Enhancing Local Life Service Recommendation with Agentic Reasoning in Large Language Models

Anonymous ACL submission

Abstract

Local life service recommendation is distinct from general recommendation scenarios due to its strong living need-driven nature. Fundamentally, accurately identifying a user's immediate living need and recommending the corresponding service are inextricably linked tasks. However, prior works typically treat them in isolation, failing to achieve a unified modeling of need prediction and service recommendation. In this paper, we propose a novel large language model based framework that jointly performs living need prediction and service recommendation. To address the challenge of noise in raw consumption data, we introduce a behavioral clustering approach that filters out accidental factors and selectively preserves typical patterns. This enables the model to learn a robust logical basis for need generation and spontaneously generalize to long-tail scenarios. To navigate the vast search space stemming from diverse needs, merchants, and complex mapping paths, we employ a curriculum learning strategy combined with reinforcement learning with verifiable rewards. This approach guides the model to sequentially learn the logic from need generation to category mapping and specific service selection. Extensive experiments demonstrate that our unified framework significantly enhances both living need prediction performance and recommendation accuracy, validating the effectiveness of jointly modeling living needs and user behaviors.

1 Introduction

Local life service platforms, such as Meituan[1], facilitate urban residents in the discovery and acquisition of surrounding physical services, including dining, accommodation, domestic help, and entertainment (Li et al., 2022; Ping et al., 2021). These platforms cover the core scenarios of daily urban life. Building upon this, the central task of local life service recommendation is to precisely recommend merchants that satisfy the user's potential living needs, taking user profiles, preferences, and spatiotemporal contexts as input. Such recommendations aim to fulfill real-world living needs and directly lead to offline physical consumption, thereby exerting a substantive influence on users' daily lives. Precise recommendations significantly reduce decision-making costs, helping individuals efficiently obtain high-quality experiences.

Distinct from general recommendation scenarios, local life services are characterized by a strong living need-driven nature. Fundamentally, living need prediction and service recommendation have an inseparable correlation. Both must be rooted in the comprehensive modeling of users and contexts, as well as a deep understanding of human need generation and behavioral laws. This correlation encourages the joint modeling of these two tasks, with the expectation that they will mutually enhance one another. However, prior works have not achieved a unified modeling of need prediction and service recommendation. Existing industrial generative methods for local life service recommendation (Han et al., 2025; Zhou et al., 2025a,b) do not explicitly model user living needs. Conversely, existing living need prediction methods (Lan et al., 2023, 2025) explicitly model needs, yet the subsequent mapping between needs and service recommendation relies on hard rules or separately trained embedding models. This lack of joint modeling constrains the accuracy of the final recommendation. This points to a significant research gap: how to construct a unified framework that jointly performs living need prediction and service recommendation.

Large Language Models (LLMs) offer a new opportunity to address this gap (Xi et al., 2023; Chen et al., 2024a; Liu et al., 2025a). LLMs possess extensive world knowledge and commonsense reasoning capabilities, enabling a deep semantic

[1] meituan.com

understanding of human living needs. Furthermore, they can unify the prediction of needs and services within a single linguistic format. By sharing the same token space, LLMs can naturally model the joint distribution of living needs and services. Nevertheless, this process presents challenges:

Firstly, constructing high-quality training data that reflects the true laws of need-driven consumption is difficult. Raw records from platforms do not centrally reflect the logic of need generation. The reasons behind recorded consumption behaviors are manifold; beyond modelable laws of living needs, records contain noise from unpredictable factors such as platform promotions. Screening for data that reflects universal laws of need generation is essential for effective model learning but constitutes a significant challenge.

Besides, the joint modeling of living needs and specific services involves navigating an extremely vast search space. This complexity stems from the high diversity of fine-grained living needs, the massive scale of available merchants, and the numerous logical paths that explain how a specific service satisfies a need. Consequently, learning to accurately map abstract need generation to specific service interaction within this expansive space presents a significant difficulty.

To resolve these challenges, we propose a novel joint modeling method. To address the first challenge, we design a clustering method based on user behavioral features. This approach filters out accidental noise and selectively preserves the behaviors of typical users, allowing the model to focus on general behavioral laws. Unlike traditional machine learning methods that require explicit memorization of long-tail data, our approach enables the LLM to learn a compact yet robust logical basis from high-quality patterns. Relying on the model's inherent reasoning capabilities, it can then effectively generalize this learned logic to long-tail scenarios. To address the second challenge, we adopt a curriculum learning approach during training. We constrain the search space by allowing the model to learn in steps. Specifically, the training phase is designed to let the model sequentially learn the laws of need generation, the mapping from need to category, and finally, specific user behavior generation. Additionally, we employ Reinforcement Learning with Verifiable Rewards (RLVR) rather than simple imitation learning to enhance the model's active reasoning ability. In the inference phase, our model adopts an agentic reasoning method, simulating the human cognitive process from living need to specific consumption behavior, thinking step-by-step to solve the recommendation problem.

Our contributions are as follows: (1) We are the first to propose the unified problem of user living need and behavior prediction in local life service recommendation. (2) We develop a novel methodology that integrates clustering-based sampling to capture robust behavioral laws and employs curriculum learning with RLVR to effectively constrain the vast search space. This enables the model to master the agentic reasoning process from abstract living needs to specific service interactions. (3) We conduct extensive experiments to verify our framework, demonstrating its superior performance over existing baselines and validating the effectiveness of each component.

2 Problem Formulation

We formulate local life service recommendation not merely as a sequential matching task, but as a hierarchical agentic decision-making process driven by explicit living needs. We define the hierarchical decision space through two intermediate granularities. The highest level consists of living needs $\mathcal{I}$, which represent high-level latent intents such as *Business Travel*. Given that such needs are typically implicit in raw click logs, we construct the ground truth set $\mathcal{I}$ by leveraging LLMs to extract explicit need descriptions from user post-transaction reviews. Bridging these abstract needs and specific merchants, we define Semantic Categories $\mathcal{C}$ as the set of coarse-grained service types, such as *Hotel*, which facilitate the mapping from intent to concrete consumption behaviors.

For each user $u \in \mathcal{U}$, we construct a comprehensive context view comprising three integral components. First, the user profile $\mathcal{P}_u$ encompasses static feature sets. Second, the historical sequence S_u records the chronological history of past interactions, denoted as $S_u = \{y_1, y_2, \ldots, y_n\}$, where each interaction $y_k = (i_k, c_k, b_k)$ is explicitly modeled as a tuple containing the living need, semantic category, and specific behavior. Finally, the spatio-temporal context X_{n+1} captures the environmental constraints for the target prediction, specifically the timestamp t_{n+1} and location ℓ_{n+1}. Formally, given the user profile $\mathcal{P}_u$, historical sequence S_u, and the target context X_{n+1}, the primary objective of HiAgentRec is to model the joint probability of the user's hierarchical decision path.

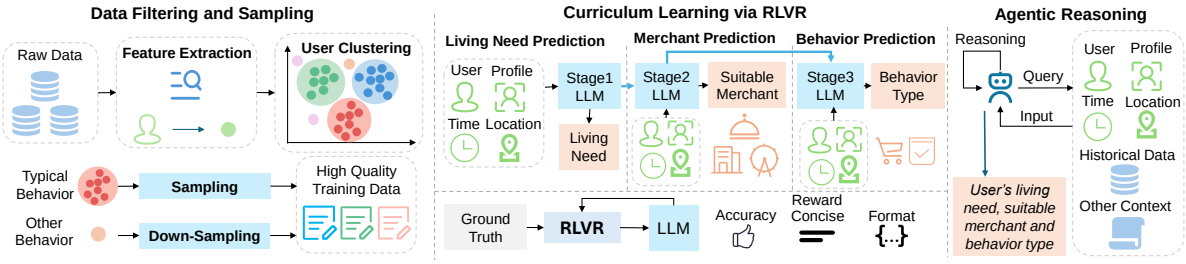

Figure 1: The overall architecture of HiAgentRec, which integrates a noise-robust data pipeline with an RLVR-optimized curriculum learning strategy for hierarchical reasoning.

2.1 Related Work

2.1.1 LLM for Recommendation

With the remarkable general capabilities demonstrated by LLMs such as ChatGPT (OpenAI, 2023), DeepSeek (Guo et al., 2025) and Qwen (Bai et al., 2024), researchers have increasingly focused on how to harness LLMs for recommender systems. Existing approaches can be broadly categorized into two paradigms.

The first treats the LLM as an enhancer. For instance, KAR (Xi et al., 2023) leverages LLMs to generate textual descriptions of users or items and enhances the representation power of traditional recommendation models via embedding augmentation strategies. The second paradigm frames recommendation as a language processing task. Works such as P5 (Geng et al., 2023), ONCE (Liu et al., 2023), UniSRec (Hou et al., 2022), and M6Rec (Cui et al., 2022) reformulate recommendation objectives into natural language prompts or generation tasks, enabling end-to-end LLM-based recommendation. However, a significant gap exists between the pretraining objective of LLMs and the optimization goals in recommendation. This mismatch often leads to challenges such as popularity bias and limited fine-grained controllability when LLMs are applied directly to recommendation tasks. (Hua et al., 2023)

To bridge this gap, fine-tuning has become a widely adopted strategy, exemplified by representative works such as BIGRec (Bao et al., 2023), Softmax-DPO (Chen et al., 2024b), and RosePO (Liao et al., 2024). More recently, researchers have explored alignment techniques better suited for complex reasoning and multi-objective optimization. GRPO (Shao et al., 2024) marks a paradigm shift toward leveraging complex, multi-level feedback signals in LLM alignment. OneRec-Think (Liu et al., 2025b) proposes a three-stage Reason–Reflect–Revise generation framework that explicitly guides the LLM to simulate the deep logic underlying user living need, item attributes, and contextual interactions. Rec-R1 (Lin et al., 2025) establishes a general closed-loop optimization framework that directly utilizes feedback from a fixed recommender to refine LLM generations, bypassing the need for costly data distillation. RecLLM-R1 (Xie et al., 2025) combines GRPO with chain-of-thought reasoning and employs a flexible multi-objective reward function to jointly optimize accuracy, diversity, and novelty. STREAM-Rec (Zhang et al., 2025) integrates GRPO with explicit reasoning paths, transforming the recommender from a black-box matcher into an interpretable, multi-step decision process. GREAM (Hong et al., 2025) enhances GRPO with structured semantic indexing and sparse reward augmentation to mitigate feedback sparsity and semantic drift in recommendation. Finally, RankGRPO (Zhu et al., 2025) adapts the GRPO framework to naturally align its reward mechanism with the rank-style output inherent to recommendation tasks. However, existing approaches largely neglect the explicit modeling of user needs, which serve as the fundamental driving force in local life service scenarios. To bridge this gap, our work introduces a joint modeling paradigm that simultaneously optimizes the prediction of underlying living needs and concrete interactions.

2.1.2 Local Life Service Recommendation

Local life service recommendation serves as a crucial bridge connecting users to offline physical services and attracts significant attention from both academia and industry. Early research primarily relies on collaborative filtering and matrix factorization techniques (Bao et al., 2015; Lian et al., 2014). Subsequent methods employ Recurrent Neural Networks (RNNs) and Transformer architectures to capture user sequential behavioral patterns, and they further integrate spatio-temporal contexts spe-

cific to local life scenarios, such as geographical location and time slots, to enhance recommendation accuracy (Feng et al., 2018; Ni et al., 2020). More recently, studies adopt generative recommendation paradigms. By encoding items into semantic identifiers or generating target items directly, these paradigms bridge the semantic gap between user context and item representations to achieve greater precision (Zhou et al., 2025a,b; Han et al., 2025).

However, these mainstream methods largely focus on mining implicit preferences from historical interaction records. Essentially, they perform behavior-to-behavior fitting and overlook the core motivation driving local life consumption, specifically the rigid and explicit living need of users. To address this limitation, another line of research introduces living need recognition and living need prediction. These approaches utilize techniques such as disentangled representation learning or graph neural networks to infer latent user living needs (Ping et al., 2021; Li et al., 2022), and recent works further attempt to explicitly model user living need states (Lan et al., 2023, 2025). Although these attempts improve system interpretability to some extent, they typically treat living need prediction and service matching as disjoint stages. The subsequent mapping process often relies on hard-coded rules or independently trained vector space retrieval, which prevents the effective translation of living need understanding into precise consumption decisions. Furthermore, while the application of large language models in recommender systems rises, typically for feature augmentation or ranking optimization, few studies achieve end-to-end joint modeling from abstract living needs to specific service consumption within a unified generative framework. Targeting this gap, our work proposes an LLM-based joint modeling approach. By simulating the human-like agentic reasoning process, our method achieves mutual enhancement between living need prediction and service recommendation.

3 Methodology

3.1 Overall Framework

In this section, we present **HiAgentRec** (**Hi**erarchical **Agent**ic **Recommendation**), an RL-enhanced agentic framework designed to jointly perform living need prediction and service recommendation. As illustrated in Figure 1, HiAgentRec orchestrates a coarse-to-fine reasoning process. Our methodology comprises three integral components:

Addressing the high noise and sparsity inherent in user behavioral data, we introduce a *Data Filtering and Sampling Pipeline* (Section 3.2). To navigate the vast solution space, we employ a Curriculum Learning Strategy optimized by RLVR (Section 3.3). To simulate the human cognitive process from living need to specific consumption behavior, we reframe the recommendation task as a multi-step decision-making process via our *Agentic Reasoning Pipeline* (Section 3.4).

3.2 Noise-Robust Data Construction

To mitigate the impact of stochastic noise and data sparsity, we propose a three-stage curation pipeline that filters accidental interactions while preserving representative long-tail patterns. We implement a clustering-based approach to disentangle unstructured noise from valid behavioral signals. By partitioning the user feature space $\{\mathbf{x}_u\}$ into K clusters using MiniBatchKMeans (Sculley, 2010), we calculate a normalized typicality score for each user:

$$z_u = \frac{\|\mathbf{x}_u - \boldsymbol{\mu}_k\|_2}{\sigma_k}, \tag{1}$$

where $\boldsymbol{\mu}_k$ and σ_k denote the centroid and intra-cluster standard deviation of cluster C_k. Users exceeding a deviation threshold are flagged as point-level outliers, while clusters falling below a minimum support threshold τ_{min} are pruned as non-representative edge cases.

To prevent the information loss in uniform sampling, we evaluate each cluster C_k based on its scale S_k, cohesion Q_k (inlier ratio), and semantic dominance. We discard low-quality clusters ($S_k < \tau_{\text{size}}$ and $Q_k < \tau_{\text{quality}}$) to eliminate noise. Conversely, we assign an enhanced sampling rate r_{high} to long-tail clusters (small S_k but high Q_k) to preserve niche behavioral patterns. The final dataset $\mathcal{D}_{\text{final}}$ is adaptively balanced to maintain structural cohesion and semantic diversity.

3.3 Curriculum Learning via RLVR

To effectively navigate the search space and bridge the semantic gap between abstract user needs and specific items, we design a curriculum learning strategy optimized by RLVR. Instead of directly optimizing the complex end-to-end mapping, we decompose the generative process into a sequence of hierarchical subtasks. The training protocol sequentially activates three optimization phases, pro-

gressively expanding the generative scope from intents to specific decision paths.

The curriculum begins with the living need alignment phase, where the model focuses solely on the high-level semantic space. In this foundational stage, the objective is to maximize the likelihood of the correct living need description. This step ensures the model grounds its reasoning in user intent without the interference of massive item candidates. Building upon this semantic foundation, the second phase, category-constrained reasoning, extends the prediction target to include service categories. The model learns to predict the business taxonomy conditioned on both the context and the generated need, optimizing the joint probability. This intermediate step bridges the gap between abstract intent and standardized platform supply. The process culminates in the full-path agentic decision phase. In this final stage, the model navigates the complete search space to identify the specific item, optimizing the full reasoning chain.

Throughout this curriculum, we employ GRPO (Guo et al., 2025) as the optimization backbone. We initialize each new phase with the model checkpoint from the preceding stage, facilitating seamless knowledge transfer. This staged policy optimization allows the model to master the complex reasoning logic step-by-step, mitigating the issues of reward sparsity in direct end-to-end training.

We design a reward function $\mathcal{R}$ to evaluate model outputs across a set of tasks For any task, the total reward is a weighted sum of matching accuracy, format validity, and response length. To ensure structural consistency and conciseness, we apply generic penalties across all tasks. The Format Reward $\mathcal{R}_{\text{fmt}}$ encourages valid JSON generation: The Length Reward $\mathcal{R}_{\text{len}}$ prevents verbose outputs. It is activated only when the prediction is correct (i.e., $\mathcal{R}_{\text{match}} > 0$) and decays over training steps to prioritize accuracy in later stages:

$$\mathcal{R}_{\text{len}} = \mathbb{I}(\text{correct}) \cdot \alpha e^{-\frac{\text{step}}{T}} \cdot \min\left(1, \max\left(0, 1 - \frac{\ell - \ell_{\min}}{\ell_{\max} - \ell_{\min}}\right)\right). \tag{2}$$

Here, $\mathbb{I}(\cdot)$ is the indicator function. This term creates a "safe zone" where $\ell \leq \ell_{\min}$ yields maximum reward, penalizing only unnecessary tokens beyond $\ell_{\min}$.

For category prediction, to account for linguistic ambiguity, we calculate the cosine similarity between the embeddings of the prediction and the ground truth. If the closest candidate in the taxonomy matches the label, a maximum reward of 1.0 is given; otherwise, the similarity score itself serves as a partial reward to guide the policy.

3.4 Agentic Reasoning

To transcend the limitations of black-box direct prediction, we propose a *Hierarchical Agentic Reasoning Pipeline* that reframes next item prediction as an autonomous decision-making process. As illustrated in the framework, the LLM functions as an agent that actively orchestrates specific function tools to narrow down the solution space.

The process is governed by a System Role that defines the agent's objective: to formulate precise recommendations by analyzing user data and context. The pipeline enforces a structured interaction protocol. Specifically, the prompt guides the agent through a three-stage reasoning process: (1) Living Need Inference, which combines user profiles with spatiotemporal context to determine intent; (2) Category Mapping, which aligns identified needs with semantic domains (e.g., Food, Entertainment); and (3) Behavior Ranking, which selects the optimal specific action based on matching scores. To facilitate automated parsing, all agent outputs are strictly constrained to a JSON schema. Due to space limitations, the complete prompt specification and template are detailed in Appendix A.8.

4 Experiments

To evaluate our framework, we conduct extensive experiments focusing on seven research questions. We first assess the overall recommendation performance (RQ1) and the contribution of individual components (RQ2). We then investigate the model's robustness, specifically its improvement for cold-start users (RQ3). Furthermore, we provide case study regarding interpretability (RQ4), the preservation of general capabilities during specialization (RQ5), the mitigation of policy collapse (RQ6), and the impact of semantic embedding models (RQ7). Due to space constraints, the detailed results for RQ3–RQ7 are provided in the Appendix.

4.1 Experimental Setup

4.1.1 Dataset

To evaluate the effectiveness of HiAgentRec, we utilize two large-scale real-world datasets collected

Table 1: Performance comparison on Shanghai (In-domain) and Beijing (Cross-City Generalization) settings. The best results are **bold**, and the second-best are underlined.

Model	Shanghai (In-domain)					Beijing (Cross-City)				
	HR@1	HR@3	HR@5	NDCG@3	NDCG@5	HR@1	HR@3	HR@5	NDCG@3	NDCG@5
GRU4Rec	0.3873	0.5182	0.5941	0.2982	0.2696	0.2840	0.4150	0.4560	0.3620	0.3810
SASRec	0.4387	0.6213	0.6984	0.5455	0.5773	0.3015	0.4381	0.4721	0.3842	0.3982
BERT4Rec	0.4685	0.6185	0.6973	0.5558	0.5881	0.3202	0.4504	0.4733	0.3963	0.4057
LLM (Base)	0.2160	0.2190	0.2240	0.2176	0.2209	0.1972	0.2120	0.2290	0.1922	0.2059
LLM+SFT	0.3879	0.5420	0.6650	0.4920	0.5280	0.2583	0.4650	0.5410	0.3820	0.4100
KAR	0.4850	0.6410	0.7150	0.5720	0.6010	0.3350	0.4620	0.4890	0.4100	0.4250
HiAgentRec	**0.6098**	**0.7420**	**0.8010**	**0.6620**	**0.6950**	**0.6210**	**0.7580**	**0.8120**	**0.6750**	**0.7080**

from a large-scale food delivery platform. The datasets specifically focus on user behaviors within Shanghai (used for in-domain evaluation) and Beijing (used for cross-city generalization), capturing the dense and complex spatiotemporal interaction patterns of metropolitan areas. The raw data comprises user logs rich in contextual information, encompassing timestamps, GPS locations, categories, and explicit user living needs (e.g., dining, tourism). The detailed statistics of both datasets are summarized in Appendix A.3.

4.1.2 Comparison methods

To evaluate the effectiveness of HiAgentRec, we compare it against a set of baselines, ranging from traditional deep sequential models to recent LLM-based approaches. To ensure a fair comparison, all models were trained and evaluated on the same dataset. Due to space limitations,, detailed descriptions of these baseline models are provided in Appendix A.2.

4.2 Overall Performance(RQ1)

We measure the performance of HiAgentRec and baselines across two settings: in-domain (Shanghai), cross-city generalization (Beijing), as shown in Table 1. HiAgentRec achieves superior performance across all metrics in both in-domain and cross-city settings. In the Shanghai dataset, our model outperforms the strongest baseline by a significant margin, improving 25.73% in HR@1 and 15.64% in NDCG@5. In the zero-shot Beijing setting, HiAgentRec demonstrates remarkable robustness, achieving a 85.37% and 66.59% gain in HR@1 and NDCG@5 respectively compared to the best-performing baselines. This stands in contrast to traditional methods, which suffer from performance degradation when transferring between cities. The reasoning framework mitigates overfitting to source-specific statistical correlations. Traditional sequential models rely on local transition patterns inherent to Shanghai's urban layout, leading to failure when spatiotemporal contexts shift. In contrast, by modeling the invariant causal chain, HiAgentRec captures universal living logic, thereby achieving robust out-of-distribution generalization.

4.3 Ablation Study(RQ2)

To systematically verify the effectiveness of each component in HiAgentRec, we conduct ablation studies focusing on three key aspects: the hierarchical agentic reasoning structure, the data curation pipeline, and the choice of semantic reward model. Furthermore, we provide a detailed analysis of the curriculum learning dynamics to validate our multi-stage training strategy.

As summarized in Table 2, we compare the full HiAgentRec model against three variants. The variant *w/o Agentic Reasoning*, which attempts to directly map context to categories without intent inference, suffers the most significant performance drop. This sharp decline confirms that decomposing the complex recommendation task into a chain is crucial. Removing the data filtering pipeline (*w/o Data Selection*) leads to a noticeable degradation in recommendation quality, evidenced by a 7.95% decrease in NDCG@5 compared to the full model. This indicates that raw behavioral logs contain substantial noise. Our filtering purifies the training signal, allowing GRPO to optimize policies based on high-quality, representative user patterns. Eliminating the semantic modeling component results in a considerable performance loss. This underscores the necessity of a semantic bridge between the model's generated text and specific item.

We further investigate the training dynamics to understand how Curriculum Learning facilitates model convergence. The model trained in the

Table 2: Ablation study on Agent Reasoning, Data Selection, and Semantic Model components. We compare the full HiAgentRec against variants without the reasoning chain, without the data filtering pipeline, and with a generic BERT encoder replacing the BGE model.

Variant	HR@1	HR@3	HR@5	NDCG@3	NDCG@5
w/o Agentic Reas.	0.4828	0.5172	0.5517	0.5045	0.5149
w/o Data Selection	0.5980	0.7220	0.7800	0.6116	0.6438
w/o Semantic Model	0.5209	0.6383	0.7475	0.6283	0.6339
HiAgentRec (Full)	**0.6098**	**0.7420**	**0.8010**	**0.6620**	**0.6950**

preceding phase serves as a superior initialization checkpoint for subsequent tasks compared to the raw LLM. We observe that the initial accuracy for category prediction improves from 0.180 to 0.184. This empirically suggests that the preliminary task of inferring user living need equips the model with a common sense understanding of the domain, thereby laying a robust foundation for fine-grained category and behavior predictions.

Table 3 illustrates the evolution of metrics during the Task 2 (Category) training phase. A compelling observation is the backward positive transfer: although Phase 2 focuses on optimizing Category output, the accuracy of the upstream *Living need* task also improves steadily, rising from 51.36% to 61.55%. This indicates that the reasoning chain is not merely a one-way dependency; the feedback from fine-grained category matching reinforces the model's understanding of user living needs.

Table 3: Evolution of Living need Accuracy, Category Hit Rate (HR), and Ranking Quality (NDCG) during Task 2 training.

Stage	Need Acc.	HR@1	HR@3	HR@5	NDCG@3	NDCG@5
After Task 1	0.5136	0.1843	0.2512	0.2890	0.1954	0.2013
20% step	0.5291	0.3992	0.5215	0.5840	0.4125	0.4253
40% step	0.5500	0.5105	0.6420	0.7054	0.5350	0.5512
60% step	0.6318	0.5315	0.6750	0.7380	0.5827	0.6129
80% step	0.6227	0.5763	0.7180	0.8052	0.6350	0.6684
After Task 2	0.6155	0.6028	0.7350	0.7945	0.6419	0.6874

4.4 Case Study(RQ4)

To illustrate the reasoning capabilities of HiAgentRec under complex scenarios, we present three representative cases in Figures 2, 3, and 4. These cases highlight HiAgentRec's ability to utilize world knowledge and logical deduction to handle cold-start and context-shifting scenarios effectively.

Case 1 (Cold-Start & Zero-Shot): This user has no historical behavior. The model relies entirely on the profile (Married, Kids) and context (19:00, Residential Area). HiAgentRec infers that 7 PM is a typical dinner preparation time for a family. It reasons that while main ingredients might be ready, Fruit serves as a healthy, high-probability supplement for children after dinner, accurately predicting the living need of Family Care.

Case 2 (Cross-Context & Habitual): This user is in a generic Scenic Area at 2:00 AM, a scenario vastly different from their usual history (Home/-Workplace). Traditional models might fail due to the location shift or recommend generic tourist services. HiAgentRec combines the context (late night + limited options) with the user's strong historical preference for Bread & Cakes (Snack). It reasons that the user needs quick hunger relief rather than a formal meal, correctly recommending a bakery item.

Case 3 (Specific Living need Recognition):This user appears at their workplace late at night (22:30). By analyzing the history of booking Economy Hotels near the workplace in the mornings, the model identifies a pattern of business travel rather than leisure. Consequently, it predicts an Accommodation living need for a budget-friendly hotel, aligning with the user's history.

CASE 1: Family Scenario

<intent>
{"predicted_intent": "Family Meal Preparation",
"reasoning_summary": "Triggered by 'Married with Kids' profile and '19:00' timestamp. The user is in a residential zone at dinner time, implying a need to provide food for the family rather than personal dining."}</intent>
<category>
{ "predicted_category": "Grocery & Fresh Produce",
"reasoning_summary": "Mapped 'Family Meal Preparation' to 'Grocery & Fresh Produce'. At this hour, preparing a full meal from scratch might be late, but purchasing supplementary items like fruits is a high-probability sub-domain demand." } </category>

Figure 2: Case Study 1: Cold-Start scenario where the agent infers based on the user's profile and time context.

CASE 2: Late Night Hunger

<intent>
{"predicted_intent": "Quick Hunger Relief ",
"reasoning_summary": "Triggered by Context 'Scenic Area at 02:00'. Normal family cooking is impossible here. The biological need is hunger, but limited facility availability constrains the intent to 'making do with a bite'." }</intent>
<category>
{ "predicted_category": "Food & Beverage (Bakery)",
"reasoning_summary": "Mapped 'Quick Hunger Relief' to 'Bakery'. In a scenic area late at night, formal dining is closed. Bakery/Convenience items are the most accessible domain for immediate consumption."}</category>

Figure 3: Case Study 2: Late-night hunger scenario showing how the agent handles cross-context reasoning.

CASE 3: Business Trip

<intent>
{"predicted_intent": "Business Stay",
"reasoning_summary": "Triggered by Spatiotemporal Pattern. User is still at 'Workplace' at '22:30', combined with historical morning bookings near work, indicating a business trip or overtime layover scenario."}</intent>
<category>
{ "predicted_category": "Accommodation (Budget)",
"reasoning_summary": "Mapped 'Business Stay' to 'Accommodation'. Given the user's history is exclusively 'Accommodation (Budget)', the domain is narrowed to budget-friendly lodging options."}</category>

Figure 4: Case Study 3: Business trip scenario demonstrating the agent's ability to identify patterns from history.

5 Deployment and Application

To address the diverse and complex service requirements in a large-scale food delivery platform, we deployed the model across two distinct high-traffic scenarios. Considering that user living needs typically exhibit spatio-temporal stability, where motivations remain relatively consistent within specific time windows and geographic areas, we implemented a near-line deployment strategy. Specifically, HiAgentRec triggers demand inference only when a shift in a user's spatio-temporal context is detected. These predictions are then integrated as a supplementary recall channel for the online recommendation pipeline.

The Huisheng (Smart Savings) prioritizes promotional value and cross-category discovery, serving as a critical traffic aggregation hub within system. By injecting intent-aware categories generated by HiAgentRec, we aimed to surface items that traditional collaborative filtering might overlook. During a rigorous online A/B test from January 14, 2026, to February 3, 2026, involving 20% of total production traffic, our proposed method achieved substantial performance gains over the highly optimized production baseline. Specifically, we observed a 13.12% increase in CTCVR, a 4.37% lift in CTR, an 8.32% improvement in CVR, and a 10.11% growth in GTV. These results underscore the model's capacity to understand user needs, effectively scaling transaction volume.

The second deployment focuses on the landing page, where the primary objective is to maintain user engagement and facilitate conversion following an initial click. During a rigorous online A/B test from January 16, 2026, to February 5, 2026, involving 20% of total production traffic, the deployment on the landing page same as Huisheng module yielded an overall CTCVR increase of 1.48%, an overall CVR increase of 1.46% and a UV New User Conversion Rate (UV-NCR) growth of 2.79%. A granular analysis across vertical business sectors reveals the model's exceptional cross-domain adaptability. UV-NCR increased by 4.04% in Leisure and Entertainment, 6.65% in Beauty and Haircare, 2.84% in Instashopping of the platform, 4.20% in In-store Dining, and 2.74% in Pharmaceutical services. These consistent improvements across diverse domains underscore the versatility and precision of the agent's reasoning in capturing multifaceted user interests.

Beyond overall performance gains, HiAgentRec demonstrates efficacy in cold-start recommendation. By leveraging the semantic reasoning of LLMs, we observed a substantial lift in conversion metrics for new users and items. Specifically, the deployment resulted in a 1.45% increase in Payment UV (Unique Visitors) and a 1.52% rise in Payment PV (Page Views). For the new user segment, the New User Payment UV grew by 3.07%, while New User Payment PV increased by 2.90%.

Overall, these experimental results demonstrate the practical value of our proposed method in large-scale industrial applications.

6 Conclusion

This paper proposes HiAgentRec, a unified framework for local life service recommendation that integrates living need prediction with agentic reasoning. By modeling the causal chain, the framework addresses the limitations of disjoint modeling in existing approaches. We introduce a clustering-based data filtering pipeline to mitigate noise and employ a curriculum learning strategy with RLVR to navigate complex search spaces. Extensive experiments on real-world datasets demonstrate that HiAgentRec consistently outperforms baselines. Notably, the model exhibits strong robustness in cross-city generalization and cold-start scenarios, validating the efficacy of the hierarchical reasoning structure. Future work will focus on incorporating multimodal information and exploring multi-objective optimization to further enhance recommendation performance.

7 Limitations

Despite the superior performance demonstrated by HiAgentRec, several limitations warrant discussion and provide directions for future work.

First, due to the computational overhead, our cur-

rent deployment is restricted to near-line scenarios rather than real-time online inference. Future work will focus on model acceleration techniques, including knowledge distillation to smaller specialized models and structured pruning, aiming to achieve online inference capabilities.

Second, the framework currently relies exclusively on textual modalities, overlooking the rich multimodal information inherent in local life services. User reviews often contain images and merchants provide visual menus and storefront photos. Incorporating such multimodal signals could enhance the model's understanding of user living needs, particularly in scenarios where text alone is insufficient. Addressing these limitations through continued research will further bridge the gap between innovation and deployment.

8 Ethical Considerations

The deployment of LLM-based recommendation systems raises important ethical considerations. LLMs may inherit societal biases present in training corpora, potentially manifesting as unfair recommendations across demographic groups. Our clustering-based data filtering could inadvertently amplify or reduce these biases. We mitigate this by auditing training data for imbalances, evaluating recommendation quality across user segments, and incorporating fairness considerations into model development. Systematic bias measurement across protected attributes remains an area for future work.

During the preparation of this manuscript, the authors used AI writing assistants solely for language polishing and coding assistance. All scientific content, analyses, and conclusions were developed and verified by the authors, who assume full responsibility for the work's integrity.

References

Jinzheng Bai and 1 others. 2024. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Jie Bao, Yu Zheng, David Wilkie, and Mohamed Mokbel. 2015. Recommendations in location-based social networks: a survey. *GeoInformatica*, 19(3):525–565.

Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Chong Chen, Fuli Feng, and Qi Tian. 2023. A bi-step grounding paradigm for large language models in recommendation systems. *Preprint*, arXiv:2308.08434.

Junyi Chen, Lu Chi, Bingyue Peng, and Zehuan Yuan. 2024a. Hllm: Enhancing sequential recommendations via hierarchical large language models for item and user modeling. *Preprint*, arXiv:2409.12740.

Yuxin Chen, Junfei Tan, An Zhang, Zhengyi Yang, Leheng Sheng, Enzhi Zhang, Xiang Wang, and Tat-Seng Chua. 2024b. On softmax direct preference optimization for recommendation. *Preprint*, arXiv:2406.09215.

Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-rec: Generative pretrained language models are open-ended recommender systems. *Preprint*, arXiv:2205.08084.

Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*, pages 1459–1468.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2023. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). *Preprint*, arXiv:2203.13366.

Daya Guo and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Ruidong Han, Bin Yin, Shangyu Chen, He Jiang, Fei Jiang, Xiang Li, Chi Ma, Mincong Huang, Xiaoguang Li, Chunzhen Jing, and 1 others. 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.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. *Preprint*, arXiv:1511.06939.

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

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. *Preprint*, arXiv:2206.05941.

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.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. *Preprint*, arXiv:1808.09781.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Xiaochong Lan, Jie Feng, Yizhou Sun, Chen Gao, Jiahuan Lei, Xinleishi Xinleishi, Hengliang Luo, and Yong Li. 2025. Open-set living need prediction with large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5454–5472.

Xiaochong Lan, Chen Gao, Shiqi Wen, Xiuqi Chen, Yingge Che, Han Zhang, Huazhou Wei, Hengliang Luo, and Yong Li. 2023. Neon: Living needs prediction system in meituan. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5292–5302.

Yinfeng Li, Chen Gao, Xiaoyi Du, Huazhou Wei, Hengliang Luo, Depeng Jin, and Yong Li. 2022. Automatically discovering user consumption intents in meituan. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3259–3269.

Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 831–840.

Jiayi Liao, Xiangnan He, Ruobing Xie, Jiancan Wu, Yancheng Yuan, Xingwu Sun, Zhanhui Kang, and Xiang Wang. 2024. Rosepo: Aligning llm-based recommenders with human values. *Preprint*, arXiv:2410.12519.

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

Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2023. Once: Boosting content-based recommendation with both open- and closed-source large language models. *Preprint*, arXiv:2305.06566.

Yueyang Liu, Jiangxia Cao, Shen Wang, Shuang Wen, Xiang Chen, Xiangyu Wu, Shuang Yang, Zhaojie Liu, Kun Gai, and Guorui Zhou. 2025a. 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, Xinchen Luo, Lejian Ren, Zixing Zhang, Qianqian Wang, Kuo Cai, Yunfan Wu, Hongtao Cheng, and 7 others. 2025b. Onerec-think: In-text reasoning for generative recommendation. *Preprint*, arXiv:2510.11639.

Jiacheng Ni, Pengpeng Zhao, Jiajie Xu, Junhua Fang, Zhixu Li, Xuefeng Xian, Zhiming Cui, and Victor S Sheng. 2020. Spatio-temporal self-attention network for next poi recommendation. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pages 409–423. Springer.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yukun Ping, Chen Gao, Taichi Liu, Xiaoyi Du, Hengliang Luo, Depeng Jin, and Yong Li. 2021. User consumption intention prediction in meituan. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 3472–3482.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

David Sculley. 2010. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. *Preprint*, arXiv:1904.06690.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang,

Weinan Zhang, Rui Zhang, and Yong Yu. 2023. Towards open-world recommendation with knowledge augmentation from large language models. *Preprint*, arXiv:2306.10933.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Yu Xie, Xingkai Ren, Ying Qi, Yao Hu, and Lianlei Shan. 2025. Recllm-r1: A two-stage training paradigm with reinforcement learning and chain-of-thought v1. *Preprint*, arXiv:2506.19235.

Junjie Zhang, Beichen Zhang, Wenqi Sun, Hongyu Lu, Wayne Xin Zhao, Yu Chen, and Ji-Rong Wen. 2025. Slow thinking for sequential recommendation. *Preprint*, arXiv:2504.09627.

Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Shiyao Wang, and 1 others. 2025a. Onerec technical report. *arXiv preprint arXiv:2506.13695*.

Guorui Zhou, Hengrui Hu, Hongtao Cheng, Huanjie Wang, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Lu Ren, Liao Yu, and 1 others. 2025b. Onerec-v2 technical report. *arXiv preprint arXiv:2508.20900*.

Yaochen Zhu, Harald Steck, Dawen Liang, Yinhan He, Vito Ostuni, Jundong Li, and Nathan Kallus. 2025. Rank-grpo: Training llm-based conversational recommender systems with reinforcement learning. *Preprint*, arXiv:2510.20150.

A Appendix

A.1 Implementation Details

We train the model for 2 epochs using a global batch size of 32, further divided into PPO mini-batches of size 128 and micro-batches of size 2 for gradient accumulation and memory efficiency.

Our base language model is Qwen2.5-3B-Instruct (Qwen et al., 2025), which we fine-tune using the Volcano Engine Reinforcement Learning for LLMs (Sheng et al., 2024). The methodology is model-agnostic and can be readily adapted to other instruction-tuned language models. To enable memory-efficient training, we employ Fully Sharded Data Parallel (FSDP). Gradient checkpointing and optimizer offloading are enabled to further reduce GPU memory consumption. The actor model is optimized using AdamW with a learning rate of 1×10^{-6}.

During policy rollouts, we generate responses using vLLM (Kwon et al., 2023) with the following sampling configuration: temperature $= 0.6$, top-$p = 0.95$, and $n = 16$ responses per prompt to ensure diverse reward estimation. For semantic matching in the reward model, we utilize the BGE-Large-Zh (Xiao et al., 2023) embedding model to compute semantic similarity between generated outputs and reference texts.

A.2 Details of Baseline Models

To comprehensively evaluate the performance of HiAgentRec, we compare it against a diverse set of representative baselines. These methods cover three distinct paradigms: traditional sequential recommendation, direct LLM-based approaches, and LLM-enhanced frameworks.

Traditional Sequential Recommendation These models rely on ID-based embeddings and deep neural networks to capture sequential patterns.

- **GRU4Rec** (Hidasi et al., 2016): A recommendation model that utilizes Gated Recurrent Units to model user interaction sequences.

- **SASRec** (Kang and McAuley, 2018): A self-attention based recommendation model that employs a unidirectional Transformer to capture long-term dependencies in user behavior.

- **BERT4Rec** (Sun et al., 2019): A sequential model based on the bidirectional Transformer

architecture. It utilizes the Cloze objective to learn bidirectional context representations.

LLM-based Recommendation These methods directly leverage LLMs for recommendation.

- **LLM (Base)**: The backbone model used with the hierarchical agentic reasoning. It predicts the next category directly based on the prompt without any parameter updates, serving as a baseline to assess the raw capability of the model.

- **LLM+SFT**: The backbone model fine-tuned via standard Supervised Fine-Tuning on the training data. It treats recommendation as a standard next-token prediction task without the agentic reasoning or reinforcement learning components.

LLM-Enhanced Recommendation These methods leverage LLMs to generate auxiliary knowledge signals, such as user profiles, item descriptions, or reasoning chains. These semantic signals are then encoded and injected into conventional sequential backbones to alleviate data sparsity and improve representation robustness.

- **KAR** (Xi et al., 2023): Knowledge Augmented Reasoning for recommendation. This method represents the LLM-as-Enhancer paradigm, which augments a BERT4Rec backbone with two distinct types of LLM-derived knowledge: (1) Factual Representation, obtained by encoding the semantic embeddings of explicit user living needs, and (2) Reasoning Representation, which captures the LLM's dynamic reasoning process as it infers user living need based on the current spatiotemporal context.

A.3 Details of dataset

The detailed statistics of both datasets are summarized in Table 4.

Table 4: Statistics of the Shanghai and Beijing Datasets.

Metric	Shanghai	Beijing
Total Users	10,302	11,889
Total Categories	422	408
Total Interactions	263,437	207,492
Avg. Sequence Length	25.57	17.45
Sparsity	93.94%	95.73%

Table 5: Performance comparison on Cold-Start users ($N = 1,943$ users with sequence length $= 2$). HiAgentRec demonstrates superior capability in handling unseen users. The best results are **bold**, and the second-best are underlined.

Model	HR@1	HR@3	HR@5	NDCG@3	NDCG@5
GRU4Rec	0.1450	0.2050	0.2501	0.1820	0.1591
SASRec	0.2676	0.2910	0.3139	0.2840	0.2933
BERT4Rec	0.2676	0.2820	0.2908	0.2750	0.3264
LLM (Base)	0.0420	0.0510	0.0610	0.0450	0.0502
LLM+SFT	0.0480	0.0650	0.0861	0.0520	0.0640
KAR	0.2950	0.3180	0.3310	0.3080	0.3450
HiAgentRec	**0.3759**	**0.4080**	**0.4260**	**0.3950**	**0.4288**

A.4 Performance on Cold-Start Users(RQ3)

To evaluate the effectiveness of our framework in cold-start scenarios, we construct a strictly defined cohort of 1,943 new users, each with only two historical interactions. We compare HiAgentRec against a range of baselines. As shown in Table 5, HiAgentRec achieves the best performance across all metrics, with an HR@5 of 0.4260 and NDCG@5 of 0.4288.

A.5 Can HiAgentRec preserve general capabilities during specialization?(RQ5)

Table 6: Accuracy of models on General Benchmark.

Benchmark	LLM	LLM+SFT	**HiAgentRec**
GSM8K	0.6808	0.5807	0.7157
MMLU	0.6656	0.6616	0.6644
IFEval	0.4122	0.4104	0.4270
MBPP	0.4633	0.4796	0.4776

We evaluate the performance of different models on general-purpose benchmarks to investigate whether our method forgets its general capabilities while specializing in a specific task. As shown in Table 6, we compare the base LLM, its supervised fine-tuned variant, and our proposed method HiAgentRec across four key dimensions: GSM8K (mathematical reasoning), MMLU (factual knowledge), IFEval (instruction following), and MBPP (code generation).

The results reveal a clear contrast in generalization behavior. LLM+SFT exhibits a notable performance drop on GSM8K, suggesting that conventional supervised fine-tuning may impair complex reasoning abilities. In contrast, HiAgentRec maintains strong performance across all benchmarks. These gains may stem from the structured prompting and consistency constraints inherent in our framework, which indirectly reinforce general rea-

soning patterns. In summary, HiAgentRec achieves task-specific specialization while retaining general purpose capabilities.

A.6 Theoretical Analysis(RQ6)

Real-world recommendation environments are characterized by stochastic user behaviors and sparse rewards. In this section, we analyze the instability of GRPO when lacking hierarchical guidance. Empirically, we observe that removing the *Living Need Inference* stage—forcing the model to predict semantic categories directly—leads to rapid policy collapse. We provide a theoretical derivation for this phenomenon and demonstrate how our hierarchical framework mitigates these risks via variance reduction.

Consider a policy π_θ optimized via GRPO. The gradient estimate for the objective function $J(\theta)$ is given by:

$$\nabla J \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t} A_t \cdot \nabla_\theta \log \pi_\theta(a_t|s_t) \quad (3)$$

In a sparse-reward setting without intermediate semantic anchors, the advantage term A_t exhibits extreme variance. When the environment yields a stochastic positive signal for a noisy action sequence a_{noise}, the gradient estimator lacks the inductive bias to distinguish valid reasoning from serendipitous noise. Consequently, the optimizer indiscriminately amplifies the probability of these shortcut patterns. As the policy overfits to these high-variance signals, the distribution converges prematurely to a deterministic state, characterized by vanishing entropy:

$$H(\pi_\theta) \to 0 \quad \Longrightarrow \quad \pi_\theta(a|s) \to \delta(a - a_{\text{degenerate}}) \quad (4)$$

This mathematical collapse manifests empirically as the generation of repetitive or nonsensical tokens ("gibberish"), as the model gets trapped in poor local optima.

HiAgentRec prevents this degeneration by decomposing the direct mapping $P(b|s)$ into a conditional probability chain: $P(b|c, i, s) \cdot P(c|i, s) \cdot P(i|s)$. This hierarchical structure leverages the *Law of Total Variance* to constrain the optimization space. The variance of the target prediction B given state S can be decomposed as:

$$\text{Var}(B|S) = \underbrace{\mathbb{E}[\text{Var}(B|C, I, S)]}_{\text{Intra-cluster Variance}} + \underbrace{\text{Var}(\mathbb{E}[B|C, I, S])}_{\text{Inter-cluster Variance}} \quad (5)$$

By explicitly supervising the *Living Need* (I) and *Category* (C) in early curriculum stages, we effectively minimize the second term (variance of the expectation). This transforms the optimization landscape: instead of searching in a high-entropy global space, the policy explores $P(b|c, i, s)$ within a significantly constricted manifold, acting as a semantic regularizer that prevents collapse into degenerate regions.

A.7 Semantic Model Comparison(RQ7)

As shown in Figure 5, adopting an approach without a semantic model yields low classification accuracy (approximately 0.39), where the raw output of LLM is directly matched against predefined business categories (e.g., Hong Kong and Taiwanese Desserts) via exact string matching. This limitation arises because LLMs tend to generate natural, descriptive, and diverse phrasings (e.g., a dessert shop specializing in Taiwanese taro balls and mango pomelo sago), whereas the ground-truth category system is highly structured and fine-grained (e.g., Hong Kong and Taiwanese Desserts, Sichuan Cuisine, Japanese Food). When the output does not explicitly contain the canonical category name, exact matching fails, even if the semantic meaning is fully aligned. Moreover, local service taxonomies often exhibit synonymy, near-synonymy, or hierarchical nesting (e.g., Desserts vs. Hong Kong and Taiwanese Desserts), further undermining the reliability of literal matching. In contrast, semantic embedding models map both LLM outputs and category labels into a shared dense vector space, enabling robust soft matching based on semantic similarity. Notably, different embedding models exhibit varying degrees of effectiveness.

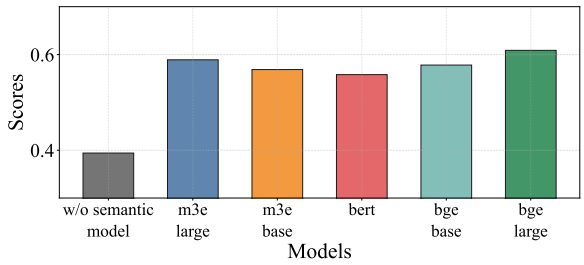

Figure 5: Comparison of Category Match Accuracy(Hit Rate@1) Across Different Semantic Models

A.8 Agentic Reasoning Prompt Specification

The complete prompt structure is illustrated in the box below:

Agentic Reasoning Pipeline

System Role:
You are an autonomous Recommendation Agent. Your objective is to formulate precise recommendations by orchestrating specific function tools to analyze user data, map semantic categories, and rank potential behaviors based on the current spatiotemporal context.

Step 1: Living need Inference

Agent Action:
Query the UserProfile MCP to retrieve long-term preferences. Combine with current Context (Time/Location) to filter candidate living needs.
Reasoning Requirement: Explain which profile feature triggered the selection of the living need from the candidate list.
Output format:

```
<intent>
{"predicted_intent": "Intent Name",
 "reasoning_summary": "Brief Reasoning"}
</intent>
```

Step 2: Category Mapping

Agent Action:
Call the Intent Parser to retrieve the predicted living need from Step 1.
Reasoning Requirement: Justify the category choice by linking the living need to specific domain availability. Semantic domains for categories:

- Food & Beverage (e.g., Chinese/Western cuisine...)
- Accommodation (e.g., luxury, budget hotels)
- Entertainment & Leisure
- Lifestyle Services (e.g., beauty, laundry)
- Grocery & Fresh Produce

Output format:

```
<category>
{"predicted_category": "Category Name",
 "reasoning_summary": "Brief Reasoning"}
</category>
```

Step 3: Behavior Ranking

Agent Action:
Call the Category Parser to retrieve the predicted category from Step 2 based on the semantic matching score.
Reasoning Requirement: Explain why this behavior ranks highest.
Output format:

```
<behavior>
{"predicted_behavior": "Behavior Name",
 "reasoning_summary": "Brief Reasoning"}
</behavior>
```