# Open-Set Living Need Prediction with Large Language Models

**Xiaochong Lan**[1,2] **Jie Feng**[1*] **Yizhou Sun**[1] **Chen Gao**[1]
**Jiahuan Lei**[2] **Xinlei Shi**[2] **Hengliang Luo**[2] **Yong Li**[1*]
[1]Department of Electronic Engineering, BNRist, Tsinghua University [2]Meituan
lanxc22@mails.tsinghua.edu.cn fengjie@tsinghua.edu.cn liyong07@tsinghua.edu.cn

## Abstract

Living needs are the needs people generate in their daily lives for survival and well-being. On life service platforms like Meituan, user purchases are driven by living needs, making accurate living need predictions crucial for personalized service recommendations. Traditional approaches treat this prediction as a closed-set classification problem, severely limiting their ability to capture the diversity and complexity of living needs. In this work, we redefine living need prediction as an open-set classification problem and propose PIGEON, a novel system leveraging large language models (LLMs) for unrestricted need prediction. PIGEON first employs a behavior-aware record retriever to help LLMs understand user preferences, then incorporates Maslow's hierarchy of needs to align predictions with human living needs. For evaluation and application, we design a recall module based on a fine-tuned text embedding model that links flexible need descriptions to appropriate life services. Extensive experiments on real-world datasets demonstrate that PIGEON significantly outperforms closed-set approaches on need-based life service recall by an average of 19.37%. Human evaluation validates the reasonableness and specificity of our predictions. Additionally, we employ instruction tuning to enable smaller LLMs to achieve competitive performance, supporting practical deployment.

## 1 Introduction

Living needs refer to the needs people generate in their daily lives for survival and well-being, from basic needs like food and accommodation to higher-level needs like socializing and hobbies. Living needs play a central role in life service platforms like Meituan[1], which serves as a bridge between millions of life service providers and nearly a billion users (Chen et al., 2022; Liu et al., 2022, 2025; Liang et al., 2024). On such platforms, user purchases are fundamentally driven by their underlying living needs - for example, when users order food delivery, it is typically motivated by their need for food. Therefore, accurate prediction of users' living needs is crucial for service recommendation, as it enables platforms to suggest services that directly address these needs, thereby improving recommendation accuracy and user experience.

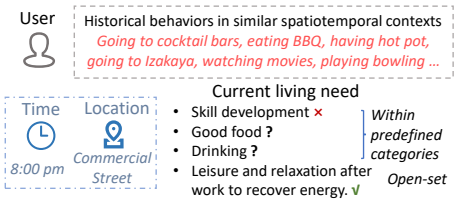



Figure 1: We define living need prediction as an open-set classification problem with unlimited classes to handle the complexity of living needs.

Living need prediction aims to predict a user's needs based on their spatiotemporal context. Previous studies have treated this as a **closed-set** classification problem, where models select from limited predefined need categories (Ping et al., 2021; Lan et al., 2023). However, this closed-set formulation has inherent limitations. First, users' needs are highly *diverse*, making it impossible to list all possibilities. Additionally, users' needs may be *vague*, such as seeking a relaxing atmosphere rather than specific venue types like restaurants or bars. Furthermore, users' needs can be *composite*, such as wanting a filling while also healthy meal. Obviously, predicting living needs within a fixed set of categories cannot handle these complexities.

In this work, we reformulate living need prediction as an open-set classification problem with unlimited classes, moving beyond the constraints of predefined categories (Zhang et al., 2021; Bao et al., 2022). Following insights that generative models show promise for open-set classification (Mundt et al., 2019) and building on recent advances (Zareian et al., 2021; Pham et al., 2024),

*Corresponding author.
[1] meituan.com



*Findings of the Association for Computational Linguistics: ACL 2025*, pages 5454–5472

we leverage large language models (LLMs) for this task. LLMs can generate unrestricted textual descriptions of living needs, naturally addressing the limitations of closed-set methods. Their extensive common sense knowledge enables deeper understanding of complex human needs. Through this approach, we aim to achieve truly open-set need prediction capable of identifying an unlimited range of living needs.

However, using LLMs for open-set living need prediction introduces additional challenges:

- First, LLMs lack understanding of individual user preference. While LLMs hold extensive general knowledge, they lack understanding of specific user preferences. To bridge this gap, we must provide LLMs with relevant user behavior data. However, achieving optimal prediction accuracy requires careful selection and providing an appropriate number of highly relevant historical records (Lin et al., 2024), making the design of effective retrieval methods crucial.
- Second, LLMs lack explicit knowledge of structured human living needs. While LLMs naturally support flexible need predictions, they sometimes produce descriptions that deviate from **needs**, like overly specific actions (e.g., *"eat a KFC burger"*). Aligning predictions with a structured living needs framework is challenging.
- Third, applying flexible need predictions in downstream tasks is difficult. The most typical application of need prediction is to recall life services that can fulfill a need. However, mapping a vast range of need types to the appropriate services is highly complex, making it challenging to model these associations effectively.

To tackle these challenges, we introduce our system, PIGEON (short for o**P**en set l**I**vin**G** n**E**eds predicti**ON**). For the first challenge, unlike traditional text embedding-based retrieval methods, we design a historical interactions-based encoder to help retrieve the most relevant historical information for the LLM to understand individual preferences. To address the second challenge, we guide the LLM to use Maslow's theory to organize platform services, creating a structured framework of users' living needs. The LLM then uses this structure to refine its initial predictions. For the third challenge, we develop a fine-tuning approach for a text embedding model, creating a recall module capable of understanding complex relationships between flexible living needs and vast life services. Additionally, to make our method deployable on platforms, we design an instruction tuning approach that enables smaller LLMs to achieve competitive performance.

We compare our method with closed-set classification approaches and LLM-based baselines on recalling appropriate life services, and the experimental results demonstrate the effectiveness of our approach. Human evaluation confirms that our method generates predictions with superior reasonableness and specificity compared to closed-set methods. Cases illustrate that our method can accurately and comprehensively predict users' needs, handling vague and composite living needs. Ablation studies further validate the contributions of each component in our design.

In summary, the main contributions of our work are as follows:

- To the best of our knowledge, we are the first to explore the open-set living need prediction problem, which is a significant yet under-addressed challenge on life services platforms.
- We propose PIGEON, an LLM-based open-set living need prediction system, which achieves state-of-the-art performance.
- We design a recall approach for open-set needs and a fine-tuning method that enables smaller LLMs to achieve competitive performance, supporting the practical deployment of our method in online environments.

## 2 Problem Formulation

Living need prediction aims to predict a user's living need at a given time and location. This prediction enhances user understanding and supports applications like life service recall. The ground-truth labels for this prediction task are derived from historical platform records, where user reviews sometimes reveal the living needs which drive the consumption. For example, a user might write, *"I came here for a quick bite and found it quite good."* Here, the user's need can be identified as *"a quick bite."* Extracting and clustering such need descriptions, then refining them with expert input, allows us to establish a set of need categories. Traditionally, this set remains limited to enable living need classification. Formally, each historical consumption instance is represented by a five-tuple $(u, t, l, n, s)$, where $u$ is the user, $t$ the time, $l$ the location[2], $n$ the need type, and $s$ the consumed service. Thus, our problem is formulated as follows:

- **Input**: $\mathcal{O}^+ = \{(u, t, l, n, s) \mid u \in U, t \in T, l \in L, n \in N, s \in S\}$, where $U, T, L, N$, and $S$

[2]In this work, location is in the form of functional zones.

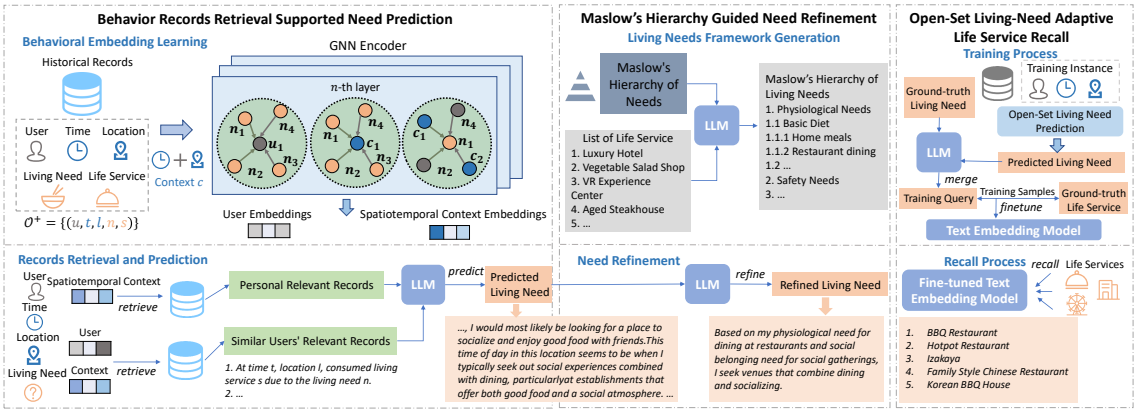



Figure 2: Architecture of our PIGEON system for open-set living need prediction and supporting recall module. First, a GNN encoder learns behavioral embeddings for users and spatiotemporal contexts, enabling retrieval of relevant historical records for LLM-based need prediction. To refine these predictions, the LLM incorporates Maslow's theory with platform services, creating a framework that aligns initial predictions with human needs. Additionally, a text embedding model is fine-tuned to link flexible need descriptions to relevant life services. To further support online deployment, the initial prompts to LLMs and their corresponding ground-truth-corrected predictions form instruction tuning pairs, which are used to fine-tune smaller LLMs.

denote the sets of users, time points, locations, need types, and life services, respectively.

- **Output**: A model that, given $u$, $t$, and $l$, can infer the user's type of living need as an unrestricted need description, covering limitless types.

## 3 Methods

In this section, we provide a detailed description of each component of our method. Figure 2 illustrates our proposed open-set living need prediction system along with the recall method that supports open-set need descriptions. The specific design of each component in our method is explained below.

### 3.1 Behavior Records Retrieval Supported Living Need Prediction

We aim to achieve open-set living need prediction with LLMs. Although LLMs possess extensive common sense, they lack specific knowledge of individual users. While explicit user profiles could provide such knowledge, they are difficult to obtain as they rely on users' active input. Instead, we leverage behavioral data that is naturally generated through user interactions. We augment LLM prediction with in-context learning from retrieved historical records (Luo et al., 2024; Wang and Lim, 2024), to help the LLM obtain knowledge about the specific user. Instead of traditional text embedding-based retrieval which only captures general text-level similarity, we propose behavioral embeddings learned from users' consumption histories. These embeddings directly encode individual preferences and contextual effects, enabling more precise retrieval of records for personalized predictions.

**Behavioral Embedding Learning** To encode the multiple relationships among users, spatiotemporal contexts, and needs, we use graph neural network (GNN) for embedding learning. First, we construct the graph. There are three types of nodes in the graph: user nodes $u$, spatiotemporal context nodes $c$, and living need nodes $n$. Since time and location jointly influence users' living needs, we combine them into unified spatiotemporal context nodes $c$. The graph has two types of weighted edges: user-need edges $(u, n)$ and context-need edges $(c, n)$. For each consumption record where user $u$ generates need $n$ in context $c$, we increment the weights of both $(u, n)$ and $(c, n)$ by 1.

Next, we describe the representation learning method on the graph. We first randomly initialize embeddings $\mathbf{e}_u^{(0)}$, $\mathbf{e}_c^{(0)}$, and $\mathbf{e}_n^{(0)}$ for users, contexts, and needs respectively. For all kinds of nodes, the propagation on the graph is as follows:

$$\mathbf{e}_i^{(k+1)} = \sum_{j \in \mathcal{N}(i)} \frac{w_{ij}}{D_i \cdot D_j} \mathbf{e}_j^{(k)},$$

where $\mathbf{e}_i^{(k)}$ is node $i$'s embedding at layer $k$, $w_{ij}$ is the weight of edge $(i, j)$, and $\mathcal{N}(i)$ denotes the set of neighboring nodes of $i$. $D_i = \sqrt{\sum_{j' \in \mathcal{N}(i)} w_{ij'}}$ and $D_j = \sqrt{\sum_{i' \in \mathcal{N}(j)} w_{i'j}}$ are normalization factors. After $K$ propagation layers, we obtain the final embedding through layer combination and normalization:

$$\mathbf{e}_i = \frac{\sum_{k=0}^{K} \mathbf{e}_i^{(k)}}{\left\| \sum_{k=0}^{K} \mathbf{e}_i^{(k)} \right\|}$$

To train the GNN encoder, we construct a Bayesian Personalized Ranking (BPR) loss:

$$\mathcal{L}_{\text{BPR}} = -\sum_{(u,c,n,n')\in\mathcal{D}} \ln \sigma\left(\mathbf{e}_{uc}^{\top}\mathbf{e}_n - \mathbf{e}_{uc}^{\top}\mathbf{e}_{n'}\right),$$

where $\sigma$ is the sigmoid function, $\mathbf{e}_{uc} = \mathbf{e}_u + \mathbf{e}_c$, and $\mathcal{D}$ represents the set of training instances $(u, c, n, n')$, with $n$ as the positive need generated by user $u$ in context $c$ and $n'$ a randomly sampled negative need. Finally, after supervised training on historical records, we obtain embeddings for users and spatiotemporal contexts that effectively capture user preferences and contextual influences.

**Records Retrieval and Prediction** Next, we utilize an LLM to predict the living need of a user in a given spatiotemporal context, based on retrieved most relevant behavioral records. While a user's own historical records are essential for personalization, relying solely on individual history has limitations: users may have unexplored needs that they haven't yet acted upon, or they may be in novel situations where relevant personal records are limited. To address these limitations, we complement individual records with historical data from users who exhibit similar behavioral patterns.

For user $u$ in context $c$, we first retrieve personal relevant records. From the user's entire historical record, we select the $K_p$ historical instances with context embeddings $\mathbf{e}_{c,i}$ most similar to the current context embedding $\mathbf{e}_c$. For the relevant historical records of similar users, we compute $\mathbf{e}_{uc} = \mathbf{e}_u + \mathbf{e}_c$ and retrieve $K_s$ historical records with embeddings $\mathbf{e}_{uc,i}$ most similar to the current $\mathbf{e}_{uc}$, from all history records. The similarity is measured using cosine similarity, for example, $\text{sim}(\mathbf{e}_{c,i}, \mathbf{e}_c) = \frac{\mathbf{e}_{c,i}^{\top}\mathbf{e}_c}{\|\mathbf{e}_{c,i}\|\|\mathbf{e}_c\|}$. Using the retrieved historical records, we then employ the LLM to predict the user's needs in the current spatiotemporal context. The specific prompt is as follows:

> *You are a user on a life service platform. At [time] in [location], what kind of living need are you most likely to have?*
> *1. Your following past behaviors are provided for reference: [Personal Relevant Records]*
> *2. Behaviors of other users in similar contexts are: [Similar Users' Relevant Records]*
> *Considering the current time, location, and preferences indicated by previous consumption behaviors, please infer and describe your potential living needs.*

Here, the personal relevant records and similar users' relevant records are provided to the LLM in textual form, including information on time, location, type of living need, and the consumed life service. For example: *"At 5 PM at home, you generated the living need for a quick meal and ordered a hamburger delivery service."* Using this prompt as input, the LLM outputs a prediction of living needs expressed in an unrestricted text format.

### 3.2 Maslow's Hierarchy Guided Living Need Refinement

Now, we have obtained a flexible text-based prediction of living needs. While the LLM allows for flexible predictions, some outputs may not align well with living **needs**. For instance, certain predictions may focus excessively on specific actions rather than genuine needs, such as *"eating a McDonald's burger"*, or suggest needs unsupported by life service platforms, like *"writing emails"*. Since living needs on service platforms naturally range from basic physiological needs to higher-level social and self-development needs, we leverage the widely-accepted Maslow's hierarchy (McLeod, 2007) as a theoretical foundation to guide the refinement of these predictions. Specifically, we use the LLM to integrate Maslow's hierarchy with the platform's service offerings, creating a structured framework that helps align predictions with genuine human living needs while preserving their flexibility. In Section A.8, we further explain the rationale for using Maslow's theory.

To construct this framework, we prompt the LLM with a list of available life services:

> *[Life Services List]*
> *These are life services on a platform that can meet human living needs. Based on this list, along with Maslow's hierarchy of needs, please generate a three-tiered framework of human living needs, ensuring that each need can be fulfilled by the listed services.*

For large-scale service lists in the production environment that exceed LLM's token limit, we either categorize services first or process them in sequential batches. Using the generated framework, we then refine the initial predictions:

> *You are a user on a life service platform. An inference about your current living need at [time] in [location] is [initial prediction].*

*Please use the following human living needs framework to further refine this inference, making it align with the framework's scope. Your response should be concise and as informative as possible, around 20 words.*
*[Human living needs framework]*

### 3.3 Open-Set Living Need-Adaptive Life Service Recall

In practical applications, predicted needs are typically used as one source in the recall stage to identify suitable life services as candidate recommendations. For closed-set methods, associations between needs and services rely on manually set rules. However, the flexibility of LLM-generated needs poses challenges for recall, as managing the complex mapping between a limitless range of needs and the services that fulfill them is nearly impossible. To address this, we design a recall approach based on a fine-tuned text embedding model, leveraging the commonsense knowledge embedded within language models to enhance recall accuracy.

In our recall method, need descriptions serve as queries to identify relevant services within the whole corpus of services. To fine-tune the text embedding model, we construct training samples from historical data by pairing actual needs with the ordered life services as ground truth. To adapt the model to flexible need descriptions as queries, we generate flexible living needs predictions for historical records and then use the LLM to refine these predictions based on the ground-truth closed-set living needs $n$. The specific prompt is:

*The ground-truth living need is [living need]; a flexible need description is: [predicted need]. Based on the ground-truth living need, please revise the flexible need description to maintain flexibility without compromising accuracy. Provide only the revised flexible need description, approximately 20 words.*

Now we obtain flexible yet accurate need descriptions to serve as training queries for recall. Each refined need description $q$ is encoded as a vector embedding $f(q)$ using a pretrained transformer-based text embedding model $f(\cdot)$. The training process optimizes the model with a triplet loss:

$$\mathcal{L} = \sum_{(q,s,s')\in\mathcal{D}} \max\left(0, \text{sim}(q,s') - \text{sim}(q,s) + \alpha\right)$$

where $\text{sim}(q,s) = \frac{f(q)^\top f(s)}{\|f(q)\|\|f(s)\|}$ is the cosine similarity between the query $q$ and the positive life service $s$ (the actual consumed service), $s'$ is a randomly selected negative service, and $\alpha$ is a margin. This loss encourages the model to embed services that can fulfill the need closer to the query than those that cannot fulfill the need, enhancing recall precision. During inference, we use the fine-tuned model $f_{\text{finetune}}(\cdot)$ to encode both queries and services, recalling services with highest similarities.

### 3.4 Domain Adaptation for LLMs

Practical deployment of our method faces computational and latency constraints, requiring high performance under limited computing resources and low latency for online deployment. In our current framework, LLM inference represents the primary computational and latency bottleneck. Therefore, we employ instruction tuning to enhance the capabilities of smaller LLMs that have lower computational requirements and latency. Our method naturally defines input-output pairs: the input being the initial prompt containing spatiotemporal context and user history, and the output being the refined training query that balances both accuracy and flexibility. After processing numerous prediction tasks with high-performance LLMs, we naturally accumulate these pairs as instruction tuning data. Using this constructed training data, we fine-tune smaller LLMs. Notably, we adopt an end-to-end training approach, aiming to train LLMs that can achieve good results in a single inference step using data generated through the two-step framework. We implement full-parameter fine-tuning using accelerate (Gugger et al., 2022).

## 4 Experimental Setup

### 4.1 Datasets

We collect two large-scale datasets from the Meituan platform, each containing 120 days of user consumption records from a different city in China. These records include user details, timestamps, locations, actual life service consumption, and living needs extracted from reviews. In these datasets, time is categorized into 48 intervals of half-hour windows, and locations are grouped into 13 categories based on functionality (e.g., residential areas, commercial streets). Following (Li et al., 2022b), we partition the datasets by time, using the first 96 days for training, the next 12 days for validation, and the final 12 days for testing. Dataset

| Dataset | #Records | #Users | #Locs | #Time | #Needs | #Services |
|---|---|---|---|---|---|---|
| **Beijing** | 6,942,123 | 37,226 | 14 | 48 | 20 | 2026 |
| **Shanghai** | 7,797,225 | 43,213 | 14 | 48 | 20 | 2155 |

Table 1: Statistics of our utilized datasets.

statistics are provided in Table 1.

### 4.2 Implementation Details

In our main experiments, we primarily use GPT-4o mini (OpenAI, 2024) as the LLM backbone for our method, chosen for its ease of use, relatively high performance, and cost-effectiveness. The cost is around $1.05 for 10,000 times of predictions. To maximize reproducibility, we set the temperature parameter to 0. We also report performance with other LLMs as backbones. For the text embedding model used as the recall module, we adopt bge-base-v1.5 (Xiao et al., 2023). We implement the GNN encoder using the PyTorch (Paszke et al., 2019) framework. We implement graph propagation efficiently using sparse matrix operations. Each reported result is the average of 5 runs. [3]

### 4.3 Evaluation Metric

Evaluating flexible need predictions poses inherent measurement challenges due to their open-ended nature. We address this by assessing prediction quality through the downstream task of life service recall, where need predictions serve as queries to retrieve relevant services. For quantitative evaluation, we adopt two widely used metrics: Recall@K and NDCG@K. Recall@K is defined as the proportion of actually consumed services that appear in the top K recalled results. NDCG@K further emphasizes that ground-truth services should be ranked high in the recalled list. While real-world recall systems focus less on NDCG, we include it here to better evaluate our recall system's ability to identify services that precisely meet user needs.

### 4.4 Comparison Methods

For life service recall, we compare our method against two types of methods: traditional closed-set living need prediction methods and LLM-based open-set need prediction methods. For closed-set methods, we include industry-standard CTR models: XGBoost (Chen and Guestrin, 2016), DeepFM (Guo et al., 2017), DCN (Wang et al., 2017), XDeepFM (Lian et al., 2018), EulerNet (Tian et al., 2023), which take user, time, and location as input features. We also evaluate graph-based models: LightGCN (He et al., 2020), HAN (Wang et al., 2019), DisenHCN (Li et al., 2022c) that capture complex relationships through graph incorporating user, time, location, need, and service nodes and propagation on the graph. To maximize the recall performance potential of closed-set need prediction methods, we utilize combinations of closed-set needs with time and locations as queries during both the training and prediction phases of the recall model.

For LLM-based open-set methods, we test zero-shot CoT (Kojima et al., 2022), which predicts solely from current context without historical data. We also include two retrieval-enhanced approaches: ReLLa (Lin et al., 2024) and LLMSRec-Syn (Wang and Lim, 2024). Both approaches leverage text embeddings for historical record retrieval, where ReLLa retrieves from individual user records while LLMSRec-Syn additionally incorporates records from users with similar behaviors. For both LLM-based methods, the recall model's training and usage are consistent with our approach, differing only in flexible need generation. A more detailed description of comparison methods is in Section A.6.

## 5 Experimental Results

We conducted extensive experiments to address the following research questions:

- **RQ1:** How is the performance of our open-set need prediction method on life service recall?
- **RQ2:** How is the quality of living need expressions generated by our method?
- **RQ3:** Does each component of our method contribute to the final performance in service recall?
- **RQ4:** How does domain adaptation improve the performance of smaller open-source models?
- **RQ5:** How does our model perform on different LLM backbones?
- **RQ6:** How do hyperparameter settings impact the performance of our method?

Due to space constraints, we present the experimental results for RQ6 in the Appendix.

### 5.1 Recall Performance (RQ1)

We measure the performance of PIGEON and baselines in the life service recall task on two datasets, as shown in Table 2. Key findings are as follows:

- **Our method achieves top performance across all metrics on both datasets.** On Beijing and

[3] Code is available at https://github.com/tsinghua-fib-lab/PIGEON

Table 2: Comparison of PIGEON and baselines on the life service recall task. Bold and underlined refer to the best and 2nd best performance. * indicates that PIGEON outperforms the best baseline with $p < 0.05$ in the paired t-test.

| Category | Model | Beijing | | | | Shanghai | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Recall@10 | Recall@20 | NDCG@10 | NDCG@20 | Recall@10 | Recall@20 | NDCG@10 | NDCG@20 |
| Closed-Set | XGBoost | 0.03106 | 0.05878 | 0.01354 | 0.02032 | 0.02164 | 0.04774 | 0.00929 | 0.01600 |
| | DeepFM | 0.03378 | 0.06486 | 0.01546 | 0.02319 | 0.06047 | 0.09230 | 0.02903 | 0.03710 |
| | DCN | 0.03176 | 0.06351 | 0.01558 | 0.02356 | 0.06122 | 0.09377 | 0.03018 | 0.03862 |
| | XDeepFM | 0.03331 | 0.06552 | 0.01623 | 0.02410 | 0.06203 | 0.09448 | 0.03081 | 0.03971 |
| | EulerNet | 0.03497 | 0.06871 | 0.01687 | 0.02485 | 0.06343 | 0.09615 | 0.03109 | 0.03927 |
| | LightGCN | 0.05510 | 0.09410 | 0.02624 | 0.03594 | 0.04375 | 0.07701 | 0.01939 | 0.02785 |
| | HAN | 0.08851 | 0.12973 | 0.04532 | 0.05574 | 0.06111 | 0.09612 | 0.02745 | 0.03620 |
| | DisenHCN | 0.09244 | 0.13347 | 0.04820 | 0.05874 | 0.07268 | 0.12335 | 0.03474 | 0.04377 |
| Open-Set | Zero-shot CoT | 0.03446 | 0.05743 | 0.01748 | 0.02330 | 0.02355 | 0.04710 | 0.01089 | 0.01680 |
| | ReLLa | 0.05878 | 0.10405 | 0.03029 | 0.04150 | 0.05347 | 0.09612 | 0.02933 | 0.03987 |
| | LLMSREC-Syn | 0.08044 | 0.12705 | 0.04421 | 0.06012 | 0.06317 | 0.10025 | 0.03738 | 0.04792 |
| | PIGEON | **0.10405*** | **0.16622*** | **0.05124*** | **0.06681*** | **0.10503*** | **0.14895*** | **0.04743*** | **0.05845*** |

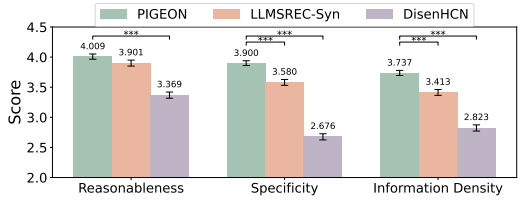



Figure 3: Human evaluation results. *** indicates $p < 0.001$ in statistical significance test.

Shanghai datasets, our method significantly outperforms the best baseline across all metrics, with relative improvements of 15.18% and 23.55%, respectively. This highlights that our flexible need predictions effectively boost life service recall, leading to more accurate recommendations.

- **The approach for historical record retrieval is crucial.** Among open-set methods, zero-shot CoT shows the lowest performance, indicating that common sense alone is insufficient for need prediction and that historical data is crucial. ReLLa and LLMSREC-Syn, which use text embeddings to retrieve records, also underperform compared to our method, underscoring the effectiveness of our behavior-based representation learning.
- **Graph-based approaches lead in closed-set methods.** Graph neural networks model complex relationships and cross-influences better than CTR models, which rely on feature interactions. This finding supports the suitability of GNN encoding in our approach.

## 5.2 Quality of Generated Prediction (RQ2)

To assess the quality of predictions generated by PIGEON, we conduct a human evaluation with 116 domain experts, who rate predictions from PIGEON, LLMSREC-Syn, and DisenHCN on reasonableness, specificity, and information density. For fair comparison, LLM outputs are constrained to approximately 10 words. Details of the questionnaire design are in Appendix Section A.4.

Results of human evaluation are shown in Figure 3. Statistical analysis shows that PIGEON generates high-quality predictions across all dimensions. While PIGEON and LLMSREC-Syn achieve comparable rationality scores (4.009 vs. 3.901, $p = 0.103$), PIGEON significantly outperforms in both specificity (3.900 vs. 3.580, $t = 5.107$, $p < 0.001$) and information density (3.737 vs. 3.413, $t = 5.062$, $p < 0.001$). Compared to DisenHCN, PIGEON shows substantial improvements across all metrics ($p < 0.001$), with particularly large gains in specificity ($t = 18.852$) and information density ($t = 13.737$).

Notably, LLM-based methods provide richer and more comprehensive need expressions that participants find more specific and actionable, while maintaining high information density despite longer descriptions. In contrast, closed-set predictions, constrained by predefined categories, often appear too generic for specific contexts.

What's more, case studies in Figure 4 demonstrate how our LLM-based approach addresses closed-set methods' limitations in capturing diverse, vague, and composite human needs.

In the first case, users ordered Brazilian barbecue, Japanese izakayas, and cocktail bars around 8 PM in commercial street. While closed-set methods categorize this as *"good food"*, PIGEON recognizes the after-work context and infers a vague need: *"relaxation and entertainment after work, to unwind and recharge"*.

In the second case, a user ordered food delivery and fresh produce around 12 PM in a residential area with gathering-related comments. PIGEON infers a composite need combining physiological

| Case 1: Vague Living Needs |
|---|
| Time: 8:00 PM, Location: Commercial Street<br>Personal historical records:<br>• At 10:00 PM, ordered Brazilian BBQ on Commercial Street due to the living need for good food.<br>• At 9:00 PM, ordered Japanese Izakaya on Commercial Street due to the living need for good food.<br>• At 8:00 PM, ordered cocktail bar in Shopping Mall due to the need for a group gathering.<br>…<br>Similar users' historical records:<br>• At 7:00 PM, ordered self-service BBQ on Commercial Street due to the living need for good food.<br>• At 10:30 PM, ordered a movie in Shopping Mall due to the living need for having fun.<br>• …<br>Prediction of PIGEON: Physiological need, leisure and relaxation after work to recover energy.<br>Prediction of DisenHCN: Good food. |
| **Case 2: Composite Living Needs** |
| Time: 12:00 PM, Location: Residential Area<br>Personal historical records:<br>• At 12:00 PM, ordered hotpot delivery in the residential area due to the living need for good food.<br>• At 11:30 AM, ordered fresh supermarket delivery in the residential area due to the living need for a group meal.<br>• At 12:00 PM, ordered fresh supermarket delivery in the residential area due to the living need for cooking at home.<br>• …<br>Similar users' historical records:<br>• At 12:00 PM, ordered burger delivery in the residential area due to the living need for good food.<br>• At 12:00 PM, ordered fresh supermarket delivery in the residential area due to the living need for cooking at home.<br>• …<br>Prediction of PIGEON: Physiological need, lunch; Social need, eating with family at home.<br>Prediction of DisenHCN: Cooking at home |

Figure 4: Cases show that our model can generate flexible living need descriptions covering vague and composite needs, while closed-set classification models cannot.

(eating lunch) and social (family gathering) aspects, while the closed-set model DisenHCN only predicts a single need like "cooking at home".

These cases show our method effectively handles vague and composite needs, demonstrating clear advantages over closed-set classification methods.

### 5.3 Ablation Study (RQ3)

To assess each component's impact on recall performance, we individually remove parts of PIGEON and present results in Table 3 and 4. The key findings are as follows:

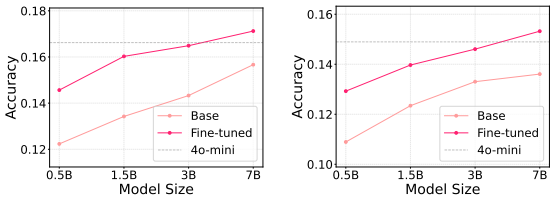
- **Behavior Records Retrieval.** Removing either personal or similar users' historical record retrieval leads to a marked drop in performance, with both types of records showing a similar degree of impact. What's more, omitting all historical records causes an even more substantial decrease, highlighting the necessity of using actual behavior data in addition to general LLM common sense.
- **Maslow Hierarchy-Guided Need Refinement.** Omitting the refinement process and relying only on the LLM's initial prediction substantially reduces performance, highlighting the importance of aligning predictions with human needs.
- **Recall Model Fine-tuning.** In our fine-tuning design, to create training queries that are both flexible and accurately aligned with life services, we first generate flexible need descriptions, then refine them using closed-set ground-truth needs from historical data. Our ablation experiments show performance drops when using either only closed-set needs or unrefined flexible needs, while using the original text embedding model without fine-tuning performs even worse. These results validate both our query construction approach and the necessity of fine-tuning for effective service recall.

Table 3: Results of ablation study on Beijing dataset.

| Model | Beijing | |
|---|---|---|
| | Recall@10 | Recall@20 |
| PIGEON | 0.10405 | 0.16622 |
| w/o history record | 0.02770 | 0.05743 |
| w/o personal history | 0.07838 | 0.13919 |
| w/o similar users' history | 0.07808 | 0.13514 |
| w/o need refinement | 0.09324 | 0.14932 |
| w/o recall model fine-tuning | 0.07095 | 0.11486 |
| w/o flexible needs in queries | 0.07500 | 0.12027 |
| w/o closed-set needs in queries | 0.09403 | 0.15176 |

Table 4: Results of ablation study on Shanghai dataset.

| Model | Shanghai | |
|---|---|---|
| | Recall@10 | Recall@20 |
| PIGEON | 0.10503 | 0.14895 |
| w/o history record | 0.02673 | 0.04838 |
| w/o personal history | 0.07257 | 0.11776 |
| w/o similar users' history | 0.06875 | 0.11585 |
| w/o need refinement | 0.09039 | 0.14131 |
| w/o recall model fine-tuning | 0.07257 | 0.10948 |
| w/o flexible needs in queries | 0.07731 | 0.11377 |
| w/o closed-set needs in queries | 0.09322 | 0.14438 |




(a) Beijing (b) Shanghai

Figure 5: Performance (Recall@20) of small LLMs on living need prediction after domain adaptation.

### 5.4 Domain Adaptation of LLMs (RQ4)

To make our system practically deployable, we design an instruction tuning approach aimed at enabling smaller LLMs to achieve usable performance, thereby reducing computational costs and latency in online deployment. We conduct experiments using Qwen2.5 series models (0.5B, 1.5B, 3B, 7B) as base models, with results shown in Figure 5. We find that through training, the models show significant performance improvements, with 3B models approaching the capabilities of GPT-4o mini, while the 7B model surpasses it. This demonstrates the effectiveness of our instruction tuning method. What's more, smaller LLMs can achieve competitive results with single-step inference, making our method more feasible for deployment.

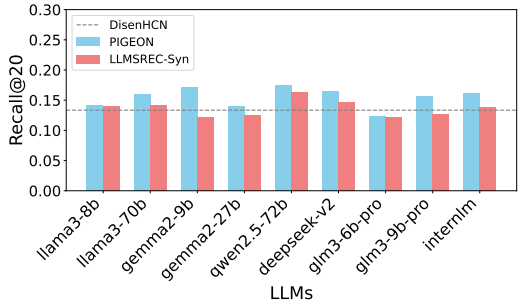



Figure 6: The performance of PIGEON and LLMSREC-Syn on various LLM backbones on the Beijing Dataset.

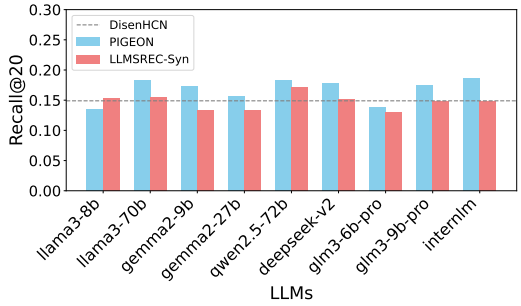



Figure 7: The performance of PIGEON and LLMSREC-Syn on various LLM backbones on the Shanghai dataset.

### 5.5 Performance on different LLMs (RQ5)

In our experiments, we use GPT-4o mini as the main backbone of PIGEON. In this section, we test PIGEON's performance with several open-source LLMs to evaluate its effectiveness across these models. We also compare our method with LLMSREC-Syn, which uses a text embedding model to vectorize user and context for record retrieval. The results on the Beijing dataset are shown in Figure 6 and 7, focusing on Recall@20.

The results indicate that even with smaller-scale open-source models, our method performs well, achieving performance comparable to or better than the best closed-set classification models. More importantly, our method outperforms the direct text embedding-based retrieval approach across all open-source models, demonstrating the effectiveness of our design.

## 6 Related Work

### 6.1 Open-Set Classification

Open-set classification extends traditional closed-set classification by handling scenarios where the model encounters unseen classes (Scheirer et al., 2014). Early methods used confidence thresholding to reject unfamiliar inputs (Bendale and Boult, 2016). Recent approaches incorporate generative models (Mundt et al., 2019) and adversarial learning (Zhang et al., 2021; Bao et al., 2022) for enhanced robustness. While recent advances in open-vocabulary object detection have shown promising results in handling unconstrained output spaces through vision-language models (Zareian et al., 2021; Gu et al., 2021; Li et al., 2022a; Pham et al., 2024), extending beyond closed-set formulation remains unexplored for living need prediction. We address this by leveraging LLMs to generate flexible need descriptions without class constraints.

### 6.2 Domain Applications of LLMs

In recent years, LLMs have flourished and been applied to various domains. Successful application areas include natural sciences (Li et al., 2025), social sciences (Gao et al., 2023, 2024; Li et al., 2023a; Piao et al., 2025; Wang et al., 2024b), urban science (Feng et al., 2025b,d,c), social media analysis (Lan et al., 2024a,b), reinforcement learning (Hao et al., 2025a,b; Xu et al., 2025), literature analysis (Hao et al., 2024), and graph learning (Li et al., 2023b,c; Zhang et al., 2024). However, these methods primarily leverage the world knowledge of LLMs, while the flexible text generation capabilities of LLMs have not been fully utilized and may even become obstacles in structured tasks. In contrast, we effectively leverage both the commonsense and flexible text generation capabilities of LLMs to achieve better human need prediction.

### 6.3 LLMs for User Modeling

User modeling focuses on understanding users' behaviors and intentions to provide personalized experiences (Feng et al., 2025a).LLMs have advanced this field through its commonsense knowledge, enabling success across tasks including profiling, recommendation, and behavior detection (Wang et al., 2024a; Liu et al., 2023). However, no existing research has utilized LLMs for predicting user living needs, making our work pioneering in this domain.

## 7 Conclusion and Future Work

In this study, we propose PIGEON, a novel open-set living need prediction system that leverages LLMs to generate flexible need descriptions unrestricted by pre-defined categories. Experimental results demonstrate that PIGEON outperforms both traditional closed-set and baseline LLM-based methods on need-based life service recall. Human evaluation validates the reasonableness and specificity of our generated need descriptions. Future work will focus on enhancing PIGEON's efficiency for large-scale industrial applications.

## 8 Limitations

While our approach demonstrates strong performance in open-set living need prediction, several limitations warrant discussion. First, the computational overhead of LLM inference operations poses scalability challenges for large-scale deployment, particularly in real-time scenarios requiring low latency. Second, our evaluation focused primarily on two cities within a single life service platform; broader validation across diverse geographical regions and service platforms would help establish the method's generalizability. Third, while our approach extends open-set classification beyond traditional boundaries, it may still be constrained by the LLM's ability to generate contextually appropriate need descriptions. Finally, the quality of predictions may vary depending on the choice of LLM backbone, as shown in our experiments with different models.

## 9 Ethics Statement

This work involves processing user behavioral data to predict living needs. We ensure all data is anonymized through multiple safeguards: user identifiers are replaced with random codes, locations are aggregated into several functional zones, and timestamps are discretized into 30-minute intervals. We further protect user privacy by removing personal identifiers from review text and filtering out unique behavioral sequences that could enable individual identification. No personally identifiable information is used in the study or retained in our datasets.

The potential risks of behavioral data analysis are significant: malicious actors could potentially use predicted needs to manipulate user choices, create targeted advertising that exploits vulnerabilities, or combine predictions with other data sources to de-anonymize users. Commercial entities might attempt to leverage these predictions for aggressive marketing or price discrimination. There's also a risk of reinforcing existing biases in user behavior or creating filter bubbles that limit user exposure to diverse services. While we have implemented data obfuscation and anonymization to help reduce these risks, continuous ethical oversight remains necessary in practical applications.

The LLM components of our system utilize the GPT-4o mini API service provided by OpenAI, in full compliance with OpenAI's terms of service and usage policies. We maintain strict data handling protocols throughout our research to prevent potential privacy breaches or misuse.

## 10 Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under grant 2024YFC3307603, in part by the China Postdoctoral Science Foundation under grant 2024M761670 and GZB20240384, in part by the Tsinghua University Shuimu Scholar Program under grant 2023SM235. This work was also supported by Meituan.

## References

Wentao Bao, Qi Yu, and Yu Kong. 2022. Opental: Towards open set temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2979–2989.

Abhijit Bendale and Terrance E Boult. 2016. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572.

Huanhuan Cao, Jinhu Jiang, Lih-Bin Oh, Hao Li, Xiuwu Liao, and Zhiwu Chen. 2013. A maslow's hierarchy of needs analysis of social networking services continuance. *Journal of Service Management*, 24(2):170–190.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Zhilong Chen, Xiaochong Lan, Jinghua Piao, Yunke Zhang, and Yong Li. 2022. A mixed-methods analysis of the algorithm-mediated labor of online food deliverers in china. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW2):1–24.

Jie Feng, Yuwei Du, Jie Zhao, and Yong Li. 2025a. Agentmove: A large language model based agentic framework for zero-shot next location prediction. In *Proceedings of the 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*.

Jie Feng, Tianhui Liu, Yuwei Du, Siqi Guo, Yuming Lin, and Yong Li. 2025b. Citygpt: Empowering urban spatial cognition of large language models. *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Jie Feng, Jinwei Zeng, Qingyue Long, Hongyi Chen, Jie Zhao, Yanxin Xi, Zhilun Zhou, Yuan Yuan, Shengyuan Wang, Qingbin Zeng, et al. 2025c. A survey of large language model-powered spatial intelligence across scales: Advances in embodied agents, smart cities, and earth science. *arXiv:2504.09848*.

Jie Feng, Jun Zhang, Tianhui Liu, Xin Zhang, Tianjian Ouyang, Junbo Yan, Yuwei Du, Siqi Guo, and Yong Li. 2025d. Citybench: Evaluating the capabilities of large language models for urban tasks. *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. 2024. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications*, 11(1):1–24.

Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2023. S3: Social-network simulation system with large language model-empowered agents. *arXiv preprint arXiv:2307.14984*.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. 2021. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*.

Qianyue Hao, Jingyang Fan, Fengli Xu, Jian Yuan, and Yong Li. 2024. Hlm-cite: Hybrid language model workflow for text-based scientific citation prediction. *arXiv preprint arXiv:2410.09112*.

Qianyue Hao, Sibo Li, Jian Yuan, and Yong Li. 2025a. Rl of thoughts: Navigating llm reasoning with inference-time reinforcement learning. *arXiv preprint arXiv:2505.14140*.

Qianyue Hao, Yiwen Song, Qingmin Liao, Jian Yuan, and Yong Li. 2025b. Llm-explorer: A plug-in reinforcement learning policy exploration enhancement driven by large language models. *arXiv preprint arXiv:2505.15293*.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Xiaochong Lan, Yiming Cheng, Li Sheng, Chen Gao, and Yong Li. 2024a. Depression detection on social media with large language models. *arXiv preprint arXiv:2403.10750*.

Xiaochong Lan, Chen Gao, Depeng Jin, and Yong Li. 2024b. Stance detection with collaborative role-infused llm-based agents. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 18, pages 891–903.

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.

Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. 2022a. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*.

Jin Li, Jie Liu, Shangzhou Li, Yao Xu, Ran Cao, Qi Li, Biye Jiang, Guan Wang, Han Zhu, Kun Gai, et al. 2021. Truncation-free matching system for display advertising at alibaba. *arXiv preprint arXiv:2102.09283*.

Nian Li, Chen Gao, Mingyu Li, Yong Li, and Qingmin Liao. 2023a. Econagent: large language model-empowered agents for simulating macroeconomic activities. *arXiv preprint arXiv:2310.10436*.

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

Yinfeng Li, Chen Gao, Quanming Yao, Tong Li, Depeng Jin, and Yong Li. 2022c. Disenhcn: Disentangled hypergraph convolutional networks for spatiotemporal activity prediction. *arXiv preprint arXiv:2208.06794*.

Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. 2023b. A survey of graph meets large language model: Progress and future directions. *arXiv preprint arXiv:2311.12399*.

Zhixun Li, Bin Cao, Rui Jiao, Liang Wang, Ding Wang, Yang Liu, Dingshuo Chen, Jia Li, Qiang Liu, Yu Rong, et al. 2025. Materials generation in the era of artificial intelligence: A comprehensive survey. *arXiv preprint arXiv:2505.16379*.

Zhixun Li, Liang Wang, Xin Sun, Yifan Luo, Yanqiao Zhu, Dingshuo Chen, Yingtao Luo, Xiangxin Zhou, Qiang Liu, Shu Wu, et al. 2023c. Gslb: The graph structure learning benchmark. *Advances in Neural Information Processing Systems*, 36:30306–30318.

Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1754–1763.

Yile Liang, Haocheng Luo, Haining Duan, Donghui Li, Hongsen Liao, Jie Feng, Jiuxia Zhao, Hao Ren, Xuetao Ding, Ying Cha, et al. 2024. Meituan's real-time intelligent dispatching algorithms build the world's largest minute-level delivery network. *INFORMS Journal on Applied Analytics*, 54(1):84–101.

Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. 2024. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. In *Proceedings of the ACM on Web Conference 2024*, pages 3497–3508.

Chang Liu, Chen Gao, Yuan Yuan, Chen Bai, Lingrui Luo, Xiaoyi Du, Xinlei Shi, Hengliang Luo, Depeng Jin, and Yong Li. 2022. Modeling persuasion factor of user decision for recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3366–3376.

Chang Liu, Huan Yan, Hongjie Sui, Haomin Wen, Yuan Yuan, Yuyang Han, Hongsen Liao, Xuetao Ding, Jinghua Hao, and Yong Li. 2025. Mrgrp: Empowering courier route prediction in food delivery service with multi-relational graph. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 364–373.

Junling Liu, Chao Liu, Peilin Zhou, Qichen Ye, Dading Chong, Kang Zhou, Yueqi Xie, Yuwei Cao, Shoujin Wang, Chenyu You, et al. 2023. Llmrec: Benchmarking large language models on recommendation task. *arXiv preprint arXiv:2308.12241*.

Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. 2024. In-context learning with retrieved demonstrations for language models: A survey. *arXiv preprint arXiv:2401.11624*.

Saul McLeod. 2007. Maslow's hierarchy of needs. *Simply psychology*, 1(1-18).

Martin Mundt, Iuliia Pliushch, Sagnik Majumder, and Visvanathan Ramesh. 2019. Open set recognition through deep neural network uncertainty: Does out-of-distribution detection require generative classifiers? In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0.

OpenAI. 2024. Gpt-4o system card.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Chau Pham, Truong Vu, and Khoi Nguyen. 2024. Lp-ovod: Open-vocabulary object detection by linear probing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 779–788.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, et al. 2025. Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society. *arXiv preprint arXiv:2502.08691*.

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.

Walter J Scheirer, Lalit P Jain, and Terrance E Boult. 2014. Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2317–2324.

Charles D Schewe. 1988. Marketing to our aging population: Responding to physiological changes. *Journal of Consumer Marketing*, 5(3):61–73.

Saroja Subrahmanyan and J Tomas Gomez-Arias. 2008. Integrated approach to understanding consumer behavior at bottom of pyramid. *Journal of Consumer Marketing*, 25(7):402–412.

Zhen Tian, Ting Bai, Wayne Xin Zhao, Ji-Rong Wen, and Zhao Cao. 2023. Eulernet: Adaptive feature interaction learning via euler's formula for ctr prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1376–1385.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2024a. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems*, 36.

Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu, et al. 2021. Milvus: A purpose-built vector data management system. In *Proceedings of the 2021 International Conference on Management of Data*, pages 2614–2627.

Jing Yi Wang, Nicholas Sukiennik, Tong Li, Weikang Su, Qianyue Hao, Jingbo Xu, Zihan Huang, Fengli Xu, and Yong Li. 2024b. A survey on human-centric llms. *arXiv preprint arXiv:2411.14491*.

Lei Wang and Ee-Peng Lim. 2024. The whole is better than the sum: Using aggregated demonstrations in in-context learning for sequential recommendation. *arXiv preprint arXiv:2403.10135*.

Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, pages 1–7.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032.

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

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, et al. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*.

Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. 2021. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14393–14402.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2024. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. *arXiv preprint arXiv:2410.02506*.

Hanlei Zhang, Hua Xu, and Ting-En Lin. 2021. Deep open intent classification with adaptive decision boundary. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14374–14382.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. 2024. Sglang: Efficient execution of structured language model programs. *Advances in Neural Information Processing Systems*, 37:62557–62583.

# A Appendix

## A.1 The Structure of Maslow's Hierarchy Guided Living Needs Framework

In our method, to ensure that predicted living needs align with actual user needs on life service platforms while maintaining flexibility, we integrate Maslow's hierarchy of needs with real-world platform services to create a structured framework. Below is the complete framework we developed, which guides our LLM in refining initial predictions to better reflect genuine human needs while preserving their diverse expressions:

- Physiological Needs
  - Basic Diet
    - Home meals
    - Restaurant dining
  - Housing Needs
    - Basic living space
    - Quality living environment
  - Daily Living Supplies
    - Essential supplies
    - Fresh groceries
- Safety Needs
  - Health Protection
    - General wellness maintenance
    - Specialized medical care
    - Preventive healthcare
  - Home Security
    - Property protection
    - Safety maintenance
  - Living Environment
    - Cleaning services
    - Environmental maintenance
- Social Belonging Needs
  - Group Activities
    - Dining and social venues
    - Leisure and entertainment activities
  - Family and Parent-Child Interaction
    - Parent-child activities and entertainment
    - Family celebrations and event organization
- Esteem Needs
  - Knowledge Development
    - Educational training
    - Arts and skill development
  - Career Growth
    - Academic advancement
    - Professional skills
- Self-Actualization Needs
  - Cultural and Artistic Pursuits
    - Cultural experiences and exhibitions
    - Artistic creation and expression
  - Travel and Experiential Activities
    - Travel and sightseeing
    - Outdoor and experiential activities

## A.2 Hyperparameter Study (RQ6)

Our method introduces two hyperparameters: $K_p$, the number of personal relevant history records retrieved, and $K_s$, the number of relevant history records from similar users. Existing work (Lin et al., 2024) reports that the quantity of historical records provided to the LLM as reference can influence prediction performance. To further investigate the finding in our task, we explore the impact of these two hyperparameters on our model's performance. The results are shown in Figure 8.

The results indicate that the optimal values for $K_s$ and $K_p$ are 5 and 10, respectively, on both datasets. Additionally, we observe that varying $K_s$ within [1, 2, 5, 10] and $K_p$ within [2, 5, 10, 20] has minimal effect on model performance. This demonstrates the stability of our method with respect to hyperparameter selection.

## A.3 Hyperparameter Settings

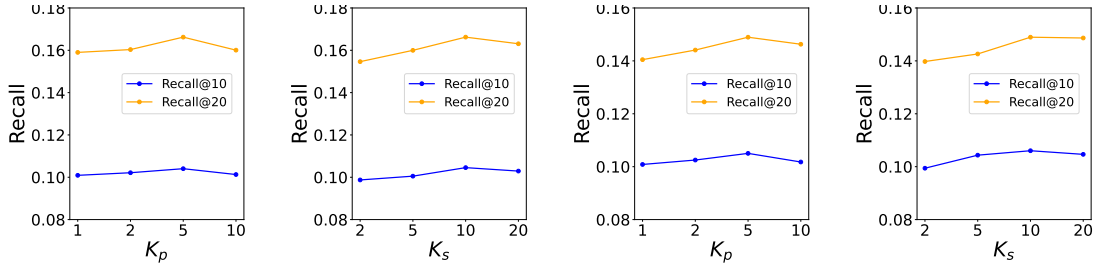
In our main method, the hyperparameters are set as follows:

- Number of personal historical records $K_p$: 5
- Number of similar users' historical records $K_s$: 10
- Number of GNN layers: 3
- GNN embedding size: 64
- GNN learning rate: $1 \times 10^{-4}$
- Triplet margin for triplet loss: 0.5
- Learning rate for triplet loss: $2 \times 10^{-5}$

## A.4 Details of Human Evaluation (RQ2)

The questionnaire uses a Likert scale format and consists of three groups of questions. Each group evaluates users' perceptions of need predictions from three models: PIGEON, LLMSREC-Syn, and DisenHCN. Each group contains 7 questions covering the five need categories from physiological to self-actualization needs, with two additional questions addressing common need combinations: physiological+social needs and physiological+safety needs. While PIGEON and LLMSREC-Syn generate predictions of approximately 10 words, DisenHCN provides fixed phrases. We evaluate three dimensions:

- Reasonableness: Assessing whether the predicted need is reasonable given the user's history and current context.
- Specificity: Evaluating whether the prediction points to needs with concrete fulfillment methods and actionability.
- Information Density: The amount of effective information conveyed per unit of text.




(a) Impact of $K_p$, Beijing (b) Impact of $K_s$, Beijing (c) Impact of $K_p$, Shanghai (d) Impact of $K_s$, Shanghai

Figure 8: Results of hyperparameter study.

Each dimension is rated on a 5-point Likert scale (1=lowest, 5=highest). To prevent bias, participants are blinded to the source models and only presented with the predicted content. To reduce cognitive load, we present only 5 behavioral records that demonstrate user preferences for each question. Here are sample questions from our survey:

(PIGEON)

Time: 2:00 PM, February 8, 2024
Location: Commercial Street
Historical Records:
- 3:00 PM, February 5, 2024: Hairdressing service at Commercial Street
- 12:30 PM, February 1, 2024: Nail service at Commercial Street
- 4:00 PM, January 28, 2024: Gym at Commercial Street
- 11:00 AM, January 25, 2024: Yoga studio at Commercial Street
- 2:30 PM, January 22, 2024: Eyelash service at Commercial Street

Prediction: Esteem needs, seeking services for appearance enhancement and fitness maintenance to boost confidence.

(LLMSREC-Syn)

[Same context and historical records as above]
Prediction: User likely seeking beauty or fitness services in the commercial street.

(DisenHCN)

[Same context and historical records as above]
Prediction: Beauty

We recruit 116 participants, including graduate students in user modeling and business professionals from life service platforms. To ensure comprehension, we provide examples and detailed explanations of each dimension at the beginning of the questionnaire. To mitigate fatigue and order effects, we randomly select 15 questions from a pool of 21 for each participant to evaluate. The average completion time is approximately 11 minutes.

### A.5 Performance Optimization and Deployment Considerations

Our method is designed for near-line deployment scenarios where second-level latency is acceptable. As described in Section 3.3, predicted needs serve as one source in the recall stage, and recall computations can be performed asynchronously. The computational flow is triggered by changes in time and location, recalculating only when time shifts to the next half-hour interval or location changes significantly, resulting in infrequent updates. While online computations often face strict latency constraints in the tens of milliseconds, near-line applications can tolerate latencies of seconds or even longer (Li et al., 2021).

Despite moderate latency requirements, we have performed comprehensive optimizations to handle potential peak loads and ensure practical deployment feasibility.

#### A.5.1 LLM Inference Optimization

We employ several strategies to reduce LLM inference latency:

**Model Fine-tuning:** We use instruction tuning on smaller LLMs. As shown in Figure 5, a fine-tuned 3B parameter model achieves performance comparable to GPT-4o mini while significantly reducing computational requirements.

**Inference Engine Optimization:** We build upon existing frameworks like vLLM (Kwon et al., 2023) and SGLang (Zheng et al., 2024) that leverage techniques including KV Cache, PagedAttention, and RadixAttention for general LLM inference optimization.

**Prompt 1:**
You are a user on a life service platform. At [8:00 PM] in [Commercial Street], what kind of living need are you most likely to have?
1. Your following past behaviors are provided for reference:
At 8:00 PM, ordered cocktail bar on Commercial Street due to the living need for social drinking.
At 8:30 PM, ordered Japanese izakaya on Commercial Street due to the living need for good food.
At 8:00 PM, ordered coffee shop at Office Area due to the living need for light refreshments.
At 8:30 PM, ordered dessert café on Commercial Street due to the living need for light refreshments.
At 7:30 PM, ordered Korean BBQ restaurant on Commercial Street due to the living need for group gathering.
2. Behaviors of other users in similar contexts are:
At 8:00 PM, ordered pub on Commercial Street due to the living need for social drinking.
At 7:00 PM, ordered café at Office Area due to the living need for light refreshments.
At 8:00 PM, ordered hotpot restaurant on Commercial Street due to the living need for group gathering.
At 8:30 PM, ordered sushi restaurant on Commercial Street due to the living need for good food.
At 8:30 PM, ordered wine bar on Commercial Street due to the living need for social drinking.
At 6:30 PM, ordered bubble tea at Office Area due to the living need for light refreshments.
At 8:00 PM, ordered yakitori shop on Commercial Street due to the living need for good food.
At 8:30 PM, ordered craft beer bar on Commercial Street due to the living need for social drinking.
At 7:30 PM, ordered BBQ restaurant on Commercial Street due to the living need for group gathering.
At 9:00 PM, ordered karaoke bar on Commercial Street due to the living need for recreation.
Considering the current time, location, and preferences indicated by previous consumption behaviors, please infer and describe your potential living needs.
**Response 1:**
At 8:00 PM on Commercial Street, I would most likely be looking for a place to socialize and enjoy good food with friends. This time of day in this location seems to be when I typically seek out social experiences combined with dining, particularly at establishments that offer both good food and a social atmosphere. The pattern suggests I value places where I can spend quality time with others while enjoying dinner or drinks, rather than just grabbing a quick meal or refreshment.
**Prompt 2:**
You are a user on a life service platform. An inference about your current living need at [8:00 PM] in [Commercial Street], is [Response 1]. Please use the following human living needs framework to further refine this inference, making it align with the framework's scope. Your response should be concise and as informative as possible, around 20 words.
**Response 2:**
Based on my physiological need for dining at restaurants and social belonging need for social gatherings, I seek venues that combine dining and socializing.

Figure 9: A working example of PIGEON.

**Targeted Prefill Optimization:** Since our average input length (approximately 400 tokens) is much larger than output length (approximately 20 tokens), the Prefill stage represents the main bottleneck. We restructure prompts to maximize common prefixes across queries without altering semantics. The optimized prompt structure is:

> *You are a user on a life service platform. Given time and location, what kind of living need are you most likely to have? Considering the current time, location, and preferences indicated by previous consumption behaviors, please infer and describe your potential living needs. Answer with around 20 words.*
> *Time: [time] Location: [location]*
> *Your following past behaviors are provided for reference: [Personal Relevant Records]*
> *Behaviors of other users in similar contexts are: [Similar Users' Relevant Records]*

We implement acceleration by explicitly caching the common prefix (the first 60 tokens), similar to System Prompt optimizations in online chat services. We verify that this rewritten prompt maintains the model's predictive performance while enabling significant acceleration.

We test performance on a fine-tuned Qwen2.5-3B model running on a single A100 GPU. Results are shown in Table 5, demonstrating significant latency reduction while maintaining accuracy. Our optimization achieves a 13.9× acceleration compared to Hugging Face Transformers inference.

Using SGLang with Prefix Caching at batch size 256, our method maintains P99 latency below 0.75 seconds while processing over 440 queries per second. There exists a trade-off between latency and QPS: larger batch sizes allow more queries to be processed in batches, increasing QPS but also increasing latency per query; smaller batch sizes reduce latency per query but decrease QPS. In actual deployment, batch size should be dynamically adjusted based on real-time load, potentially leveraging queuing theory. For latency-critical scenarios, setting batch size to 1 achieves an average latency

Table 5: Performance comparison of different inference optimization strategies on a single A100 GPU. P50/P99 refer to median and 99th percentile latency; QPS denotes queries per second.

| Method | Prefix Cache | Batch Size = 64 | | | Batch Size = 128 | | | Batch Size = 256 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | P50 (s) | P99 (s) | QPS | P50 (s) | P99 (s) | QPS | P50 (s) | P99 (s) | QPS |
| HF Transformers | - | 2.224 | 2.238 | 29.26 | 4.186 | 4.215 | 31.27 | 8.166 | 8.202 | 32.08 |
| vLLM | No | 0.298 | 0.501 | 210.80 | 0.427 | 0.441 | 303.60 | 0.706 | 0.881 | 354.57 |
| vLLM | Yes | 0.282 | 0.479 | 221.33 | 0.405 | 0.428 | 319.32 | 0.670 | 0.841 | 375.42 |
| SGLang | No | 0.260 | 0.435 | 232.96 | 0.362 | 0.374 | 329.31 | 0.629 | 0.758 | 415.43 |
| SGLang | Yes | 0.250 | 0.416 | 249.18 | 0.355 | 0.356 | 342.95 | 0.579 | 0.713 | 446.31 |

of 157ms.

Crucially, the optimizations we implement are strictly lossless, incurring no loss in accuracy. In practical applications, further acceleration can be achieved by incorporating techniques such as quantization and FP8 mixed-precision inference. Simple calculations indicate that using at most 8 A100 80G GPUs, our system can handle over 10 million queries per hour with no accuracy loss, demonstrating that online deployment is entirely feasible considering the inherently infrequent computation requirements of this method.

#### A.5.2 Other Component Optimizations

**Behavioral Embedding Learning and Retrieval:** User and context embeddings can be pre-computed offline, incurring no additional inference latency. Vector similarity search leverages mature open-source solutions like Milvus (Wang et al., 2021), which achieves throughputs of 15,000-20,000 QPS for nearest neighbor search on millions of vectors using cost-effective hardware while maintaining recall rates above 0.95. We plan to deploy using such established solutions, with potential targeted optimizations to further enhance performance.

**Life Service Recall:** The recall process, mapping flexible need descriptions to services, can also achieve low latency. Experiments on a consumer-grade GPU (Nvidia 4090) show that vectorizing 10,000 predicted need descriptions (output limited to approximately 20 words by prompt constraints) using the fine-tuned bge-base-en-v1.5 model takes an average of 8.92ms per description. Service vectors can be pre-computed, and matching need embeddings to the nearest service vectors can be efficiently handled by systems like Milvus, typically completing within 10ms.

These comprehensive optimizations demonstrate that online deployment is entirely feasible while maintaining the system's prediction accuracy and meeting practical performance requirements for large-scale industrial applications.

### A.6 Details of Comparison Methods

Here, we give a more detailed discussion of our utilized comparison methods.

Closed-set prediction methods:

- **XGBoost** (Chen and Guestrin, 2016): A widely used gradient-boosting model that builds an ensemble of decision trees to capture nonlinear interactions between user, time, and location features. Known for its robustness and strong generalization capabilities, XGBoost is effective in CTR tasks where feature relationships are complex and varied.

- **DeepFM** (Guo et al., 2017): This model integrates a deep neural network with a factorization machine (FM) to learn both low- and high-order feature interactions, such as the cross-features between user, time, and location. DeepFM effectively captures intricate dependencies without requiring extensive feature engineering, making it a strong baseline for closed-set need prediction.

- **DCN** (Wang et al., 2017): The Deep & Cross Network (DCN) combines a deep network and a cross network, which explicitly models cross-feature interactions in a multi-layered structure. By learning high-level feature combinations, DCN can model complex dependencies among user, time, and location, increasing the prediction accuracy within fixed need sets.

- **XDeepFM** (Lian et al., 2018): Extending DeepFM, XDeepFM includes a Compressed Interaction Network (CIN) to capture complex, higher-order feature interactions across layers. This component enhances the model's ability to learn subtle cross-feature patterns between user, time, and location, making it a powerful model for structured need prediction tasks.

- **EulerNet** (Tian et al., 2023): A recent CTR model optimized for large-scale recommendation, EulerNet efficiently handles high-dimensional feature inputs. It incorporates a

multi-layered structure and leverages feature embeddings to support large-scale applications with diverse features, providing competitive performance in need prediction tasks.

- **LightGCN** (He et al., 2020): An effective and concise Graph Convolutional Network (GCN) model. In our implementation, LightGCN constructs graphs with user, spatiotemporal context, living needs, and life services as nodes. The graph includes four types of edges: user-living need edges, user-life service edges, spatiotemporal context-living need edges, and spatiotemporal context-life service edges. The training method for LightGCN follows the same approach as the encoder training described in the paper. By focusing on core graph embeddings and omitting complex transformation layers, LightGCN efficiently captures direct relationships, making it effective for need prediction based on user and context.

- **HAN** (Wang et al., 2019): The Heterogeneous Attention Network (HAN) constructs a multi-relational graph, incorporating diverse types of nodes and edges (e.g., user, time, location, need) into a unified structure. HAN uses attention mechanisms to weigh different types of relationships, allowing it to learn the relative importance of each edge in predicting closed-set needs within varied contexts. In our implementation, we construct meta-paths including user-need-user, user-service-user, and context-need-context to capture different semantic relationships, then apply node-level and semantic-level attention to learn comprehensive representations.

- **DisenHCN** (Li et al., 2022c): DisenHCN utilizes disentangled representations within a hypergraph framework, breaking down user preferences along dimensions like time and location. By isolating preference factors in separate subspaces, DisenHCN captures the unique aspects of user behavior in each dimension, providing a more refined prediction of user needs.

Open-set prediction methods:

- **Zero-Shot CoT** (Kojima et al., 2022): The Chain-of-Thought (CoT) zero-shot approach leverages the LLM's generalization capability to predict needs based solely on current time and location, bypassing the need for historical data. This method serves as a baseline to evaluate how well the model can deduce user needs purely from situational context.

- **ReLLa** (Lin et al., 2024): ReLLa enhances LLM-based predictions by employing a text embedding model to convert user behavior history and spatiotemporal context into a vectorized format. During prediction, ReLLa retrieves the most relevant historical user records based on the current time and location, improving need prediction accuracy by leveraging similar past behaviors.

- **LLMSRec-Syn** (Wang and Lim, 2024): Extending ReLLa, LLMSRec-Syn retrieves similar historical behaviors not only from the target user but also from other users with similar patterns. This method leverages both individual and similar users' behavior, enhancing the open-set need prediction process by incorporating both user-specific and community-based context.

### A.7 A Working Example of PIGEON

Here, we provide a complete working example of PIGEON, as shown in Figure 9. It can be seen that first, a comprehensive and multi-faceted need description was generated. Then, PIGEON generates a more concentrated and structured need expression. This refined need expression can be used for downstream applications such as life service recall.

### A.8 Rationale for Using Maslow's Hierarchy

The choice of Maslow's hierarchy as our theoretical foundation is motivated by several key considerations. First, Maslow's model provides a comprehensive yet structured framework that naturally aligns with the hierarchical organization of life services on platforms. For instance, food delivery services correspond to physiological needs, while educational services map to higher-level growth needs. This natural alignment facilitates more systematic refinement of LLM predictions.

Second, Maslow's theory has been extensively validated in consumer behavior research (Subrahmanyan and Tomas Gomez-Arias, 2008; Schewe, 1988; Cao et al., 2013), demonstrating strong correlations between its need levels and consumer purchase patterns. This empirical foundation makes it particularly relevant for our task of predicting needs that drive service consumption, as it helps ensure our predictions align with established patterns of human motivation and consumption behavior.

While simpler prompting approaches could potentially guide the LLM to avoid overly specific

predictions, Maslow's framework offers additional benefits by providing a structured way to organize and validate need predictions against established psychological principles. Our ablation studies show modest but consistent improvements with this approach, suggesting it helps maintain prediction quality while adding theoretical grounding.

### A.9 Cost Analysis

In our main experiment, we used GPT-4o-mini as the base large language model. The costs per 10,000 predictions are approximately:

- Zero-shot CoT: $0.09
- ReLLa: $0.21
- LLMSREC-Syn: $0.67
- PIGEON: $1.05

### A.10 The List of Closed-Set Living Needs

For traditional closed-set living needs prediction, we extracted needs from user reviews and categorized them into 21 types under expert guidance. These living needs are:

1. Good food
2. A quick meal
3. Cooking at home
4. A group meal
5. Social drinking
6. Business travel
7. Personal grooming
8. Group gathering
9. Home supplies
10. Physical fitness
11. Pet companionship
12. Light refreshments
13. Family bonding
14. Vehicle service
15. Recreation
16. Body relaxation
17. Beauty
18. Leisure travel
19. Child education
20. Skill development
21. Daily essentials