

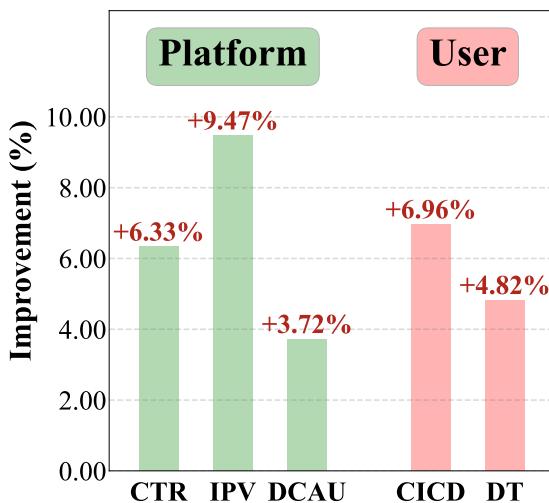
RecGPT Technical Report

RecGPT Team

Recommender systems are among the most impactful applications of artificial intelligence, serving as critical infrastructure connecting users, merchants, and platforms. However, most current industrial systems remain heavily reliant on historical co-occurrence patterns and log-fitting objectives—i.e., optimizing for past user interactions without explicitly modeling user intent. This log-fitting approach often leads to overfitting to narrow historical preferences, failing to capture users' evolving and latent interests. As a result, it reinforces filter bubbles and long-tail phenomena, ultimately harming user experience and threatening the sustainability of the whole recommendation ecosystem.

To address these challenges, we rethink the overall design paradigm of recommender systems and propose RecGPT, a next-generation framework that places user intent at the center of the recommendation pipeline. By integrating large language models (LLMs) into key stages of user interest mining, item retrieval, and explanation generation, RecGPT transforms log-fitting recommendation into an intent-centric process. To effectively align general-purpose LLMs to the above domain-specific recommendation tasks at scale, RecGPT incorporates a multi-stage training paradigm, which integrates reasoning-enhanced pre-alignment and self-training evolution, guided by a Human-LLM cooperative judge system. Currently, RecGPT has been fully deployed on the Taobao App. Online experiments demonstrate that RecGPT achieves consistent performance gains across stakeholders: users benefit from increased content diversity and satisfaction (e.g., CICD +6.96%, DT +4.82%), merchants and the platform gain greater exposure and conversions (e.g., CTR +6.33%, IPV +9.47%, DCAU +3.72%). These comprehensive improvement results across all stakeholders validates that LLM-driven, intent-centric design can foster a more sustainable and mutually beneficial recommendation ecosystem.

Overall Performance



Matthew Effect

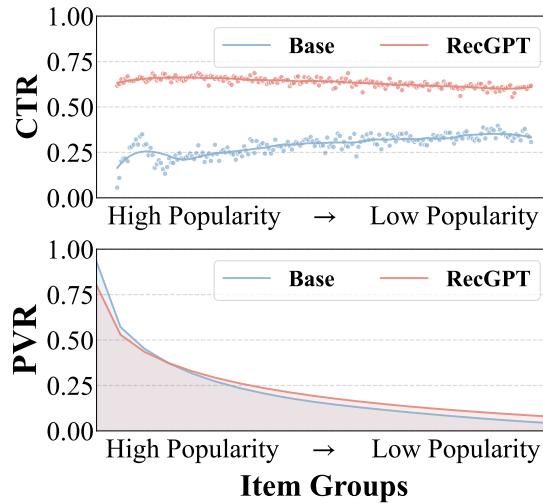


Figure 1 | Online performance of RecGPT in the “Guess What You Like” scenario on Taobao APP’s homepage. The left figure shows the overall performance improvements of RecGPT compared to the baseline system across key metrics, including Click Through Rate (CTR), Item Page View (IPV), Daily Click Active Users (DCAU), per-user Clicked Item Category Diversity (CICD), and user Dwell Time (DT). The right figure illustrates the CTR and Page View Rate (PVR) distributions of RecGPT and the baseline system across product groups of different popularity levels. For the purpose of protecting business confidentiality, the commercial metrics (such as CTR and PVR) have been normalized.

Contents

1	Introduction	3
2	RecGPT Workflow	4
2.1	User Interest Mining	6
2.2	Item Tag Prediction	11
2.3	Item Retrieval	16
2.4	Personalized Explanation Generation	19
3	Human-LLM Cooperative Judge	23
3.1	LLM-as-a-Judge	24
3.2	Human-in-the-Loop	25
4	Evaluation	25
4.1	Evaluation Setup	26
4.2	Online A/B Test	26
4.3	Human vs. LLM-as-a-Judge	27
4.4	Case Studies	29
4.5	User Experience Investigation	30
5	Conclusion, Limitations, and Future Directions	32
References		34
A	Contributions	37
B	Prompts	38
C	Implementation Details of Curriculum Learning-based Multi-task Fine-tuning	40

1. Introduction

Recommender systems have become pervasive across modern digital ecosystems, ranging from e-commerce portals such as Taobao and Amazon to content platforms like YouTube and TikTok, fundamentally reshaping how people discover and consume information (Lü et al., 2012). An ideal recommender system should match a user’s (often implicit) intent with the most relevant item or content, allowing the user to obtain the maximum experience value with minimal effort (Resnick and Varian, 1997). When this alignment is achieved, it forms a positive feedback loop that benefits all stakeholders: users enjoy satisfying experiences, merchants see increased sales, and platforms gain sustained traffic and revenue. The crux of realizing this vision, however, lies in accurately inferring and modeling the user’s true intent behind the long and rapidly changing behavior traces.

Over the past two decades, both academia and industry have pursued this vision through relentless optimization of *feature engineering* and *model architecture*. Feature representations have progressed from hand-crafted statistics to sequential and cross features, and most recently to ultra-long behavior modeling (Schifferer et al., 2020). Model architectures have advanced from factorization machines (Rendle, 2010) to deep matching networks (Zhang et al., 2019), graph neural models (Wu et al., 2022), and the latest generative Transformer backbones (Deng et al., 2025). Although these efforts have delivered remarkable business gains, they remain fundamentally limited by the co-occurrence patterns found in historical logs—they essentially “*learn clicks from clicks*”. Lacking an explicit understanding of user interest, such log-fitting methods tend to reinforce what similar users have already consumed, amplifying *Filter Bubble* effects (Nguyen et al., 2014; Wang et al., 2022) and further marginalizing long-tail sparsity (i.e., *Matthew Effects* (Gao et al., 2023; Liu and Huang, 2021)). Bridging this gap calls for a new modeling paradigm that can look beyond surface-level correlations and reason about the motivations that drive user behavior.

The recent advent of large language models (LLMs) (Zhao et al., 2023), especially those with strong reasoning capabilities, has opened a promising pathway to transcend the limitations of purely log-fitting recommendation. Thanks to their broad world knowledge, fine-grained semantic understanding, and step-wise reasoning abilities, LLMs can help accurately and comprehensively analyze user potential interests and explicitly reason about why a user may want an item. Although a growing body of work has begun to use LLMs to enhance recommender systems, most studies are limited to small, offline benchmarks, and cannot be applied in real-world recommendation environments (Wu et al., 2024; Zhao et al., 2024). How to effectively integrate LLMs into large-scale industrial recommender systems to truly understand and mine user intent—so as to overcome the limitations of log-fitting recommendation—remains largely underexplored.

To fill this gap, we introduce **RecGPT**, a production-scale framework that integrates three reasoning LLMs into the core of an industrial recommendation pipeline, forming a closed loop of “User Interest Mining → Item Tag Prediction → Item Retrieval → Explanation Generation” (Figure 2). Specifically, RecGPT first employs a *User-Interest LLM* to comprehensively analyze users’ lifelong behavior history and explicitly generate a concise, natural-language profile of current interests. A second *Item-Tag LLM* then reasons over those interests to generate fine-grained item tags that describe the items the user is most likely seeking. These tags are injected into the item-retrieval stage, expanding the conventional user-item dual-tower matcher into a user-item-tag tri-tower architecture. Consequently, only items that align with the inferred user intent are passed on to the downstream ranking and re-ranking cascade. By turning user behavior logs into a dynamically updated intent signature, RecGPT reshapes candidate generation from collaborative filtering to interest-enhanced process, improving recall relevance and long-tail coverage without changing the downstream infrastructure. Finally, a *Recommendation-Explanation LLM* attaches cached, natural-language user-friendly explanations to final recommended items, completing the loop from intent discovery to transparent delivery.

Our main contributions are summarized as follows:

★ RecGPT has been fully deployed online in the “*Guess What You Like*” scenario on Taobao APP’s homepage, achieving significant performance improvements across multiple stakeholders. From the user perspective, our system enhances CICD by **6.96%** and DT by **4.82%**, indicating effective discovery of users’ latent and diverse interests beyond historical interaction patterns, effectively breaking through information bubbles and expanding recommendation boundaries. For merchants and the platform, notable improvements are observed in CTR (**+6.33%**), IPV (**+9.47%**), and DCAU (**+3.72%**), reflecting substantial commercial value. Additionally, RecGPT effectively alleviates the Matthew effect by providing more equitable exposure opportunities across diverse merchants, ultimately establishing a **win-win-win ecosystem** for users, merchants, and the platform.

★ Unlike traditional collaborative filtering approaches that “learn clicks from clicks”, we leverage LLMs’ world knowledge and reasoning capabilities to explicitly mine users’ latent interests from behavioral patterns, shifting from surface-level correlations to deep profile analysis and preference modeling. To the best of our knowledge, ***we are the first to deploy a reasoning-enhanced hundred-billion-scale recommendation foundation model in industrial applications serving over a billion consumers and items***, which powerfully validates and advances the practical potential and value of large language models for recommender systems.

★ To enable effective LLM integration in large-scale industrial recommender systems, we develop a systematic multi-stage training framework that addresses the unique challenges of adapting general-purpose LLMs to recommendation-specific tasks. Our approach progresses from reasoning-enhanced pre-alignment to self-training evolution, leveraging LLM-as-a-Judge capabilities for automated data quality curation and model evaluation. This framework enables a **progressive transition from manual expert review to a Human-LLM cooperative judge system**, significantly accelerating model iteration cycles while maintaining rigorous quality standards.

2. RecGPT Workflow

In this section, we present the overall workflow of RecGPT, as illustrated in Figure 2. The core idea of RecGPT is to leverage large language models to empower different stages of the recommendation pipeline, including user interest understanding, item prediction, and generating user-friendly recommendation explanations for final results. We introduce three corresponding LLM modules: \mathcal{LM}_{UI} serves user interest mining tasks, \mathcal{LM}_{IT} handles item tag prediction tasks, and \mathcal{LM}_{RE} generates recommendation explanations. Furthermore, to map the items (referred to as *item tags* in this paper) predicted by \mathcal{LM}_{IT} to specific items within the in-domain item corpus, we propose a tag-aware semantic relevance retrieval method. This approach leverages the deep semantic understanding derived from LLM-generated item tags, which captures user intent through reasoning-based analysis rather than surface-level feature matching. By integrating these LLM-driven semantic insights with traditional collaborative filtering signals, our item retrieval method effectively balances semantic relevance and collaborative patterns, enabling both exploration of users’ potential diverse preferences and exploitation of their established behavioral patterns.

Overall, the RecGPT workflow consists of the following components:

- ◆ **User Interest Mining** (Section 2.1): Through \mathcal{LM}_{UI} , we conduct explicit interest mining on users’ lifelong multi-behavior sequences to identify diverse user interest patterns.
- ◆ **Item Tag Prediction** (Section 2.2): Based on user interest mining results, we use \mathcal{LM}_{IT} to predict item tags that represent users’ potential preference distributions.
- ◆ **Item Retrieval** (Section 2.3): The tag-aware semantic retrieval method maps predicted tags to

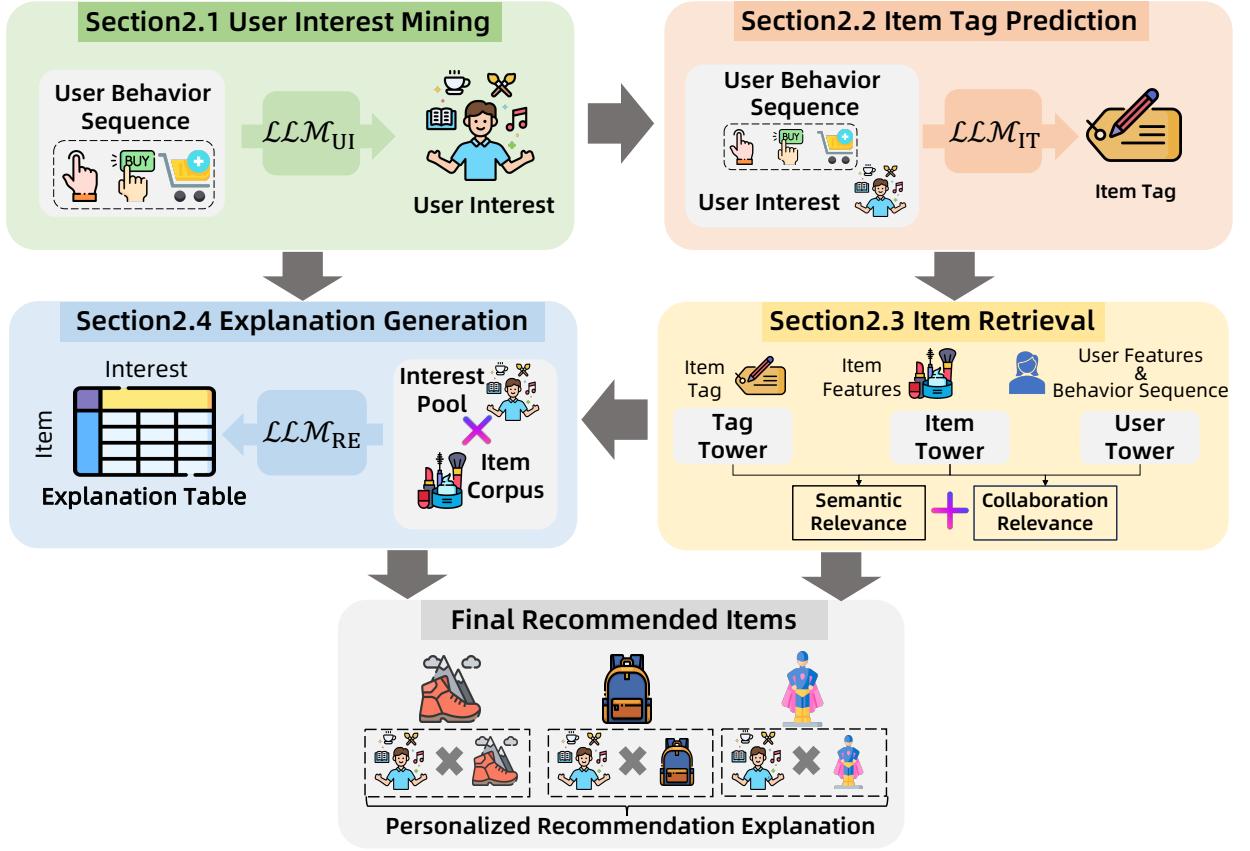


Figure 2 | The overall workflow of RecGPT. \mathcal{LLM}_{UI} , \mathcal{LLM}_{IT} , and \mathcal{LLM}_{RE} represent LLMs for user interest mining, item tag prediction, and recommendation explanation generation, respectively. RecGPT employs “user interest mining → item tag prediction” to identify potential items matching user preferences, then performs item retrieval through tag-aware semantic relevance combined with behavioral collaboration to map inferred tags to specific in-domain items. Finally, \mathcal{LLM}_{RE} generates user-friendly recommendation explanations based on the retrieval results and user interests.

specific items while incorporating user behavioral collaborative signals to balance semantic and collaborative relevance.

- ◆ **Recommendation Explanation Generation (Section 2.4):** \mathcal{LLM}_{RE} synthesizes user interests and recommended items to generate personalized and user-friendly explanations that resonate with individual user preferences, improving system transparency and user experience.

In contrast to traditional recommendation algorithms that rely on latent features and final user feedback for optimization, RecGPT employs explicit text-based modeling through large language models across pipeline stages. This approach provides two key advantages: First, it enables interpretable monitoring of intermediate processes and model performance at each stage. Second, it facilitates expert knowledge integration through process-level supervision, allowing targeted optimization of individual components. By decomposing the workflow into manageable sub-tasks with clear input-output relationships, this methodology simplifies end-to-end optimization while enabling independent evaluation and refinement.

2.1. User Interest Mining

The fundamental goal of recommender systems is to understand users' personalized preferences through their historical interaction behaviors and achieve item product recommendations. However, existing recommendation algorithms rely solely on fixed, statistical implicit user features, making it difficult to explicitly model users' dynamic and complex interests. To overcome these fundamental limitations, we introduce **Generative User Profiling**, a novel approach that harnesses the powerful reasoning capability of LLMs to revolutionize user interest modeling. However, despite their promising potential in natural language understanding, several key challenges hinder LLMs' effectiveness in user interest mining:

(1) Context Window Limitations. Real-world user behavioral histories in recommender systems exhibit vast scale and complexity. Within Taobao's e-commerce platform, users possess over 37k historical behavior records on average. These extensive behavioral sequences pose significant challenges to current LLMs limited by 128k-token context windows.

(2) Domain Knowledge Gaps. Despite broad world knowledge, LLMs lack specialized understanding of domain-specific features in platforms like Taobao. This knowledge gap hinders the models' ability to effectively extract and abstract user interests from raw interaction data at an expert level.

To overcome these challenges, we first develop ***Reliable Behavioral Sequence Compression*** to preserve critical temporal information while reducing input length, enabling better adaptation to LLM context window constraints (Section 2.1.1). Building upon this, we present a multi-stage ***Task Alignment*** framework for the ***User-Interest LLM \mathcal{LLM}_{UI}*** to enhance domain-specific user interest mining capabilities (Section 2.1.2).

2.1.1. Reliable Behavioral Sequence Compression

To enhance the reliability and effectiveness of user behavior sequences, we first employ a ***Reliable Behavior Extraction*** method to filter out noise and redundant information from user behaviors. Furthermore, to accommodate ultra-long user behavior sequences within the context window limitations of LLMs, we develop a ***Hierarchical Behavior Compression*** method for heterogeneous behavior compression.

Reliable Behavior Extraction. To ensure that user interaction behaviors accurately reflect genuine user interests, we first extract reliable signals from large-scale, multi-source, multi-behavior user sequences as the data foundation for user interest modeling. We define the following reliable behaviors (using Taobao e-commerce platform as an example):

- ◆ **Intentional Feedback Behaviors** include high-engagement actions such as “favorites”, “purchases”, “add-to-cart”, and deliberate click behaviors like “detailed product views” and “reviews reading”, demonstrating strong user interest through direct engagement actions or focused clicking patterns that reflect deep attention to item details and provide robust signals for modeling user preferences and purchase intentions.
- ◆ **Search Behaviors** comprise actions such as “search queries” for product discovery. These behaviors represent deliberate exploration efforts, revealing user intentions toward specific product categories or attributes.

Note that we exclude ordinary product clicking behaviors, as these actions may contain considerable noise and are less effective at reflecting user interests compared to the defined reliable behaviors.

Hierarchical Behavior Compression. To accommodate the ultra-long user behavior sequences, we develop a hierarchical compression method that compresses multi-source heterogeneous behaviors into a unified sequence format. Specifically, we employ compression strategies at both the item level and sequence level, which are designed to reduce the input length while preserving essential information for interest mining.

(1) **Item-level Compression.** Considering that raw item information contains substantial redundant and irrelevant details, using this raw data as input to LLMs would result in low information density and excessive token consumption. Therefore, we first compress the relevant information for each item. Here, we prompt the LLM to compress detailed item information while preserving core attributes such as item name, category, brand, and other essential features.

(2) **Sequence-level Compression.** After compressing individual item information, we further compress the user’s behavioral sequences through a two-step aggregation process. We first partition the user’s behavior sequence into different time periods (daily partitions for behaviors within one month, monthly partitions for behaviors spanning multiple months, and yearly partitions for behaviors exceeding one year). Our compression method operates through two complementary aggregation steps: (1) **Step 1: Temporal-Behavioral Aggregation.** We use “time-behavior type” pairs as keys to group all items that the user interacted with during specific time periods through specific behaviors. (2) **Step 2: Item-based Reverse Aggregation.** We then reverse this mapping by using item sequences as keys to aggregate their corresponding time-behavior type combinations. For example, items that frequently appear together across different time periods and behaviors are grouped, with their associated temporal-behavioral contexts preserved. This dual aggregation process produces compressed behavioral sequences in the following format (specific cases can be referred to in Section 4.4):

“Time₁ (Behavior₁, Behavior₂, …), Time₂ (Behavior₁, Behavior₃, …), … | Item₁, Item₂, …”

This representation efficiently captures both temporal behavioral patterns and item co-occurrence relationships while significantly reducing overall prompt sequence length.

Through the above behavior extraction and compression process, we obtain multi-source heterogeneous user behavior sequences $\mathcal{B}_u = [B_{u,1}, B_{u,2}, \dots, B_{u,|\mathcal{B}_u|}]$ with higher information density that are both refined and reliable, where each behavior $B_{u,i}$ contains multiple interactions within the same time period, along with their corresponding interaction behavior types and associated item information. We empirically demonstrate that the proposed reliable behavior compression method effectively accommodates 98% of user behaviors within the 128k-token context window of large language models, compared to only 88% coverage achieved by uncompressed sequences. Furthermore, this compression approach improves interest inference efficiency by 29%, significantly reducing both inference time and computational costs while maintaining complete behavior representation.

2.1.2. Task Alignment for User Interest Mining

To enhance User Interest LLM \mathcal{LLM}_{UI} capability in interest mining task, we design a multi-stage task alignment framework with the following stages to develop a human-aligned \mathcal{LLM}_{UI} :

Stage 1: Curriculum Learning-based Multi-task Fine-tuning. We first design 16 preparatory subtasks (containing 16.3k training samples) to enhance general-purpose LLMs’ domain-specific foundational abilities. These subtasks develop key competencies across multiple dimensions, such as key information extraction, complex user profile analysis, and causal reasoning. To ensure stable capability enhancement, inspired by curriculum learning principles (Bengio et al., 2009; Pentina et al., 2015; Soviany et al., 2022; Wang et al., 2021), we organize subtasks through topological sorting based on difficulty levels and dependency relationships, progressively guiding the model to master

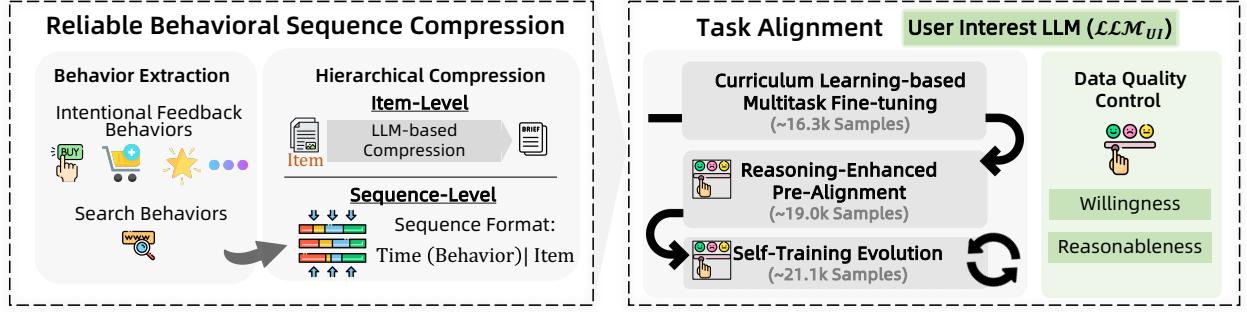


Figure 3 | Illustration of the user interest mining module. The left figure demonstrates the compression processing of lifelong user behavioral sequences, including behavior extraction and hierarchical behavior compression. The right figure shows the multi-stage task alignment framework for user interest mining and data quality control standards.

complex tasks. Relevant experimental details are provided in Appendix C.

Stage 2: Reasoning-Enhanced Pre-alignment. We leverage the advanced reasoning capabilities of DeepSeek-R1 (Guo et al., 2025a) to generate high-quality training data for interest mining. Through careful manual curation, we distill the initial 90.0k generated samples into a refined 19.0k high-quality dataset. This dataset serves as the foundation of knowledge distillation, allowing the user interest LLM LLM_{UI} to achieve performance comparable to the teacher model via pre-alignment fine-tuning.

Stage 3: Self-Training Evolution. To further enhance the model’s capability ceiling, we propose a self-training paradigm that enables continuous self-evolution. In this stage, the model generates its own training data and uses these self-generated samples for iterative optimization, creating a feedback loop for capability improvement. During this self-training process, we collect 21.1k high-quality samples that drive the model’s evolution. To efficiently filter these self-generated outputs and evaluate model performance at low cost, we adopt a *Human-LLM collaborative paradigm* with *LLM-as-a-Judge* capabilities for data quality control and assessment. This collaborative framework significantly improves curation efficiency while reducing manual annotation costs. We provide a comprehensive introduction to the Human-LLM collaborative judge system in Section 3.

In what follows, we focus on explaining the **Prompt Engineering** strategy and **Data Quality Control** protocols for the interest mining task. Additionally, we present extensive **Human Evaluation Experiments** to demonstrate the effectiveness of our self-training method and provide practical **Online Deployment** details.

Prompt Engineering. Our carefully designed prompt template takes compressed behavior sequences (*cf.* Section 2.1.1) and personal attributes as user information, instructing the LLM to generate diverse yet precise interest profiles. To improve generation accuracy, the template incorporates *Chain-of-Thought* (CoT) reasoning that guides the model through explicit logical steps instead of direct interest prediction. The prompt template structure is shown in Prompt 2.1.2, where the placeholders **{User Attributes}** and **{Compressed Reliable Behavioral Sequences}** are instantiated with user-specific contextual data. Detailed specifications for **{Other Interest Mining Requirements}** and **{Other Constraints}** are provided in Appendix B due to space limitations. The **{Matched Interest Pool}** is filled with a dynamic collection of matched interests.

User Interest Mining Prompt Template

Role

You are a shopping guide for an e-commerce platform. Based on users' behavioral history, you need to accurately and comprehensively analyze their potential interests and preferences.

Input

User Attribute: **{User Attributes}**

User Behavioral Information: **{Compressed Reliable Behavioral Sequences}**

Mandatory Requirements

Task Requirements **{Interest Mining Requirements}**

Task Constraints: **{Constraints}**

Preset Interest List

{Matched Interest Pool}

Output Format

(Detailed output format requirements)

Following the prompt template design, we formalize the user interest mining process as follows. Given a target user's attributes \mathcal{A}_u extracted from user-provided information (e.g., age, gender, location), compressed behavioral sequence \mathcal{B}_u , and the pre-configured matched candidate interest pool \mathcal{I}_m , we leverage the User-Interest LLM \mathcal{LLM}_{UI} to perform inference:

$$\mathcal{I}_u = \mathcal{LLM}_{UI}(\mathcal{A}_u, \mathcal{B}_u, \mathcal{I}_m | \mathcal{P}_{UI}), \quad (1)$$

where \mathcal{P}_{UI} represents the user interest mining prompt template, and $\mathcal{I}_u = \{I_{u,1}, I_{u,2}, \dots, I_{u,|\mathcal{I}_u|}\}$ denotes the set of potential user interests inferred from the given user information. This approach enables the model to synthesize user attributes, behavioral patterns, and candidate interests through CoT-based reasoning to mine personalized interest profiles.

Data Quality Control. To mitigate potential LLM hallucination and inaccuracies, we employ multi-dimension reject sampling for training data preparation. The evaluation criteria for correct (✓) and incorrect (✗) interests include 2 dimensions:

- ◆ **Willingness.** This criterion evaluates whether the identified interests genuinely reflect voluntary user preferences rather than external obligations. Genuine interests embody intrinsic motivation and active exploration, distinguishing them from necessity-driven behaviors.
 - ✓ **Spontaneity.** The interest originates from voluntary affection and personal choice, which is driven by curiosity, passion, or fulfillment that aligns with one's values, experiences, or aspirations. Such interests reflect authentic user preferences and intrinsic motivation.
 - ✗ **Necessity.** Behaviors misclassified as interests may actually stem from external pressures, survival needs, or situational requirements (e.g., career-related skill acquisition). These represent functional demands rather than genuine personal interests.
- ◆ **Reasonableness.** This criterion ensures that inferred user interests have sufficient behavioral evidence for support, maintaining reliability and accuracy in interest identification. We establish four correlation degrees to evaluate interest reasonableness:
 - ✓ **Strong Correlation.** The inferred interest and observed behaviors demonstrate clear, logical connections with compelling evidence. The behavioral patterns strongly support the interest inference with minimal ambiguity.

Table 1 | Criteria for Evaluating Model-Generated Interests. **Correct** indicates interests that meet both Willingness and Strong Correlation criteria, while **Incorrect** includes four categories: Necessity, Weak Correlation, No Correlation and Hallucination. “Both ✓ → ✓” means when both criteria are satisfied, the interest is considered correct, while “Any ✗ → ✗” means if any criterion is violated, the interest is deemed incorrect.

Label	Evaluation Criteria	Example	Why ✓ or ✗
Correct (Both ✓ → ✓)	Spontaneity (Willingness)	[Interest] <i>tennis</i> .	Tennis is an interest from affection. (Willingness ✓)
	Strong Correlation (Reasonableness)	[Reason] <i>User purchased a tennis racket</i> .	Reason and the interest are related reasonably. (Strong Correlation ✓)
Incorrect (Any ✗ → ✗)	Necessity (Willingness)	[Interest] <i>Purchasing household necessities</i> . [Reason] <i>Most purchases are daily cleaning items</i> .	Purchasing household items is a daily need, not a hobby.
	Weak Correlation (Reasonableness)	[Interest] <i>Yoga</i> . [Reason] <i>Purchases include yoga pants</i> .	May indicate clothing hobby, not yoga preference.
	No correlation (Reasonableness)	[Interest] <i>Model making</i> . [Reason] <i>User searched for many blankets</i> .	Interest and reasoning are unrelated.
	Hallucination (Reasonableness)	[Interest] <i>Smart home</i> . [Reason] <i>User purchased smart home accessories</i> .	User history has no smart home-related activities.

- ✗ **Weak Correlation.** User behaviors show partial connection to the inferred interest but lack sufficient evidence. For example, purchasing a tennis skirt may provide some indication of interest in tennis but cannot conclusively establish this preference.
- ✗ **No Correlation.** User behaviors exhibit no meaningful connection to the inferred interest. For instance, purchasing “The Prince of Tennis” merchandise does not indicate interest in tennis sports, as it likely reflects interest in the anime rather than the sport.
- ✗ **Hallucination.** The generated interest has no behavioral evidence, representing unfounded associations fabricated by the LLM.

Using the above data quality control protocol, we retain data exhibiting strong user intent and clear correlations for training, while filtering out data with weak intent, poor correlations, or hallucinations as low-quality samples that could introduce noise and bias into the learning process. Through this rigorous data curation process, we ensure consistently high data quality across both pre-alignment and self-training stages, thereby improving model performance on user interest mining tasks.

Human Evaluation Experiments. To validate the effectiveness of our multi-stage alignment framework, we conduct human evaluation on user interest mining performance across different models. We evaluate two foundation LLMs: DeepSeek-R1 and Qwen3-14B (hereafter referred to as **Qwen3**-

Table 2 | Human-evaluated pass rates for different models on user interest mining task. The best performance is highlighted in **bold**.

Model	DeepSeek-R1	Qwen3-Base	Qwen3-SFT	TBStars-SFT
Pass Rate (%)	70.00	59.74	77.28	74.39

Base), alongside our multi-stage aligned models: Qwen3 (hereafter referred to as **Qwen3-SFT**) and TBStars-42B-A3.5 (hereafter referred to as **TBStars-SFT**), a sparse Mixture-of-Experts (MoE) large language model internally developed by Taobao that activates only 3.5B parameters per inference. The generated interest is considered “passed” only when it satisfies all data quality evaluation criteria outlined before, including both willingness and reasonableness dimensions.

As shown in Table 2, DeepSeek-R1 achieves a 70.00% pass rate, significantly outperforming Qwen3-Base at 59.74%. The performance gap demonstrates that reasoning-enhanced LLMs possess superior contextual understanding capabilities for mining user interests from ultra-long behavioral sequences. Furthermore, our multi-stage aligned Qwen3-SFT achieves the highest performance at 77.28%, substantially outperforming its base counterpart. The experimental results validate the effectiveness of our alignment framework in enhancing domain-specific interest mining capabilities.

For practical online deployment, TBStars-SFT achieves 74.39% pass rate after full-parameter fine-tuning with the collected high-quality data, demonstrating significant improvement over both baseline models (*i.e.*, DeepSeek-R1 and Qwen3-Base) while maintaining superior performance-efficiency balance due to its sparse architecture. This characteristic makes it particularly suitable for large-scale online recommendation scenarios where both accuracy and inference speed are critical.

Online Deployment. We utilize the model \mathcal{LLM}_{UI} offline to predict users’ interest preferences, with an average of **16.1** predicted interests per user. During online deployment, we perform iterative model optimization and refresh user interests every two weeks to ensure the timeliness of the user interest, while precisely capturing the dynamic changes in users’ personalized interests.

2.2. Item Tag Prediction

In this section, we explore how to leverage large language models to guide item tag prediction based on inferred user profiles. To adapt the world knowledge of LLMs to the specific product domain, similar to the User Interest LLM \mathcal{LLM}_{IT} , we first perform multi-stage **Task Alignment** to ensure \mathcal{LLM}_{IT} can effectively understand and process product-related contextual information (Section 2.2.1). Next, we introduce a **Incremental Learning** method that enables the model to continuously adapt to evolving user interests and new product trends (Section 2.2.2).

2.2.1. Task Alignment for Item Tag Prediction

While foundation large language models show impressive general capabilities, they prove inadequate when directly applied to personalized item prediction tasks due to the domain-specific requirements of recommender systems. To overcome this limitation, we adopt a two-stage alignment process similar to the Item-Tag LLM approach. This process employs **Reasoning-Enhanced Pre-Alignment** and **Self-Training Evolution** to enhance \mathcal{LLM}_{IT} with domain-aware product understanding. Here, we focus on the **Prompt Engineering** and **Data Quality Control** protocols specifically designed for the item tag prediction task. Furthermore, we provide extended **Human Evaluation Experiments** to demonstrate the effectiveness of our alignment approach in improving model performance.

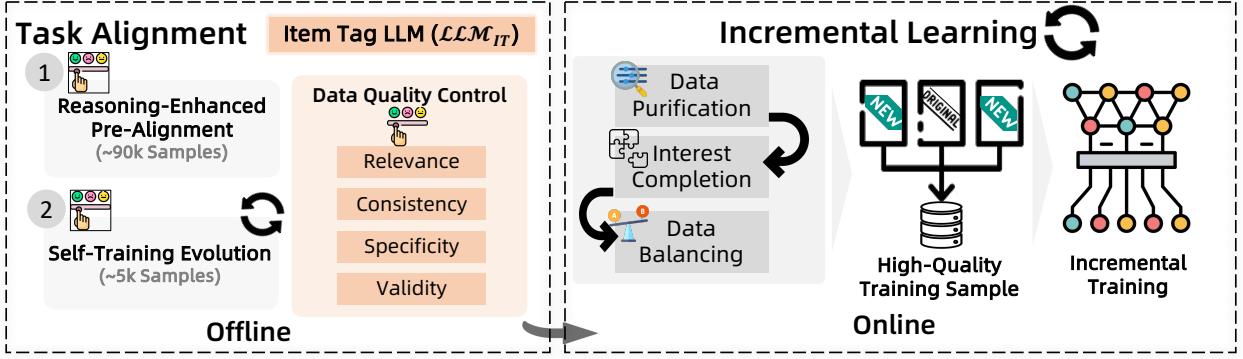


Figure 4 | Illustration of the item tag prediction module. The left figure demonstrates the two-stage alignment process for the item tag LLM and data quality control standards, while the right figure shows data cleaning and incremental learning based on real online user feedback data.

Prompt Engineering. We require the Item-Tag LLM \mathcal{LLM}_{IT} to predict tag sets in the “**Modifier + Core-Word**” format (e.g., “Outdoor waterproof non-slip hiking boots”), based on provided user profile information and multi-behavior interaction sequences (such as clicks, purchases, searches). We employ CoT-based tag reasoning to fully leverage the reasoning capabilities of large language models. Additionally, we incorporate the following constraints in our prompts to ensure the generated tag sets meet the practical requirements of recommender systems as follows:

- ◆ **Interest Consistency:** Generated tags are constrained to maintain alignment with user interests, thereby preventing recommendations that contradict established user preference profiles.
- ◆ **Diversity Enhancement:** A minimum of 50 tags is enforced to guarantee diverse recommendation across broad categories, mitigating filter bubble phenomena.
- ◆ **Semantic Precision:** Tag generation is restricted to semantically focused descriptions, eliminating vague or overly broad categorizations that compromise recommendation accuracy and user experience quality.
- ◆ **Temporal Freshness:** The generated tag should prioritize novel product categories while systematically avoiding repetitive recommendations of recently engaged items, ensuring both temporal relevance and diversified suggestions.
- ◆ **Seasonal Relevance:** Temporal context is integrated into the tag generation process to produce seasonally appropriate recommendations aligned with the provided timestamp, enhancing user satisfaction through contextually aware suggestions.

Through the above multi-constraint prompting strategies, \mathcal{LLM}_{IT} generates a list of triplets containing **(Tag, Associated Interest Preference, Rationale)** for subsequent item retrieval. The detailed tag prediction prompt template structure is presented in Prompt 2.2.1, where **{User Attributes}**, **{User Interests}**, **{Click Behavior Sequence}**, **{Purchase Behavior Sequence}**, **{Search Behavior Sequence}**, and **{Extra Information}** are placeholders for the related user profile and behavior records, which are dynamically filled in during the model inference process. Regarding **{Recommendation Principles}**, **{Recommendation Requirements}**, **{Quantity Requirements}**, and **{Strict Prohibitions}** in the prompt, due to space constraints, we omit the detailed descriptions here and provide specifications in the Appendix B.

Item Tag Prediction Prompt Template

Role

You are a professional product recommendation specialist for the Taobao app.

Input

User Attributes: {Generated User Attributes}

User Interests: {Generated User Interests}

User Behavior Information

Click Behavior Sequence: {Click Behavior Sequence} | Purchase Behavior Sequence: {Purchase Behavior Sequence} | Search Behavior Sequence: {Search Behavior Sequence} | Extra Information: {Extra Information}

Mandatory Requirements

Task Requirements: **Recommendation Principles (✓)** | {Recommendation Principles} | **Recommendation Requirements (✓)** | {Recommendation Requirements} | **Quantity Requirements (✓)** | {Quantity Requirements} | **Strict Prohibitions (✗)** | {Strict Prohibitions}

Output Format

(Detailed output format requirements)

Following the prompt template design, we formalize the item tag prediction process as follows. Given the inferred user profile information, including user attributes \mathcal{A}_u and interests \mathcal{I}_u , along with user multi-behavior interaction sequences \mathcal{S}_u (comprising click behaviors, purchases, and search queries), we leverage the Item Tag LLM to predict item tags that the user might interact with next:

$$\mathcal{T}_u = \mathcal{LLM}_{IT}(\mathcal{A}_u, \mathcal{I}_u, \mathcal{S}_u | \mathcal{P}_{IT}), \quad (2)$$

where \mathcal{P}_{IT} represents the item tag prediction prompt template, and $\mathcal{T}_u = \{T_{u,1}, T_{u,2}, \dots, T_{u,|\mathcal{T}_u|}\}$ denotes the predicted set of item tags that the user is likely to interact with, inferred from the joint analysis of user interaction history and profile information.

Data Quality Control. To align \mathcal{LLM}_{IT} with human consistency and enable it to function like a real shopping assistant, we also introduce multi-dimensional rejection sampling to achieve high-quality training sample filtering:

- ◆ **Relevance:** Evaluates whether the generated tags are directly aligned with the user's associated interests. This criterion measures the model's capacity to genuinely understand and accurately predict user needs by assessing whether the tag matches the specified interest.
- ◆ **Consistency:** Assesses whether the item tag is generated with explicit reference to the user's profile information and historical behavioral data. This criterion focuses on whether the model's reasoning process incorporates authentic user context rather than fabricating or ignoring the given user information, ensuring that the generated tags are grounded in real user data.
- ◆ **Specificity:** Evaluates tag specificity to avoid generic term like "fashion sports equipment" that lead to imprecise product retrieval.
- ◆ **Validity:** Determines whether the predicted tags correspond to an actual existing product, preventing non-existent tag generation.

Based on the above multi-dimensional reject sampling criteria, we conduct strict quality control on the model-generated tags. Specifically, if a tag meets all criteria, it is labeled as a qualified sample for

Table 3 | Criteria for Model-Generated Item Tags, where **Correct** indicates tags that meet all criteria, while **Incorrect** includes four categories: Weak Relevance, Low Consistency, Low Specificity, and Invalid Tag. “All ✓ → ✓” means when all criteria are satisfied, the tag is considered correct, while “Any ✗ → ✗” means if any criterion is violated, the tag is deemed incorrect.

Label	Evaluation Criteria	Example	Why ✓ or ✗
Correct (All ✓ → ✓)	Strong Relevance High Consistency High Specificity Valid Tag	[Tag] <i>Foldable Pet Case.</i> [Interest] <i>Cat Ownership.</i> [Reason] <i>User has a cat and focuses on portable storage.</i>	Direct match to user need and history. (Relevance ✓ & Consistency ✓) Tag is specific and real. (Specificity ✓ & Validity ✓)
Incorrect (Any ✗ → ✗)	Weak Relevance	[Tag] <i>Embroidery bird-and-flower pattern silk pillowcase.</i> [Interest] <i>Skincare.</i>	No direct link to skincare.
	Low Consistency	[Tag] <i>Calcium supplement powder for elderly health.</i> [attribute age] <i>20.</i> [historical behaviors] <i>None.</i>	User is too young and no relevant history.
	Low Specificity	[Tag] <i>Mountaineering outdoor sports equipment.</i>	Too broad as a tag.
	Invalid Tag	[Tag] <i>Smart coaster.</i>	Product does not exist.

training; if any criterion is not satisfied, the tag is marked as an unqualified sample. Table 3 shows the specific reject sampling criteria and examples. Through this approach, we ensure that \mathcal{LLM}_{IT} can perform two-stage alignment on high-quality items that meet human evaluation standards, improving the accuracy and reliability of tag prediction.

Human Evaluation Experiments. To validate the effectiveness of our task alignment approach for item tag prediction, we conduct human evaluation on model-generated item tags according to the aforementioned criteria. A predicted tag is considered qualified only when it satisfies all evaluation standards. We compare four models: DeepSeek-R1, Qwen3-Base, Qwen3-SFT, and TBStars-SFT, where Qwen3-SFT and TBStars-SFT are full-parameter fine-tuned models based on our multi-stage task alignment framework. As shown in Table 4, several key insights can be drawn from the results:

(1) Qwen3-Base achieves only 33.70% pass rate, demonstrating substantial limitations when directly applying base LLMs to item tag prediction tasks. This poor performance indicates that foundation models lack sufficient domain knowledge and task-specific adaptation capabilities.

(2) Both aligned models, Qwen3-SFT (84.80%) and TBStars-SFT (88.80%), significantly outperform the DeepSeek-R1 (80.00%). These results validate that our knowledge distillation from strong models and self-training evolution approach enables smaller-scale language models to progressively approach and eventually exceed the performance of reasoning language models.

(3) TBStars-SFT achieves the highest performance at 88.80% pass rate, substantially outperforming all other models. Beyond superior accuracy, the additional low-latency inference advantage of TBStars-SFT makes it particularly suitable for industrial recommender systems where both prediction quality and computational efficiency are necessary for practical deployment.

Table 4 | Human-evaluated pass rates for different models on item tag prediction task. The best performance is highlighted in **bold**.

Model	DeepSeek-R1	Qwen3-Base	Qwen3-SFT	TBStars-SFT
Pass Rate (%)	80.00	33.70	84.80	88.80

2.2.2. Incremental Learning

To better adapt to dynamic user interests and data distribution shifts in online environments (such as seasonal changes), we adopt a bi-weekly *Incremental Learning (IL)* method for updating the $\mathcal{LLM}_{\text{IT}}$. During each update cycle, we select users' online interaction records (e.g., clicks, purchases) from the past 14 days as the data source for incremental training. However, real-world data presents two critical challenges: (1) **Substantial Noise**: e.g., accidental clicks or promotional artifacts, that misrepresent genuine preferences, and (2) **Inherent Imbalance**: dominant interest tags may skew model training—potentially degrading recommendation diversity, exacerbating filter bubbles, and reinforcing Matthew effects. To address these dual challenges, we design the following three-step process to handle online user behavior data:

Step 1. Data Purification. Following the data quality criteria for relevance and timeliness outlined in Section 2.2.1, we employ the QwQ-32B (Team, 2025) as an automated judge for data cleaning. Specifically, for relevance, we analyze the consistency between user behaviors and their underlying interests, filtering out low-quality interaction records that do not align with user preferences. For timeliness, we focus on whether user behaviors satisfy seasonal requirements, *i.e.*, whether the products are suitable for the current season or the upcoming season. This approach ensures high-quality training data by minimizing noise behaviors from random clicks and transient behaviors.

Step 2. Interest Completion. To construct structured training data, we need to map users' valid interaction behaviors to triplet outputs in the form of (*Tag, Associated Interest Preference, Rationale*). Specifically, we use QwQ-32B to perform deep reasoning based on given information (including user profiles, historical behaviors, and requirement prompts) to infer underlying interest preferences and the corresponding justifications that support user behaviors. Besides, since we can obtain real user interaction behaviors in this context, we directly use the item titles as tags. Through this approach, we can transform user behavioral data into structured data samples suitable for model training.

Step 3. Data Balancing. In this step, we design a two-stage data resampling strategy to address the inherent imbalance in online user behavior data. Specifically, in the first stage, for each user, we first randomly select behavioral records corresponding to 80 item tags to ensure training data diversity and representativeness while improving training efficiency. In the second stage, we further utilize a pre-trained Tag-to-Cate model $\phi(\cdot)$ to convert these item tags into corresponding category labels (note that the number of categories is much smaller than the number of tags), and perform secondary sampling based on this, ensuring that the number of samples for each category is roughly equal (in our experimental setup, we sample at most 2 samples per category) to achieve data balance.

Through the above process, we ultimately obtain high-quality, diversified incremental online training data. This incremental learning strategy not only helps the $\mathcal{LLM}_{\text{IT}}$ learn the dynamic changes in users' latest preferences and product knowledge, but also effectively improves the model's generalization capability and recommendation accuracy, avoiding repeated recommendations of duplicate and outdated products, and achieving weekly optimization for industrial RS.

Table 5 | Performance comparison of tag prediction accuracy before and after using incremental learning (IL). The best performance is highlighted in **bold**.

Model	TBStars-SFT (w/o IL)	TBStars-SFT (w/ IL)
HR@30	0.3671	0.3776 (+1.05%)

Incremental Learning Effectiveness Evaluation To validate the effectiveness of incremental learning, we leverage real online user interaction behaviors for verification. Specifically, we utilize the \mathcal{LLM}_{IT} to predict 30 item tags for each user’s historical sequence according to Eq. (2), and employ the Tag-to-Gate model $\phi(\cdot)$ to convert these tags $\mathcal{T}_u = \{T_{u,1}, T_{u,2}, \dots, T_{u,30}\}$ into specific predefined product category $C_u^{\text{pred}} = \{C_{u,1}, C_{u,2}, \dots, C_{u,30}\}$. We design the HR@30 metric to evaluate item tag prediction accuracy, which is formulated as:

$$\text{HR}@30 = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbb{I}(C_u^{\text{gt}} \in C_u^{\text{pred}}),$$

where \mathcal{U} represents the set of test users, C_u^{gt} denotes the product category of user u ’s actual next interacted item, C_u^{pred} represents the set of predicted categories converted from the 30 predicted item tags for user u , and $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if the predicted category set contains the true next interaction category, and 0 otherwise.

To demonstrate the practical value of our incremental learning approach, we conduct a comparative analysis using TBStars-SFT, the foundation recommendation model currently deployed in our online system. We evaluate the prediction accuracy before and after applying incremental training on real-world data. As presented in Table, we observe that the model fine-tuned with cleaned and balanced datasets from online interactions achieves a notable 1.05% improvement in HR@30 compared to the baseline model without incremental learning. This improvement is particularly significant considering the scale and complexity of real-world recommendation scenarios, where even marginal gains can translate to substantial business impact. The results validate the effectiveness of our incremental learning strategy in adapting to evolving user preferences and emerging product trends.

2.3. Item Retrieval

While the LLM-generated item tags provide rich semantic understanding of user preferences, a critical challenge emerges: these abstract semantic representations cannot be directly mapped to specific products within the target domain. The gap between high-level semantic concepts and concrete item characteristics necessitates an effective bridge mechanism for practical recommendation deployment. To address this challenge, we introduce a tag-aware method that seamlessly connects semantic understanding with item retrieval. Furthermore, recognizing that collaborative knowledge derived from user-item interactions contains valuable behavioral patterns complementary to semantic signals, we integrate collaborative filtering mechanisms to enhance retrieval effectiveness. This leads to a unified **User-Item-Tag Retrieval Framework** that synergistically combines semantic reasoning capabilities with collaborative behavioral insights, ultimately improving both the accuracy and efficiency of online recommender systems.

In the following sections, we present the overall architecture of user-item-tag retrieval framework (Section 2.3.1), the collaborative-semantic enhanced optimization algorithm (Section 2.3.2) and the online inference methodology (Section 2.3.3).

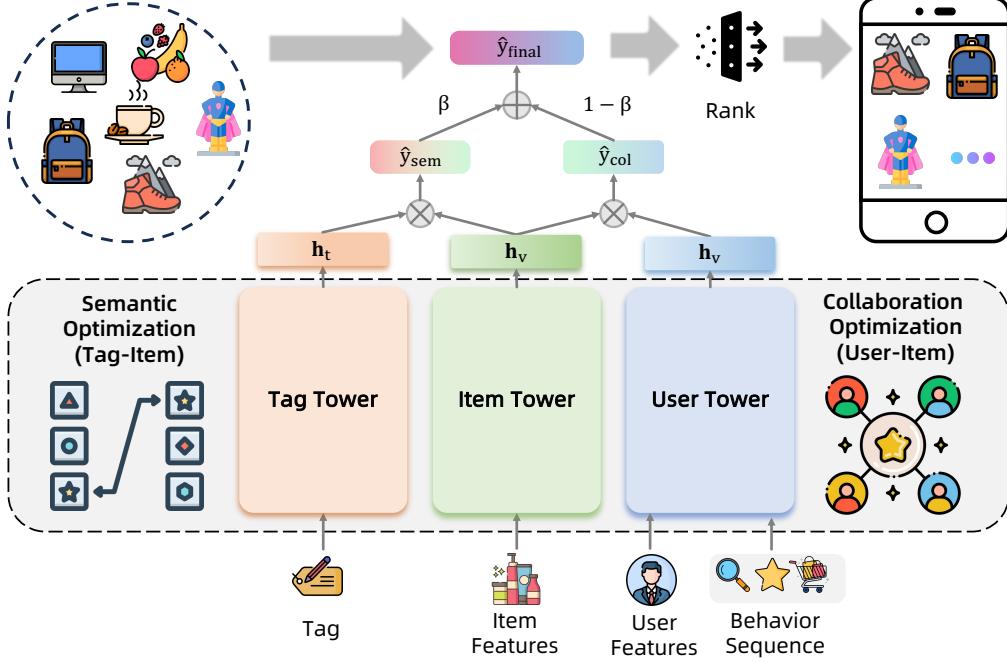


Figure 5 | User-Item-Tag Retrieval Framework. The framework jointly optimizes tag tower and item tower for semantic enhancement, while combining user tower and item tower for collaborative optimization. Based-on collaborative-semantic relevance scoring, retrieved items are aligned with user interests at the source level, with filtered items subsequently fed into the ranking stage.

2.3.1. Overall Architecture

The overall architecture of TAR is illustrated in Figure 5. The framework consists of three parallel towers: the Item Tower, User Tower, and Tag Tower. We introduce the details of each tower as follows:

Item Tower: Given an item v , we define its feature set as $\mathcal{F}_v = \{\mathcal{F}_v^{\text{sparse}}, \mathcal{F}_v^{\text{dense}}\}$, where $\mathcal{F}_v^{\text{sparse}} = \{\text{item id}, \text{category}, \text{brand}, \dots\}$ represents sparse categorical features and $\mathcal{F}_v^{\text{dense}} = \{\text{price}, \text{sales}, \dots\}$ denotes continuous numerical features. The feature embedding layer transforms both sparse and discretized dense features into uniform dense vectors:

$$\begin{aligned} \mathbf{e}_j^{\{\text{sparse}\}} &= \text{EMB}(f_j^{\{\text{sparse}\}}), \quad f_j^{\{\text{sparse}\}} \in \mathcal{F}_v^{\{\text{sparse}\}} \\ \mathbf{e}_k^{\{\text{dense}\}} &= \text{EMB}(\text{Discretize}(f_k^{\{\text{dense}\}})), \quad f_k^{\{\text{dense}\}} \in \mathcal{F}_v^{\{\text{dense}\}} \end{aligned}$$

where $\text{EMB}(\cdot)$ denotes the embedding operation, and $\text{Discretize}(\cdot)$ converts continuous features into discrete representations. The item tower then applies a Deep Neural Network (DNN) to learn the item representation:

$$\mathbf{h}_v = \text{DNN}_{\text{item}}([\mathbf{e}_1^{\{\text{sparse}\}}; \mathbf{e}_2^{\{\text{sparse}\}}; \dots; \mathbf{e}_1^{\{\text{dense}\}}; \mathbf{e}_2^{\{\text{dense}\}}; \dots])$$

User Tower: The user tower captures user preferences through multi-behavioral sequence modeling. For user u , the input features include user ID and multi-behavior interaction sequences: $\mathcal{F}_u = \{\text{user id}, \mathcal{S}_u^{\text{click}}, \mathcal{S}_u^{\text{purchase}}, \dots\}$, where $\mathcal{S}_u^{\text{behavior}}$ represents the chronological sequence of user interactions for a specific behavior type. For each behavior sequence, we apply mean pooling over the item representations to obtain the sequence representation $\mathbf{s}_u^{\text{behavior}}$.

The user representation is then computed by concatenating the user ID embedding with the

pooled sequence representations:

$$\mathbf{h}_u = \text{DNN}_{\text{user}}([\text{EMB}(id_u); \mathbf{s}_u^{\text{click}}; \mathbf{s}_u^{\text{purchase}}; \dots])$$

Tag Tower: The tag tower transforms the item tag T into dense representations:

$$\mathbf{h}_t = \text{DNN}_{\text{tag}}(\text{MEAN}([\text{EMB}(w_1); \text{EMB}(w_2); \dots; \text{EMB}(w_n)]))$$

where w_i represents the i -th token in the tokenized tag sequence $T = [w_1, w_2, \dots, w_n]$, and $\text{MEAN}(\cdot)$ denotes the mean pooling operation.

Our framework generates two complementary prediction scores:

$$\begin{aligned}\hat{y}_{\text{col}} &= \mathbf{h}_u^T \mathbf{h}_v \quad (\text{Collaborative Score}), \\ \hat{y}_{\text{sem}} &= \mathbf{h}_t^T \mathbf{h}_v \quad (\text{Semantic Score}).\end{aligned}$$

The score \hat{y}_{col} captures behavioral collaborative patterns through user-item interaction modeling, while the score \hat{y}_{sem} leverages tag-item semantic relevance to understand preference reasoning.

2.3.2. Optimization

In this section, we introduce the optimization objective functions for user-item collaborative modeling and tag-item semantic modeling, respectively.

Collaborative Optimization. The collaborative optimization objective is to maximize the likelihood of positive user-item interactions while minimizing the likelihood of negative interactions. Specifically, we treat items clicked by users as positive samples and unclicked items as negative samples, employing negative sampling from the latter to perform contrastive learning-based optimization. The optimization objective is formulated as follows:

$$\mathcal{L}_{\text{col}} = - \sum_{(u,v) \in \mathcal{D}} \log \frac{\exp(\mathbf{h}_u^T \mathbf{h}_v)}{\exp(\mathbf{h}_u^T \mathbf{h}_v) + \sum_{v' \in \mathcal{V}^-} \exp(\mathbf{h}_u^T \mathbf{h}_{v'})}$$

where \mathcal{D} represents the set of positive user-item pairs, and \mathcal{V}^- denotes the set of sampled negative items for each user.

Semantic Optimization. In contrast to collaborative optimization, semantic optimization aims to maximize the semantic relevance between tags generated based on predicted user preferences and items. We adopt a similar contrastive learning-based optimization, where the positive samples are the tags of clicked items, and the negative samples are randomly sampled tags from other items:

$$\mathcal{L}_{\text{tag}} = - \sum_{(t,v) \in \mathcal{D}} \log \frac{\exp(\mathbf{h}_t^T \mathbf{h}_v)}{\exp(\mathbf{h}_t^T \mathbf{h}_v) + \sum_{v' \in \mathcal{V}^-} \exp(\mathbf{h}_t^T \mathbf{h}_{v'})}$$

where \mathcal{V}^- denotes sampled negative items for each tag.

Furthermore, to prevent overfitting to descriptive tag features, we introduce a category contrastive loss function to enhance semantic discrimination within item categories. Specifically, for each original tag-item pair (t, v) from \mathcal{D} , we sample items from the same category as v to serve as positive samples and items from different categories as negative samples:

$$\mathcal{L}_{\text{cate}} = - \sum_{(t,v) \in \mathcal{D}} \sum_{v^+ \in C_v^+} \log \frac{\exp(\mathbf{h}_t^T \mathbf{h}_{v^+})}{\exp(\mathbf{h}_t^T \mathbf{h}_{v^+}) + \sum_{v^- \in C_v^-} \exp(\mathbf{h}_t^T \mathbf{h}_{v^-})}$$

where C_v^+ represents the set of sampled items from the same category as item v , and C_v^- represents the set of sampled negative items from different categories than item v . This category-aware contrastive learning encourages the model to learn fine-grained semantic distinctions within categories while maintaining clear boundaries across different categories.

The final optimization objective of TAR is formulated as:

$$\mathcal{L}_{\text{TAR}} = \mathcal{L}_{\text{col}} + \underbrace{\alpha \mathcal{L}_{\text{tag}} + (1 - \alpha) \mathcal{L}_{\text{cate}}}_{\mathcal{L}_{\text{sem}}},$$

where α is a hyperparameter that balances the contributions of tag and category contrastive losses. We set $\alpha = 0.5$ in our experiments, indicating equal importance for both losses.

2.3.3. Online Inference

During the inference phase, we dynamically fuse the outputs of the user tower and tag tower to achieve controllable recommendations with collaborative-semantic relevance. Specifically, we first compute the output vectors \mathbf{h}_u and \mathbf{h}_t from the user tower and tag tower (where tags are predicted by $\mathcal{LLM}_{\text{IT}}$), and then perform weighted fusion:

$$\mathbf{h}_{\text{fuse}} = \beta \mathbf{h}_u + (1 - \beta) \mathbf{h}_t$$

where β is a hyperparameter controlling the fusion ratio between user tower and tag tower outputs. This fused representation is used for item retrieval from the candidate pool. Essentially, it is equivalent to computing the final matching score as a weighted sum of the collaborative and semantic scores:

$$\hat{y}_{\text{final}} = \beta \hat{y}_{\text{col}} + (1 - \beta) \hat{y}_{\text{sem}},$$

where the collaborative and semantic signals are balanced according to the fusion weight. The resulting matching scores are then utilized in the downstream recommendation pipeline. This dynamic fusion mechanism enables flexible control over the balance between collaborative filtering signals and semantic understanding, allowing the system to adapt to different recommendation scenarios while maintaining both behavioral relevance and semantic coherence.

2.4. Personalized Explanation Generation

Beyond enhancing the matching between candidate items and user interests, RecGPT also introduces a recommendation explanation generation module to further elevate user experience in recommender systems. This module generates personalized explanations for recommended items, helping users better understand recommendation outputs by answering the fundamental question: “*why is this product recommended to me*”. Below, we detail the **Task Alignment** for the Recommendation-Explanation LLM $\mathcal{LLM}_{\text{RE}}$ and the **Offline Production** strategy to meet online low-latency requirements.

2.4.1. Task Alignment for Recommendation Explanation Generation

Similar to the item tag prediction tasks, we adapt large language models for recommendation explanation generation through two-stage training. We first pre-train the model using reasoning-enhanced teacher datasets generated by DeepSeek-R1 (**Reasoning-Enhanced Pre-Alignment**), followed by training on self-generated data that are subject to rigorous quality control through human or LLM-Judge filtering (**Self-Training Evolution**), ultimately achieving human-aligned explanation generation performance. In this section, we focus on the **Prompt Engineering** and **Data Quality Control** protocols specifically designed for this task. Additionally, we provide detailed **Human Evaluation Experiments** to validate the effectiveness of our alignment approach.

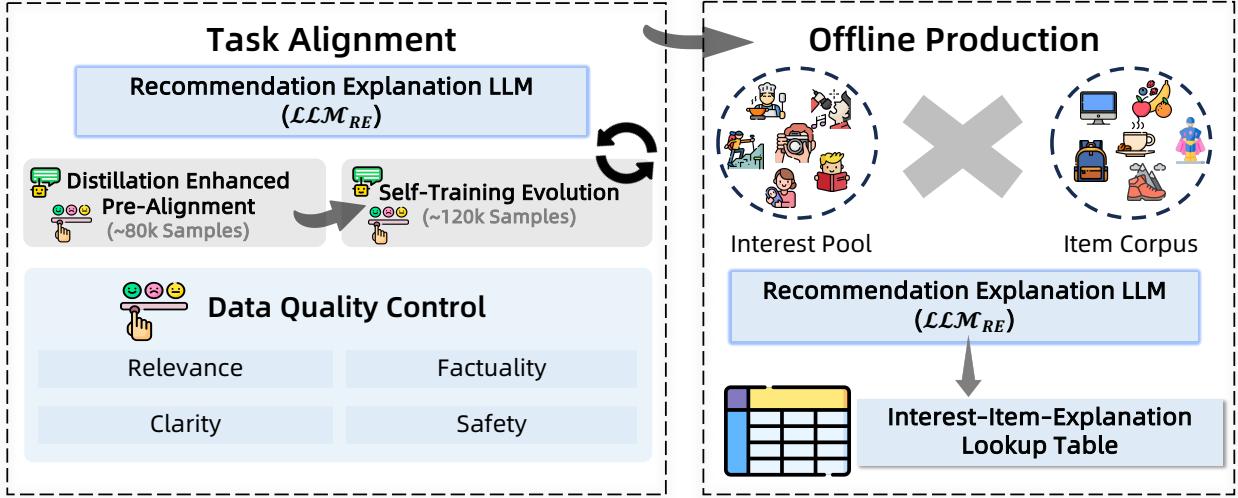


Figure 6 | Illustration of the recommendation explanation generation task. The left figure demonstrates the task alignment process and data quality control protocols, while the right figure shows the offline production of interest-item-explanation tables using the recommendation-explanation LLM.

Prompt Engineering. Given user interest sets and relevant recommended item information (such as item tags, titles), we instruct \mathcal{LLM}_{RE} to execute the following two steps to generate reasonable recommendation explanations:

1. **Context Understanding.** Analyze the given input information to understand user interests and item characteristics.
2. **Explanation Generation.** Based on the above analysis, if reasonable correlations exist between recommended products and user interests, generate conversational phrases that present these connections while maintaining an approachable tone; otherwise, generate recommendation explanations primarily based on the product's inherent qualities.

Through these steps, the model \mathcal{LLM}_{RE} can generate personalized recommendation explanations based on user interests and item information, helping users understand recommendation results and enhancing user experience. The simplified prompt template is shown as Prompt 2.4.1, where the placeholders **{User Interest}**, **{Date Information}**, and **{Item Information}** are instantiated with specific user interests, current date, and recommended item information, respectively. The placeholders **{Context Understanding}**, and **{Explanation Generation}** are populated with the two reasoning steps mentioned above, designed to leverage the CoT-based reasoning capabilities of LLMs to generate reasonable explanations, while **{Recommendation Principles}** and **{Strict Prohibitions}** are filled with pre-configured requirements and constraints. Due to space constraints, we provide details about the full prompt template in the Appendix B.

Recommendation Explanation Generation Prompt Template

Role

Generate personalized recommendation explanations based on user profiles and recommended items. The explanations must satisfy the following requirements.

Input

User Interest: **{User Interest}**

Current Date: **{Date Information}**

Item Information: {Item Information}

Core Reasoning Steps

{Context Understanding} | {Explanation Generation}

Mandatory Requirements

Recommendation Principles (✓)

{Recommendation Principles}

Strict Prohibitions (✗)

{Strict Prohibitions}

Output Format

(Detailed output format requirements)

Based on the aforementioned prompt engineering design, we formalize the explanation generation process as follows: Given user interest \mathcal{I}_u and item information Info_v , we utilize the recommendation explanation LLM to generate personalized explanations for the recommended items:

$$E_u = \mathcal{LLM}_{\text{RE}}(\mathcal{I}_u, \text{Info}_v | \mathcal{P}_{\text{RE}}), \quad (3)$$

where E_u represents the generated explanation for user u , $\mathcal{LLM}_{\text{RE}}(\cdot)$ denotes the recommendation explanation model, \mathcal{I}_u denotes the user's interests, Info_v contains the relevant information about item v (e.g., item tags, titles), and \mathcal{P}_{RE} represents the prompt template.

Table 6 | Criteria for rejecting model-generated explanations. In this example, the user's interest is *outdoor travel*, and the product is a *backpack*.

Label	Evaluation Criteria	Example	Why ✓ or ✗
Accept (Both ✓ → ✓)	Strong Relevance	<i>Roam mountains rivers backpack journey companion.</i>	Aligning interests with use. (Relevance & Factuality ✓)
	Verified Factuality		Brief, poetic, no privacy leaked. (Clarity & Safety ✓)
	Full Clarity		
	Proven Safety		
Reject (Any ✗ → ✗)	Weak Relevance	<i>Office backpack document carry convenience.</i>	Ignores user's outdoor interest.
	Unverified Factuality	<i>Bag cutproof fireproof lasts forever.</i>	Exaggerated false claims.
	Limited Clarity	<i>Bag bag good bag buy bag good.</i>	Repetitive nonsense.
	Unproven Safety	<i>MsZhang time-limited offer bag quick buy.</i>	Privacy leak + hard sell.

Data Quality Control. To ensure the model's instruction-following capability, we also introduce multi-dimensional rejection sampling to achieve high-quality training sample filtering:

- ♦ **Relevance:** Alignment between the explanation and both the characteristics of the recommended item and the user's interests.
- ♦ **Factuality:** Accuracy of the explanation in reflecting the item's actual features and functionality.

- ◆ **Clarity:** Quality of text fluency, grammatical correctness, and stylistic expression.
- ◆ **Safety:** Absence of sensitive or personally identifiable information in the generated content.

Positive and negative examples of the above criteria are shown in Table 6. We consider samples with generated explanations that meet the above criteria as qualified samples, while those that do not meet the criteria are regarded as unqualified samples. During the recommendation explanation LLM alignment process, we employ both human evaluation and LLM auto-evaluation to filter training samples, improving the model’s instruction-following ability and generation quality.

Table 7 | Human-evaluated pass rates for different models on recommendation explanation generation task. The best performance is highlighted in **bold**.

Model	DeepSeek-R1	Qwen3-Base	Qwen3-SFT
Pass Rate (%)	92.7	30.0	95.8

Human Evaluation Experiments. To validate the effectiveness of our task alignment approach for recommendation explanation generation, we conduct human evaluation on model-generated explanations according to the multi-dimensional criteria outlined above. An explanation is considered qualified only when it satisfies all evaluation standards including relevance, factuality, clarity, and safety. We compare three models: DeepSeek-R1, Qwen3-Base, and Qwen3-SFT, where Qwen3-SFT represents our multi-stage aligned LLMs.

As shown in Table 7, our experimental analysis reveals several important findings:

(1) Qwen3-Base demonstrates insufficient performance in generating high-quality recommendation explanations that meet industry standards, with a pass rate of only 30%. This limitation stems from the lack of domain-specific knowledge and task-oriented instruction-following capabilities required for personalized explanation generation in recommendation scenarios.

(2) DeepSeek-R1 achieves superior performance compared to Qwen3-Base with 92.7% pass rate, benefiting from its larger parameter scale and enhanced reasoning capabilities. The model’s deep thinking abilities enable better understanding of user-item relationships and generation of more coherent explanations that align with user interests and item characteristics.

(3) Our aligned Qwen3-SFT model demonstrates substantial improvement in adapting to multi-dimensional explanation generation requirements, achieving the highest pass rate of 95.8%. Through training on carefully curated high-quality recommendation explanation samples via reasoning-enhanced pre-alignment and self-training evolution, the model effectively learns to generate explanations that satisfy relevance, factuality, clarity, and safety criteria simultaneously. This comprehensive alignment makes it well-suited for online deployment requirements where both explanation quality and computational efficiency are essential for large-scale recommender systems.

2.4.2. Offline Production

Due to the excessive computational overhead of generating recommendation explanations for each user–item pair in real-time online scenarios, it becomes challenging to meet the low-latency requirements of industrial recommender systems. To address this issue, we design an interest-based offline explanation production method.

Specifically, we start from the user interest set and leverage the collected tag–interest association pairs (see Section 2.2.1). We utilize a pretrained Tag-to-Cate model $\phi(\cdot)$ to map the predicted item

tag T to specific item categories, which can be formalized as:

$$C \leftarrow \phi(T),$$

where C represents the mapped item category of the item tag. Since each item can be mapped to its corresponding item category, we can establish associations between user interests and individual items through their shared categories. This creates pairing relationships between user interests and specific items within the same category. Importantly, this approach generates only matched *interest-item* pairs rather than exhaustive *user-item* combinations, significantly reducing the target scope.

Based on this framework, we perform offline explanation generation for all matched interest-item pairs, creating a comprehensive **Interest-Item-Explanation Lookup Table**. During online recommendation, we efficiently retrieve the corresponding explanations from this precomputed table by matching the currently recommended items with the user's interest set. This approach enables real-time explanation delivery while dramatically reducing computational overhead compared to generating explanations for all possible user-item combinations.

3. Human-LLM Cooperative Judge

To ensure that large language models meet human subjective expectations in recommendation generation tasks (*i.e.*, user interest mining, item tag prediction, and recommendation explanation generation), we manually curate training samples generated by DeepSeek-R1 or our self-trained models to align with human standards. However, scaling up manual evaluation through crowdsourced annotation is impractical in real-world industrial environments due to prohibitive costs and lengthy development cycles. Inspired by the excellent performance of **LLM-as-a-Judge** approaches across various natural language understanding and generation tasks (Chen et al., 2024; Gu et al., 2024; Tang et al., 2025; Zheng et al., 2023), we adopt this paradigm by leveraging LLMs as intelligent judges to achieve automated evaluation, aiming to reduce evaluation costs and improve efficiency.

However, we empirically find two critical challenges that hinder the effectiveness of LLM-Judges:

- **Cognitive Bias:** Unlike straightforward evaluation tasks like harmfulness assessment, recommendation systems require understanding complex user behaviors, product characteristics, and operational strategies. This demands domain-specific knowledge and contextual awareness beyond basic reasoning capabilities. Native LLMs often exhibit cognitive biases due to knowledge limitations and pre-training biases (Dai et al., 2024; Schroeder and Wood-Doughty, 2024; Son et al., 2024; Ye et al., 2024), compromising their evaluation reliability.
- **Temporal Misalignment:** The dynamic nature of recommendation ecosystems creates a fundamental mismatch between static LLM judges and evolving real-world conditions. This temporal discrepancy manifests through three critical dimensions:
 - *Evolving User Behavior Patterns* – emerging interaction trends and shifting preference distributions that deviate from historical training data.
 - *Dynamic Item Characteristics* – introduction of new product categories, features, and attributes that were not present during judge training.
 - *Updated Evaluation Criteria* – evolving business strategies, market expectations, and quality standards that continuously redefine the evaluation criteria.

The cumulative effect of these temporal dynamics progressively undermines the evaluation capabilities of static LLM judges, introducing systematic biases for different generation tasks.

To address these issues, we propose a Human-LLM Cooperative Judge System. The core idea is to enhance task-specific evaluation capabilities through collaborative cooperation between human

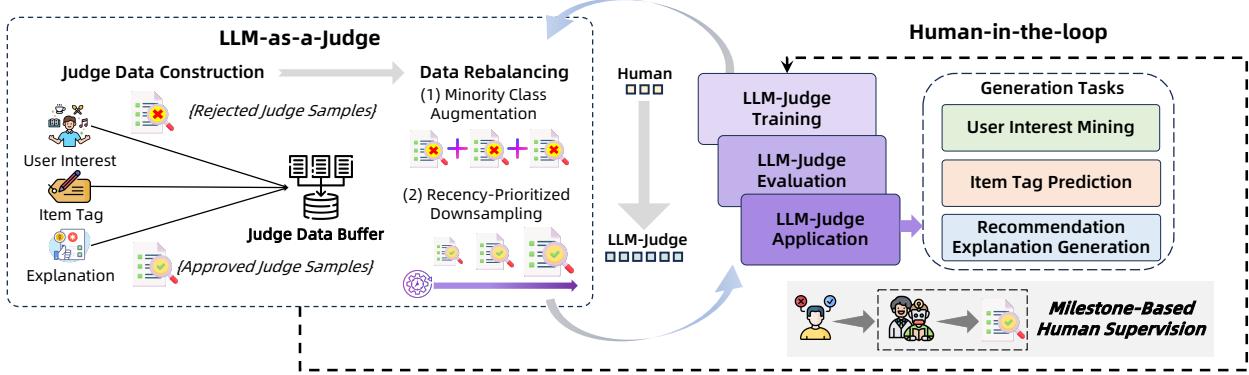


Figure 7 | Human-LLM Cooperative Judge System. Multi-round human judgment data from three generative tasks (user interest mining, item tag prediction, recommendation explanation generation) are collected into a Judge Data Buffer, with data balancing through minority class augmentation and recency-prioritized downsampling. The LLM-Judge is trained and deployed upon reaching accuracy thresholds, complemented by periodic human performance evaluation. This judge system facilitates a gradual transition from human curation to LLM-Human Cooperative curation.

experts and LLM-Judge, while integrating human-in-the-loop supervision that monitors performance milestones and triggers realignment with evolving data distributions and task requirements when needed. In the following sections, we will provide detailed introductions to the two key components: **LLM-as-a-Judge** (Section 3.1) and **Human-in-the-Loop** (Section 3.2).

3.1. LLM-as-a-Judge

To enhance the alignment between LLM-based Judges and human evaluators in recommendation generation tasks, we develop a human-annotated evaluation dataset for LLM instruction fine-tuning.

Dataset Construction. Specifically, we first categorize the evaluation tasks across different generation tasks and assessment criteria into the following two types:

- **Binary Classification Evaluation**, e.g., in item tag prediction task, we employ a binary {Yes, No} evaluation scheme for “Relevance”, determining whether the tags are relevant to user interests.
- **Multi-level Evaluation**, e.g., in recommendation explanation generation task, we adopt a multi-level evaluation scheme {Excellent, Good, Bad} for “Truthfulness”, assessing how well the generated recommendation explanations align with factual information.

Furthermore, we collect judge training data from the following sources:

- **Pre-alignment Data:** Reasoning-enhanced data generated by DeepSeek-R1 during the pre-alignment phase.
- **Self-training Data:** Self-Generated samples produced from the task-specific LLM across multiple iterative rounds during the self-training phase.

We conduct human annotation on data from both sources for quality assessment according to different evaluation criteria specific to their respective tasks. The annotated samples and results are stored in a **Judge Data Buffer**, which is subsequently used to fine-tune corresponding LLM-Judges.

Data Rebalancing Strategy. However, in practice, we observed severe class imbalance in the collected judge training data, leading to significant *Majority Class Bias* in the trained judge models. Under the *Empirical Risk Minimization (ERM)* principle ([Johnson and Khoshgoftaar, 2019](#)), models predominantly learn from majority classes during training, thereby neglecting minority class characteristics and compromising model generalization and evaluation accuracy. To address this challenge, we design a data rebalancing strategy for judge model training, comprising the following steps:

(1) **Minority Class Augmentation:** For underrepresented classes, we cumulatively utilize samples from multiple previous rounds of human annotation to augment data for these categories.

(2) **Recency-Prioritized Downsampling:** For dominant classes, we employ a temporal decay-based downsampling strategy that prioritizes the most recent evaluation samples while gradually incorporating earlier samples, effectively balancing sample quantities across different classes.

We empirically find that our proposed resampling strategy effectively improve the evaluation performance, particularly in evaluation accuracy for minority classes. This approach effectively enhances model generalization capabilities and prevents learning collapse. Note that our current approach primarily focuses on *data-level* balancing strategies. Future work will explore *model-level* balancing techniques, such as cost-sensitive learning ([Elkan, 2001](#); [Fernández et al., 2018](#)).

3.2. Human-in-the-Loop

While LLM-as-a-Judge systems demonstrate significant advantages in cost-effectiveness and evaluation efficiency, their reliability faces critical challenges due to dynamic data distribution shifts. LLM-based Judges become not only increasingly unreliable in quality assessments but also struggle to adapt to evolving evaluation standards when encountering emerging user behavior patterns or novel product characteristics.

To address these limitations, we propose a *Milestone-Based Human Supervision* framework that integrates human-in-the-loop validation. Specifically, during major version updates: *First*, we collect expert annotations on recent generation samples; *Second*, we perform systematic comparisons between LLM Judge evaluations and human assessments. When detecting substantial performance degradation, we conduct continuous training through targeted fine-tuning of the LLM-Judge using newly annotated data. This dual approach ensures sustained alignment with evolving data distributions while maintaining operational efficiency.

By combining *automated LLM-as-a-judge evaluation* with *strategic human-in-the-loop oversight*, we establish a robust human-LLM cooperative judgment system. This hybrid framework enables reliable large-scale data curation and model performance monitoring, achieving an optimal balance between evaluation accuracy and operational efficiency.

4. Evaluation

In this section, we first introduce the experimental setup of RecGPT, including user group selection of online serving, training infrastructure, and implementation details ([Section 4.1](#)). We then present the overall performance of RecGPT in online A/B testing, analyzing its impact on users, merchants, and the platform ([Section 4.2](#)). We examine the consistency between LLM-as-a-Judge and human evaluators in different recommendation generation tasks ([Section 4.3](#)). Additionally, we demonstrate real-world case studies ([Section 4.4](#)) and conduct user surveys ([Section 4.5](#)) with online users to capture user feedback and reflect the changes brought by RecGPT.

4.1. Evaluation Setup

User Group Selection. We conducted a one-month online A/B experiment by deploying RecGPT to the “Guess What You Like” (**Guess**) scenario on Taobao’s homepage. The experiment targeted the top one-third of active users, with both control and experimental groups each allocated 1% of the traffic. Users in the experimental group received recommendations generated from RecGPT system, while those in the control group continued using the existing base recommender system.

Infrastructure. To optimize computational efficiency and resource utilization, we leverage FP8 quantization and KV caching techniques. Our distributed training leverages a Megatron-based framework, enabling efficient processing of ultra-long user behavior sequences and scalable model training. These infrastructure optimizations resulted in a 57% improvement in inference speed.

Implementation Details. We initially used **Qwen3-14B (Qwen3)** (Yang et al., 2025) as the base model for RecGPT training and dataset accumulation. For user interest mining and item tag prediction tasks, we adapt to the lightweight deployment requirements of online services by training an integrated model based on **TBStars-MoE-42B-A3.5B (TBStars)** using high-quality training data from the Qwen3 training process. This approach activates only 3.5B parameters per inference request, enabling efficient online service. For explanation generation tasks, we continue using the Qwen3-14B model for both training and inference to ensure high quality and accuracy of the generated results.

4.2. Online A/B Test

Evaluation Metrics. We evaluate our online performance across the following dimensions:

(1) User Experience:

- **Dwell Time (DT):** The average time users spend on the recommended items.
- **Exposure Item Category Diversity (EICD):** The diversity of item categories exposed to users.
- **Clicked Item Category Diversity (CICD):** The diversity of item categories that users click on.

(2) Platform Benefits:

- **Item Page Views (IPV):** The number of times item pages are viewed from recommendations.
- **Click-Through Rate (CTR):** The ratio of clicks to impressions for recommended items.
- **Daily Click Active Users (DCAU):** The number of unique users who perform at least one click action on recommended items daily.
- **Add-To-Cart (ATC):** The number of items added to the cart from recommendations.

The online A/B test results based on TBStars are presented in Table 8, from which we can observe the following key findings:

◆ **From the user experience perspective,** RecGPT significantly improves user dwell time (DT) and product category diversity (EICD and CICD) by 4.82%, 0.11%, and 6.96%, respectively, by leveraging LLMs’ world knowledge and reasoning capabilities to capture users’ diverse interest preferences beyond traditional interaction-based methods. Our approach harnesses semantic understanding to infer latent user interests and identify subtle connections between user actions and underlying preferences, enabling recommendations across broader categories while maintaining relevance. The substantial improvement in category diversity demonstrates successful mitigation of the filter bubble effect through uncovering latent preferences, while increased dwell time indicates enhanced user engagement through more serendipitous yet relevant recommendations, ultimately improving user satisfaction and platform experience.

Table 8 | The performance improvement of RecGPT compared to the baseline in the online A/B test conducted from June 17 to June 20, 2025.

Scenario	Metrics (Improvement)						
	DT	EICD	CICD	IPV	CTR	DCAU	ATC
Guess	+4.82%	+0.11%	+6.96%	+9.47%	+6.33%	+3.72%	+3.91%

◆ **From the platform perspective**, RecGPT demonstrates substantial improvements across key engagement metrics. The 9.47% increase in IPV reflects enhanced user engagement depth, indicating that users are exploring more products per session due to the system’s ability to surface genuinely interesting and relevant items that capture their diverse preferences. The 6.33% boost in CTR demonstrates improved recommendation precision, as users are more likely to click on items that align with their interests captured through our LLM-powered interest modeling and item tag prediction, reducing wasted impressions and improving content relevance. The 3.72% rise in DCAU signifies improved user retention and platform stickiness, showing that more users are motivated to actively engage with recommendations on a daily basis rather than passively browsing.

◆ **From the merchant perspective**, RecGPT effectively mitigates the Matthew effect by promoting fairer exposure distribution across merchants of varying scales and popularity levels. As illustrated in the top of Figure 1, our approach demonstrates more uniform CTR performance across different item popularity groups compared to the baseline system. While the baseline system exhibits disproportionate exposure allocation toward high-popularity items, leading to concentration bias that limits competitive opportunities for less popular merchants, RecGPT achieves consistently higher and more stable click-through rates across different popularity groups. This indicates that less popular items receive meaningful exposure opportunities without sacrificing overall performance. Furthermore, as shown in the bottom of Figure 1, the Page View Rate (PVR) distribution reveals that RecGPT effectively flattens the long-tail distribution, providing increased visibility for merchants with lower-popularity items. This redistribution creates more equitable market opportunities, enabling smaller merchants to compete more effectively while maintaining the platform’s overall engagement quality, fostering a healthier and more sustainable marketplace ecosystem.

These comprehensive improvements demonstrate that RecGPT successfully creates a **win-win-win outcome for all ecosystem stakeholders**. By mitigating both filter bubbles for users and the Matthew effect for merchants, our approach enhances user satisfaction through diverse discovery experiences while ensuring fairer market opportunities for merchants of all scales, ultimately strengthening platform health through increased engagement and transaction volume. This multi-stakeholder value creation establishes a virtuous feedback loop where improved user experiences drive higher retention and activity, generating richer behavioral data that enables more precise recommendations, which in turn boost merchant performance and platform growth, creating sustainable incentives for continued ecosystem optimization and long-term competitive advantage.

4.3. Human vs. LLM-as-a-Judge

Evaluation Setup. To validate the effectiveness of LLM-as-a-Judge methods for recommendation generation tasks, we conduct comprehensive evaluations across three tasks: *User Interest Mining*, *Item Tag Prediction*, and *Recommendation Explanation Generation*. We employ Qwen3 as our base Judge model, referred to as Qwen3-Judge, and enhance its performance through Supervised Fine-Tuning (SFT) on collected human judgment data, referred to as Qwen3-Judge-SFT. For evaluation standards, each generation output is assessed across multiple criteria using either binary classification

(where only “Yes” responses indicate passing for that criterion) or multi-level assessment (where only “Excellent” and “Good” ratings constitute passing for that criterion). Each recommendation generation result is considered qualified only when it passes all evaluation criteria simultaneously.

Evaluation Metrics. We utilize Accuracy (ACC), Precision, Recall, and F1 Score as our evaluation metrics to quantify the agreement between the *LLM-as-a-Judge* and *Human Annotators*. Higher metric scores indicate stronger alignment between the two.

Table 9 | Performance comparison between LLM-based Judge models and human expert evaluations across three recommendation generation tasks. Qwen3-Judge-Base represents the original LLM judge model, while Qwen3-Judge-SFT denotes the fine-tuned version trained on human judgment data. The best results are highlighted in **bold**.

Task	Judge Model	ACC	Precision	Recall	F1
User Interest Mining	Qwen3-Judge-Base	0.6777	0.6742	0.9777	0.7968
	Qwen3-Judge-SFT	0.7689	0.7996	0.8575	0.8275
Item Tag Prediction	Qwen3-Judge-Base	0.8741	0.9310	0.9196	0.9253
	Qwen3-Judge-SFT	0.9308	0.9714	0.9463	0.9587
Explanation Generation	Qwen3-Judge-Base	0.5677	0.8753	0.5677	0.6657
	Qwen3-Judge-SFT	0.8976	0.9067	0.8976	0.9016

Experimental Results. The experimental results are shown in Table 9, from which we can draw the following conclusions:

◆ The baseline Qwen3-Judge-Base model shows varying performance across different tasks, achieving 87.41% accuracy for item tag prediction, 67.77% for user interest mining, and 56.77% for recommendation explanation generation. This performance hierarchy reflects the inherent evaluation complexity of each task, with item tag prediction involving relatively objective assessment criteria, while explanation generation requires sophisticated evaluation of content quality, relevance, and factuality. These results demonstrate that vanilla LLMs lack sufficient capability to serve as effective judges for domain-specific recommendation tasks.

◆ The task-aligned Qwen3-SFT-Judge model significantly outperforms the baseline across different tasks and most metrics. Notable accuracy improvements include explanation generation (56.77% to 89.76%), user interest mining (67.77% to 76.89%), and item tag prediction (87.41% to 93.08%). Similar enhancement patterns are observed in precision, recall, and F1 scores. These comprehensive improvements demonstrate that supervised fine-tuning with human judgment data effectively bridges the alignment gap between automated evaluation and human assessment standards, enabling reliable automated evaluation across diverse recommendation scenarios.

These results demonstrate that LLMs can effectively serve as automated judges for recommendation generation tasks by leveraging their powerful human-like reasoning capabilities. Through alignment with task-specific human judgment data, our fine-tuned judge models achieve sufficient accuracy to replace costly and time-consuming manual evaluation processes. Traditional human evaluation, while providing high-quality assessments, suffers from scalability limitations, extended evaluation cycles, and high operational costs that are incompatible with the rapid development demands of modern enterprises. In contrast, our automated LLM-based evaluation framework enables efficient quality assessment at scale, significantly accelerating the iteration cycle for industrial development

while maintaining evaluation reliability, thus providing a practical solution for continuous model improvement and deployment in production environments.

4.4. Case Studies



Figure 8 | Case Studies of RecGPT in the Taobao App's *Guess What You Like* scenario.

Figure 8 illustrates the comprehensive workflow of RecGPT through a representative user case, demonstrating its effectiveness in interpreting complex behavioral patterns and generating contextually relevant recommendations tailored to user interests. The example features a 30-year-old female user from Hangzhou whose extensive *three-year behavioral history* includes diverse purchasing, searching, and browsing activities, indicating distinct preferences for *traditional Chinese aesthetics* and **modern parenting needs**.

Through systematic analysis of the user's historical activities, such as searches for qipao dresses, women's ink-painting clothing sets, baby sun hats, children's adjustable desks, and glowing spinning toys, the **User Interest Mining** module identifies two primary areas of interest: "*Fashion styling*" and

“Parenting and baby care”. These referred interests reflect the system’s ability to detect meaningful thematic patterns within seemingly unrelated behavioral data. Subsequently, the **Item Tag Prediction** component translates these broad interest categories into specific product-related tags, such as “Linen-blend Wide-leg Coordinates” and “Baby Bath Temperature Sensor”. These tags effectively capture her preference for stylish yet comfortable fashion and her practical concern for child safety.

The **User-Item-Tag Retrieval** framework utilizes these tags to select relevant products matching her varied interests. The **Personalized Recommendation Explanation** module then generates personalized rationales, clearly linking the recommended items to her behavioral history. For example, explanations such as “Hangzhou summer atmosphere new releases” seamlessly incorporate her geographic context and seasonal fashion preferences, while “Temperature control for mom’s peace of mind” directly addresses her emphasis on infant safety. Further context-specific explanations like “Summer outfits refreshing and stylish” and “This sun-safe piece is just right for baby” resonate with her simultaneous interest in personal style and child protection, completing a **sophisticated closed-loop system** that effectively translates behavioral insights into meaningful recommendations.

This practical case underscores RecGPT’s core strength: employing task-specific large language models aligned with extensive world knowledge and logical reasoning to reveal users’ hidden and diverse interests while maintaining relevance. Unlike traditional collaborative filtering, which relies only on user interactions, RecGPT’s LLM-driven approach semantically interprets behaviors, uncovering implicit connections, such as associating traditional fashion interests with cultural identity and linking parenting concerns with safety awareness. This knowledge-driven approach expands recommendation possibilities beyond past interactions, ensuring recommendations remain diverse yet personally meaningful and contextually precise.

4.5. User Experience Investigation

Objective To systematically validate the effectiveness of the RecGPT in improving recommendation quality, we conduct a comprehensive user study focusing on two critical dimensions:

- **Diversity Assessment:** We evaluate whether RecGPT significantly enhances recommendation diversity by reducing repetition of items with similar brands, categories, or attributes, thereby providing users with richer and more varied choices.
- **User Perception Assessment:** We quantitatively measure improvements in user-perceived recommendation quality through structured feedback collection, with particular emphasis on redundancy perception (e.g., “Do you feel the recommendations are repetitive?”).

Implementation Details

- **Participant Selection:** We randomly select 500 active users to ensure comprehensive coverage across different demographics, including varied age groups, genders, and interest profiles.
- **Experimental Setup:**
 - **Control Group:** Users receive recommendations generated by the baseline algorithm.
 - **Treatment Group:** Users receive recommendations from the RecGPT-enhanced system.
- **Evaluation Methodology:**
 1. We use a three-evaluator consensus mechanism where only unanimous decisions are counted as valid responses, ensuring high reliability and minimizing subjective bias.
 2. The evaluation follows a structured three-step process:

- **Historical Review:** Evaluators examine each user's complete behavioral history (purchases, clicks, interactions, exposures) in chronological order to understand authentic user preferences and browsing patterns.
- **Recommendation Analysis:** Evaluators review the system-generated recommendation lists for both control and treatment groups.
- **Redundancy Assessment:** From the user's perspective, evaluators determine whether obvious redundancy exists in the recommendations and, when present, identify the primary sources of repetition (e.g., dominant product categories, repeated brand patterns, or similar attribute clusters).
- **Data Analysis:** We conduct comparative analysis of redundancy scores and user satisfaction metrics between treatment and control groups to comprehensively assess RecGPT's impact on recommendation diversity and overall business performance.

Questionnaire: Perceived Duplication on the Page

1. While browsing the current page, do you perceive any duplication?
 - A Yes, I feel it is duplicated.
 - B No, I do not feel any duplication.
 - C Skip (layout issue, etc.)
2. (Multiple choice) If you do perceive duplication, where does it mainly come from?
 - A The current main page
 - B The left-hand list
 - C Other, please specify
3. (Multiple choice) If the duplication comes from the **current page**, what are the main sources?
 - A Too many SKUs with similar specs / prices
 - B Too many variations under the same colour
 - C Overall page looks repetitive
4. (Multiple choice) If the duplication comes from the **current page**, which types of products are involved?
 - A Exactly the same style
 - B Similar style
 - C Same series
 - D Others (e.g. same colour tone, design style)
5. (Multiple choice) If the duplication comes from the **left-hand list**, what are the main sources?
 - A Repeated items with similar specs
 - B Repeated items under the main list
 - C Repeated tags within the same series
 - D Overlap with the hero image items
 - E Other, please specify
6. (Multiple choice) If the duplication comes from the **left-hand list**, which types of products are involved?
 - A Exactly the same style

- B Similar style
- C Same series
- D Others (e.g. same colour tone, design style)

7. Please describe any other reasons why you feel the current page is repetitive.

Experimental Results Experimental results demonstrate that RecGPT effectively reduces recommendation redundancy across multiple evaluation metrics. Human evaluators identified fewer repetitive items in the RecGPT system, with the repetition rate decreasing from 37.1% to 36.2% compared to the baseline. This improvement is most pronounced within the top 4 recommendation slots, where similar product clustering decreased substantially from 27.7% to 25.3%, indicating that RecGPT successfully diversifies recommendations in the positions where users focus their attention most.

A notable finding emerged when analyzing advertisement influence on perceived redundancy. The treatment group exhibited more balanced ad distribution patterns, and when ad cards were excluded from the analysis, the reduction in perceived repetition became substantially more pronounced. Specifically, the redundancy improvement nearly doubled from 0.88% (with ads included) to 1.57% (without ads), with this effect being most evident within the top 8 recommendation positions.

These findings indicate that RecGPT demonstrates clear advantages in enhancing recommendation diversity and reducing user-perceived repetition. The benefits become particularly apparent when controlling for advertisement interference, suggesting that the model's core recommendation capabilities effectively address content homogenization challenges in practical deployment scenarios.

5. Conclusion, Limitations, and Future Directions

In this paper, we propose RecGPT, a novel recommender system framework that leverages the world knowledge and logical reasoning capabilities of large language models to achieve intent-centered personalized recommendations. RecGPT conducts generative user profiling analysis on users' lifelong multi-behavior sequences and infers users' potential interest distribution through item tag prediction. Additionally, RecGPT enhances system transparency and user experience by generating personalized recommendation explanations. To align large language models with recommendation domain knowledge, we employ a progressive approach spanning from distillation-based pre-alignment using strong reasoning language models to self-training model evolution. We also transition from expert supervision to an automated Human-LLM cooperative evaluation system, significantly improving both the cost-effectiveness and efficiency of model optimization. Through comprehensive online experiments conducted on Taobao, a real-world e-commerce platform, we validate RecGPT's effectiveness across user experience, commercial conversion, and platform health metrics, demonstrating mutual benefits for users, merchants, and the platform ecosystem.

Although RecGPT has demonstrated promising performance in A/B tests, there are still some limitations and areas for improvement:

◆ **Modeling Ultra-Long User Sequences:** Handling ultra-long user behavior sequences presents significant challenges for our current model. First, *the computational burden is substantial*, as model training and inference become prohibitively expensive when processing extensive user histories, with approximately 2% of sequences still exceeding our 128K token limit. Second, *maintaining accuracy across such lengthy sequences proves difficult*, as the model may inadvertently focus on irrelevant noise within user behaviors rather than meaningful interest patterns, resulting in

biased user understanding. To address these limitations, we plan to explore advanced sequence modeling techniques specifically designed for LLMs, emphasizing improved ***Context Engineering*** that can dynamically optimize long-term and short-term memory management, context selection, and information compression for user behavior sequences.

♦ **Multi-objective Joint Learning with Reinforcement Learning:** Currently, RecGPT relies on supervised learning with periodic model updates, facing two key limitations. First, this static training approach struggles to adapt effectively to continuously evolving user preferences and product characteristics in real-world e-commerce environments. Second, different generation tasks are trained separately without achieving ideal joint optimization, despite their potential for mutual reinforcement as they collectively serve the final recommendation goal. To address these challenges, we plan to develop ***Reinforcement Learning (RL)***-based multi-objective joint optimization that utilizes online user feedback data as unified optimization signals. For implementation, we will leverage ROLL ([Wang et al., 2025](#)), a scalable library designed for large-scale reinforcement learning optimization. This approach will enable joint training across all generation tasks while simultaneously optimizing multiple objectives such as user engagement, conversion rates, and long-term platform health, leading to improved model adaptation through better utilization of real-world user interactions.

♦ **End-to-End LLM-as-a-Judge Judge System:** Existing RecGPT evaluation frameworks primarily focus on individual task quality assessment, necessitating separate training data for different evaluation dimensions. This approach leads to a fragmented evaluation process that lacks comprehensive contextual understanding and fails to holistically evaluate multiple aspects simultaneously. To address these limitations, we plan to develop an end-to-end LLM-as-a-Judge system incorporating ***Reinforcement Learning from Human Feedback (RLHF)*** ([Casper et al., 2023](#); [Kaufmann et al., 2024](#); [Kirk et al., 2023](#); [Lee et al., 2023](#)) to train LLM-Judge with human feedback for integrated multi-task assessments. Additionally, we will explore inference-scaling generative reward models ([Chen et al., 2025](#); [Guo et al., 2025b](#); [Liu et al., 2025](#)), allowing dynamic allocation of computational resource during inference to enhance evaluation effectiveness and achieve more nuanced pipeline assessments.

How to effectively leverage large language models in real-world industrial recommender systems has attracted significant research attention since the emergence of ChatGPT. As one of the early successful attempts to fully deploy large language models in real applications, RecGPT serves billions of users and products, demonstrating the tremendous potential of LLMs-for-RS. As large language models continue to evolve and application scenarios expand, we will continue exploring how to better utilize their powerful reasoning and generation capabilities to enhance the intelligence level of recommender systems, conducting meaningful and practical research and practices.

References

- Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.
- S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.
- B. Chen, X. Gao, C. Hu, P. Yu, H. Zhang, and B.-K. Bao. Reasongrm: Enhancing generative reward models through large reasoning models. *arXiv preprint arXiv:2506.16712*, 2025.
- D. Chen, R. Chen, S. Zhang, Y. Wang, Y. Liu, H. Zhou, Q. Zhang, Y. Wan, P. Zhou, and L. Sun. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. In *Forty-first International Conference on Machine Learning*, 2024.
- S. Dai, C. Xu, S. Xu, L. Pang, Z. Dong, and J. Xu. Bias and unfairness in information retrieval systems: New challenges in the llm era. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6437–6447, 2024.
- J. Deng, S. Wang, K. Cai, L. Ren, Q. Hu, W. Ding, Q. Luo, and G. Zhou. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*, 2025.
- C. Elkan. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pages 973–978. Lawrence Erlbaum Associates Ltd, 2001.
- A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera. Cost-sensitive learning. In *Learning from imbalanced data sets*, pages 63–78. Springer, 2018.
- C. Gao, K. Huang, J. Chen, Y. Zhang, B. Li, P. Jiang, S. Wang, Z. Zhang, and X. He. Alleviating matthew effect of offline reinforcement learning in interactive recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 238–248, 2023.
- J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.
- D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.
- J. Guo, Z. Chi, L. Dong, Q. Dong, X. Wu, S. Huang, and F. Wei. Reward reasoning model. *arXiv preprint arXiv:2505.14674*, 2025b.
- J. M. Johnson and T. M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of big data*, 6(1):1–54, 2019.
- T. Kaufmann, P. Weng, V. Bengs, and E. Hüllermeier. A survey of reinforcement learning from human feedback. 2024.
- R. Kirk, I. Mediratta, C. Nalmpantis, J. Luketina, E. Hambro, E. Grefenstette, and R. Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*, 2023.

- H. Lee, S. Phatale, H. Mansoor, K. R. Lu, T. Mesnard, J. Ferret, C. Bishop, E. Hall, V. Carbune, and A. Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. 2023.
- Y. C. Liu and M. Q. Huang. Examining the matthew effect on youtube recommendation system. In 2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), pages 146–148. IEEE, 2021.
- Z. Liu, P. Wang, R. Xu, S. Ma, C. Ruan, P. Li, Y. Liu, and Y. Wu. Inference-time scaling for generalist reward modeling. arXiv preprint arXiv:2504.02495, 2025.
- L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou. Recommender systems. Physics reports, 519(1):1–49, 2012.
- T. T. Nguyen, P.-M. Hui, F. M. Harper, L. Terveen, and J. A. Konstan. Exploring the filter bubble: the effect of using recommender systems on content diversity. In Proceedings of the 23rd international conference on World wide web, pages 677–686, 2014.
- A. Pentina, V. Sharmanska, and C. H. Lampert. Curriculum learning of multiple tasks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5492–5500, 2015.
- S. Rendle. Factorization machines. In 2010 IEEE International conference on data mining, pages 995–1000. IEEE, 2010.
- P. Resnick and H. R. Varian. Recommender systems. Communications of the ACM, 40(3):56–58, 1997.
- B. Schifferer, C. Deotte, and E. Oldridge. Tutorial: feature engineering for recommender systems. In Proceedings of the 14th ACM Conference on Recommender Systems, pages 754–755, 2020.
- K. Schroeder and Z. Wood-Doughty. Can you trust llm judgments? reliability of llm-as-a-judge. arXiv preprint arXiv:2412.12509, 2024.
- G. Son, H. Ko, H. Lee, Y. Kim, and S. Hong. Llm-as-a-judge & reward model: What they can and cannot do. arXiv preprint arXiv:2409.11239, 2024.
- P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe. Curriculum learning: A survey. International Journal of Computer Vision, 130(6):1526–1565, 2022.
- J. Tang, J. Zhang, Z. Tian, X. Feng, L. Wang, and X. Chen. Hf4rec: Human-like feedback-driven optimization framework for explainable recommendation. arXiv preprint arXiv:2504.14147, 2025.
- Q. Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL <https://qwenlm.github.io/blog/qwq-32b/>.
- W. Wang, F. Feng, L. Nie, and T.-S. Chua. User-controllable recommendation against filter bubbles. In Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval, pages 1251–1261, 2022.
- W. Wang, S. Xiong, G. Chen, W. Gao, S. Guo, Y. He, J. Huang, J. Liu, Z. Li, X. Li, et al. Reinforcement learning optimization for large-scale learning: An efficient and user-friendly scaling library. arXiv preprint arXiv:2506.06122, 2025.
- X. Wang, Y. Chen, and W. Zhu. A survey on curriculum learning. IEEE transactions on pattern analysis and machine intelligence, 44(9):4555–4576, 2021.

- L. Wu, Z. Zheng, Z. Qiu, H. Wang, H. Gu, T. Shen, C. Qin, C. Zhu, H. Zhu, Q. Liu, et al. A survey on large language models for recommendation. *World Wide Web*, 27(5):60, 2024.
- S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37, 2022.
- A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. [arXiv preprint arXiv:2505.09388](#), 2025.
- J. Ye, Y. Wang, Y. Huang, D. Chen, Q. Zhang, N. Moniz, T. Gao, W. Geyer, C. Huang, P.-Y. Chen, et al. Justice or prejudice? quantifying biases in llm-as-a-judge. [arXiv preprint arXiv:2410.02736](#), 2024.
- S. Zhang, L. Yao, A. Sun, and Y. Tay. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38, 2019.
- W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al. A survey of large language models. [arXiv preprint arXiv:2303.18223](#), 1(2), 2023.
- Z. Zhao, W. Fan, J. Li, Y. Liu, X. Mei, Y. Wang, Z. Wen, F. Wang, X. Zhao, J. Tang, et al. Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6889–6907, 2024.
- L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Appendix

A. Contributions

Core Contributors

Chao Yi
Dian Chen
Gaoyang Guo
Jiakai Tang[†]
Jian Wu
Jing Yu
Mao Zhang
Sunhao Dai[†]
Wen Chen
Wenjun Yang
Yuning Jiang
Zhujin Gao

Kan Liu
Lang Tian
Liang Rao
Longbin Li
Lulu Zhao
Na He
Peiyang Wang
Qiqi Huang
Tao Luo
Wenbo Su
Xiaoxiao He
Xin Tong
Xu Chen[†]
Xunke Xi
Yang Li
Yaxuan Wu
Yeqiu Yang
Yi Hu
Yinnan Song
Yuchen Li
Yujie Luo
Yujin Yuan
Yuliang Yan
Zhengyang Wang
Zhibo Xiao
Zhixin Ma
Zile Zhou
Ziqi Zhang

Contributors

Bo Zheng
Chi Li
Dimin Wang
Dixuan Wang
Fan Li
Fan Zhang
Haibin Chen
Haozhuang Liu
Jialin Zhu
Jiamang Wang
Jiawei Wu
Jin Cui
Ju Huang
Kai Zhang

[†] Renmin University of China

The listing of authors is in alphabetical order based on their first names.

B. Prompts

User Interest Mining Prompt Template

Role

You are a shopping guide for an e-commerce platform. Based on users' behavioral history, you need to accurately and comprehensively analyze their potential interests and preferences.

Input

User Profile: {Generated User Attributes}

User Behavioral Information:

Click Behavior Sequence: {Click Behavior Sequence} | Purchase Behavior Sequence: {Purchase Behavior Sequence}

| Search Behavior Sequence: {Search Behavior Sequence} |

Extra Information: {Extra Information}

Mandatory Requirements

Interest Mining Principles(✓)

- 1.Differentiate long-term interests from short-term usage scenarios.
- 2.Mine high-confidence interests based on diverse behaviors.
- 3.Validate interest credibility through temporal distribution patterns and eliminate non-sustained behaviors.

Interest Mining Requirements (✓)

- 1.Cross-reference gender, age, and other attributes to exclude gift-related scenarios inconsistent with user characteristics.

- 2.Categorize results into listed interests and extended interests.

Quantity Requirements (✓)

Reason over 10 interests.

Task Constraints (✗)

- 1.Avoid selecting interest categories unrelated to user behavior.

- 2.Avoid exclude daily consumables.

- 3.When evaluating, do not rely on single actions or short-term concentrated purchases.

Preset Interest List

{Matched Interest Pool}

Output Format

```
[
  {
    "ID": "matched_interest_01",
    "Interest": "xxx",
    "Stage": "yyy",
    "Reason": "zzz",
  },
  ...
]
```

Item Tag Prediction Prompt Template

Role

You are a professional product recommendation specialist for the Taobao app.

Input

User Attributes: {Generated User Attributes}

User Interests: {Generated User Interests}

User Behavior Information

Click Behavior Sequence: {Click Behavior Sequence} | Purchase Behavior Sequence: {Purchase Behavior Sequence} | Search Behavior Sequence: {Search Behavior Sequence} | Extra Information: {Extra Information}

Mandatory Requirements

Task Requirements:

Recommendation Principles (✓)

- 1.Combine user profiles and behavior sequences.
- 2.Focus primarily on long-term and recent behaviors.
- 3.Pay attention to current time and seasonal factors.

Recommendation Requirements (✓)

- 1.Provide specific item descriptions without mentioning brand or model, but avoid overly broad descriptions.
- 2.Associate each item with a user's interest preference.
- 3.Give the recommendation rationale, linking it to the user's relevant attributes and historical behavior.

Quantity Requirements (✓)

Recommend 50 products.

Strict Prohibitions (✗)

- 1.Items that the user has clicked on, purchased, or searched for within the past month should not be recommended.
- 2.Avoid using broad or vague descriptive terms such as “smart”, “modular”, or “set” in item descriptions.

Output Format

```
[
  {
    "Item Tag": "xxx",
    "Interest": "yyy",
    "Reason": "zzz",
  },
  ...
]
```

Recommendation Explanation Generation Prompt Template

Role

Generate personalized recommendation explanations based on user profiles and recommended items. The explanations must satisfy the following requirements.

Input

User Interest: {User Interest}

Current Date: {Date Information}

Item Information: {Item Information}

Core Reasoning Steps

1. **Context Understanding:** Extract core item characteristics from the item title and extra information.

2. **Explanation Generation:** Synthesize creative and impactful recommendation explanations through analytical interpretation of input information.

Mandatory Requirements

Recommendation Principles (✓)

1. **Length:** 6–10 characters (exclusive of punctuation/spaces)

2. **Style:** Naturally fluent phrasing with concrete descriptions

3. **Expression:** Incorporate humor, wit, and digital-native flair; metaphorical or homophonic techniques are permitted

Strict Prohibitions (✗)

1. Fabricating functional benefits, materials, or brand attributes

2. Meaningless generic terms (e.g., "practical," "versatile," "elegant")

3. Slogan-style clichés (e.g., "essential gadget")

4. Exaggerated claims (e.g., "100% effective")

5. Near-duplication of product titles

Output Format

```
{
    "Explanation": "xxx"
}
```

C. Implementation Details of Curriculum Learning-based Multi-task Fine-tuning

Directly applying large language models to user interest mining in e-commerce domains presents significant challenges that limit their effectiveness:

- **Open Interest Space and Uncertainty:** Users' interest landscapes are expansive, and their affinity for previously unseen interests exhibits high uncertainty, making it difficult to effectively introduce new interests to users.
- **Task Complexity Disparity:** Industrial-scale LLMs lack deep understanding of the massive, rapidly evolving item corpora on online platforms. Off-the-shelf LLMs cannot effectively capture domain-specific behavioral patterns.

To bridge this gap between general LLM capabilities and domain-specific requirements, we propose a **Curriculum Learning-based Multi-task Fine-tuning (CL-MFT)** approach. This framework guides the model through progressively challenging tasks, beginning with fundamental e-commerce concepts and advancing to sophisticated interest inference and recommendation generation. By following this structured learning progression, the model develops robust domain knowledge while maintaining

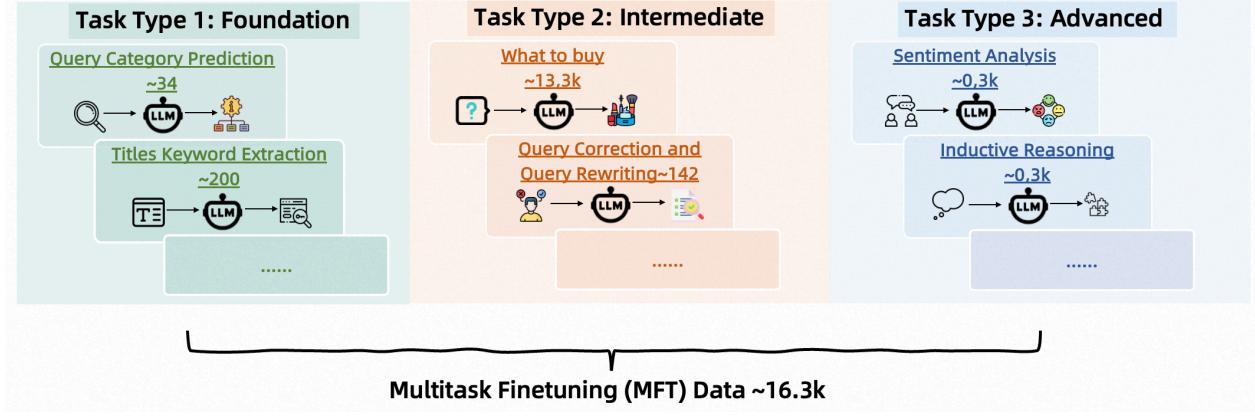


Figure 9 | Curriculum learning-based multi-task fine-tuning framework with foundation, intermediate, and advanced levels, progressively guiding LLMs to develop capabilities for solving complex e-commerce domain tasks.

strong generalization capabilities for complex user behavior modeling and cold-start scenarios.

Our curriculum design (see Figure 9) organizes tasks into three progressive difficulty levels that mirror natural learning progression: **foundation**, **intermediate**, and **advanced**. The complete task list is presented in Table 10.

At the **foundation level**, we establish core competencies through tasks that directly connect user behavior with purchase intent. Specifically, *Query Category Prediction* teaches the model to map search terms to product categories, building essential connections between user actions and interest signals. *Query-Item Relevance Judgment* develops the ability to evaluate matches between search queries and product titles, uncovering users' attribute-specific preferences. *Key Information Extraction from Item Titles* trains the model to identify crucial product characteristics (e.g., retro style, silent operation), forming the foundation for comprehensive interest profiling.

The **intermediate level** advances to tasks requiring sophisticated profile integration. For example, The *E-commerce What-to-Buy* task challenges the model to analyze contextual queries (e.g., 0-1 year old baby clothes) and infer broader interest domains (e.g., maternal and infant products) while considering user demographic information. *Query Correction and Query Rewriting* develop refinement capabilities, transforming imprecise searches into targeted intent (e.g., “cream-style bathroom cabinet” → “cream-colored solid wood bathroom cabinet”), thereby improving interest matching precision.

The **advanced level** introduces complex reasoning through *Causal Reasoning* and *Inductive Reasoning* tasks that decode behavioral patterns (e.g., “waterproof headphones” → “outdoor sports affinity”). *Keyword Extraction* and *Sentiment Analysis* extract nuanced insights from user reviews to validate and refine interest inferences. Finally, *Product Description Generation* task synthesizes all acquired capabilities to produce content that precisely aligns with novel user interests.

This staged training approach introduces low-complexity, diverse tasks first, reducing the model's learning burden and laying the groundwork for subsequent interest mining. It ultimately trains the base LLM to transition from simple matching to comprehensive reasoning for interest discovery.

Table 10 | Curriculum Learning-based Multi-task List. Each task is categorized by type and subtask, with the number of samples used for training.

Task Type	Subtask	Count
Foundation	Query Category Prediction	34
Foundation	Query-Item Relevance	100
Foundation	Key Information Extraction from Item Titles	200
Foundation	Item Key Points Extraction	100
Intermediate	E-commerce <i>What to Buy</i>	13.3k
Intermediate	E-commerce Concept Explanation	200
Intermediate	Unique Selling Proposition	200
Intermediate	Query Correction	44
Intermediate	Query Rewriting	108
Advanced	Recommandation	300
Advanced	Keyword Extraction	300
Advanced	Sentiment Analysis	300
Advanced	Text Classification	300
Advanced	Inductive Reasoning	300
Advanced	Deductive reasoning	212
Advanced	References	300