improve accuracy or F1 score and may slightly degrade performance, suggesting that excessive historical interactions can introduce noisy or less relevant signals. In contrast, Amazon Music exhibits relatively stable performance across different his-

Setting	Yelp				Amazon Music				Amazon Books			
	MAE	RMSE	Acc (%)	F1 (%)	MAE	RMSE	Acc (%)	F1 (%)	MAE	RMSE	Acc (%)	F1 (%)
DUET w/o RL	0.8283	1.3893	48.53	41.15	0.7322	1.1430	57.18	52.48	0.8741	1.4760	51.83	50.35
DUET (full)	**0.5126**	**0.9485**	**61.23**	**55.18**	**0.3937**	**0.7564**	**67.96**	**63.89**	**0.4612**	**0.9089**	**64.38**	**59.27**

Table 5: Effect of RL-based optimization (using Qwen 3 (8B)).

tory lengths, indicating that user preferences in this domain are less sensitive to history truncation.

Necessity of RL-based Optimization. We compare the full model with a simplified variant without RL (DUET w/o RL) to isolate the effect of RL. As shown in Table 5, removing RL component leads to substantial performance degradation across all datasets (e.g., Yelp Accuracy drops from 61.23% to 48.53%), indicating that the gains cannot be attributed to prompt design alone.

Without RL, the generator reduces to a static mapping from interaction history to textual profiles, lacking adaptive selection of relevant signals. In contrast, RL optimizes profile construction under reward feedback, resulting in more discriminative representations.These results demonstrate that RL-based optimization is essential for DUET .

4.4 Semantic Analysis of Generated Profiles

To better understand the source of performance gains, we analyze whether the generated profiles exhibit meaningful semantic properties rather than serving as intermediate textual artifacts. We introduce two complementary metrics to characterize semantic compatibility and grounding.

Semantic Alignment. We measure the embedding-level similarity between generated user and item profiles using `all-mpnet-base-v2` from Sentence-Transformers (Reimers and Gurevych, 2019). For each user–item pair, we compute cosine similarity:

$$\text{Align}(u, i) = \frac{\mathbf{e}_u \cdot \mathbf{e}_i}{\|\mathbf{e}_u\| \|\mathbf{e}_i\|} \tag{3}$$

where $\mathbf{e}_u$ and $\mathbf{e}_i$ denote the embedding vectors of the generated user and item profiles. Higher values indicate stronger semantic compatibility between modeled user preferences and item characteristics.

Coverage (Faithfulness). We measure token-level grounding as:

$$\text{Cov} = \frac{|\text{Tokens}(\text{profile}) \cap \text{Tokens}(\text{history})|}{|\text{Tokens}(\text{profile})|} \tag{4}$$

This metric quantifies how much of the generated profile is supported by historical textual evidence. We separately report the coverage scores for user profiles and item profiles respectively.

As shown in Table 6, DUET achieves the highest semantic alignment across all datasets while maintaining comparable coverage. Compared to existing methods, which either rely on extractive compression (e.g., KAR (Xi et al., 2024)) or generate free-form descriptions with weaker grounding (e.g., RLMRec (Ren et al., 2024) and PALR (Yang et al., 2023)), DUET consistently attains stronger alignment without sacrificing coverage. For instance, LG (Wang et al., 2025) achieves relatively high alignment but exhibits less stable grounding across datasets, while RLMRec attains higher coverage at the cost of weaker semantic alignment. In contrast, DUET maintains both high alignment and mid-to-high coverage, suggesting a more effective balance between semantic abstraction and evidence preservation. Thus, DUET distills interaction-relevant information into compact and semantically coherent representations grounded in historical interactions.

4.5 Case Study

Figure 4 presents a representative case study that demonstrates how semantically aligned user and item profiles enable accurate prediction that transcends the limitations of sparse raw interaction histories. In the user profile, initially scattered and fragmented preference cues are systematically distilled into a stable and coherent preference structure favoring *funk, soul, and progressive rock* (orange), coupled with a pronounced emphasis on *musical complexity* and *historical significance* (purple). The corresponding item profile exhibits striking symmetry, characterizing the album as an exemplary *funk-rock* composition (orange) and positioning it as a *defining and culturally influential release of the 1970s* (purple). This example shows 519 that the learned profiles capture meaningful preference–attribute correspondence that is difficult to recover from individual reviews alone.

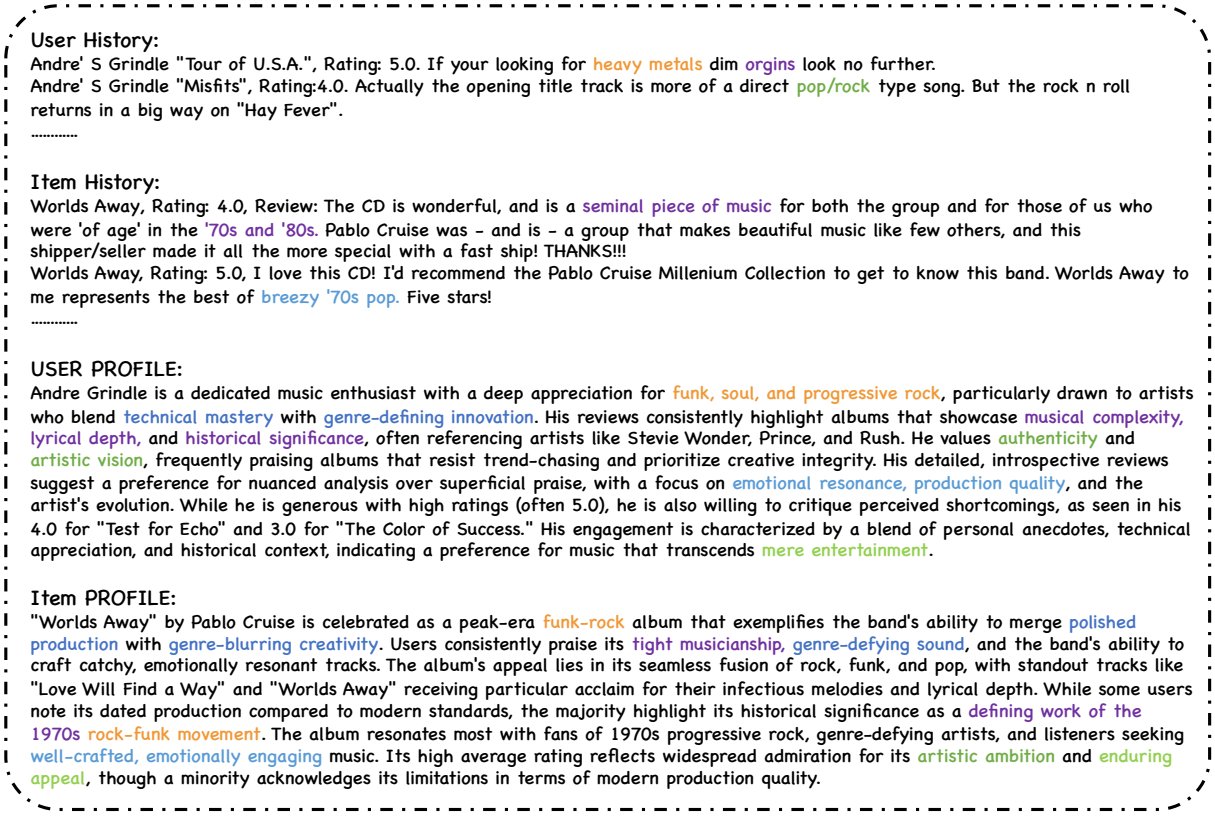

Figure 4: Illustration of the mutual correspondence between user and item. The highlighted regions demonstrate that user preferences summarized in the user profile align with the key attributes extracted in the item profile, which provides complementary information beyond raw histories and thus improves prediction accuracy.

Method	Yelp			Amazon Music			Amazon Books		
	Align	User Cov.	Item Cov.	Align	User Cov.	Item Cov.	Align	User Cov.	Item Cov.
10H	0.3902	/(1.00)	/(1.00)	0.3577	/(1.00)	/(1.00)	0.2604	/(1.00)	/(1.00)
KAR(Xi et al., 2024)	0.4932	0.1745	0.1823	0.4807	0.2170	0.2391	0.5702	0.1877	0.1544
RLMRec(Ren et al., 2024)	0.4010	0.2646	**0.3773**	0.3931	0.3526	**0.4889**	0.4436	0.2994	**0.3506**
PALR(Yang et al., 2023)	0.4715	**0.3675**	0.1917	0.4938	0.2197	0.2390	0.5216	0.2050	0.2267
LG(Wang et al., 2025)	0.5709	0.2378	0.2682	0.5109	0.2546	0.3749	0.5506	**0.3883**	0.3364
R4Rec(Fang et al., 2025)	0.4882	0.2348	0.2330	0.4208	0.3328	0.3595	0.4946	0.2831	0.2949
Ours	**0.6382**	0.2880	0.3429	**0.5947**	**0.4002**	0.4482	**0.7287**	0.3457	0.3127

Table 6: Semantic alignment and coverage of generated profiles. Cov. denotes the fraction of tokens grounded in historical text.

5 Conclusion

In this paper, we propose DUET , a closed-loop framework for jointly generating user and item textual profiles for recommendation. Unlike prior methods that rely on fixed templates or independently constructed profiles, DUET treats profile generation as an exploration problem and aligns representations directly with downstream recommendation performance.

Specifically, DUET integrates cue-based initialization, adaptive strategy construction, and feedback-driven joint optimization to produce flexible yet task-aligned user–item profiles. By optimizing both profiles in a shared semantic space, the framework reduces semantic mismatch and captures interaction-relevant signals that are difficult to recover from raw histories or static prompts alone.

Experiments on multiple real-world datasets demonstrate that DUET consistently outperforms strong baselines under different backbone models, validating the effectiveness of joint profiling and reinforcement learning–based optimization. These results suggest that adaptive, interaction-aware textual profiles provide a promising direction for more effective interpretable and performance-oriented recommendation systems.

Limitations

Despite its effectiveness, our approach has several limitations. First, the proposed framework relies on large language models for both profile generation and downstream recommendation, which introduces additional computational overhead during training and inference. While our experiments show consistent gains across different backbones, the overall efficiency may be constrained when scaling to very large user or item sets. Second, the quality of the generated profiles is inherently dependent on the underlying LLMs and prompting strategies. Variations in model capacity or prompt sensitivity may lead to differences in profile stability, which we do not explicitly control in the current design. Finally, our evaluation focuses on text-rich recommendation scenarios where sufficient historical reviews are available. The effectiveness of the proposed strategy in domains with extremely sparse textual signals or in non-textual modalities remains to be further explored.

References

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.

Ting Chen, Wei-Li Han, Hai-Dong Wang, Yi-Xun Zhou, Bin Xu, and Bin-Yu Zang. 2007. Content recommendation system based on private dynamic user profile. In *2007 International conference on machine learning and cybernetics*, volume 4, pages 2112–2118. IEEE.

Yue Chen, Minghua He, Fangkai Yang, Pu Zhao, Lu Wang, Yu Kang, Yifei Dong, Yuefeng Zhan, Hao Sun, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2025. Warriormath: Enhancing the mathematical ability of large language models with a defect-aware framework. *Preprint*, arXiv:2508.01245.

Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *Preprint*, arXiv:2502.18965.

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

Yi Fang, Wenjie Wang, Yang Zhang, Fengbin Zhu, Qifan Wang, Fuli Feng, and Xiangnan He. 2025. Reason4rec: Large language models for recommendation with deliberative user preference alignment. *Preprint*, arXiv:2502.02061.

Minghua He, Yue Chen, Fangkai Yang, Pu Zhao, Wenjie Yin, Yu Kang, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2025. Execoder: Empowering large language models with executability representation for code translation. *Preprint*, arXiv:2501.18460.

Minjie Hong, Zirun Guo, Yan Xia, Zehan Wang, Ziang Zhang, Tao Jin, and Zhou Zhao. 2025a. Apo: Enhancing reasoning ability of mllms via asymmetric policy optimization. *Preprint*, arXiv:2506.21655.

Minjie Hong, Yan Xia, Zehan Wang, Jieming Zhu, Ye Wang, Sihang Cai, Xiaoda Yang, Quanyu Dai, Zhenhua Dong, Zhimeng Zhang, and 1 others. 2025b. Eager-llm: Enhancing large language models as recommenders through exogenous behavior-semantic integration. In *Proceedings of the ACM on Web Conference 2025*, pages 2754–2762.

Minjie Hong, Zetong Zhou, Zirun Guo, Ziang Zhang, Ruofan Hu, Weinan Gan, Jieming Zhu, and Zhou Zhao. 2025c. Generative reasoning recommendation via llms. *Preprint*, arXiv:2510.20815.

Jihwan Jeong, Yinlam Chow, Guy Tennenholtz, Chih-Wei Hsu, Azamat Tulepbergenov, Mohammad Ghavamzadeh, and Craig Boutilier. 2023. Factual and personalized recommendations using language models and reinforcement learning. *Preprint*, arXiv:2310.06176.

Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2023. A critical study on data leakage in recommender system offline evaluation. *ACM Trans. Inf. Syst.*, 41(3):75:1–75:27.

Jiacheng Lin, Tian Wang, and Kun Qian. 2025. Rec-r1: Bridging generative large language models and user-centric recommendation systems via reinforcement learning. *Preprint*, arXiv:2503.24289.

Aiwei Liu, Minghua He, Shaoxun Zeng, Sijun Zhang, Linhao Zhang, Chuhan Wu, Wei Jia, Yuan Liu, Xiao Zhou, and Jie Zhou. 2025. Wedlm: Reconciling diffusion language models with standard causal attention for fast inference. *Preprint*, arXiv:2512.22737.

Hongtao Liu, Fangzhao Wu, Wenjun Wang, Xianchen Wang, Pengfei Jiao, Chuhan Wu, and Xing Xie. 2019. Nrpa: Neural recommendation with personalized attention. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1233–1236.

Wensheng Lu, Jianxun Lian, Wei Zhang, Guanghua Li, Mingyang Zhou, Hao Liao, and Xing Xie. 2024. Aligning large language models for controllable recommendations. *arXiv preprint arXiv:2403.05063*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *EMNLP*.

Zhaopeng Qiu, Xian Wu, Jingyue Gao, and Wei Fan. 2021. U-BERT: pre-training user representations for improved recommendation. In *AAAI*, pages 4320–4327. AAAI Press.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3982–3992.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3464–3475.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2025. Defining and characterizing reward hacking. *Preprint*, arXiv:2209.13085.

Harald Steck. 2019. Embarrassingly shallow autoencoders for sparse data. In *Proceedings of The World Wide Web Conference (WWW)*, pages 3251–3257.

Chao Sun, Yaobo Liang, Yaming Yang, Shilin Xu, Tianmeng Yang, and Yunhai Tong. 2024. Rlrf4rec: Reinforcement learning from recsys feedback for enhanced recommendation reranking. *arXiv preprint arXiv:2410.05939*.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformer Reinforcement Learning.

Lu Wang, Di Zhang, Fangkai Yang, Pu Zhao, Jianfeng Liu, Yuefeng Zhan, Hao Sun, Qingwei Lin, Weiwei Deng, Dongmei Zhang, Feng Sun, and Qi Zhang. 2025. Lettingo: Explore user profile generation for recommendation system. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, New York, NY, USA. Association for Computing Machinery.

Ye Wang, Jiahao Xun, Minjie Hong, Jieming Zhu, Tao Jin, Wang Lin, Haoyuan Li, Linjun Li, Yan Xia, Zhou Zhao, and 1 others. 2024. Eager: Two-stream generative recommender with behavior-semantic collaboration. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3245–3254.

Triyanna Widiyaningtyas, Indriana Hidayah, and Teguh B Adji. 2021. User profile correlation-based similarity (upcsim) algorithm in movie recommendation system. *Journal of Big Data*, 8(1):52.

J. Wu. 2023. Computational understanding of user interfaces.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards open-world recommendation with knowledge augmentation from large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 12–22.

Fan Yang, Zheng Chen, Ziyan Jiang, Eunah Cho, Xiaojiang Huang, and Yanbin Lu. 2023. Palr: Personalization aware llms for recommendation. *arXiv preprint arXiv:2305.07622*.

Jiarui Zhang. 2024. Guided profile generation improves personalization with llms. *arXiv preprint arXiv:2409.13093*.

A Experiment Setup

A.1 Data Curation

Table 7: Statistical details of the evaluation datasets.

Dataset	#Train	#Valid	#Test	#User	#Item
Music	43,071	3,271	1,296	4,183	2,660
Book	71,972	6,144	5,541	13,863	13,515
Yelp	51,497	4,757	4,328	8,453	13,426

We conduct experiments on three widely used real-world datasets:

- **Amazon Music (Music)**: This refers to the "Digital Music" subset of the well-known Amazon Product dataset[3], which records rich user reviews, ratings, and textual information about items, such as titles, across a broad range of product categories, on the Amazon platform.2

- **Amazon Book (Book)**: This refers to the "Book" subset of the Amazon Product dataset.

- **Yelp:** This refers to the Yelp Open dataset[4], which includes user reviews, ratings for businesses such as restaurants and retail shops, as well as textual information about the businesses. It is widely used in recommendation tasks (Qiu et al., 2021).

We use the entire Music dataset for experiments, while for the Book and Yelp datasets, we utilize only a subset due to their large size. For the Book dataset, we use data from the last two months, and for the Yelp dataset, we use data from the last six months. For each dataset, we split it into training, validation, and test sets based on the timestamps of interactions, ensuring that test interactions occur after all training and validation interactions to prevent information leakage (Ji et al., 2023).

Regarding data filtering, following prior work (Liu et al., 2019), we adopt a 5-core setting to filter the data and exclude cold-start users and items—those not appearing in the training set—from the validation and test sets. The statistical details of the processed dataset are provided in Table 7.

[3] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon/links.html.

[4] https://business.yelp.com/data/resources/open-dataset/.

A.1.1 Implementation Details

In our experiments, we primarily employ Qwen3-8B (Team, 2025) and LLaMA3 8B Instruct (Dubey et al., 2024; Touvron et al., 2023) as both the recommendation model and the profile generation model. The training process is implemented using the TRL (von Werra et al.). Key hyperparameters, such as batch size and learning rate, are determined through grid search to achieve optimal performance. More details can be found in our code.

A.2 Baseline Prompts

KAR Prompt

Task: Analyze user preferences based on business reviewing history
Input: {user_history} - User's business reviewing history with sentiments over time
Instructions:
1. Analyze the user's preferences considering business names and categories
2. Take into account sentiment patterns over time
3. Provide clear explanations based on reviewing history details
4. Consider other pertinent factors that may influence preferences

PALR Prompt

Task: Summarize user preferences using keywords.
Input: {user_history} - historical businesses with user sentiments.
Output Format: An itemized list ranked by importance.
Template:

- KEY_WORD_1: "HISTORY_BUSINESS_1", "HISTORY_BUSINESS_2"
- KEY_WORD_2: "HISTORY_BUSINESS_3"

Instructions:

1. Extract key preference indicators from user interaction history.
2. Rank keywords by importance.

RLMRec Prompt

Role: Business recommendation assistant
Task: Determine business types a user is likely to enjoy
Input Format:
• Title: Business name
• Categories: Business categories
• Sentiment: User sentiment toward business

Output Requirements:
1. JSON format only
2. Structure:
{
"summarization": "Types of businesses user likely enjoys" ($\leq$100 words),
"reasoning": "Brief explanation for summarization" (no word limit)
}
3. No additional text outside JSON

Input: INTERACTION ITEMS: {user_history}

LG Prompt

You will serve as an assistant to help me generate a user profile based on this user's sentiments history to better understand this users' interest and thus predict his/her sentiment about a target item. I will provide you with some behavior history of the user in this format: [item attributes and sentiment]. The user profile you generate should contain as much useful content as possible to help predict the user's sentiment towards a new business.
USER HISTORY: {user_history}.
PROFILE YOU GENERATE:

R4Rec Prompt (Reasoner)

User Review History
$\langle H_u$ organized as below$\rangle$
1. Title of Item 1
Positive Aspects: [Aspect 1], [Aspect 2], ...
Negative Aspects: [Aspect 1], [Aspect 2], ...
User Preference Elements: [Preference 1], [Preference 2], ...
2. Title of Item 2
Positive Aspects: [Aspect 1], [Aspect 2], ...
Negative Aspects: [Aspect 1], [Aspect 2], ...
User Preference Elements: [Preference 1], [Preference 2], ...
...
Item Review History by Other Users
$\langle H_i$ organized in the same format as above$\rangle$
Task: Analyze whether the user will like the new Music i based on the user's preferences and the item's features. Provide your rationale in one concise paragraph.

A.3 Example of Cue

The following examples illustrate how raw signals are distilled into cues. As shown on the top, user cues emphasize historical preferences, while item cues highlight metadata and user-group patterns. Together, they provide minimal but informative hypotheses for profile exploration.

Examples of User Cues
"enjoys retro puzzle games" — derived from repeated engagement with classic titles.
"prefers concise product reviews" — inferred from a pattern of short, direct comments.
"tends to give high ratings but rarely comments" — highlighting consistency but limited feedback.

Examples of Item Cues
"lightweight trail-running shoes" — derived from product metadata.
"popular among budget-conscious users" — inferred from purchase patterns.
"stylized with retro aesthetics" — extracted from item descriptions.

B Additional Experiments

B.1 Non-LLM Baseline via Extractive Summarization

To examine whether the gains of DUET stem from improved semantic representations rather than generic text generation, we introduce a non-LLM baseline based on extractive summarization. Specifically, we apply TextRank (Mihalcea and Tarau, 2004) to select salient sentences from user histories and construct user profiles without using any generative model.

The extracted summaries are then fed into the same downstream predictor for rating estimation. This baseline isolates the effect of readable summarization from representation learning.

Table 8 shows that TextRank improves over simple history truncation (10H), but remains signifi-

cantly worse than DUET across all datasets. This indicates that coherent summaries alone are insufficient, and that the gains of DUET arise from learned semantic abstraction rather than extractive compression.

B.2 Ranking under Hard Negative Sampling

To construct a more challenging ranking scenario, we replace random negatives with hard negatives generated by a collaborative filtering model. Specifically, for each user, we retrieve high-scoring items from an EASE(Steck, 2019) model that the user has not interacted with, and combine them with the ground-truth item to form the candidate set.

Table 2 reports the results. Compared to random sampling, performance decreases for all methods due to increased difficulty, while DUET consistently maintains the best performance across datasets. This indicates that the improvements are robust and not limited to trivial ranking scenarios.

B.3 Robustness under Preference Diversity

We further analyze the robustness of DUET under varying levels of user preference diversity. We use the variance of historical ratings as a proxy for preference stability: low variance indicates consistent preferences, while high variance corresponds to diverse or potentially conflicting signals.

We partition users into three groups based on percentile thresholds (bottom 33%, middle 33%, top 33% of rating variance) and evaluate performance within each group.

As shown in Table 9, performance degrades smoothly as preference diversity increases across all datasets. Importantly, the degradation is gradual rather than catastrophic, indicating that DUET remains stable under heterogeneous or noisy interaction histories.

C The Use of Large Language Models

We used a Large Language Model (LLM) only as a writing assistant to polish the language of the manuscript (*e.g.*, grammar refinement, style adjustment, and clarity improvement). The research ideas, methodology design, experiments, and analysis were entirely conceived, implemented, and validated by the authors without reliance on the LLM. The LLM did not contribute to research ideation, experimental design, or result interpretation.

Method	Yelp				Amazon Music				Amazon Books			
	MAE	RMSE	Acc (%)	F1 (%)	MAE	RMSE	Acc (%)	F1 (%)	MAE	RMSE	Acc (%)	F1 (%)
10H	0.8283	1.3893	48.53	41.15	0.7322	1.1430	57.18	52.48	0.8741	1.4760	51.83	50.35
TextRank	0.8104	1.1323	50.51	30.23	0.5914	0.8223	61.19	29.68	0.6328	0.8854	59.59	29.48
Ours	**0.5126**	**0.9485**	**61.23**	**55.18**	**0.3937**	**0.7564**	**67.96**	**63.89**	**0.4612**	**0.9089**	**64.38**	**59.27**

Table 8: Comparison with a non-LLM extractive summarization baseline (TextRank).

Variance Group	Yelp					Amazon Music					Amazon Books				
	#Samp.	MAE	RMSE	Acc (%)	F1 (%)	#Samp.	MAE	RMSE	Acc (%)	F1 (%)	#Samp.	MAE	RMSE	Acc (%)	F1 (%)
Stable	1530	0.4017	0.8664	71.76	68.96	432	0.2368	0.5990	82.56	86.89	1876	0.3171	0.8403	76.64	72.56
Moderate	1354	0.4773	0.8309	60.09	46.26	436	0.3921	0.6628	64.16	47.02	1827	0.4495	0.8059	66.73	55.45
Diverse	1444	0.6632	1.1198	51.13	45.95	428	0.5536	0.9607	57.10	59.27	1838	0.6199	1.0603	49.52	46.35

Table 9: Performance of DUET under different levels of user preference diversity (measured by rating variance). #Samp. denotes the number of samples in each group.