NEON: Living Needs Prediction System in Meituan

Xiaochong Lan
Department of Electronic Engineering, BNRist, Tsinghua University
Beijing, China
lanxc22@mails.tsinghua.edu.cn

Chen Gao†
Department of Electronic Engineering, BNRist, Tsinghua University
Beijing, China
chgao96@gmail.com

Shiqi Wen
Meituan
Beijing, China
wenshiqi@meituan.com

Xiuqi Chen
Meituan
Beijing, China
chenxiuqi@meituan.com

Yingge Che
Meituan
Beijing, China
cheyingge@meituan.com

Han Zhang
Meituan
Beijing, China
zhanghan56@meituan.com

Huazhou Wei
Meituan
Beijing, China
weihuazhou@meituan.com

Hengliang Luo
Meituan
Beijing, China
luohengliang@meituan.com

Yong Li
Department of Electronic Engineering, BNRist, Tsinghua University
Beijing, China
liyong07@tsinghua.edu.cn

ABSTRACT

Living needs refer to the various needs in human's daily lives for survival and well-being, including food, housing, entertainment, etc. At life service platforms that connect users to service providers, such as Meituan, the problem of living needs prediction is fundamental as it helps understand users and boost various downstream applications such as personalized recommendation. However, the problem has not been well explored and is faced with two critical challenges. First, the needs are naturally connected to specific locations and times, suffering from complex impacts from the spatiotemporal context. Second, there is a significant gap between users' actual living needs and their historical records on the platform. To address these two challenges, we design a system of living **NE**eds predicti**ON** named NEON, consisting of three phases: feature mining, feature fusion and multi-task prediction. In the feature mining phase, we carefully extract individual-level user features for spatiotemporal modeling, and aggregated-level behavioral features for enriching data, which serve as the basis for addressing two challenges, respectively. Further, in the feature fusion phase, we propose a neural network that effectively fuses two parts of features into the user representation. Moreover, we design a multitask prediction phase, where the auxiliary task of needs-meeting way prediction can enhance the modeling of spatiotemporal context. Extensive offline evaluations verify that our NEON system can effectively predict users' living needs. Furthermore, we deploy NEON into Meituan's algorithm engine and evaluate how it enhances the three downstream prediction applications, via large-scale online A/B testing. As a representative result, deploying our system leads to a 1.886% increase *w.r.t.* CTCVR in Meituan homepage recommendation. The results demonstrate NEON's effectiveness in predicting fine-grained user needs, needs-meeting way, and potential needs, highlighting the immense application value of NEON.

CCS CONCEPTS

• **Information systems** → **Information systems applications**;

KEYWORDS

Living Needs Prediction; Deep Neural Networks; Multi-task Learning

ACM Reference Format:
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 (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3580305.3599874

1 INTRODUCTION

Living needs are the various needs generated by individuals in their daily routines for daily survival and well-being. Typical living needs include necessities such as food, housing, and personal care as well as leisure activities such as entertainment, for which there exists a diverse array of life service providers. Meituan is a large platform connecting customers to life service providers, in which users can meet almost all kinds of living needs. Unlike traditional information systems such as e-commerce websites, where users can only purchase products (*i.e.*, meeting one kind of living needs), Meituan allows access to various living services, such as booking

†Chen Gao is the corresponding author (chgao96@gmail.com).

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599874

https://www.metiuan.com

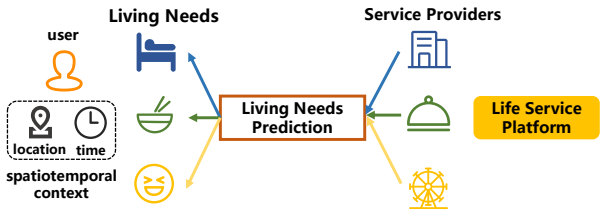

Figure 1: Living needs prediction aims to predict the specific living need of a user given the spatiotemporal context. By predicting users' living needs, life service platforms can recommend life services that can fulfill these needs.

a hotel, ordering takeout , etc. Moreover, an accurate understanding of users' needs can significantly improve user experience, for example, by enhancing the downstream recommendation tasks. Generally speaking, generating a specific kind of living need is naturally accompanied by a specific time or location, such as ordering food delivery at the office. Thus, the problem of *living needs prediction* can be defined as predicting the type of needs of the target user's spatiotemporal context, as illustrated in Figure 1.

For this new problem from the real-world scenario, there are two closely-related research topics, demand forecasting [10, 18, 29, 32, 34] and spatiotemporal activity prediction [3, 7, 15, 19, 41]. Demand forecasting is dedicated to predicting the quantity of a product or service that consumers will purchase. However, these works focus on the problem of aggregated demand prediction with times-series modeling, ignoring the needs of the specific user, time, and location. As for spatiotemporal activity prediction, the existing works focus mainly on either online activities such as App usage, or offline activities such as location visitation, while individuals' living needs can be fulfilled both in-store (offline) and via delivery (online), leading to a completely new problem.

There exist two critical challenges for living needs prediction as follows.

- **The impact of spatiotemporal context is complex.** As discussed above, for the same user, living needs at different locations or times are totally different. For example, the user in Figure 1 eats food delivery at noon and watches a movie at night. Additionally, the lifestyles, *i.e., the spatiotemporal pattern of living needs*, are extremely various for different users, further making it more difficult to model the spatiotemporal context.
- **There is a significant gap between users' actual living needs and their historical records on the platform.** Typically, users will face various kinds of situation in their life, and will generate multiple living needs. But they may only choose to satisfy one or a few of them on the platform, leading to a significant gap between their actual living needs and their historical records. We refer to the needs that can not be observed from historical records as *potential needs*. The case here is different from that in most recommender systems, where the actual interests and the historical record generally do not differ greatly for users, leading to another critical challenge.

To address these challenges, in this work, we described our deployed NEON system (short for living **NE**eds predicti**ON**) in Meituan, which includes three phases: **feature mining**, **feature fusion**, and **multitask prediction**. First, in the feature mining phase, to address the first challenge, we carefully design the spatial and temporal features for individual-level users, and to address the second challenge, we extract the behavioral-pattern features for group-level users. Second, in the feature fusion phase, we develop a feature-fusion neural network that combines internal preferences, impact from spatiotemporal context, and group behavior patterns to generate user representations, addressing both challenges. Last, as the complement to the main task of living needs prediction, we introduce the auxiliary task of needs-meeting way prediction to enhance the model's learning of spatiotemporal context, further addressing the first challenge.

The proposed NEON system plays a critical role in Meituan's recommendation engine with various downstream applications, including homepage recommendation, *Guess-you-like* page recommendation, and message pop-up recommendation, which requires the different-aspect ability of living needs prediction. After the deployment of the proposed system, we obtain stable and significant gains in three applications, providing strong real-world evidence for NEON's effectiveness from different perspectives.

The contribution of this work can be summarized as follows.

- To the best of our knowledge, we take the first step to study the problem of living needs prediction, which is a critical problem in real-world life service platforms but has not been well explored.
- We proposed the NEON system, which includes three phases of feature mining, feature fusion layer, and multitask prediction, which well addresses the two challenges, the complex impact of spatiotemporal context and missing behavioral data.
- We deploy NEON in Meituan's recommendation engine and conduct large-scale online A/B tests on three typical downstream applications, along with extensive offline evaluations. Offline experimental results verify that NEON can accurately predict users' living needs. The downstream evaluations strongly confirm NEON's high application value, with significant performance improvement in three downstream applications, among which a representative result is a 1.886% increase *w.r.t.* CTCVR for Meituan homepage recommendation.

2 PROBLEM STATEMENT

As we discussed above, in users' daily life, they generate various living needs, such as eating, accommodation, entertainment, beauty, etc. These needs can be fulfilled by life service providers in the city, which can be accessed through platforms connecting life service providers to customers. To enhance the user experience, it's crucial for these platforms to accurately predict users' needs and recommend appropriate services. This leads to the problem of living needs prediction.

As defined in the introduction, living needs prediction aims to predict the specific living needs of a user given the spatiotemporal context in which they are located (in the following we also refer to this as given the *user scene*). To clearly define the problem, with the help of experts, we divide all the living needs that users can satisfy on the platform into ten categories, shown in Table 1. We use $\mathcal{N}$ as the symbol for the set of all living needs. In response to these needs, all the life services on Meituan are also divided into 10 categories. The problem can be defined as follows.

Input: A dataset O^+ of real-world life service consumption records that reflect users' living needs. Each instance in the dataset tells the

Table 1: 10 types of living needs that can be satisfied in Meituan

Living Needs	Ordering food delivery, Eating in a restaurant, Booking a hotel, Buying medicine, Specialty shopping online, Hair-dressing, Grocery shopping online, Beauty, Tourism and Entertainment

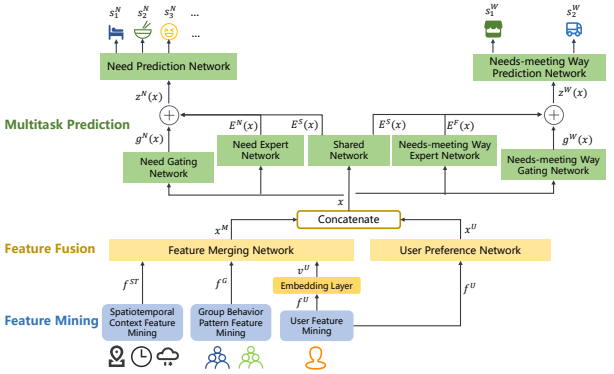

Figure 2: Illustration of living needs prediction system NEON.

kind of life service a specific user purchases, which indicates the specific living need n ($n \in \mathcal{N}$) of the user in a specific user scene i.
Output: A model to estimate the probability that a user will generate the living need n in user scene i, formulated as $f(i, n|O^+)$. Here $f(\cdot)$ denotes the function that the model aims to learn.

3 OUR NEON SYSTEM

To address the challenges mentioned in the introduction, we develop the NEON system made up of three phases: feature mining, feature fusion layer, and multitask prediction. First, in the feature mining phase, we address the first challenge by carefully designing spatiotemporal features for individual-level users and address the second challenge by extracting behavioral-pattern features for group-level users. The feature fusion phase then employs a feature-fusion neural network to seamlessly integrate internal preferences, spatiotemporal context impact, and group behavior patterns to generate complete user representations, overcoming both challenges. Last, in the multitask prediction stage, to enhance the model's understanding of spatiotemporal context, we introduce an auxiliary task of needs-meeting way prediction to the main goal of predicting living needs, providing additional support in addressing the first challenge. The deep feature fusion layer and the multi-task prediction parts of our system are illustrated in Figure 2.

3.1 Feature Mining

First of all, we use features that directly reflect user traits, such as users' profile, their recent behavior sequence, and their historical behaviors, as inputs for the model.

As mentioned above, for a specific user, his living needs are greatly affected by the spatial and temporal scenarios in which he is located. For example, *on a rainy midday, a person at work probably has the need to order food delivery; but on a sunny noon, he/she may have another need of going out to eat in the restaurant.* This complexity and variability of human living needs, driven by the flux of time and space, pose a considerable challenge in accurately modeling the impact of the spatiotemporal context. In order to tackle this challenge, we incorporate spatiotemporal context features as an integral part of our system's input.

What's more, on the platform, users may have potential needs with sparse or even non-existent history records. For example, *a person who never buys medicine on the platform may have a cold and need to buy medicine online one day.* Such potential needs are difficult for the model to grasp. To address the challenge of modeling potential needs, we introduce group behavior pattern features to help the model learn the potential living needs of users.

Below we give a detailed description of the three categories of features.

3.1.1 User Features. This group of features includes user profiles and user history behavior sequences.

- **User profiles f_p^U.** The user's profile, including their age, gender, etc.
- **User recent online behavior sequence f_{rb}^U.** The sequence of items recently clicked by the user in the platform; the sequence of items recently ordered by the user in the platform.
- **User aggregated historical online behavior f_{hb}^U.** The percentage of times users buy each type of life service.
- **User offline visitation record f_{ov}^U.** The 50 most visited POIs (point of interest) by the user in the last six months; the 50 most visited AOIs (area of interest) by the user in the last six months.

We concatenate all the mentioned features above to get a sparse user feature vector f^U, formulated as follows:

$$f^U = \left[f_p^U, f_{rb}^U, f_{hb}^U, f_{ov}^U \right]. \tag{1}$$

3.1.2 Spatiotemporal Context Features. Users' living needs are greatly affected by time, location, and other environmental factors. Thus, we introduce spatiotemporal context features as part of the input of our system to help our system model the complex impact of spatiotemporal context, which can be listed as follows.

- **Time f_t^{ST}.** Current time period. More than one time period feature of different granularity is applied, including hour, day, whether it is a holiday, etc.
- **Location f_l^{ST}.** The POI (point of interest) embedding of the user's real-time location; the AOI (area of interest) embedding of the user's real-time location; the city embedding of the user's real-time location. The location features are hourly real-time features.
- **Weather f_w^{ST}.** Weather information for the user's city or region, including wind, humidity, temperature, and weather type (sunny, rainy, snowy, etc.). Weather features are refined to hourly granularity.
- **Travel state f_{ts}^{ST}.** Information about whether the user is located in his/her resident city. Possible states include *based in resident city*, *about to travel*, and *on travel*.

The dense spatiotemporal context feature vector f^{ST} is created by concatenating all previously mentioned context features, formulated as follows:

$$f^{ST} = \left[f_t^{ST}, f_l^{ST}, f_w^{ST}, f_{ts}^{ST} \right]. \tag{2}$$

3.1.3 Group Behavior Pattern Features. We introduce group behavior pattern features to supplement the sparse individual behavior of users, in order to assist in identifying the potential living needs of individual users.

- **Group aggregated behavior f_a^G.** We first segment users into groups based on their profiles. In each group, we get the group aggregated behavior by calculating the percentage of views, clicks, and purchases of each type of life service among all views, clicks, and purchases initiated by the group. For each user, the group aggregated behaviors of the groups the user belongs to are used as features. For example, a middle-aged person has group aggregated behaviors feature of middle-aged users and other groups he/she is in.
- **Popularity in the current time period f_{ct}^G.** We cut all time into time periods according to different criteria, such as whether it is a holiday, if it is morning, noon, or night, etc. Then in each time period, we calculate the popularity of each type of life service by calculating the percentage of times the life service is viewed/clicked/purchased among all views/clicks/purchases happening in this time period. We determine the time periods in which the current time is located, and use the popularity of each type of life service in these time periods as a feature. For example, if the user opens the app on Christmas night, popularity on holiday and popularity at night of each kind of life service are set as features.
- **Group behavior pattern in spatiotemporal context f_{st}^G.** By discovering group preferences in different spatiotemporal contexts, we further capture more fine-grained group behavior patterns. We calculate the percentage of views/clicks/purchases of each type of life service initiated by each group in each kind of spatiotemporal context. These fine-grained patterns are used as features of the model. For example, the group preference of middle-aged people at work at noon on working days are used to enrich the representations of every individual within this demographic in such spatiotemporal scenario.
- **User behaviors augmented by inter-need correlation f_{ic}^G.** There is an inherent association across different types of users' living needs. This association can be leveraged to improve prediction performance. For example, *a user who frequently purchases hairdressing services may also be inclined to purchase beauty services.* We use the association rule mining algorithm to analyze the co-occurrence of different life service categories, filter out high-correlation relationships, and employ them to augment user behavior as input features.

We combine all previously mentioned group behavior pattern features to generate the dense group behavior pattern feature vector f^G, which is formulated as follows:

$$f^G = \left[f_a^G, f_{ct}^G, f_{st}^G, f_{ic}^G \right]. \tag{3}$$

3.2 Feature Fusion Layer

As mentioned in Section 2, we refer to a user in a specific spatiotemporal context as a user scene. For a user scene i, after feature mining, we have dense spatiotemporal features f_i^{ST}, dense group pattern features f_i^G, and sparse user features f_i^U. For brevity of presentation, we omit the subscript i in some of the expressions below. We designed a feature fusion layer to integrate these features into the input of the subsequent prediction module.

We first set up an embedding layer, which processes the high-dimensional sparse user feature vector f^U into a low-dimensional dense vector v^U. To address the challenges of complex impact of spatiotemporal context and users' potential needs, we mine spatiotemporal features and group behavior pattern features in the feature mining phase, respectively. With these features as input, we use a feature merging network to model the interaction between spatiotemporal contexts, group behavior patterns, and users as follows,

$$x^M = h^M \left(\left[f^{ST}, f^G, v^U \right] \right), \tag{4}$$

where $[\cdot]$ denotes concatenation operation. Here h^M is the feature merging network, which merges three information sources of spatiotemporal contexts, group behavior patterns, and user preference into a fusion representation x^M.

Moreover, users have their own internal characteristics that are independent of the spatiotemporal scene they are in and the group they belong to. To model the internal characteristics of users, we generate a representation as follows,

$$x^U = h^U \left(f^U \right), \tag{5}$$

where h^U denotes the user preference network that turns raw user features into dense user preference representation. We then concatenate the two parts of representations into the full representation of the user scene:

$$x = \left[x^M, x^U \right]. \tag{6}$$

In brief, we design a feature fusion layer to tackle both challenges by considering the influence of spatiotemporal context, incorporating group behavior patterns, as well as extracting individual preferences.

3.3 Multitask Prediction

We further design a prediction module which takes user scene representation as input to predict users' living needs. The module is tasked with two objectives: *fine-grained need prediction* and *needs-meeting way prediction*. Fine-grained need prediction is to predict the specific living need of the user. Neets-meeting way prediction is to predict the preferred way of the user to meet their needs.

Specifically, among the ten kinds of needs which we mentioned in the problem formulation, there are two ways to satisfy the needs: in-store and via-delivery. In other words, consumers can choose to satisfy their needs by visiting a physical store or by ordering online and then receiving goods via delivery. Each type of the 10 needs can be classified into one of two categories, in-store needs or via-delivery needs. We show the classification in Table 2. Actually, needs-meeting way prediction is to predict whether the preferred way of the user to meet their needs is in-store or via-delivery.

Users' preferences for needs-meeting ways are strongly affected by the spatiotemporal context. For example, *a person at work during lunchtime on a weekday is more likely to have the need to order food delivery (via delivery), while the same person in a shopping district on a weekend evening is more likely to visit a store for a meal (in-store).* With this in mind, we include the needs-meeting way prediction task which is jointed trained with the main task of need

Table 2: The classification of the 10 living needs that can be satisfied on Meituan

Living needs that can be satisfied via delivery	Specialty shopping online, Grocecy shopping online, Ordering food delivery, Buying medicine
Living needs that can be satisfied in store	Eating in a restaurant, Hotel, Hair-dressing, Beauty, Tourism, Entertainment

prediction to enhance the model's ability to learn spatiotemporal context information. Next, we describe how we get the prediction results of the two tasks. We use y^W and y^N to denote the prediction result of needs-meeting way and specific need. $y^k, k \in \{W, N\}$ can be generated as follows,

$$\begin{aligned} y^k &= t^k(z^k), \\ \text{where } z^k &= g^k(x)_0 E^k(x) + g^k(x)_1 E^S(x). \end{aligned} \tag{7}$$

Here t^k is the prediction neural network for task k. There are a variety of choices in the specific structure of the neural network. In Section A.1, we will state our specific choice. To avoid verbosity, we will use *network* to replace *neural network* in the following text. The output of t^N, y^N, is the scores of ten types of living needs, and the output of t^W, y^W, is the scores of in-store and via-delivery needs-meeting ways. We use s^N_{im} to denote the score of need m for user scene i, and use s^W_{in} to denote the score of needs-meeting way n for user scene i. E^k is the expert network [22, 35] for task k. E^S is the shared network between the two tasks. E^S is responsible for generating general representations that are common to both tasks, while E^k is responsible for learning task-specific representations that are more fine-tuned to the specific task k. The gating network g_k determines what proportion of information input each task's prediction network receives from the shared network and the expert network. We formulate the gating network as follows,

$$g^k(x) = \text{Softmax}(W_k x), \tag{8}$$

where $W_k \in \mathbb{R}^{2\times d}$ is trainable weights for task k. The gating network takes x as input, and outputs the relative importance of the shared and task-specific representations for a given tasking, allowing the model to selectively attend to the most relevant information and improve its performance. In summary, to address the complexity of spatiotemporal context impact, we introduce an auxiliary task of needs-meeting way prediction which is jointly trained with the main task of fine-grained living needs prediction to enhance our system's learning of spatiotemporal context. The multitask prediction module in our system produces a score for each living need and needs-meeting way.

3.4 Model Training

In this section we describe how our system is trained. Corresponding to the two tasks, we design two parts of loss. We design need prediction loss taking into account the fact that the frequency of different needs arising in users' lives is different. For example, *a user may need to order food delivery for lunch every workday, but rarely need to buy medicine.* In order to address the class imbalance issue for different living needs, we propose using a multi-class focal loss which can decrease the effect of needs with a high volume of training data on the final prediction loss. The need prediction loss can be formulated as follows,

$$\text{Loss}_{\text{need}} = -\sum_{i\in O}\left(\sum_{n=1}^{10}\left(1-q^N_{in}\right)^{\gamma}\chi^N_{in}\log\left(q^N_{in}\right)\right), \tag{9}$$

$$\text{where } q^N_{in} = \text{Softmax}\left(s^N_{in}\right) = \frac{e^{s^N_{in}}}{\sum_n e^{s^N_{in}}}. \tag{10}$$

Here O is the training set, s^N_{in} is the score of living need n for user scene i, γ is the hyperparameter which decides the importance of difficult samples, χ^N_{in} is 1 if n is the ground truth need for user scene i, else it is 0. For the needs-meeting way prediction task, we use BCE loss as prediction loss. We formulate it as follows,

$$\text{Loss}_{\text{way}} = -\sum_{i\in O}\left(\sum_{m=1}^{2}\chi^W_{im}\log\left(q^W_{im}\right)\right), \tag{11}$$

$$\text{where } q^W_{im} = \text{Softmax}\left(s^W_{im}\right) = \frac{e^{s^W_{im}}}{\sum_m e^{s^W_{im}}}. \tag{12}$$

Here O is the training set, s^W_{im} is the score of needs-meeting way m (in store or via delivery)for user scene i. χ^W_{im} is 1 if m is the ground truth needs-meeting way for user scene i, else it is 0. In our system, the feature integration module and multitask prediction module are trained end to end. The entire loss function is:

$$\text{Loss} = \lambda_1\text{Loss}_{\text{need}} + \lambda_2\text{Loss}_{\text{way}}. \tag{13}$$

λ_1 and λ_2 are hyperparameters that control the importance of the two parts of loss.

4 OFFLINE EVALUATION

4.1 Experimental Settings

4.1.1 Dataset. We conduct an offline experiment on a real-world dataset at a billion-scale. The dataset comprises a sampling of all 2022 purchase records on the platform, based on the percentage of purchases for each type of life service. It consists of over 7 billion actual purchase records from 65 million users. The details of datasets are provided in Appendix A.2.

4.1.2 Metrics. We design three metrics, namely Sort Accuracy (SA), Via-delivery Sort Accuracy (VDSA), and In-store Sort Accuracy (ISSA), to measure the performance of our system and baseline systems in predicting living needs. SA measures the overall accuracy of sorting living needs based on their scores. VDSA focuses on the accuracy of predicting needs that are satisfied via delivery, while ISSA focuses on the accuracy of predicting needs for in-store scenarios. Detailed definitions of these metrics are provided in Appendix A.3.

4.1.3 Baselines. To illustrate the effectiveness of our system, we compare it with two baselines widely in actual production environments, including **DIN [42]**, **DNN [5]**, **DCN [38]**, **ESMM [24]**, and **MMOE [22]**. We will provide a detailed description of these baselines in Appendix A.4.

Table 3: Offline experimental performance of NEON and baselines.

Method	VDSA	ISSA	SA
DIN	0.9044	0.7467	0.8700
DNN	0.9060	0.7500	0.8718
DCN	0.9051	0.7466	0.8701
ESMM	0.9080	0.7476	0.8708
MMOE	0.9097	0.7573	0.8757
NEON	**0.9175**	**0.8277**	**0.9070**
Improvement	0.86%	9.30%	3.57%

Table 4: Performance of NEON with/without multitask prediction.

	VDSA	ISSA	SA
w.o.	0.9089	0.8084	0.8968
with	0.9175	0.8277	0.9070
Improvement	0.95%	2.39%	1.14%

4.2 Overall Performance

We test the performance of our proposed system and baselines on the living needs prediction task, and show the results in Table 3. We can have the following observations.

- **Our system steadily outperforms all baselines on all metrics.** The improvement of our system compared to the best baseline is 0.86%, 9.30%, and 3.57% *w.r.t.* VDSA, ISSA, and SA, respectively. The significant performance gain confirms the effectiveness of our system on the living needs prediction task. Furthermore, such a significant improvement in the ability to predict living needs will result in a huge benefit in real-world production scenarios, which will be further confirmed through online evaluation.
- **Our system achieves greater improvement on ISSA.** The task of predicting users' in-store living needs is relatively difficult, since the in-store consumption data is sparser, and the relationships between spatiotemporal context and in-store needs are more complex. On this task, all methods perform the worst, and our system outperforms baselines on a large margin, with an improvement of 9.30% *w.r.t.* ISSA. This further confirms our model's ability to tackle the complex impact of spatiotemporal context and discovering potential needs.

4.3 Ablation Study

As mentioned in Section 3.3, we introduce the task of needs-meeting way prediction to enhance the system's learning of spatiotemporal context. To study the effectiveness of the multitask prediction design, we remove it from our system to observe the impact of the design on the system performance. Specifically, we change the system structure by removing the needs-meeting way prediction network t^W and taking the sum of z^N and z^W as the input of the need prediction network t^N, and test the performance of the changed system. The results are shown in Table 4. The results show that our proposed system outperforms the system without multitask prediction design. Our system performs better *w.r.t.* VDSA by 1.16%, *w.r.t.* ISSA by 2.39%, and *w.r.t.* SA by 1.14%. The significant performance improvement confirms the validity of the multitask prediction design.

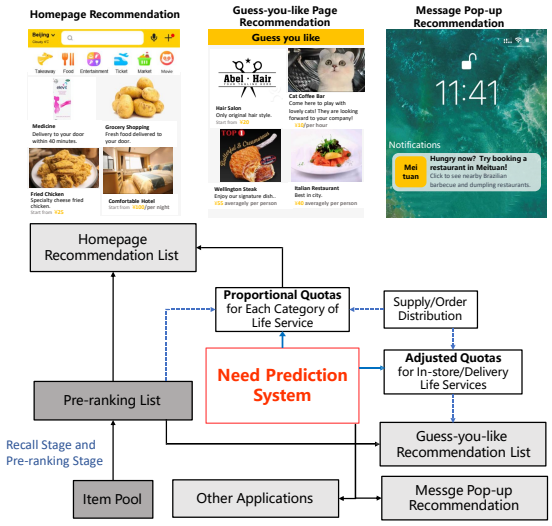

Figure 3: Online deployment of NEON in Meituan Recommendation Engine. In summary, NEON helps to generate the quotas of categories of life services in the recommendation list (homepage recommendation and *guess-you-like* recommendation), or decide the one category to be recommended to the user (message pop-up recommendation).

5 ONLINE EVALUATION

For life service platforms such as Meituan, understanding and predicting users' living needs are important in all business scenarios. In this section, we evaluate the performance of three downstream recommendation tasks when deploying NEON into Meituan's recommender engine, and the illustration of deployment is shown in Figure 3. Specifically, the three typical applications are homepage recommendation, *Guess-you-like* recommendation, and message pop-up recommendation. The performance of NEON on these applications reflects its effectiveness on fine-grained need prediction, needs-meeting way prediction, and potential need prediction, respectively. We elaborate on the online testing of three tasks one by one as follows.

5.1 Homepage Recommendation

In this section, we evaluate the performance of NEON when deployed to homepage recommendation, which reflects its ability on fine-grained living need prediction.

5.1.1 Deployment Scenario. Typically, users open Meituan mobile APP to meet their certain living need. In the overall recommendation list on homepage offered by Meituan, users probably only focus on the items which belong to the type of life service that can fulfill their living needs, and choose one item from the items. Whether an item is in the category of life service that the user *need* is at least as important as whether the user *likes* the item. To emphasize the importance of recommending *needed* items, the engine follows a two-step approach to generate the final recommendation

Table 5: Results of A/B tests on homepage recommendation

Metric	w/o NEON	with NEON	Improvement
CTR	3.0601×10^{-2}	3.0672×10^{-2}	+0.230%
CVR	1.1497×10^{-1}	1.1687×10^{-1}	+1.652%
CTCVR	3.5183×10^{-3}	3.5846×10^{-3}	+1.886%

list from the pre-ranking result. For a user scene, it first decides the quotas for all the ten categories of life services and then generates the final list according to the quotas and the prediction scores of the items. The supply quotas are generated based on the scores our system outputs along with other criteria.

In short, our NEON system is used to generate the *quotas* of different kinds of life services in the homepage recommendation list.

5.1.2 Experiment Setting. We will first give a detailed description of how our system is used in generating the quotas for each category of life services. At first, we recall local life service items into the recall pool using various strategies, such as popularity and collaborative filtering. Then the engine outputs a preliminary recommendation list based on the recall pool, called pre-ranking list. After that, we take the Softmax normalized scores output by our system as the proportional quotas for each category of life service. We further adjust the quotas taking the proportion of each category in the pre-ranking list, supply distribution by category, and order distribution by category into account. The generated quotas are the proportion of each category in the lists received by users. The lists are generated considering both quotas of categories and prediction scores of items.

We compare the homepage recommendation performance of our whole recommendation engine with and without our system through online A/B tests. The tests last for one week and involved around 4.5 million users.

In our tests, the users are randomly divided into two buckets of similar size and assigned different methods for calculating quotas for each category. Specifically, for the first group, we use the method described previously, while for the second group, we generate quotas based only on the proportion of each category in the pre-ranking list, supply distribution by category, and order distribution by category. We maintain consistency across all other modules to ensure a fair comparison.

For metrics, we use *Click Through Rate* (CTR), *Conversion Rate* (CVR), *Click to Conversion Rate* (CTCVR) to measure the quality of the final recommendation list, which are widely-used measurements [21, 24, 39].

5.1.3 Performance. The results of our A/B tests are shown in Table 5. From the results, we can have the following observations:

- There is a significant improvement with respect to all metrics. The increase *w.r.t.* CTR and CVR are 0.230% and 1.653%, respectively, which is a notable improvement.
- The increase *w.r.t.* CTCVR is 1.886%. CTCVR indicates how likely users are to purchase the recommended items. Such improvement can result in a substantial rise in total consumption on the platform.
- The rise *w.r.t.* GTV-CC is 3.627%. The remarkable uplift demonstrates the outstanding capability of our system in predicting users' living needs that has no historical record.

Table 6: KLD of different time periods between real user order distribution and quotas given with/without NEON.

Time Period	Kullback-Leibler Divergence		Improvement
	w/o NEON	with NEON	
0-4	0.2578	0.2201	-14.62%
5-8	0.2237	0.1907	-14.75%
9-10	0.2179	0.1902	-12.71%
11-12	0.2408	0.2138	-11.21%
13-16	0.2323	0.2053	-11.62%
17-19	0.2353	0.2096	-10.92%
20-24	0.2333	0.1995	-14.49%

We further calculate the *Kullback-Leibler divergence* [16] (KLD) between real user order distribution by category and the average proportional allocation given by the online engine with/without our system. Lower KLD indicates the quotas match better with real user order distribution, which can be regarded as real user needs distribution. The results of different time periods throughout the day are shown in Table 6. The time periods are separated based on the business characteristics of the platform during each hour. Similar hours are grouped within a single period.

From the results, it can be seen that in all time periods, the Kullback-Leibler Divergence between the actual user order distribution by category and the average proportional allocation generated by the online engine with NEON is significantly less than that without our NEON. The percentage of decrease (improvement) is 12.90% on average. This indicates that the quotas produced by the online engine with our system are more aligned with the actual user consumption distribution, or the real-life user living needs, compared to the ones generated without our system. This can be regarded as evidence of our model's effectiveness in addressing the intricate impact of spatiotemporal context, and further confirms our system's strong ability for predicting fine-grained user needs.

5.2 Guess-you-like Recommendation

This section assesses NEON's performance in guess-you-like recommendation, highlighting its ability on needs-meeting way prediction.

5.2.1 Deployment Scenario. Meituan designs a *Guess-you-like* page, which user may be guided to during their leisure time. Typically, users browse this page to satisfy their *unnecessary* living needs, such as entertainment or beauty, etc. In this page, they purchase items they need and also *like*. To provide more choices for users in the categories of life services they need, the recommendation list should have more items in those categories. In this page, we assume that users are more concerned with being recommended items they like. Therefore, we do not maintain the proportion of a category in the "Guess-you-like" recommendation list just because there is a slight possibility that it is essential, as users can use modules other than Guess-you-like such as the search button to find necessary life services. To ensure enough items are in the needed category, the online engine generates new quotas for Guess-you-like page based on the quotas for homepage recommendation list and the in store/via delivery score output by our system.

Table 7: Overall results of A/B tests on *Guess-you-like* page recommendation.

Metric	w/o NEON	with NEON	Improvement
CTR	9.5707×10^{-2}	9.5647×10^{-2}	-0.063%
CVR	1.3568×10^{-1}	1.3606×10^{-1}	+0.280%
CTCVR	1.3014×10^{-2}	1.2985×10^{-2}	+0.218%
NFR-UV	1.0128×10^{-4}	1.0294×10^{-4}	+1.605%
NFR-PV	8.3283×10^{-5}	8.6726×10^{-5}	+3.342%

In summary, the needs-meeting way prediction results of NEON are used to further adjust the *quotas* of different life services in *Guess-you-like* recommendation list.

5.2.2 Experiment Setting. We conduct online A/B tests involving about 7 million users over a period of two weeks. We randomly divide users into two buckets, each of which has a similar amount of users, and assign them different methods of generating the quotas of categories in the Guess-you-like page. Specifically, for the first bucket we adopt the same strategy as in the homepage recommendation. For the second bucket, we calculated the score of both needs-meeting ways, and turn higher the quotas of categories whose needs-meeting way gets a higher score. As for metrics, we use CTR, CVR, CTCVR, *Negative Feedback Rate for Unique Visitor* (NFR-UV), *Negative Feedback Rate for Page View* (NFR-PV) to measure the quality of the recommendation list in the Guess-you-like page. NFR-UV and NFR-PV emphasize that the recommendation list should not contain items users dislike.

5.2.3 Performance. Under the aforementioned settings, we conduct extensive online A/B experiments. The results are shown in Table 7. We list our observations as follows:

- The performance gain *w.r.t.* CVR, CTCVR, and OV are 0.280%, 0.218%, and 0.310%, respectively. Such improvement resulting from adjusting the quotas with the assistance of the needs-meeting way prediction module can lead to a significant increase in the consumption amount.
- The NFR-UV and NFR-PV decrease by 1.57% and 3.31% respectively, indicating that the recommendation engine is able to recommend fewer items that users dislike in the Guess-you-like page by adjusting the quotas with the aid of the needs-meeting way prediction module in our system.

We further calculate the performance increase on some popular categories of life services in Guess-you-like page. The results are shown in Table 8. From the result, we can observe that:

- For all these categories there is average relative improvement of 3.801% *w.r.t.* CVR and 3.765% *w.r.t* CTCVR. With such improvements, all these categories can enjoy a notable increase in order volume on Meituan platform.
- Among all these categories, the rise on Kids category is the most significant, which is 8.723% *w.r.t.* CVR and 9.691% *w.r.t.* CTCVR.
- There are also slight decreases *w.r.t.* CTR of 0.034% for Food Delivery, -0.449% for Beauty, and -0.595% for Hotel. These decreases are within the typical fluctuations observed in the market.

The above results, which demonstrate the successful implementation of our system in guess-you-like recommendation, further illustrate the efficacy of our system, particularly in needs-meeting way prediction.

Table 8: The A/B tests results on several popular categories of life services in *Guess-you-like* page.

Category	Metric	w/o NEON	with NEON	Improvement
Food	CTR	1.8298×10^{-2}	1.8292×10^{-2}	-0.034%
	CVR	2.8841×10^{-1}	2.8987×10^{-1}	+0.506%
	CTCVR	5.2774×10^{-3}	5.3023×10^{-3}	+0.472%
Beauty	CTR	1.8769×10^{-2}	1.8684×10^{-1}	-0.449%
	CVR	1.2583×10^{-3}	1.2889×10^{-3}	+2.427%
	CTCVR	2.3616×10^{-4}	2.4081×10^{-4}	+1.967%
Kids	CTR	1.7717×10^{-2}	1.7874×10^{-2}	+0.890%
	CVR	7.8825×10^{-3}	8.5701×10^{-3}	+8.723%
	CTCVR	1.3965×10^{-1}	1.5319×10^{-4}	+9.691%
Hotel	CTR	1.6567×10^{-2}	1.6469×10^{-2}	-0.595%
	CVR	4.1454×10^{-2}	4.2923×10^{-2}	+3.546%
	CTCVR	6.8678×10^{-1}	7.0690×10^{-4}	+2.930%

5.3 Message Pop-up Recommendation

In this section, we test our model's performance in message pop-up recommendation to observe its potential need prediction ability.

5.3.1 Deployment Scenario. In the case where a user has a living need that has never or rarely been fulfilled on Meituan platform, he/she probably won't launch Meituan mobile app, as they may not be aware that this need can be satisfied on Meituan or may not be accustomed to fulfilling this need on Meituan. So, if our engine runs in the background of the phone and detects the user's potential living need, it will send a message pop-up to the user with a recommendation for a life service solution, if the user allows it. The message pop-up recommends the user a life service that is in the category of the highest score output by our system.

In brief, with the ability of potential needs prediction, NEON is deployed for message pop-up recommendation.

5.3.2 Experiment Setting. We conduct online A/B tests involving one million users over a period of two weeks. The users are randomly divided into two buckets of similar volume and are assigned different strategies for selecting items to be sent in message pop-ups. For the first bucket, the message pop-up recommends items that belong to the category with the highest score output by our system. For the second bucket, in each hour of the day, we calculate the popularity and average CTR of each category. We distribute the traffic for message pop-ups to the categories with the most popularity and highest average CTR.

We use CTR, CVR, CTCVR, OV, *Number of Cold-start Customers* (NCC) to measure the performance of our system in potential need prediction. NCC represents the number of customers who purchase a lifestyle service that they have never acquired on the platform previously.

5.3.3 Performance. The results of the online A/B tests are shown in Table 9. We can observe that:

- By replacing the algorithm based on popularity and average CTR with our living need prediction system NEON to determine the category of recommendation in message pop-up, all metrics show

Table 9: Results of A/B tests on message pop-up recommendation.

Metric	CTR	CVR	CTCVR	OV	NCC
Improv.	+8.21%	+78.64%	+85.71%	+95.92%	+74.26%

significant improvement. CTR, CVR, CTCVR, and OV increase by 8.21%, 78.64%, 85.71%, and 95.92%, respectively.
- NCC increases by 74.26%, indicating that the online deployment of our system in message pop-up recommendations results in a 74.26% increase in the number of customers purchasing a life service that they have not previously acquired on the Meituan platform, via the message pop-up feature.

Our system is able to accurately detect the potential needs of users for a specific lifestyle service, even in instances where they have never previously purchased that service on the Meituan platform, and accordingly deliver targeted message pop-ups to them. This result strongly demonstrates the exceptional capability of our system in predicting users' potential needs.

In summary, the success achieved by our NEON system on three downstream recommendation tasks proves its effectiveness on fine-grained need prediction, needs-meeting way prediction, and potential need prediction, respectively. The significant performance gain in real applications can lead to huge benefits.

6 RELATED WORK

As discussed above, in this work we explore the problem of living needs prediction, defined as predicting the specific living **needs** of a user given the **spatiotemporal** context. Thus, there are two closely related research topics: demand forecasting and spatiotemporal activity prediction.

6.1 Demand Forecasting

Demand forecasting aims at predicting the quantity of a product or service that consumers will purchase. It helps in making informed decisions on inventory management, production scheduling, pricing strategy, etc. The problem of demand forecasting is broad and multifaceted, affecting many different industries, including restaurant [18], manufacturing [32], retail [29], tourism [34], energy [10], transportation [36], etc.

To address the problem of demand forecasting, researchers have proposed various methods which can be broadly classified into three categories: statistical models [8, 11, 18, 26, 37], machine learning models [2, 23, 30, 31], and deep learning models [6, 14, 17, 28]. Statistical models, such as exponential smoothing [33], are well-suited for long-term demand forecasting as they are based on historical trends and patterns. However, they are not adept at handling variations or outliers in the data, making them unsuitable for volatile or short-term demand forecasting. On the other hand, machine learning models for demand forecasting, such as Random Forest based models [1, 27], are efficient at short-term demand forecasting, but their performance drops when it comes to long-term forecasting. Deep learning models such as LSTM based models [17] and GAN based models [12] have the ability to capture complex patterns and dependencies in the data, making them suitable for both short-term and long-term demand forecasting. However, they require a large amount of data to work well.

Existing demand forecasting methods can not handle the problem of living needs prediction. These methods focus on the overall demand for a particular product or service in a market, but in this work, we aim at predicting the need of a specific consumer. What's more, demand forecasting methods predict demands in several months or years, while in this work we predict a user's need at a specific time and location.

6.2 Spatiotemporal Activity Prediction

Spatiotemporal activity prediction aims to predict the activity of a user at a given time and location. Previous works have employed various methods to perform spatiotemporal activity prediction. One popular approach is to build a tensor using historical data and then conduct tensor factorization to learn intrinsic association [3, 7, 41]. For example, Fan *et al.* [7] propose to integrate tensor factorization with transfer learning for online activity prediction. Additionally, WDGTC [20] proposes a low-rank tensor decomposition and completion framework for passenger flow prediction by introducing L1-norm and Graph Laplacian penalties. Recently, researchers have introduced Graph Convolutional Networks [15] (GCNs) to achieve high performance in spatiotemporal activity prediction. For example, SA-GCN [40] develops a Graph Convolutional Network with meta path-based objective function for App-usage prediction task. Furthermore, DisenHCN [19] utilizes a heterogeneous hypergraph to model fine-grained user similarities, resulting in significant performance gains.

However, existing works on spatiotemporal activity prediction focus on predicting the specific activities of people, while our work focuses on the general living needs which are the driving force behind specific consumption behaviors. What's more, these works mainly focus on either online or offline activities, but in our work, we predict living needs can be satisfied both in store (offline) and via delivery (online) by different kinds of life services, which is beyond the capabilities of existing methods.

7 CONCLUSION AND FUTURE WORK

In this work, we approach the new problem of living needs prediction, which is critical in life services platforms. We present the NEON system in Meituan, consisting of three phases, feature mining, feature fusion, and multitask prediction. Large-scale online A/B testing in three downstream applications, along with extensive offline evaluation, strongly confirm the effectiveness of our system. As for future work, we plan to test NEON's performance in more downstream applications.

ACKNOWLEDGEMENT

This work is supported in part by National Key Research and Development Program of China under 2022YFB3104702. This work is supported in part by National Natural Science Foundation of China under 62272262, 61971267, and 61972223. This work is supported in part by a grant from the Guoqiang Institute, Tsinghua University under 2021GQG1005. This work is supported in part by Beijing National Research Center for Information Science and Technology. This work is also supported by Meituan.

REFERENCES

[1] Raza Abid Abbasi, Nadeem Javaid, Muhammad Nauman Javid Ghuman, Zahoor Ali Khan, and Shujat Ur Rehman. 2019. Short term load forecasting using XGBoost. In *Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33*. Springer, 1120–1131.

[2] K Aishwarya, N Rao, Nikita Kumari, Akshit Mishra, and MR Rashmi. 2020. Food Demand Prediction using machine learning. *International Research Journal of Engineering and Technology (IRJET)* 7 (2020).

[3] Preeti Bhargava, Thomas Phan, Jiayu Zhou, and Juhan Lee. 2015. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In *WWW*. 130–140.

[4] Chen Cheng, Haiqin Yang, Irwin King, and Michael Lyu. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 26. 17–23.

[5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[6] Brendan Andrew Duncan and Charles Peter Elkan. 2015. Probabilistic modeling of a sales funnel to prioritize leads. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1751–1758.

[7] Yali Fan, Zhen Tu, Yong Li, Xiang Chen, Hui Gao, Lin Zhang, Li Su, and Depeng Jin. 2019. Personalized context-aware collaborative online activity prediction. *UbiComp* 3, 4 (2019), 1–28.

[8] Jamal Fattah, Latifa Ezzine, Zineb Aman, Haj El Moussami, and Abdeslam Lachhab. 2018. Forecasting of demand using ARIMA model. *International Journal of Engineering Business Management* 10 (2018), 1847979018808673.

[9] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 315–323.

[10] Luis Hernandez, Carlos Baladron, Javier M Aguiar, Belén Carro, Antonio J Sanchez-Esguevillas, Jaime Lloret, and Joaquim Massana. 2014. A survey on electric power demand forecasting: future trends in smart grids, microgrids and smart buildings. *IEEE Communications Surveys & Tutorials* 16, 3 (2014), 1460–1495.

[11] Jakob Huber, Alexander Gossmann, and Heiner Stuckenschmidt. 2017. Cluster-based hierarchical demand forecasting for perishable goods. *Expert systems with applications* 76 (2017), 140–151.

[12] Amir Mahmud Husein, Muhammad Arsyal, Sutrisno Sinaga, and Hendra Syahputa. 2019. Generative adversarial networks time series models to forecast medicine daily sales in hospital. *Sinkron: jurnal dan penelitian teknik informatika* 3, 2 (2019), 112–118.

[13] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. pmlr, 448–456.

[14] Zeynep Hilal Kilimci, A Okay Akyuz, Mitat Uysal, Selim Akyokus, M Ozan Uysal, Berna Atak Bulbul, and Mehmet Ali Ekmis. 2019. An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain. *Complexity* 2019 (2019).

[15] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[16] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[17] Balakrishnan Lakshmanan, Palaniappan Senthil Nayagam Vivek Raja, and Viswanathan Kalathiappan. 2020. Sales demand forecasting using LSTM network. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Springer, 125–132.

[18] Agnieszka Lasek, Nick Cercone, and Jim Saunders. 2016. Restaurant sales and customer demand forecasting: Literature survey and categorization of methods. *Smart City 360°: First EAI International Summit, Smart City 360°, Bratislava, Slovakia and Toronto, Canada, October 13-16, 2015. Revised Selected Papers 1* (2016), 479–491.

[19] Yinfeng Li, Chen Gao, Quanming Yao, Tong Li, Depeng Jin, and Yong Li. 2022. DisenHCN: Disentangled Hypergraph Convolutional Networks for Spatiotemporal Activity Prediction. *arXiv preprint arXiv:2208.06794* (2022).

[20] Ziyue Li, Nurettin Dorukhan Sergin, Hao Yan, Chen Zhang, and Fugee Tsung. 2020. Tensor completion for weakly-dependent data on graph for metro passenger flow prediction. In *proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 4804–4810.

[21] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2020. Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2636–2645.

[22] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1930–1939.

[23] Shaohui Ma, Robert Fildes, and Tao Huang. 2016. Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra-and inter-category promotional information. *European Journal of Operational Research* 249, 1 (2016), 245–257.

[24] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.

[25] Enrico Palumbo, Giuseppe Rizzo, Raphaël Troncy, Elena Baralis, Michele Osella, and Enrico Ferro. 2018. Knowledge graph embeddings with node2vec for item recommendation. In *The Semantic Web: ESWC 2018 Satellite Events: ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers 15*. Springer, 117–120.

[26] Rahul Priyadarshi, Akash Panigrahi, Srikanta Routroy, and Girish Kant Garg. 2019. Demand forecasting at retail stage for selected vegetables: a performance analysis. *Journal of Modelling in Management* 14, 4 (2019), 1042–1063.

[27] B Sri Sai Ramya and K Vedavathi. 2020. An advanced sales forecasting using machine learning algorithm. *International Journal of Innovative Science and Research Technology* 5, 5 (2020), 342–345.

[28] Kamran Raza. 2017. Prediction of Stock Market performance by using machine learning techniques. In *2017 International conference on innovations in electrical engineering and computational technologies (ICIEECT)*. IEEE, 1–1.

[29] Shuyun Ren, Hau-Ling Chan, and Tana Siqin. 2020. Demand forecasting in retail operations for fashionable products: methods, practices, and real case study. *Annals of Operations Research* 291 (2020), 761–777.

[30] Dennis Reynolds, Imran Rahman, and William Balinbin. 2013. Econometric modeling of the US restaurant industry. *International Journal of Hospitality Management* 34 (2013), 317–323.

[31] Kimberly F Sellers and Galit Shmueli. 2010. Predicting censored count data with COM-Poisson regression. *Robert H. Smith School Research Paper No. RHS-06-129* (2010).

[32] Mahya Seyedan and Fereshteh Mafakheri. 2020. Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities. *Journal of Big Data* 7, 1 (2020), 1–22.

[33] Juliana C Silva, Manuel C Figueiredo, and Ana C Braga. 2019. Demand forecasting: A case study in the food industry. In *Computational Science and Its Applications–ICCSA 2019: 19th International Conference, Saint Petersburg, Russia, July 1–4, 2019, Proceedings, Part III 19*. Springer, 50–63.

[34] Haiyan Song, Richard TR Qiu, and Jinah Park. 2019. A review of research on tourism demand forecasting: Launching the Annals of Tourism Research Curated Collection on tourism demand forecasting. *Annals of Tourism Research* 75 (2019), 338–362.

[35] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 269–278.

[36] Theodore Tskeris and Charalambos Tsekeris. 2011. Demand forecasting in transport: Overview and modeling advances. *Economic research-Ekonomska istraživanja* 24, 1 (2011), 82–94.

[37] Jiaxing Wang, GQ Liu, and Lu Liu. 2019. A selection of advanced technologies for demand forecasting in the retail industry. In *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)*. IEEE, 317–320.

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

[39] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang. 2020. Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 2377–2386.

[40] Yue Yu, Tong Xia, Huandong Wang, Jie Feng, and Yong Li. 2020. Semantic-aware spatio-temporal app usage representation via graph convolutional network. *UbiComp* 4, 3 (2020), 1–24.

[41] Vincent W Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Twenty-fourth AAAI conference on artificial intelligence*.

[42] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

A APPENDIX FOR REPRODUCIBILITY

A.1 Model Implementation Details

In this section we provide implementation details of our system. Due to the massive amount of training data and the high demand for low latency in our online system, we simplify each network within the framework. In our final implementation, feature merging network h^M is a one-layer fully-connected network of 120 output units (120D FC). User preference network h^U is a 340D FC. Shared network E^S and expert networks E^W/E^N are 256D FCs. Need prediction network t^N is a 10D FC. In-store/delivery classification network t^W is a two-layer perceptron, which has 10 hidden units and 2 output units. Complexifying these neural networks can help us further optimize performance, but at the same time, it would lead to an increase in system latency. The activation function of all mentioned networks is ReLU [9]. We adopt batch normalization [13] right after h^M, h^U, E^S, E^P, and E^N. Following such implementation, we conduct rich online and offline experiments to prove the effectiveness of our model, which are shown in Section 4 and Section 5.

A.2 Dataset

We conduct our offline experiment on a real-world dataset at the scale of billions. The dataset is a sampling of all purchase records in 2022 on the platform. We sample the records according to the percentage of purchases of each kind of life service. The dataset includes more than 7 billion real purchase records from 65 million users. Each instance in the dataset includes user profile, time, location, and other real-time environmental factors, and the kind of life service the user purchase. The type of life service consumed by the user reflects their actual living need in the spatiotemporal context. Following existing works [4, 25], we randomly sample 80% of the dataset as the training set, and 20% as the test set.

A.3 Metrics

We design a metric named Sort Accuracy (SA) to measure the performance of systems on our problem. The metric SA can be defined as follows. For user scene i, our system outputs scores for all kinds of living needs. We sort the living needs by their scores and get a list. We define *Relative Ranking Error* as the difference between the actual ranking position and the ideal ranking position (the first position) of the ground truth need, and further define *Maximum Ranking Error* as the maximum relative ranking error any system can give for a user scene (the number of categories of user needs - 1). For example, in our system which handles 10 types of living need, for user scene i, if the ground truth need is ranked third, then the relative ranking error is 2 (2 = 3-1), and the maximum ranking error is 9 (9 = 10-1). Then Sort Accuracy (SA) can be defined as follows,

$$\text{SA} = \text{average}_{i \in T} \left(1 - \frac{\text{Relative Ranking Error}}{\text{Maximum Ranking Error}}\right) \tag{14}$$

where T denotes the testing set on which the metric is calculated. We first calculate 1−Relative Ranking Error/Maximum Ranking Error of every user scene in the testing set and then compute the average. We then define Via-delivery Sort Accuracy (VDSA) and In-store Sort Accuracy (ISSA) to measure systems' performance on delivery and in-store living needs.

$$\text{VDSA} = \text{average}_{i \in T_{VD}} \left(1 - \frac{\text{Relative Ranking Error}}{\text{Maximum Ranking Error}}\right) \tag{15}$$

$$\text{ISSA} = \text{average}_{i \in T_{IS}} \left(1 - \frac{\text{Relative Ranking Error}}{\text{Maximum Ranking Error}}\right) \tag{16}$$

T_{VD} and T_{IS} denote sets of testing samples where the ground truth needs-meeting way is via delivery and in store, respectively. In the following, we will use these metrics to measure the living needs prediction performance of our system and baseline systems.

A.4 Baselines

To illustrate the effectiveness of our system, we compare it with two baselines widely in actual production environments, including **DIN [42]**, **DNN [5]**, **DCN [38]**, **ESMM [24]**, and **MMOE [22]**. We will provide a detailed description of these baselines in the appendix. DIN is a recommendation algorithm that leverages deep neural networks to analyze users' historical behavior and make predictions about their potential interests. It uses an attention-based mechanism to weigh the importance of different historical behaviors for predicting the current interest of a user. As for DNN, We follow the design of Wide & Deep learning to build a Deep Neural Network (DNN) based system for our task. It can learn both simple and complex relationships in the data. DCN is proposed to keep the benefits of a DNN model while introducing a cross network that is more efficient in learning certain bounded-degree feature interactions. It applies feature crossing at each layer, and it doesn't require manual feature engineering, adding minimal extra complexity to the DNN model. ESMM estimates post-click conversion rates for recommendation systems by using sequential user actions and a feature representation transfer learning strategy to alleviate sample selection bias and data sparsity. MMoE is a multi-task learning approach that learns to model task relationships from data. It adapts the Mixture-of-Experts (MoE) structure to multi-task learning by sharing the expert submodels across all tasks, while also having a gating network trained to optimize each task. For a clear comparison, the input features for all baselines are kept the same as those for our model.

A.5 Additional Ablation Study

To gain a deeper understanding of the impact of each component design of our model, we also conduct ablation studies with a particular focus on the effects of the group behavior pattern features and the spatiotemporal context features. When removing the group behavior pattern features from the model input, the performance decreased by 1.06%. When removing the spatiotemporal context features from the model input, the performance decreased by 1.46%. These results provide compelling evidence of the pivotal role that these two features play in accurately predicting users' daily living needs.