

Awakening Dormant Users: Modeling Intent Evolution via LLM-based Functional Role Reasoning for Generative Recommendation

Huishi Luo^{1,†,‡}, Shuokai Li^{2,†}, Hanchen Yang², Zhongbo Sun^{2,*},
Haojie Ding², Boheng Zhang², Zijia Cai², Renliang Qian², Fan Yang²,
Tingting Gao², Chenyi Lei², Wenwu Ou², Fuzhen Zhuang^{1,*}

¹ School of Artificial Intelligence, Beihang University, Beijing, China

² Kuaishou Technology, Beijing, China

hsluo2000@buaa.edu.cn, lisk260@gmail.com

Abstract

Awakening dormant users, who remain engaged but exhibit low conversion, is a key driver for incremental GMV growth in e-commerce platforms. Existing methods often underperform since they typically rely on single-step estimation of an item’s intrinsic value (e.g., immediate click probability). This overlooks the *instrumental effect* of items, where specific interactions act as triggers to shape latent intent and drive subsequent decisions along a conversion trajectory. To bridge this gap, we propose **RoleGen**, a framework that synergizes a Conversion Trajectory Reasoner with a Generative Collaborative Executor. Specifically, the LLM-based Reasoner explicitly models the context-dependent *Functional Role* of items to reconstruct intent evolution. It further uses counterfactual role exploration to mitigate interest collapse and generate diverse conversion-oriented candidates. These reasoned candidate items are then injected into the Executor, which is optimized via a feedback-driven co-training mechanism to ensure grounded execution. Extensive offline experiments and online A/B testing on the Kuaishou e-commerce platform show a 5.82% gain in Recall@1 and a 6.71% increase in online order volume, confirming RoleGen’s effectiveness in activating dormant users.

1 Introduction

In large-scale e-commerce platforms, a substantial fraction of active users remains engaged but has not made any purchases over an extended period. We term these *dormant users*. Empirical statistics from the Kuaishou e-commerce platform reveal they account for over 40% of Daily Active Users



Figure 1: Context-dependent instrumental effect in conversion trajectories. The same item (juice) may play different functional roles under different user intents, leading to distinct downstream conversions.

(DAU) and more than 35% of total Page Views (PV), yet contribute less than 20% of total orders. Despite their low current monetization, their high engagement implies considerable latent conversion potential. Accurately modeling the latent purchase intent of dormant users and *awakening* them into buyers, is therefore a key driver of incremental GMV growth.

Awakening dormant users remains relatively under-explored in industrial recommendation. This task is often formulated as a variation of cold-start recommendation. Traditional approaches typically rely on content-based feature enrichment (Huang et al., 2023), knowledge transfer techniques such as meta-learning (Zhu et al., 2021) and domain adaptation (Xu et al., 2025), or even LLMs (Zhang et al., 2025b) to alleviate data sparsity and distribution shift between cold and active users.

However, directly applying these methods to awaken dormant users often yields suboptimal results. Fundamentally, they model the *single-step, point-wise effect* of recommending individual items, focusing solely on predicting immediate responses. Yet, this fails to capture real-world consumption journeys, where an item’s utility extends beyond its immediate conversion probability, i.e., *intrinsic value*. Instead, items often exert an *instrumental effect* on subsequent decisions: even if item *A* is not purchased, it may still act as a trigger, shaping the user’s latent intent and facilitating conversions on other items. Consider the example in Figure 1

[†]Both authors contributed equally to this work.

[‡]Work done during an internship at Kuaishou Technology

*Corresponding authors: zhuangfuzhen@buaa.edu.cn, szb17@tsinghua.org.cn

for a user pursuing a healthy lifestyle. Although the vegetable juice does not convert, it triggers a shift in the user’s underlying intent: freshly blended juice at home is healthier and more cost-effective than packaged drinks, which directly motivates the purchase of the blender. Therefore, **it is essential to model an item’s instrumental effect along the user’s conversion trajectory**, rather than merely maximizing the standalone click-through rate (CTR) of individual items. This perspective is particularly vital for dormant users with weak purchase intentions, for whom conversions often require multiple items to act in concert throughout the intent evolution, rather than any single item acting in isolation.

Identifying these latent instrumental items is challenging. Traditional attribution analyses (Zeng et al., 2025; Bencina et al., 2025) fail due to the severe noise and delayed feedback in dormant users’ prolonged browsing sequences. Given the reasoning capabilities of LLMs, leveraging their inherent world knowledge offers a promising avenue to infer latent user intents from such sparse signals. Existing methods, such as RecGPT (Yi et al., 2025) and Align3GR (Ye et al., 2026), have explored integrating LLMs into recommendation. However, these works primarily utilize LLMs to infer static user profiles (Liu et al., 2025d) or enrich item-level tags (Yi et al., 2025). Fundamentally, they still focus on evaluating the intrinsic value of items in isolation and lack explicit modeling of items’ instrumental effects on downstream conversions, which dynamically depend on the interplay among items, user intents, and consumption context.

To bridge this gap, we propose RoleGen, a dual-module framework comprising an LLM-based **Conversion Trajectory Reasoner** and a **Generative Collaborative Executor**. (1) The Reasoner, fine-tuned from a general LLM, explicitly models the **Functional Role** of items, characterizing how each item dynamically influences the user’s intent within a specific context. It employs **counterfactual inference** to explore potential conversion paths, effectively mitigating the Matthew effect (Yi et al., 2025), and outputs Semantic Identifiers (SIDs) as candidate next items. (2) The Executor is a decoder-only generative model that combines robust sequence modeling with high inference efficiency, enabling the direct incorporation of the Reasoner’s SID-level outputs as guidance. (3) We further introduce a closed-loop collaborative training strategy: Reasoner counterfactually infers Functional Roles → Executor executes recommendation → system collects feedback

→ Reasoner reflects and refines. This dual-module synergy enables the system to progressively capture the underlying logic of user intent evolution and substantially improve dormant user conversions. Our main contributions are summarized as:

- We propose RoleGen, a unified framework that synergizes an LLM-based Reasoner with a Generative Collaborative Executor via a collaborative feedback–reflection closed-loop mechanism.
- We introduce Functional Role Reasoning to capture the context-dependent instrumental effects of items. The Reasoner further performs counterfactual trajectory inference to simulate alternative conversion paths, thereby enhancing the diversity of generated candidate items.
- Extensive experiments on a large-scale e-commerce platform demonstrate significant performance gains (offline Recall@1 +5.82% and online order volume +6.71%). Moreover, RoleGen effectively mitigates the Matthew effect by enhancing long-tail item exposure.

2 Related Work

The full literature review is in Appendix A.

Generative Recommendation Generative recommendation reformulates user-item interactions as a sequence generation task (Wang et al., 2025). Recent advances (Rajput et al., 2023; Deng et al., 2025; Ye et al., 2026) use *Semantic IDs (SIDs)* to encode items into discrete tokens, significantly improving scalability and transferability. **LLM-Enhanced Recommendation** LLMs have been widely explored to alleviate data sparsity in recommendation (Lin et al., 2025; Team et al., 2026), mainly through representation enhancement (Liu et al., 2025a,c) or data augmentation (Sun et al., 2025; Wei et al., 2024). These methods typically use LLMs as semantic feature extractors or isolated data generators. In contrast, RoleGen trains an LLM-based Reasoner to explicitly infer user intent evolution and couples such reasoning with collaborative execution through reasoning injection and behavioral reflection. **Recommendation for Dormant Users** Re-engaging dormant users is challenging due to sparse feedback and temporal preference drift (Zhang et al., 2025a). Existing methods focus on latent preference recovery (Chang et al., 2023), temporal continuity (Lee et al., 2026), behavioral simulation (Huang et al., 2024), or knowledge graph augmentation (Yang et al., 2024). However, they mainly address static profile completion or item-level cold starts, while overlooking the *instrumen-*

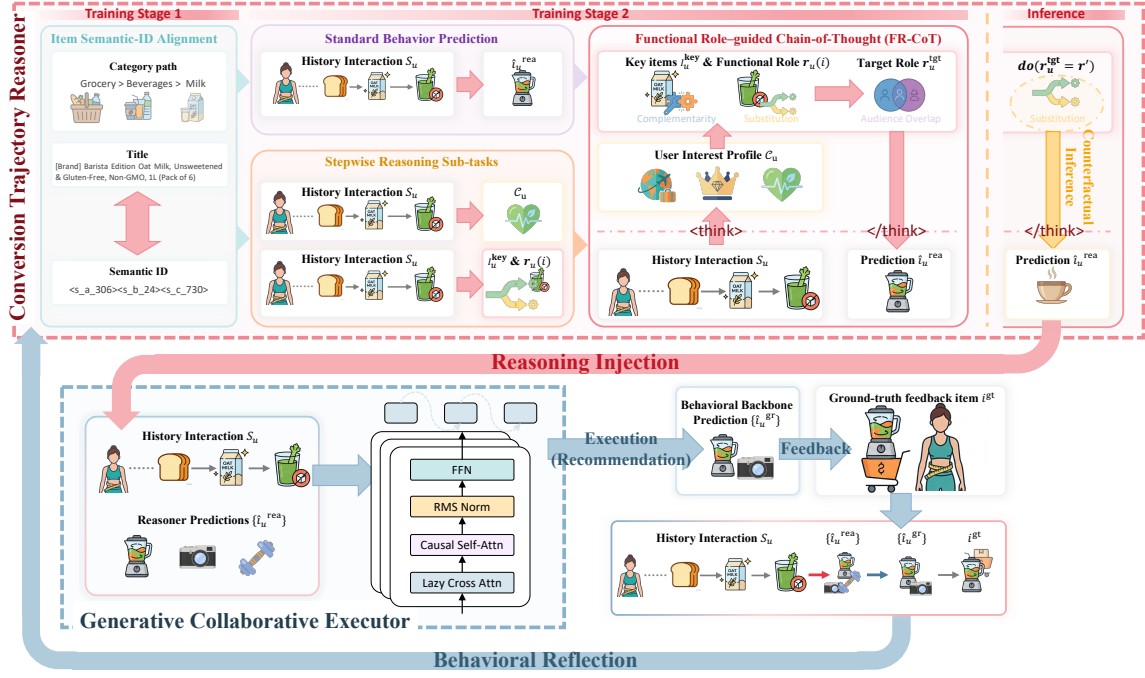


Figure 2: The overall architecture of RoleGen. It consists of two synergistic modules: (1) The **Conversion Trajectory Reasoner** (top), which reasons the latent *Functional Role Trajectory* from sparse user behaviors and employs counterfactual-inspired role sampling to explore diverse conversion paths; and (2) The **Generative Collaborative Executor** (bottom), which grounds these reasoning signals into precise item retrieval through a generative recommender. The two modules are optimized through a collaborative “reasoning–execution–feedback–reflection” closed-loop training strategy.

tal effect of items in driving conversion trajectories. RoleGen addresses this gap through functional-role trajectory reasoning.

3 Method

3.1 Problem Formulation

Dormant Users. Let $\mathcal{U}$ and $\mathcal{I}$ denote the user and item sets. User u ’s chronological interaction sequence is $S_u = [(i_k, b_k, t_k)]_{k=1}^{N_u}$, where $i_k \in \mathcal{I}$, behavior type $b_k \in \{\text{view, click, purchase}\}$, and t_k is the timestamp. A user u is defined as *dormant* ($\mathcal{U}_0 \subset \mathcal{U}$) if S_u contains no purchase actions within the past ΔT (e.g., 30 days) from the reference time.

Functional Role Trajectory. We extract key decision-driving items $I_u^{\text{key}} = [\tilde{i}_1, \dots, \tilde{i}_{M_u}]$ from S_u and abstract them into a functional role trajectory $R_u = [r_u(\tilde{i}_m)]_{m=1}^{M_u}$. Each role $r_u \in \mathcal{R}$ explicitly captures the context-dependent instrumental utility (e.g., complementarity, substitution) of the item toward the final conversion.

Task Formulation. Given a dormant user’s sequence S_u , our goal is to predict a purchase-oriented target item. We achieve this by reasoning R_u as the intent anchor, and generating recommendations conditioned on both S_u and the reasoning signals derived from R_u .

3.2 Overview

Figure 2 shows RoleGen, a reasoning-augmented generative recommendation framework for dormant users. It consists of two coupled components.

(1) Conversion Trajectory Reasoner. This LLM-based module is trained with multi-task SFT (Ye et al., 2026), and designed to distill the decision logic behind dormant-to-active transitions. Given a dormant user’s history S_u , the Reasoner infers a functional-role trajectory R_u that summarizes the user’s latent intent evolution, and generates preliminary candidates $\hat{I}_u^{\text{rea}}$ through structured reasoning and counterfactual role exploration.

(2) Generative Collaborative Executor. This module functions as the recommendation execution model, formulated as a generative collaborative filtering model. It leverages the Reasoner’s candidates $\hat{I}_u^{\text{rea}}$ as explicit guidance to accurately generate the final target item $\hat{I}_u^{\text{exe}}$.

Feedback-Driven Co-Training. To bridge semantic planning and collaborative execution, the framework employs a reflection-based closed loop. The Reasoner provides forward guidance to the Executor, while the Executor’s predictions $\hat{I}_u^{\text{exe}}$ and the subsequent ground-truth user feedback are fed back into the Reasoner to trigger a *reflection* mechanism.

3.3 Functional Role Modeling

Accurately attributing the influence of historical interactions to final purchases is crucial for dormant-user conversion. However, severe noise and delayed feedback in dormant users’ prolonged browsing sequences make traditional attribution methods unreliable (Zeng et al., 2025; Bencina et al., 2025). To address this, we leverage LLMs to semantically model an item’s instrumental contribution, introducing the *Functional Role Trajectory* as a structured abstraction of user intent evolution.

Contextual Intent Role. For each user sequence S_u , we define a user interest profile $\mathcal{C}_u = [c_k]$ based on historical item categories, where c_k denotes the k -th category in user u ’s behavioral interest ranking. To connect $\mathcal{C}_u$ with items, we construct a category-to-category (C2C) decision knowledge graph from large-scale interaction logs, where an advanced LLM (e.g., DeepSeek-R1 (Guo et al., 2025)) pseudo-labels category pairs by jointly considering category semantics and collaborative-filtering signals. We define three relations: *Complementarity*, *Substitution*, and *Audience Overlap*.

Based on user interest $\mathcal{C}_u$ and the C2C graph, we identify an ordered key-item subsequence $I_u^{\text{key}} = [\tilde{i}_1, \dots, \tilde{i}_{M_u}]$ from S_u to serve as reasoning anchors, together with their corresponding contextual intent roles r_{rel} . Specifically, an item i is selected as key only if its category $c(i)$ relates to a dominant historical category in $\mathcal{C}_u$. Its *Contextual Intent Role* is then formally defined as:

$$\begin{aligned} c_m^* &= \arg \max_{c \in \mathcal{C}_u: \text{Rel}_{c \rightarrow c(\tilde{i}_m)}} \text{Score}_u(c), \\ r_{\text{rel}}(\tilde{i}_m; \mathcal{C}_u) &= (c_m^*, \text{Rel}_{c_m^* \rightarrow c(\tilde{i}_m)}), \quad \tilde{i}_m \in I_u^{\text{key}}. \end{aligned} \quad (1)$$

Here $\text{Score}_u(c)$ denotes the behavior-weighted interest score of user u for category c , and $c(i)$ denotes the category of item i .

Intrinsic Attribute Roles. To further characterize these key items, we augment them with static inherent properties across three dimensions: (1) *Market Popularity* r_{pop} : Booming, Evergreen, Long-tail; (2) *Replenishment Nature* r_{repl} : FMCG (Fast-Moving Consumer Goods, e.g., tissue) vs. Durables; and (3) *Decision Cost* r_{cost} : Trial, Core, Premium. These roles provide platform-specific priors on how an item is typically decided upon.

Finally, an item i ’s functional role $\mathbf{r}_u(i) \in \mathcal{R}$ is defined as a multi-dimensional discrete vector:

$$\mathbf{r}_u(i) = (r_{\text{rel}}(i; \mathcal{C}_u), r_{\text{pop}}(i), r_{\text{repl}}(i), r_{\text{cost}}(i)). \quad (2)$$

The functional role trajectory $R_u = [\mathbf{r}_u(\tilde{i}_m)]_{m=1}^{M_u}$ summarizes how the user’s needs and decision context evolve along the conversion path: for example,

items with $r_{\text{repl}}(i) = \text{FMCG}$ and $r_{\text{cost}}(i) = \text{Trial}$ typically correspond to a “Daily Necessity” intent.

Crucially, our work differs from prior CoT approaches in recommendation (Yi et al., 2025; Team et al., 2026; Tang et al., 2026). Existing methods typically summarize *one-sided* information (e.g., disjoint item tags or static user profiles). In contrast, our Functional Role captures *bilateral interaction information*, a cross-feature synthesis of item characteristics and user interests. This bilateral coupling provides a more faithful and transferable signal, enabling the LLM to reason about *why* an item may facilitate conversion, rather than merely *what* they interact with.

3.4 Reasoner Training

3.4.1 Stage 1: Item Semantic-ID Alignment

A prerequisite for LLM-based next-item recommendation is aligning its general semantic space with the platform’s discrete item IDs. We expand the LLM vocabulary to include the hierarchical Semantic ID (SID) codebook deployed in Kuaishou’s production environment. To establish a bidirectional mapping, we construct a multi-task instruction-tuning dataset using item metadata (e.g., title, category). The alignment encompasses **Item Indexing** (metadata $\rightarrow$ SID) and **Item Profiling** (SID $\rightarrow$ metadata). To prevent the catastrophic forgetting of linguistic capabilities, general instruction data (Peng et al., 2023; Luo et al., 2025) is jointly mixed during training.

3.4.2 Stage 2: Functional-Role-guided Structured Reasoning (FR-CoT)

Given user u ’s profile and sequence S_u , FR-CoT generates a CoT reasoning trace wrapped in `<think>` tags followed by the target SID. The CoT adheres to a three-step reasoning logic:

- (1) distilling the user’s current interest profile $\mathcal{C}_u$;
- (2) identifying key items I_u^{key} and their functional roles to reconstruct the role trajectory R_u ; and
- (3) explicitly inferring the functional role of the target item, denoted as $\hat{\mathbf{r}}_u^{\text{tgt}}$.

Crucially, predicting $\hat{\mathbf{r}}_u^{\text{tgt}}$ before the target item ID serves as an intent anchor, which effectively prunes the search space and constrains generation to items consistent with the inferred conversion logic.

To enforce reasoning robustness, we jointly optimize the main FR-CoT objective with a mix of three auxiliary SFT tasks: (1) **Semantic alignment replay** to prevent forgetting Stage-1 semantics. (2) **Stepwise reasoning sub-tasks** decompose the full

CoT into independent mappings (e.g., $S_u \rightarrow I_u^{\text{key}}$, and $(S_u, I_u^{\text{key}}) \rightarrow R_u$), guiding the model to separately internalize intermediate concepts. (3) **Standard behavior prediction** (e.g., $S_u \rightarrow \text{target SID}$) to train the model with direct collaborative filtering capabilities. Prompts are provided in Appendix F.

3.5 Counterfactual Functional Role Inference

For dormant users with sparse and noisy interactions, standard maximum likelihood estimation (MLE) often overfits dominant but possibly stale interests, leading to interest collapse and self-reinforcing the Matthew effect. Token-level decoding strategies (e.g., Beam Search) offer limited remedies, as diversity remains structurally bounded once the high-level intent anchor $\hat{r}_u^{\text{tgt}}$ is determined. To break this loop, we argue that exploration must occur at the *intent level* rather than the *item level*.

We introduce **Counterfactual Functional Role Inference**, which replaces the predicted target role $\hat{r}_u^{\text{tgt}}$ with a sampled counterfactual role r' from a compact role set derived from the user’s trajectory and successful conversion patterns. Here, counterfactual inference denotes a counterfactual-inspired strategy that samples alternative target roles for intent-level exploration, rather than causal-effect estimation. By reusing the FR-CoT prompt under this counterfactual premise, the Reasoner generates candidates along alternative but logic-consistent conversion paths, providing structured diversity beyond token-level beam search (details in Appendix B.2).

3.6 Generative Execution and Co-Training

Directly using LLMs for online serving is impractical due to high inference latency and potential drift from platform-specific collaborative-filtering signals. Therefore, we instantiate a decoder-only **Generative Collaborative Executor** built upon the OneRec-v2 (Zhou et al., 2025) architecture, reformulating recommendation as an efficient, hierarchical token-generation task.

Furthermore, we establish a bidirectional closed-loop strategy to synergize the Reasoner and Executor. Specifically, a *forward reasoning injection* explicitly guides the Executor’s generation using the Reasoner’s counterfactual predictions, while a *backward behavioral reflection* continuously calibrates the Reasoner’s semantic logic using ground-truth feedback from the Executor’s online exposures. This co-training paradigm tightly couples LLM world knowledge with actual collaborative

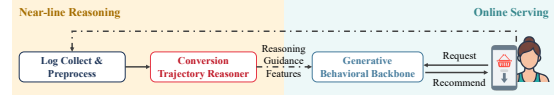


Figure 3: Online Deployment of RoleGen.

evidence, supporting grounded execution for dormant users (Appendix B.4).

3.7 Implementation and System Deployment

RoleGen has been deployed on the Kuaishou e-commerce platform, which serves over 700 million monthly active users (MAU). As illustrated in Figure 3, the deployment architecture follows an asynchronous collaborative design: the Reasoner is scheduled for weekly updates and inference, while the generative executor supports online streaming updates and real-time serving. Implementation details are provided in Appendix C.

4 Experiments

RQ1: How does RoleGen perform compared with strong baselines for dormant users? **RQ2:** How do the key design components of RoleGen, in particular the Reasoner and its Functional-Role-guided reasoning, contribute to the overall performance? **RQ3:** How effectively and robustly does the Reasoner guide the Executor to break the self-reinforcing feedback loop? **RQ4:** How does RoleGen perform in real-world, large-scale online deployment?

4.1 Experimental Setup

Datasets. We evaluate our framework on large-scale, real-world datasets collected from the Kuaishou e-commerce platform. Specifically, we utilize 22M recently converted users to train the Reasoner, and subsequently generate counterfactual candidates for 14M dormant users to fine-tune the Executor. The final evaluation is conducted on a test set comprising over 7M positive interactions. More details are in Appendix D.1.

Metrics. (1) **Hit Item@K (HI@K):** Proportion of test instances where the ground-truth item is among the top- K candidates. (2) **Hit/MRR SID_l @K (HS_l/MS_l @K):** The hit rate and Mean Reciprocal Rank of the ground-truth l -th level SID at cutoff K . (3) **Loss:** The generative training loss L_{Gen} (Eq. (6)). **Models.** We compare the following models in our experiments: (1) **SASRec** (Kang and McAuley, 2018) is a state-of-the-art sequential recommendation model that captures long-term dependencies within interaction sequences. (2) **TIGER** (Rajput

Table 1: Comparison of offline evaluation results on item hitrate on the production e-commerce platform.

Category	Models	HI@1	HI@10	HI@100	HI@500	HI@2000	Loss
Offline SeqRec	SASRec	1.86%	2.93%	5.11%	14.37%	23.18%	-
	TIGER	4.73%	7.88%	14.86%	19.53%	29.49%	-
Production Retrieval	I2I	4.38%	8.12%	17.82%	21.68%	31.76%	-
	U2I	3.19%	4.26%	6.95%	15.74%	27.66%	-
Ours	w/o Rea	5.11%	10.26%	20.54%	31.26%	35.16%	3.2950
	RoleGen	10.93%	14.89%	24.07%	33.98%	37.13%	3.0443

Table 2: Comparison of offline evaluation results on SID Hitrate & MRR across Generative Methods.

Models	HS ₁ @1	HS ₂ @1	HS ₃ @1	MS ₁ @10	MS ₂ @10	MS ₃ @10
TIGER	19.83%	33.47%	50.24%	27.35%	46.59%	60.16%
w/o Reasoner	24.51%	38.09%	56.32%	31.55%	51.43%	64.93%
RoleGen	30.12%	41.77%	56.41%	38.72%	53.35%	65.28%

et al., 2023) is a pioneering generative recommendation framework, which introduces semantic IDs via RQ-VAE and generates target SIDs autoregressively. (3) *U2I* & *I2I* are the major online retrieval methods deployed on the e-commerce platform, serving full-scale online traffic, and accounting for more than 80% of online orders. (4) *w/o Rea* is the Executor trained without Reasoner intervention. (5) *RoleGen* is the complete framework proposed in this paper. Detailed implementation settings for all models are provided in Appendix D.2.

4.2 Offline Evaluation (RQ1)

Tables 1 and 2 report offline performance on dormant users under item-level and SID-level metrics. RoleGen consistently outperforms all baselines, improving HI@1 from 5.11% to 10.93% over the generative Executor without Reasoner and reducing the training loss from 3.2950 to 3.0443, demonstrating the effectiveness of functional-role-guided reasoning and co-training. The strong performance of w/o Reasoner over conventional sequential and production retrieval baselines also verifies the advantage of the hierarchical SID-based generative paradigm for large-scale item generation. Moreover, RoleGen achieves clear gains on SID hitrate and MRR, especially at coarse SID levels, indicating that our bidirectional co-training strategy effectively synchronizes the Reasoner’s high-level semantic intent planning with the Executor’s collaborative filtering.

4.3 Ablation Study (RQ2)

To evaluate the contribution of each component in RoleGen, we conduct ablation studies on both the Executor and the Reasoner. As shown in Section 4.2, adding the Reasoner to the generative Executor consistently improves item-level and SID-level performance, indicating that counterfactual

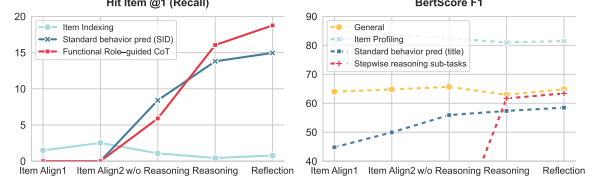


Figure 4: Ablation study of RoleGen.

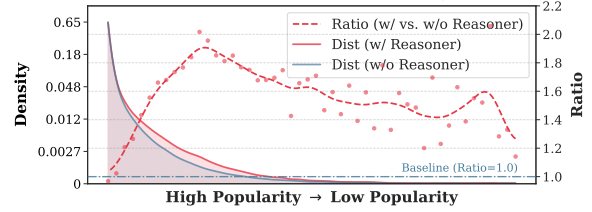


Figure 5: Analysis of Matthew effect mitigation. The exposure ratio rises above 1.0 for long-tail items.

functional-role reasoning provides useful supplementary signals for dormant users with sparse behaviors. For the Reasoner, Fig. 4 compares the progressive training stages, including SID alignment, standard behavior prediction, functional-role-guided reasoning, and behavioral reflection. Item prediction is evaluated by Hit Item, while text generation tasks are evaluated by BertScore (Liu et al., 2025d). The results show that each stage brings incremental gains, and the improvement from **w/o Reasoning** to **Reasoning** verifies that Functional-Role-guided CoT helps the model better capture dormant users’ latent conversion intent.

4.4 Generalization Analysis (RQ3)

We conduct two analyses to demonstrate that world knowledge reasoning yields novel recommendations and superior generalization on long-tail interactions. First, we partition items into popularity buckets and calculate the recommendation ratio (*w/ Reasoner* vs. *w/o Reasoner*). As shown in Fig. 5, the Reasoner significantly boosts long-tail coverage while reducing concentration on the most popular items, effectively mitigating the Matthew Effect. Second, we perform an out-of-distribution (OOD) evaluation, defining an OOD interaction as one where the target item’s top-level category (SID₁) is absent from the user’s

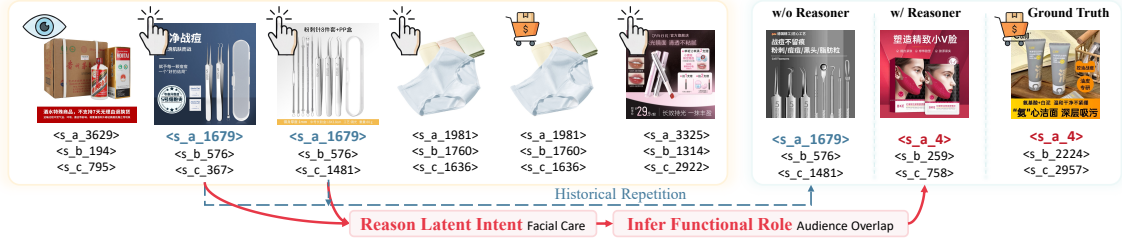


Figure 6: Case study on intent generalization. Unlike the baseline which suffers from interest collapse, our RoleGen leverages “Audience Overlap” functional roles to break the self-reinforcing loop, achieving precise Level-1 SID prediction.

Table 3: OOD test on functional role reasoning. The world knowledge shows more generalization on OOD data, which helps provide novel items.

Models	HS ₁ @10	HS ₁ @20	HS ₂ @10	HS ₂ @20
w/o Reasoner	2.06%	4.33%	10.97%	15.88%
RoleGen	3.12%	5.25%	20.53%	28.48%

historical sequence. Table 3 demonstrates that the Reasoner generates significantly more generalizable recommendations, especially in the OOD scenario.

4.5 Case Study (RQ4)

To provide a more intuitive comparison, we present a representative case study in Figure 6. Historically, the user’s interactions were primarily focused on intimate apparel and cosmetics. However, the ground truth reveals a shift in the user’s latent intent towards facial care products. The “w/o Reasoner” baseline, constrained by historical patterns, myopically recommends repeated items and fails to escape the self-reinforcing feedback loop. In contrast, RoleGen successfully infers the latent intent and recommends a face mask. This demonstrates that RoleGen achieves superior generalization aligned with the user’s evolving intent, effectively breaking the filter bubble.

4.6 Online Evaluation (RQ4)

We further validate RoleGen through an online A/B test with 10% production traffic over two weeks. All users are randomly partitioned into control and treatment groups, with assignments fixed throughout the testing period regardless of subsequent conversion events. Table 4 reports the metric improvements evaluated on a fixed cohort of users identified as dormant before the experiment. RoleGen improves Daily Active Customers (DAC), CTCVR, and order volume by **+5.57%**, **+5.38%**, and **+6.71%**, respectively. As a diagnostic reference, OneRec-Think (Liu et al., 2025d) yields a statistically insignificant **+1.20%** order-volume lift

Table 4: Online A/B testing improvements over the production baseline on dormant users ($p < 0.05$).

DAC	CTCVR	Order Volume
+5.57%	+5.38%	+6.71%

Table 5: Online improvements in diversity and novelty metrics for dormant-user conversions.

Metric	Online Lift
Category Coverage	+2.439% daily, +2.459% weekly
Novelty Conversion	+2.314%

($p = 0.41$). These results demonstrate that RoleGen improves conversion-oriented retrieval for dormant users in the production system.

Furthermore, Table 5 presents the improvements in online diversity and novelty of the resulting purchases. Specifically, Category Coverage is defined as the average number of distinct fine-grained categories purchased per user. Novelty Conversion measures the volume of valid orders originating from out-of-distribution categories (i.e., categories without any click history in the preceding 14 days). These results indicate that RoleGen’s performance gains are accompanied by broader purchase coverage, effectively breaking the filter bubble and facilitating user activation beyond historical browsing boundaries.

5 Conclusion

This paper proposes RoleGen, a novel reasoning enhanced retriever to re-engage dormant users. In contrast to previous methods that focus on isolated point-wise item prediction, RoleGen first extracts functional role trajectories via structured CoT, subsequently integrating the inferred user intent into the Generative Collaborative Executor. Finally, we establish a closed-loop mechanism feedback-reflection to synergize the semantic Reasoner with the Executor. Extensive offline and online experiments verify the effectiveness of RoleGen.

Limitations

Despite the strong experimental evidence for the effectiveness of RoleGen in deployed dormant-user awakening, several limitations warrant discussion.

The first limitation concerns data availability. Our experiments are mainly conducted on platform-specific industrial data rather than public benchmarks. This choice is closely tied to the distinction between *dormant-user* awakening and conventional *cold-start* recommendation. Unlike cold-start users, dormant users still exhibit moderate browsing activity despite the absence of recent purchases. Therefore, the primary challenge is not the absolute absence of data, but the sparsity of conversion-oriented intent. Commonly used pre-filtered public datasets rarely preserve the massive “browsed-but-unconverted” logs needed to identify this setting. Evaluating on such datasets would largely collapse our task into generic cold-start prediction. In addition, RoleGen relies on industrial signals that are typically unavailable in public datasets. SID construction requires multi-modal item information, while the category-relation graph depends on explicit item categories and large-scale behavioral evidence. To improve transparency, we provide detailed implementation guidelines in Appendix C, which may support partial reproduction and evaluation in comparable industrial environments.

Another practical boundary is the scope of LLM-based baselines. We reproduce a limited set of representative methods, such as OneRec-v2 and OneRec-Think. Extending the comparison to more LLM-based recommender systems under the same serving constraints is costly in an industrial setting. Some methods depend on infrastructure that is not directly transferable. For example, RecGPT (Yi et al., 2025) requires a mature tag system and a human-in-the-loop annotation pipeline with 21.1K samples. A more extensive comparison will become feasible once comparable implementations and equivalent deployment budgets are available.

A further limitation concerns the counterfactual inference module in RoleGen, which should not be interpreted as a strict causal inference method. It is better viewed as a counterfactual-inspired sampling strategy for intent-level exploration, where alternative target roles are sampled to guide the generation of more diverse candidate items (Section 3.5). A rigorous causal formulation would require estimating the individual causal effect of each historical

item and validating the estimates with randomized treatment data, which is beyond the scope of this work.

Finally, our Functional Role annotations are pseudo-labels generated by an LLM-assisted, platform-informed pipeline, rather than ground truth for users’ actual latent needs. Accordingly, agreement with these labels measures consistency with the predefined role taxonomy and should not be interpreted as direct validation of users’ true decision rationales or intent evolution. The intrinsic role dimensions are item- or category-level attributes, whereas the Contextual Intent Role is user-dependent and is jointly determined by the item category and the user’s interaction-derived interest profile. It captures a cross-feature synthesis of item characteristics and user interests. We therefore distinguish role-label consistency from independent validation of latent user needs.

Ethical Considerations

RoleGen is designed to improve dormant-user awakening in deployed recommender systems. As with other recommendation algorithms, such deployment may affect user experience and consumption behavior. Therefore, the method should be used to improve recommendation relevance and user experience, rather than to encourage excessive or misleading consumption. At the same time, RoleGen does not introduce new categories of ethical risk beyond those already present in large-scale recommender systems.

For data privacy, all user behavior logs used in this work are processed within the scope permitted by the platform’s user agreement and data-governance requirements, and are used only for recommender-system development aimed at improving user experience. Direct personal identifiers are not used for modeling. The reported statistics are aggregated, and no individual-level behavior trace is disclosed.

GenAI Usage Disclosure

Generative AI tools (e.g., ChatGPT, Gemini) were used only to assist with language polishing and the drafting of repetitive implementation code. All manuscript content and code were reviewed and verified by the authors. The authors take full responsibility for the accuracy and originality of all material in this work.

Acknowledgments

This research work is supported by the National Key Research and Development Program of China under Grant NO. 2024YFF0729003, the National Natural Science Foundation of China under Grant NOs. 62176014, the Fundamental Research Funds for the Central Universities.

References

- John Bencina, Erkut Aykutlug, Yue Chen, Zerui Zhang, Stephanie Sorenson, Shao Tang, and Changshuai Wei. 2025. Lidda: Data driven attribution at linkedin. *arXiv preprint arXiv:2505.09861*.
- Bo Chang, Alexandros Karatzoglou, Yuyan Wang, Can Xu, Ed H Chi, and Minmin Chen. 2023. Latent user intent modeling for sequential recommenders. In *Companion Proceedings of the ACM Web Conference 2023*, pages 427–431.
- Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. 2024. When large language models meet personalization: Perspectives of challenges and opportunities. *World Wide Web*, 27(4):42.
- Jiawei Cheng, Jingyuan Wang, Yichuan Zhang, Jiahao Ji, Yuanshao Zhu, Zhibo Zhang, and Xiangyu Zhao. 2025. Poi-enhancer: an llm-based semantic enhancement framework for poi representation learning. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI’25/IAAI’25/EAAI’25. AAAI Press.
- Ziqiang Cui, Yunpeng Weng, Xing Tang, Fuyuan Lyu, Dugang Liu, Xiuqiang He, and Chen Ma. 2025. Comprehending knowledge graphs with large language models for recommender systems. In *Proceedings of the 48th international ACM SIGIR conference on research and development in information retrieval*, pages 1229–1239.
- Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*.
- Hongfei Ge, Yuanchun Jiang, Jianshan Sun, Kun Yuan, and Yezheng Liu. 2025. Llm-enhanced composed image retrieval: An intent uncertainty-aware linguistic-visual dual channel matching model. *ACM Transactions on Information Systems*, 43(2):1–30.
- Hao Gu, Rui Zhong, Yu Xia, Wei Yang, Chi Lu, Peng Jiang, and Kun Gai. 2025. R4ec: A reasoning, reflection, and refinement framework for recommendation systems. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, RecSys ’25, page 411–421, New York, NY, USA. Association for Computing Machinery.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, et al. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.
- Ruidong Han, Bin Yin, Shangyu Chen, He Jiang, Fei Jiang, Xiang Li, Chi Ma, Mincong Huang, Xiaoguang Li, Chunzhen Jing, et al. 2025. Mtgr: Industrial-scale generative recommendation framework in meituan. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 5731–5738.
- Wenyue Hua, Lei Li, Shuyuan Xu, Li Chen, and Yongfeng Zhang. 2023. Tutorial on large language models for recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1281–1283.
- Feiran Huang, Zefan Wang, Xiao Huang, Yufeng Qian, Zhetao Li, and Hao Chen. 2023. Aligning distillation for cold-start item recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 1147–1157.
- Feiran Huang, Zhenghang Yang, Junyi Jiang, Yuanchen Bei, Yijie Zhang, and Hao Chen. 2024. Large language model interaction simulator for cold-start item recommendation. *arXiv preprint arXiv:2402.09176*.
- Silong Huang and Zichen Liu. 2025. The impact of personalized recommendation systems on consumer purchase decisions under data law frameworks: An empirical study based on e-commerce user behavior data. *Journal of Organizational and End User Computing (JOEUC)*, 37(1):1–28.
- Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.
- Alpamis Kutlimuratov, Akmalbek Bobomirzaevich Abdusalomov, Rashid Oteniyazov, Sanjar Mirzakhilov, and Taeg Keun Whangbo. 2022. Modeling and applying implicit dormant features for recommendation via clustering and deep factorization. *Sensors*, 22(21):8224.
- Gyuseok Lee, Hyunsik Yoo, Junyoung Hwang, SeongKu Kang, and Hwanjo Yu. 2026. Capturing user interests from data streams for continual sequential recommendation. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, WSDM ’26, page 313–323, New York, NY, USA. Association for Computing Machinery.

- Houyi Li, Zhihong Chen, Chenliang Li, Rong Xiao, Hongbo Deng, Peng Zhang, Yongchao Liu, and Haihong Tang. 2021. Path-based deep network for candidate item matching in recommenders. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1493–1502.
- Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, et al. 2025. How can recommender systems benefit from large language models: A survey. *ACM Transactions on Information Systems*, 43(2):1–47.
- Qidong Liu, Xian Wu, Yejing Wang, Zijian Zhang, Feng Tian, Yefeng Zheng, and Xiangyu Zhao. 2025a. Llmser: large language models enhancement for long-tailed sequential recommendation. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.
- Ruochen Liu, Hao Chen, Yuanchen Bei, Zheyu Zhou, Lijia Chen, Qijie Shen, Feiran Huang, Fakhri Karay, and Senzhang Wang. 2025b. Filterllm: Text-to-distribution llm for billion-scale cold-start recommendation. *Preprint*, arXiv:2502.16924.
- Yueyang Liu, Jiangxia Cao, Shen Wang, Shuang Wen, Xiang Chen, Xiangyu Wu, Shuang Yang, Zhaojie Liu, Kun Gai, and Guorui Zhou. 2025c. Llm-alignment live-streaming recommendation. *Preprint*, arXiv:2504.05217.
- Zhanyu Liu, Shiyao Wang, Xingmei Wang, Rongzhou Zhang, Jiaxin Deng, Honghui Bao, Jinghao Zhang, Wuchao Li, Pengfei Zheng, Xiangyu Wu, Yifei Hu, Qigen Hu, Xinchun Luo, Lejian Ren, Zixing Zhang, Qianqian Wang, Kuo Cai, Yunfan Wu, Hongtao Cheng, and 7 others. 2025d. Onerec-think: In-text reasoning for generative recommendation. *Preprint*, arXiv:2510.11639.
- Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*.
- Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.
- Haohao Qu, Wenqi Fan, Zihuai Zhao, and Qing Li. 2025. Tokenrec: Learning to tokenize id for llm-based generative recommendations. *IEEE Transactions on Knowledge & Data Engineering*, 37(10):6216–6231.
- Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Keshavan, Trung Vu, Lukasz Heidt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. Recommender systems with generative retrieval. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.
- Yuqi Sun, Qidong Liu, Haiping Zhu, and Feng Tian. 2025. Llmser: Enhancing sequential recommendation via llm-based data augmentation. *Preprint*, arXiv:2503.12547.
- Jianting Tang, Dongshuai Li, Tao Wen, Fuyu Lv, Dan Ou, and Linli Xu. 2026. Large reasoning embedding models: Towards next-generation dense retrieval paradigm. In *Proceedings of the ACM Web Conference 2026*, WWW '26, page 8115–8126, New York, NY, USA. Association for Computing Machinery.
- OneRec Team, Biao Yang, Boyang Ding, Chenglong Chu, Dunju Zang, Fei Pan, Han Li, Hao Jiang, Honghui Bao, Huanjie Wang, Jian Liang, Jiangxia Cao, et al. 2026. Onereason technical report. *Preprint*, arXiv:2606.06260.
- Hao Wang, Wei Guo, Luankang Zhang, Jin Yao Chin, Yufei Ye, Huifeng Guo, Yong Liu, Defu Lian, Ruiming Tang, and Enhong Chen. 2025. Generative large recommendation models: Emerging trends in llms for recommendation. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 49–52, New York, NY, USA. Association for Computing Machinery.
- Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, page 806–815, New York, NY, USA. Association for Computing Machinery.
- Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2024. A survey on large language models for recommendation. *World Wide Web*, 27(5):60.
- Xiaolong Xu, Hongsheng Dong, Haolong Xiang, Xiyuan Hu, Xiaoyong Li, Xiaoyu Xia, Xuyun Zhang, Lianying Qi, and Wanchun Dou. 2025. C2lrec: causal contrastive learning for user cold-start recommendation with social variable. *ACM Transactions on Information Systems*.
- Shenghao Yang, Weizhi Ma, Peijie Sun, Min Zhang, Qingyao Ai, Yiqun Liu, and Mingchen Cai. 2024. Common sense enhanced knowledge-based recommendation with large language model. In *International Conference on Database Systems for Advanced Applications*, pages 381–390. Springer.
- Xiaoyong Yang, Yadong Zhu, Yi Zhang, Xiaobo Wang, and Quan Yuan. 2020. Large Scale Product Graph Construction for Recommendation in E-commerce. *Preprint*, arXiv:2010.05525.
- Wencai Ye, Mingjie Sun, Shuhang Chen, Wenjin Wu, and Peng Jiang. 2026. Align³ gr: Unified multi-level

- alignment for llm-based generative recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 16154–16162. Association for the Advancement of Artificial Intelligence (AAAI).
- Chao Yi, Dian Chen, Gaoyang Guo, Jiakai Tang, Jian Wu, Jing Yu, Mao Zhang, Sunhao Dai, Wen Chen, Wenjun Yang, Yuning Jiang, Zhujin Gao, Bo Zheng, Chi Li, Dimin Wang, Dixuan Wang, Fan Li, Fan Zhang, Haibin Chen, and 35 others. 2025. [Recgpt technical report](#). *Preprint*, arXiv:2507.22879.
- Hyunsik Yoo, Ting-Wei Li, SeongKu Kang, Zhining Liu, Charlie Xu, Qilin Qi, and Hanghang Tong. 2025. [Continual low-rank adapters for llm-based generative recommender systems](#). *Preprint*, arXiv:2510.25093.
- Xiangyu Zeng, Amit Jaspal, Bin Liu, Goutham Panneeru, Kevin Huang, Nicolas Bievre, Mohit Jaggi, Prathap Maniraju, and Ankur Jain. 2025. Click a, buy b: Rethinking conversion attribution in e-commerce recommendations. In *International Workshop on Computational Advertising*, pages 113–124. Springer.
- Weizhi Zhang, Yuanchen Bei, Liangwei Yang, Henry Peng Zou, Peilin Zhou, Aiwei Liu, Yinghui Li, Hao Chen, Jianling Wang, Yu Wang, Feiran Huang, Sheng Zhou, Jiajun Bu, Allen Lin, James Caverlee, Fakhri Karray, Irwin King, and Philip S. Yu. 2025a. [Cold-start recommendation towards the era of large language models \(llms\): A comprehensive survey and roadmap](#). *Preprint*, arXiv:2501.01945.
- Wenlin Zhang, Chuhan Wu, Xiangyang Li, Yuhao Wang, Kuicai Dong, Yichao Wang, Xinyi Dai, Xiangyu Zhao, Huifeng Guo, and Ruiming Tang. 2025b. Llmtrerec: Unleashing the power of large language models for cold-start recommendations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 886–896.
- Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyang Luo, Zhangchi Feng, and Yongqiang Ma. 2024. [Llamafactory: Unified efficient fine-tuning of 100+ language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.
- Guorui Zhou, Hengrui Hu, Hongtao Cheng, Huanjie Wang, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Lu Ren, Liao Yu, Pengfei Zheng, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Ruiming Tang, Shiyao Wang, Shujie Yang, Tao Wu, and 56 others. 2025. [OneRec-V2 Technical Report](#). *Preprint*, arXiv:2508.20900.
- Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and*

Development in Information Retrieval, pages 1167–1176.

Appendix Contents

A	Extended Related Work	12
A.1	Generative Recommendation . . .	12
A.2	LLM-Enhanced Recommendation	12
A.3	Recommendation for Dormant Users	12
B	Method Details	13
B.1	Functional Role Construction . . .	13
B.2	Counterfactual Functional Role Inference	14
B.3	Generative Collaborative Executor	14
B.4	Co-Training Strategy: Reasoning Injection and Behavioral Reflection	14
B.5	Training Strategy	15
B.5.1	Training of the Reasoner. .	15
B.5.2	Training of the Generative Collaborative Executor. . .	15
C	Implementation and System Deployment	15
D	Experimental Setup	16
D.1	Training Data Preparation	16
D.2	Configurations and Hyperparameters	16
E	Additional Experimental Results	17
E.1	SID-Level Ranking Performance (RQ1)	17
E.2	Contribution of Functional Roles (RQ2)	17
E.2.1	Functional-Role Ablation	18
E.2.2	Consistency of Predicted Target Functional Roles .	18
E.3	Counterfactual Sampling for Diversity (RQ3)	19
F	Prompt Template	19

A Extended Related Work

This appendix expands the related work that is briefly summarized in the main text due to space constraints. It first covers generative and LLM-enhanced recommendation, then narrows to dormant-user reactivation.

A.1 Generative Recommendation

Generative recommendation treats user behaviors and items as discrete tokens and produces predictions in an autoregressive manner (Wang et al., 2025). Recent studies introduce *semantic ID* representations to bridge large item spaces with generative models. Representative works such

as TIGER (Rajput et al., 2023), MTGR (Han et al., 2025), Tokenrec (Qu et al., 2025), and OneRec (Deng et al., 2025) encode items into hierarchical or discrete semantic tokens, transforming recommendation into a structured token generation problem. By significantly reducing vocabulary size and imposing semantic constraints, these methods improve scalability, generalization, and cross-task transferability.

A.2 LLM-Enhanced Recommendation

The emergence of large language models (LLMs) has opened new opportunities to alleviate data sparsity by leveraging their strong semantic understanding and generative capabilities (Hua et al., 2023; Lin et al., 2025; Chen et al., 2024; Wu et al., 2024). Existing LLM-based recommendation approaches can be broadly categorized into two paradigms: (1) LLM-based representation enhancement, which leverages LLMs to produce semantic embeddings for items or users through offline pipelines (Liu et al., 2025a,c; Cheng et al., 2025; Ge et al., 2025). The resulting representations are then fused with collaborative signals in downstream recommendation models. Despite their success, most approaches integrate LLM outputs as side information without tightly coupling LLM reasoning with the recommendation process. (2) LLM-based data augmentation, where LLMs are employed to generate synthetic user-item interactions or pseudo behavioral sequences (Sun et al., 2025; Wei et al., 2024; Liu et al., 2025b; Gu et al., 2025). While effective in enriching training data, these methods typically offer limited controllability over generated behaviors, and are susceptible to hallucinated interactions. Consequently, the ability of LLMs to infer and structure high-level user intent beyond observed interactions remains largely underexplored.

A.3 Recommendation for Dormant Users

Dormant users are previously active individuals with rich historical interactions but prolonged non-conversion, making them a critical yet under-explored group (Kutlimuratov et al., 2022). However, due to temporal preference drift (Zhang et al., 2025a), these historical signals often become stale or misaligned with current user intent and platform dynamics, making effective re-engagement particularly challenging. Existing studies on dormant user recommendation can be broadly categorized into three paradigms. (1) *Latent Preference Recovery* methods aim to infer hidden user intent from histor-

ical behaviors, using techniques such as VAE-based intent modeling (Chang et al., 2023) or deep matrix factorization over dormant user clusters (Kutlimuratov et al., 2022). (2) *Temporal Continuity Modeling* focuses on retaining long-range interests across inactivity gaps. Representative methods, such as CSTRec (Lee et al., 2026) and PESO (Yoo et al., 2025), employ continual learning and a balance between stability and plasticity to prevent catastrophic forgetting. Recent work explores (3) *LLM-based Behavioral Augmentation* for dormant users. By leveraging the world knowledge and reasoning ability of large language models, these approaches enrich sparse behavior sequences with inferred preferences or synthesized semantic context (Huang and Liu, 2025). ColdLLM (Huang et al., 2024) leverages a customized LLM-based simulator to generate user-item interactions between users and cold items, warming cold items and demonstrating LLMs’ potential in sparse settings. Beyond behavior simulation, CSRec (Yang et al., 2024) and CoLaKG (Cui et al., 2025) enhance recommendation by leveraging LLMs to enrich knowledge graphs—via common-sense knowledge fusion and semantic graph embeddings, respectively, hereby strengthening relational signals under data sparsity settings.

Despite their promise, existing LLM-enhanced approaches largely consider standalone items in isolation and overlook how an item contributes to the user’s conversion trajectory. In contrast, our work introduces a reasoning-augmented generative framework that explicitly models user intent evolution through structured functional role trajectories, enabling more principled exploration and re-engagement beyond surface-level behavior recovery.

B Method Details

This section expands the method components that are abbreviated in the main text.

B.1 Functional Role Construction

Intrinsic Attribute Roles. These roles characterize the inherent static properties of an item. We distill them into three dimensions:

- **Market Popularity** $r_{\text{pop}}(i)$: reflects the item-level collective user preference. We discretize popularity into three levels based on category-normalized item sales:
 - *Booming* (high-velocity trends),

- *Evergreen* (consistent sellers), and
- *Long-tail* (niche demands).

- **Replenishment Nature** $r_{\text{repl}}(i)$: captures the category-level recurrence frequency of the underlying user need. Based on the aggregate sales volume of the item’s category, we categorize items into:
 - *FMCG* (Fast-Moving Consumer Goods, e.g., tissue) versus
 - *Durables* (e.g., appliances).
- **Decision Cost** $r_{\text{cost}}(i)$: reflects the cognitive effort and perceived risk associated with the purchase. Considering both the raw price and its category price distribution, we classify items into:
 - *Trial*: low-cost items that facilitate trust-building and impulsive conversion;
 - *Core*: standard consumption choices; and
 - *Premium*: high-price items that typically require repeated comparison and deliberation (e.g., expensive electronics).

Contextual Intent Role. Intrinsic roles alone cannot capture how an item relates to the user’s dynamic intent in the current sequence S_u . Thus, we introduce a contextual relational role $r_{\text{rel}}(i; \mathcal{C}_u)$ defined with respect to the user’s cumulative interest profile $\mathcal{C}_u$. First, we aggregate behavior-weighted scores over historical categories in the interaction sequence S_u , and obtain an ordered interest profile $\mathcal{C}_u = [c_k]_{k=1}^{L_u}$, where c_k denotes the k -th category in user u ’s behavioral interest ranking and $L_u = |\mathcal{C}_u|$. Second, we construct a domain-specific e-commerce decision knowledge graph from large-scale interaction logs on our platform, encoding semantic and behavioral relations between categories. We define three relation types:

- *Complementarity* (scenario completion),
- *Substitution* (functional alternatives), and
- *Audience Overlap* (appealing to similar user personas).

For example, the edge “Phone $\rightarrow$ Phone Stand” is labeled as a *Complementarity* relation. Finally, the item’s **Contextual Intent Role** $r_{\text{rel}}(i; \mathcal{C}_u)$ is determined by the dominant relation between $c(i)$ and the high-scoring categories in the user’s interest profile $\mathcal{C}_u$ via this graph:

$$r_{\text{rel}}(i; \mathcal{C}_u) = (c^*, \text{Rel}_{c^* \rightarrow c(i)}),$$

$$c^* = \arg \max_{c \in \mathcal{C}_u} (\text{Score}_u(c) \cdot \mathbb{I}[\exists \text{Rel}_{c \rightarrow c(i)}]), \quad (3)$$

where $c(i)$ denotes the category of item i , $\text{Rel}_{c^* \rightarrow c(i)}$ represents the relation type on the edge, and $\text{Score}_u(c)$ denotes the behavior-weighted interest score of user u for category c .

B.2 Counterfactual Functional Role Inference

Given the functional-role representation above, **Counterfactual Functional Role Inference** diversifies recommendations by intervening on the intent anchor in the compact functional role space rather than on the full token sequence. Instead of strictly conditioning on the FR-CoT prediction $\hat{r}_u^{\text{tgt}}$, we apply a counterfactual intervention $do(r_u^{\text{tgt}} = r')$. The counterfactual role r' is sampled from a candidate set of target roles $\tilde{\mathcal{R}}_u^{\text{tgt}}$ constructed by combining: (i) roles already observed along the role trajectory R_u ; and (ii) globally frequent conversion roles mined from historically successful transitions. We then reuse the FR-CoT prompt while substituting the target role with the counterfactual one r' , and let the Reasoner predict the corresponding next item $\hat{i}_u^{\text{rea}}$:

$$\underbrace{(u, S_u)}_{\text{Context}}, \underbrace{\langle \text{think} \rangle C_u}_{\text{Intent}}, \underbrace{R_u}_{\text{Trajectory}}, \underbrace{r' \sim P(\tilde{\mathcal{R}}_u^{\text{tgt}})}_{\text{CF Intervention}}, \langle / \text{think} \rangle \xrightarrow{\text{Reasoner}} \hat{i}_u^{\text{rea}}. \quad (4)$$

Conceptually, this process instructs the LLM: “If the user were to act under target role r' , which item would be most likely to lead to conversion?” Subsequently, we employ beam search decoding to generate a diverse set of candidate items $\{\hat{i}_u^{\text{rea}}\}$ based on this counterfactual premise.

This counterfactual mechanism provides two key benefits:

- **Logic-constrained exploration.** Intervening on the target role yields *structured diversity*. The model explores alternative conversion paths (e.g., *Trial* $\rightarrow$ *Premium* upgrade vs. *Trial* $\rightarrow$ *FMCG* replenishment), while remaining consistent with the user’s inferred intent and the upstream FR-CoT reasoning.
- **Efficient multi-candidate generation.** Since the intervention operates in a low-cardinality role space rather than recomputing entire reasoning traces, multiple counterfactual candidates can be generated in parallel with minimal additional cost, effectively breaking repetitive self-reinforcing loops for dormant users and offering a cost-effective exploration–exploitation trade-off.

B.3 Generative Collaborative Executor

Although the Reasoner captures user intent, relying solely on it for online serving is suboptimal. LLM predictions often drift from implicit collaborative-filtering (CF) patterns, and their prohibitive inference latency hinders real-time response, while caching strategies inevitably introduce data staleness. To address these limitations, we employ a Generative Collaborative Executor as the execution engine, utilizing the Reasoner as a knowledge-guided planner.

Specifically, we instantiate the Executor using a state-of-the-art decoder-only generative architecture (Zhou et al., 2025). By representing items as hierarchical SIDs, this approach transforms recommendation into a structured token generation task. This formulation significantly lowers the learning difficulty compared to traditional flat softmax over the full item space and facilitates the seamless injection of the Reasoner’s guidance.

In the Executor, we concatenate multiple user-side features (e.g., static profile, interaction history) and linearly map them into a unified *Context* representation T . To ensure efficiency, T is normalized via RMSNorm to yield static key-value pairs (k, v) , which serve as a persistent memory cache accessible to the decoder throughout the entire generation process. The decoder consists of L_{decoder} stacked transformer blocks that generate the target item’s SIDs autoregressively. Crucially, to optimize inference, each block employs a lazy cross-attention mechanism (Zhou et al., 2025) that attends to the shared static cache (k, v) without redundant projections, followed by standard causal self-attention and position-wise FFNs.

B.4 Co-Training Strategy: Reasoning Injection and Behavioral Reflection

The co-training strategy connects semantic reasoning with collaborative execution through a bidirectional loop between the Reasoner and the Executor. In this loop, the Reasoner provides high-level guidance for Executor generation, while interaction feedback from the Executor refines the Reasoner’s logic.

Forward Guidance via Reasoning Injection.

A primary challenge in the execution module is the scarcity of reliable signals for dormant users. To alleviate this, we inject the Reasoner’s counterfactual reasoning into the Executor. For user u , we encode the Reasoner’s predictions $\{\hat{i}_u^{\text{rea}}\}$ into struc-

tured **Reasoning Guidance Features**, including (i) the raw candidate item list; and (ii) hierarchical semantic statistics, such as the positional patterns of coarse-grained SIDs (e.g., the first two layers). These features are concatenated into the Executor’s input context, guiding the generation of the final recommendation set $\mathcal{I}_u^{\text{gr}}$ with explicit semantic navigational cues.

Backward Alignment via Behavioral Reflection.

To further harmonize the collaboration between the semantic Reasoner and the Executor, we introduce a *Reflection* mechanism that utilizes the Executor as a collaborative mirror to rectify the Reasoner. We construct a post-hoc instruction-tuning task using collected online logs. Specifically, given the original user profile and behavior sequence S_u , the two modules sequentially produce their recommendations $\{\hat{i}_u^{\text{rea}}\}$ and $\{\hat{i}_u^{\text{gr}}\}$. From the online exposure of $\{\hat{i}_u^{\text{gr}}\}$, we can observe the ground-truth feedback item i^{gt} . We then fine-tune the Reasoner to recover i^{gt} from this enriched context via the following instruction tuning task:

$$(S_u, \{\hat{i}_u^{\text{rea}}\}, \{\hat{i}_u^{\text{gr}}\}) \xrightarrow{\text{Reasoner Reflection}} i^{\text{gt}}. \quad (5)$$

Through this reflection process, the Reasoner iteratively calibrates its semantic logic against actual collaborative evidence, fostering a virtuous cycle that synchronizes reasoning with recommendation execution.

B.5 Training Strategy

B.5.1 Training of the Reasoner.

We first expand the vocabulary of the base LLM to incorporate the hierarchical SIDs, adding tokens corresponding to the three semantic levels. To ensure robust adaptation, we adopt a progressive optimization strategy. In the item alignment phase (Section 3.4.1), we exclusively unfreeze the input embedding layer and the output language modeling head (lm_head). This aligns the newly added SIDs with the pre-trained semantic space while preserving the model’s general linguistic knowledge. Subsequently, for the remaining instruction tuning phases, we perform full-parameter fine-tuning. The model is optimized using the standard SFT objective, minimizing the auto-regressive negative log-likelihood loss on the target response tokens.

Specifically, during the item alignment phase of the Reasoner, we employ a **curriculum strategy**. Instead of one-shot training, we align items in a progressive manner:

- **Global Alignment:** We first train on the full item set to establish a broad cognitive foundation, ensuring the model covers the entire distribution of item semantics;
- **High-Quality Refinement:** We then fine-tune exclusively on a subset of high-quality items (filtered by interaction feedback).

Crucially, this strategy prevents noisy, low-quality items from being treated indiscriminately with high-quality ones, thereby minimizing noise propagation and enhancing both the accuracy and robustness of semantic alignment.

B.5.2 Training of the Generative Collaborative Executor.

To capture comprehensive behavioral patterns on our platform and avoid performance degradation caused by data sparsity, we first train the Generative Collaborative Executor on logs from *all* users, including both dormant and active users. Following (Zhou et al., 2025), we adopt an autoregressive cross-entropy objective over the three semantic-ID tokens of each target item:

$$\mathcal{L}_{\text{Gen}} = -\frac{1}{L} \sum_{l=1}^L \log p(\text{SID}_l(i^{\text{gen}}) \mid \text{BOS}, \text{SID}_{<l}(i^{\text{gen}}), \text{Context}), \quad (6)$$

where $L = 3$ denotes the level of the hierarchical SID, $\text{SID}_l(i^{\text{gen}})$ represents the l -th token of the target item’s SID, and BOS is the begin-of-sentence token. The Context comprises both static user attributes and the interaction history (Zhou et al., 2025).

Upon completion of this general training, we further specialize the model for dormant-user recommendation via Reasoner-guided warm-start fine-tuning. We filter for dormant instances and augment their input context with Reasoning Guidance Features (Appendix B.4). The Executor is then further fine-tuned on this augmented dataset using the same objective in Eq. (6).

C Implementation and System Deployment

RoleGen has been deployed on a large-scale e-commerce platform that serves hundreds of millions of monthly active users (MAU). As illustrated in Figure 3, the deployment architecture follows an asynchronous collaborative design: the Reasoner is scheduled for weekly updates and inference, while the Executor supports online streaming updates and real-time serving.

- **Near-line Reasoning Lane.** The Reasoner operates in a near-line environment to handle high-latency inference tasks.

1. **Initialization & Updating:** For the initial deployment, we collect logs from the past three months of successful conversions of dormant users and perform SFT following Section B.5.1. After deployment, we update the Reasoner on a weekly basis by aggregating newly collected logs of successful dormant-user conversions, predictions from the Executor, and subsequent user feedback. These signals are mixed with historical data for continual fine-tuning.
2. **Inference:** On a weekly basis, we employ vLLM to perform counterfactual inference (beam size=25) for all dormant users. The derived semantic signals are structured into knowledge-guided features and cached by user ID for fast lookup at online serving.

- **Online Serving Lane.** The Executor serves real-time traffic as one of the retrieval channels. Upon an incoming request from user u , the Executor retrieves the pre-computed Reasoner features for u from the near-line cache, integrates them with the real-time context and the latest interaction sequence, and performs next-item generation with a beam size of 1024. The generated candidates are then passed to downstream coarse-ranking and fine-ranking stages for final exposure. Thanks to this decoupled design, the online serving path benefits from deep functional-role CoT reasoning signals while maintaining strictly low latency ($P99 < 53\text{ms}$), ensuring a seamless user experience.

D Experimental Setup

D.1 Training Data Preparation

We construct the SFT dataset for the Reasoner by sampling users who transitioned from a dormant state to successful conversion within the past three months. This large-scale dataset comprises over 22M users and approximately 28M items, incorporating three types of interactions: {view, click, purchase}. To control label imbalance and focus on high-confidence signals, we down-sample view events while retaining all click and purchase interactions. Inspired by curriculum learning, we adopt a two-stage strategy in the item alignment phase (Section 3.4.1). Initially, we conduct one

epoch of alignment on the entire item set to establish broad semantic coverage. Subsequently, we perform an additional epoch restricted to high-quality items whose orders exceed 5, resulting in about 9M items. Considering the computational overhead of the Reasoner, we truncate each interaction sequence to keep only the most recent 128 high-quality items.

After the Reasoner is trained, we execute counterfactual inference with beam size 25 for about 14M dormant users, obtaining for each user a candidate set of items $\{\hat{i}_u^{\text{rea}}\}$ and the corresponding *Reasoning Guidance Features*. The enriched features from these 14M users are then utilized to further fine-tune the pre-trained Generative Collaborative Executor.

D.2 Configurations and Hyperparameters

This subsection details the configuration and training protocols for RoleGen and the baseline models. We first present the baseline settings:

- **SASRec (Kang and McAuley, 2018):** A state-of-the-art sequential recommendation model that employs self-attention mechanisms to capture long-term dependencies within user interaction sequences.
- **TIGER (Rajput et al., 2023):** A pioneering generative recommendation framework. It introduces semantic IDs via RQ-VAE and formulates recommendation as a sequence-to-sequence generation task, predicting target SIDs autoregressively.
- **U2I & I2I:** The production-tier retrieval methods currently deployed on the production e-commerce platform, including both U2I-type and I2I-type methods. U2Is (two-tower based model) and I2Is (like Swing (Yang et al., 2020) and PDN (Li et al., 2021)) are the major online methods, serving full-scale online traffic, and providing over 80% online exposures, clicks, and orders.
- **w/o Reasoner:** The base version of our proposed Executor, trained on the entire platform user population without the intervention of the Reasoner.
- **RoleGen:** The complete framework proposed in this paper. It fine-tunes the Executor by incorporating reasoning guidance features and employing the synergistic training strategy (Appendix B.4).

Table 6: SID-level NDCG results. The **w/o Reasoner** variant is the Executor trained without Reasoner intervention and follows the same OneRec-v2-style generative backbone.

Model	SID Level	NDCG@1	NDCG@10	NDCG@50	NDCG@100
w/o Reasoner	SID ₁	0.2451	0.3574	0.3918	0.4053
RoleGen	SID ₁	0.3012	0.4387	0.4746	0.4858
w/o Reasoner	SID ₂	0.3809	0.5571	0.5922	0.6018
RoleGen	SID ₂	0.4177	0.5943	0.6254	0.6264
w/o Reasoner	SID ₃	0.5632	0.6805	0.6971	0.7036
RoleGen	SID ₃	0.5641	0.6827	0.6988	0.7046

Both the Reasoner and the Executor utilize a shared vocabulary derived from an in-house Semantic ID (SID) codebook deployed in the production environment. This codebook employs a hierarchical structure with 3 layers, containing 4,096 tokens per layer.

Conversion Trajectory Reasoner. The Reasoner is initialized from Qwen-8B and fine-tuned on a cluster of 32 GPUs, each providing 80 GB of GPU memory. The training framework is developed upon LLaMA Factory (Zheng et al., 2024), using a learning rate of 1.0×10^{-4} . In the *Item Semantic-ID Alignment* phase, we expand the original LLM vocabulary by adding 12,290 tokens, covering the three-level SIDs (3×4096) and special boundary tokens (<sid_begin>, <sid_end>). Given the relatively short prompt length in this stage ($P99 < 130$ tokens), we set the global batch size to 256 and exclusively optimize the embed_tokens and lm_head layers. In the subsequent *Instruction Tuning with Structured Reasoning* phase, we perform full-parameter fine-tuning. Due to the inclusion of CoT content, the prompt length increases significantly ($P99 < 2,900$ tokens); consequently, we adjust the global batch size to 32. During counterfactual inference, we employ beam search with a beam width of 25 and a temperature of 0.95 to generate diverse counterfactual candidates.

Generative Collaborative Executor. The Executor is instantiated as a decoder-only architecture with approximately 0.1 billion parameters. It consists of 6 decoder layers and 4 attention heads. We train the Executor with a global batch size of 128 and a peak learning rate of 1.0×10^{-4} , following a cosine decay schedule. The maximum context length is truncated to 128 interaction records.

Baseline Configurations. For SASRec, we set the embedding dimension to 128, the number of self-attention blocks to 2, and the number of at-

tention heads to 2. We use a dropout rate of 0.3 and a learning rate of 1.0×10^{-3} . For **TIGER**, to ensure a fair comparison, we utilize the same in-house SID system described above, thereby bypassing the original RQ-VAE tokenization. The generative retrieval component is instantiated as a Transformer-based encoder-decoder architecture with 8 layers and 6 heads trained with a learning rate of 1.0×10^{-4} and a dropout rate of 0.1.

E Additional Experimental Results

This section reports the ranking performance of RoleGen. Regarding its internal modules, we analyze the efficacy of Functional Roles during the reasoning process and explore the impact of counterfactual inference on recommendation diversity.

E.1 SID-Level Ranking Performance (RQ1)

We report NDCG@K under the SID-level setup of Table 2 as a position-sensitive complement to Hit Rate and MRR. It compares two variants: **w/o Reasoner**, the base generative backbone (Executor) without the Reasoner, and **RoleGen**, the complete proposed framework. As the Executor is instantiated with the OneRec-v2 architecture, the **w/o Reasoner** variant also corresponds to the offline performance of the generative backbone without trajectory reasoning. For each semantic granularity, NDCG@K is computed for SID_l. This metric measures how well the model ranks the ground-truth SID_l within the top-*K* candidates, given the ground-truth prefix SID_{<l}. As shown in Table 6, RoleGen yields substantial improvements, particularly at SID₁ and SID₂. These NDCG gains align with the Hit Rate and MRR improvements reported in the main experiments.

E.2 Contribution of Functional Roles (RQ2)

This subsection evaluates Functional Roles at two points in the Reasoner: their effect on key-item lo-

Table 7: Functional-Role ablation study of the LLM-based Reasoner. HS₁₂ denotes joint localization over the first two SID levels.

Model	Key-item Localization			Behavior Prediction			
	HI	HS ₁	HS ₁₂	HI@1	HI@5	HI@20	HI@25
w/o FR	56.97%	63.55%	58.42%	9.79%	21.33%	32.11%	33.33%
w/ FR	65.53%	72.48%	67.15%	13.80%	26.39%	36.99%	38.26%

Table 8: Predicted target functional roles compared with constructed target-role annotations.

Evaluation Dimension	Metric	Score
Attribute-Level Accuracy	Popularity accuracy	50.4%
	Decision Cost accuracy	48.7%
	Replenishment accuracy	67.9%
Strict Exact Match	All role dimensions predicted correctly simultaneously	20.6%
Text-Level Consistency	ROUGE-L between predicted and reference roles	69.93
	BLEU-4 between predicted and reference roles	48.75

Table 9: Offline trade-off of counterfactual sampling in the Reasoner. The last row applies CF-S only to the tail 5 candidates among 25 beam outputs. Hit SID_{1,2} denotes joint matching over the first two SID levels.

Beam Setting	Hit SID ₁	Hit SID _{1,2}	Hit Item	Avg. Unique SID ₁
1	24.5%	15.4%	12.4%	1.00
5	45.1%	32.2%	27.6%	3.19
10	53.9%	40.1%	34.9%	5.34
20	61.2%	46.8%	41.0%	8.73
25 w/o CF-S	63.7%	49.0%	44.3%	9.82
25 with last-5 CF-S	63.1%	48.8%	43.7%	11.63

calization and target prediction, and the consistency of predicted target-role anchors.

E.2.2 Consistency of Predicted Target Functional Roles

E.2.1 Functional-Role Ablation

We evaluate the contribution of Functional Roles by comparing two Reasoner variants: **w/ FR**, which follows the three-step FR-CoT reasoning logic described in Section 3.4, and **w/o FR**, where both R_u and $\hat{r}_u^{\text{tgt}}$ are removed from the full CoT process. The LLM-based Reasoner is evaluated on two tasks: key-item localization ($S_u \rightarrow I_u^{\text{key}}$) and standard behavior prediction ($S_u \rightarrow \text{target SID}$). As shown in Table 7, **w/ FR** yields consistent improvements. Conceptually, the detailed FR attributes provide structured intermediate CoT constraints, reducing ambiguity in intent decoding and improving the transfer from sparse trajectories to candidate generation. Empirically, FR improves the recall of key-item identification by 8.56 absolute points in HI, and this stronger intent localization propagates to standard behavior prediction.

We evaluate predicted target functional roles ($\hat{r}_u^{\text{tgt}}$ in Section 3.4) on approximately 10k samples to measure how well the intermediate role anchors match constructed annotations. The reference target-role annotations are constructed by the same functional-role pipeline used for the stepwise FR-CoT supervision. Table 8 reports both exact attribute accuracy and text-level consistency against these annotations. While the strict exact match reflects the inherent difficulty of predicting precise multi-dimensional intents for dormant users with extremely sparse signals, the text-level consistency metrics remain reasonably high. These results suggest that the Reasoner provides useful intermediate inferences for downstream logic-constrained guidance, while leaving room for more fine-grained role calibration.

Table 10: Prompts for instructing the Reasoner in item indexing.

Instruction
You are a semantic indexing module responsible for mapping product metadata from an e-commerce platform to discrete Semantic ID (SID) tokens.
Prompt
Based on the following product information, generate one Semantic ID (SID) token that represents the item. Title: {item_title} Price: {item_price} Category Path: {category_path} Output only the SID token.
Target
{multi_level_sid}

E.3 Counterfactual Sampling for Diversity (RQ3)

We measure how CF-S changes hit rates and SID-level candidate diversity on approximately 10k users, with the Reasoner beam size fixed to 25 for the CoT-based target item prediction task. Applying CF-S to all target functional roles substantially improves diversity (relative +71% in Coverage-SID₁@25), but reduces exact-item hit rates for top-ranked items ($K < 15$). Given this trade-off, our production deployment applies CF-S only to the tail 20% of the beam results, i.e., the last 5 of 25. Table 9 reports this hybrid strategy: standard decoding is used for the top 20 candidates, while CF-S is applied exclusively to the tail 5. These results show a modest trade-off in hit rates when CF-S is applied to the tail 20% (e.g., Hit Item drops from 44.3% to 43.7%), together with a noticeable increase in intent diversity: the average distinct number of SID₁ rises from 9.82 to 11.63, a relative gain of 18.4%. Overall, the results suggest that CF-S improves Reasoner output diversity while preserving most of the retrieval accuracy. The online diversity results in Section 4.6 are full-framework online measurements, not an isolated CF-S-only A/B test.

F Prompt Template

Tables 10–15 present the detailed prompts used to instruct the Reasoner. They cover SID alignment, item prediction, and functional-role reasoning.

Table 11: Prompts for instructing the Reasoner in item profiling.

Sub Task 1: SID → title	
Instruction	You are an e-commerce index-to-text mapper. Given an internal Semantic ID (SID) token, generate the corresponding product title in the long, keyword-rich style commonly used on Chinese e-commerce platforms.
Prompt	Please generate the original e-commerce product title for the following SID. SID: {multi_level_sid} Title:
Target	{item_title}
Sub Task 2: SID → category path	
Instruction	You are a category prediction module responsible for decoding the product category from Semantic ID (SID) tokens in an e-commerce recommendation system.
Prompt	Given a product Semantic ID (SID), infer its category path in the format: "Level-1 Category > Level-2 Category > Level-3 Category". SID: {multi_level_sid} Category Path:
Target	{category_path}

Table 12: Prompts for instructing the Reasoner in standard behavior prediction.

Sub Task 1: Sequence → next item SID	
Instruction	You are a senior expert in e-commerce recommendation algorithms. Task: Based on the user’s profile and historical interaction sequence, predict the SID of the next item under the user’s {target_behavior} behavior. Steps: 1. Analyze the user’s long-term interest tags and spending level. 2. Observe recent trends and potential demand shifts in the interaction history. 3. Exclude items already appearing in the historical sequence, and recommend a new SID that best matches the user’s current intent.
Prompt	User interactions are recorded as (behavior, item SID) pairs. Based on the user profile and recent interaction sequence, predict the next item the user will interact with. Profile: {user_profile_text} History: {interaction_seq_text} Output only one item SID token, which must be a new item not appearing in the history.
Target	{target_item_token}
Sub Task 2: Sequence → next item title	
Instruction	You are an e-commerce recommendation assistant. Predict the title of the next product the user is likely to {target_behavior}, and make sure it is a new product not appearing in the history. Generate a title that follows Chinese e-commerce naming conventions, including the product category, core selling point, and descriptive modifiers.
Prompt	User Profile: {user_profile_text} Interaction History: {interaction_seq_text} Output only the title of the next product.
Target	{target_item_title}

Table 13: Prompts for instructing the Reasoner in Functional Role-guided Chain-of-Thought.

Instruction You are an e-commerce recommendation expert specializing in user intent understanding. Your task is to accurately predict the SID of the next item for the user’s {target_behavior} based on their profile and interaction history. Please strictly follow the steps below and conduct your reasoning within <think> tags: Profile Anchoring: Summarize the user’s core preference categories. Functional Role Analysis: Scan the interaction history to identify the functional roles of key items (e.g., high-frequency consumables, long-tail durables, novelty-seeking items). Intent Evolution Inference: Analyze the temporal evolution pattern of these roles, such as moving from a durable main product to a complementary accessory, and infer the current demand gap accordingly. Decision Alignment: Based on the inferred demand gap, select the most aligned item SID. After completing your reasoning, start a new line and output only the predicted item SID. Ensure the recommendation is novel and does not duplicate any item in the historical sequence.
Prompt User Profile: {user_profile_text} Interaction Sequence: {interaction_seq_text} Please reason: What functional need is reflected by the key items the user has recently engaged with? Based on this need, what is the most reasonable next item? Output your reasoning process followed by the SID.
Target <think> {think_content} </think> {target_item_token}

Table 14: Prompts for instructing the Reasoner in stepwise reasoning sub-tasks.

Sub Task 1: Key Items Extraction	
Instruction	You are an e-commerce data analyst. Analyze the user profile and interaction history to extract the key items most informative for predicting the user's next <code>{target_behavior}</code>. Selection criteria: Retain items that are either strongly aligned with the user's core preferences or indicative of recent shifts in demand. Output format: List the SIDs of the selected items in chronological order of interaction, one SID per line.
Prompt	User Profile: <code>{user_profile_text}</code> Interaction History: <code>{interaction_seq_text}</code> Please filter the interaction sequence based on the user profile to identify Key Items:
Target	<code>{key_items_list_str}</code>
Sub Task 2: Functional Role Interpretation.	
Instruction	You are a recommendation expert on an e-commerce platform. Analyze the Functional Role of the given key items. Incorporate the items' intrinsic attributes, including sales volume, price, and durability, together with the user's core preference categories to determine their roles in the user's decision path. Role analysis dimensions: Sales frequency: (high_freq, mid_freq, low_freq), Price level: (high_price, mid_price, low_price), Durability: (durable_cate, freq_cate), and the relationship to the user's core preference categories, such as same-category substitution, complementary pairing, or peer-group following. Output format: Attributes: <code>{sales/price/durability}</code>, relationship to <code>{user_preference_category}</code> is <code>{relationship_type}</code>.
Prompt	User Information: <code>{user_profile_text}</code> Interaction Sequence: <code>{interaction_seq_text}</code> Key Items: <code>{key_items_list_str}</code> Infer the functional roles provided by these key items:
Target	<code>{key_items_and_role_no_beh}</code>
Sub Task 3: Joint Reasoning & Intent Modeling.	
Instruction	You are an expert in e-commerce intent reasoning. Analyze the user's interaction history and build a logical chain leading to the user's next <code>{target_behavior}</code> decision. Clue Extraction: Remove noisy interactions from the sequence and identify key items that reflect the user's core intent. Role Decoding: Based on the user profile, analyze the Functional Roles of these key items. Intent Modeling: Predict the Functional Role of the item for the user's next <code>{target_behavior}</code>. Output requirement: Directly output a structured analysis report without producing the final SID.
Prompt	User Profile: <code>{user_profile_text}</code> Interaction History: <code>{interaction_seq_text}</code> Reasoning Task: Identify key items, analyze their functional roles, and ultimately predict the functional role of the next item.
Target	<code>{think_content}</code>

Table 15: Prompts for instructing the Reasoner in reasoning-based decision alignment.

Instruction You are a recommendation decision system for an e-commerce platform. Based on the provided user profile, interaction history, and Functional Role analysis, predict the SID of the item for the user's next {target_behavior}. Decision logic: The predicted item must fill the demand gap identified in the analysis and match the user's preferences over price and item attributes. Output only one SID, and do not repeat any item in the historical sequence.
Prompt User Profile: {user_profile_text} Interaction History: {interaction_seq_text} Intent Reasoning Analysis: {key_items_and_role} Functional Role of the next item: {target_role} Based on the above analysis, the SID of the next item is:
Target {target_item_token}