Predicting User Behavior in Smart Spaces with LLM-Enhanced Logs and Personalized Prompts

Yunpeng Song, Jiawei Li, Yiheng Bian, Zhongmin Cai*

MOE KLINNS Lab, Xi'an Jiaotong University, Xi'an, China
yunpengs@xjtu.edu.cn, zmcai@sei.xjtu.edu.cn

Abstract

Enhancing the intelligence of smart systems, such as smart homes, smart vehicles, and smart grids, critically depends on developing sophisticated planning capabilities that can anticipate the next desired function based on historical interactions. While existing methods view user behaviors as sequential data and apply models like RNNs and Transformers to predict future actions, they often fail to incorporate domain knowledge and capture personalized user preferences. In this paper, we propose a novel approach that incorporates LLM-enhanced logs and personalized prompts. Our approach first constructs a graph that captures individual behavior preferences derived from their interaction histories. This graph effectively transforms into a soft continuous prompt that precedes the sequence of user behaviors. Then our approach leverages the vast general knowledge and robust reasoning capabilities of a pretrained LLM to enrich the oversimplified and incomplete log records. By enhancing these logs semantically, our approach better understands the user's actions and intentions, especially for those rare events in the dataset. We evaluate the method across four real-world datasets from both smart vehicle and smart home settings. The findings validate the effectiveness of our LLM-enhanced description and personalized prompt, shedding light on potential ways to advance the intelligence of smart space.

Introduction

In today's society, smart devices have become an indispensable part of our daily lives, significantly enriching our life experiences. However, despite the variety of functionalities these smart devices offer, they largely operate independently of one another, lacking deep inter-connectivity. This isolation hinders the formation of truly "smart" systems. Take the smart vehicle scenario as an example: on a cold workday morning, a driver might open their car door, followed by the need to adjust the seat position, turn on the heating, set the navigation to plan a congestion-free route, and play their favorite music. While these functions are straightforward, manually executing each one can be tedious. Enabling a simple action, like opening a door, to automatically trigger all related tasks would significantly enhance the user experience in smart settings. Simplifying the interaction between users and smart devices to allow the devices to understand and automatically adapt to users' needs in various scenarios represents an urgent yet challenging goal (Desolda, Ardito, and Matera 2017).

This vision intersects with two main areas of research: human activity prediction and sequential recommendation. In human activity prediction, previous works in the context of smart space attempt to build models from historical interaction data between users and their environments, with the goal of automating the recognition of users' intentions and planning the next move (Tax 2018; Maharjan et al. 2019; Mohamed, El-Kilany, and El-Tazi 2022; Dunne, Morris, and Harper 2023). For instance, deep learning models have been employed to analyze various sensor data within smart homes to forecast subsequent user activities (Krishna et al. 2018). On the other hand, sequential recommendation focuses on the online activity context suggesting content that users may find engaging based on their behaviors (Hou et al. 2022; Li et al. 2023). This includes analyzing sequences of user online interactions, such as movie watching, web browsing, clicking, bookmarking, and purchasing, to tailor recommendations of potentially interesting content, like movies or products.

However, in real-world smart spaces, existing methods face challenges that differ markedly from those in recommendation systems. While recommendation systems can access extensive side information, which includes the names, brands, and attributes of purchased items (Li et al. 2023), and the content of movies that interest users, data logs from devices and sensors in smart spaces are often sparse or even incomplete. Consequently, earlier methods that employed one-hot encoding to represent different events as unrelated IDs frequently missed the inherent connections between related actions. For example, opening a refrigerator and preheating an oven are both linked to the intention of eating, yet are treated as unrelated by one-hot encoding. This approach fails to recognize the natural associations between activities, leading to a disconnect in representations, particularly for low-frequency actions that are poorly represented.

Additionally, due to the limited number of users in previous smart space datasets (Tax 2018; Park et al. 2022; Wang et al. 2022b; Dunne, Morris, and Harper 2023), prior research in smart spaces typically employs a single model to predict a range of user actions, neglecting the individ-

---

*Corresponding author.

ual preferences that result in behavioral differences. For instance, the routine of going to sleep could involve actions like drawing the curtains and switching off the lights for some, while for others, it might include playing soft music and setting an alarm. This variation highlights the inadequacy of a one-size-fits-all approach in predicting user actions without considering individual preferences, which is crucial for effective smart space automation.

To address the challenges in predicting user behavior within smart spaces, we propose a novel approach that incorporates LLM-enhanced logs and personalized prompts. This method taps into the vast general knowledge and robust reasoning capabilities of a pretrained LLM, using in-context learning to enrich the oversimplified and incomplete log records. By enhancing these logs semantically, we can better understand the user's actions and intentions. Additionally, our approach constructs a graph that captures individual behavior preferences derived from their interaction histories. This graph effectively transforms into a soft continuous prompt that precedes the sequence of user behaviors. Through prompt tuning, the model is guided to make varied predictions about user behavior based on this prompt.

We primarily evaluate our approach through experiments conducted on two large-scale, real-world datasets of user interactions with smart vehicles, complemented by two additional datasets focusing on smart home settings. The experimental outcomes demonstrate the superior performance of our approach when compared to state-of-the-art methods in terms of predicting a user's next move within smart vehicle contexts. By incorporating LLMs for description augmentation, our approach significantly boosts performance in scenarios involving low-frequency events. Furthermore, our framework demonstrates robust adaptability, effectively handling cold-start scenarios with new users and seamlessly transitioning to smart home contexts. The contributions are summarized as follows:

- We introduce a novel prompt-based approach for predicting user behavior, which adapts dynamically to individual user preferences, making it a pioneering application of prompt tuning for customized predictions within the smart space context.

- Utilizing a pre-trained LLM, our method significantly enhances event representations by augmenting them with semantic descriptions, particularly effective for addressing rare events.

- We validate our approach with extensive experiments on datasets from two different domains, demonstrating its effectiveness in accurately predicting user behavior.

Related Work

This section provides an overview of the existing research in three main areas: human activity prediction within smart environments, sequential recommendation systems for applications such as movies, e-commerce, and other services, and prompt tuning for recommendations.

Human Activity Prediction

Human activity prediction in smart environments aims to enhance the user experience by reducing the need for direct interaction and enabling more proactive support (Minor and Cook 2017; Tax 2018; Almeida and Azkune 2018; Yang et al. 2020). Research in this domain has primarily leveraged data-driven approaches, employing machine learning algorithms to analyze sensor-generated event logs that capture daily user interactions. These logs are annotated with activity labels to train prediction models. Early efforts (Minor, Doppa, and Cook 2015, 2017) utilized simple predictors with manually crafted features for prediction given sequences of past events. Advancements in the field have introduced sophisticated models like LSTM and GRU networks for discrete sequence modeling (Tax 2018; Krishna et al. 2018). Alternative methods employing CNN (Wang et al. 2022b) and DNN (Park et al. 2022) have been proposed to extract local features from the status of all sensors, using LSTM networks for the next activity prediction. Song et al. (Song et al. 2024a) further employed a Transformer decoder to incorporate the multi-source data for generating automation routines. However, these studies primarily overlook the semantic associations between signals and the variations in user behavior preferences by utilizing ID-based methods for modeling events and user profiles. In contrast, our approach enhances signal semantic descriptions using a pre-trained LLM and employs a prompt-based method to distinctively represent individual user behavior preferences.

Sequential Recommendation

Sequential recommendation systems are designed to anticipate a user's next point of interest, which could range from movies to products, by analyzing their past interactions (Wang et al. 2022a; Hou et al. 2023; Rajput et al. 2023; Dong et al. 2024). Early research in this field utilized Markov Chains to understand the sequence of item transitions (Shani et al. 2005). However, the integration of deep learning has led to the development of more advanced techniques. One of the first significant contributions in this area was GRU4Rec (Hidasi et al. 2015), which introduced RNNs to the sequential recommendation domain. SASRec (Kang and McAuley 2018) and Bert4Rec (Sun et al. 2019) applied transformers equipped with self-attention mechanisms to capture the dynamic interplay between items. Fastformer (Wu et al. 2021) utilizes additive attention to achieve linear complexity, accelerating the Transformer model. DuoRec (Qiu et al. 2022) and CL4SRec (Xie et al. 2022) explored contrastive learning to refine the process of learning item embeddings. More recently, UniSRec (Hou et al. 2022) and RecFormer (Li et al. 2023) incorporate textual attributes of products into models to enrich the representations of items further. However, these studies primarily concentrate on online behaviors, utilizing datasets with relatively uniform behavioral objectives such as Amazon Product[1] for purchasing or browsing items or MovieLens[2] for watching movies, which differ significantly from our focus.
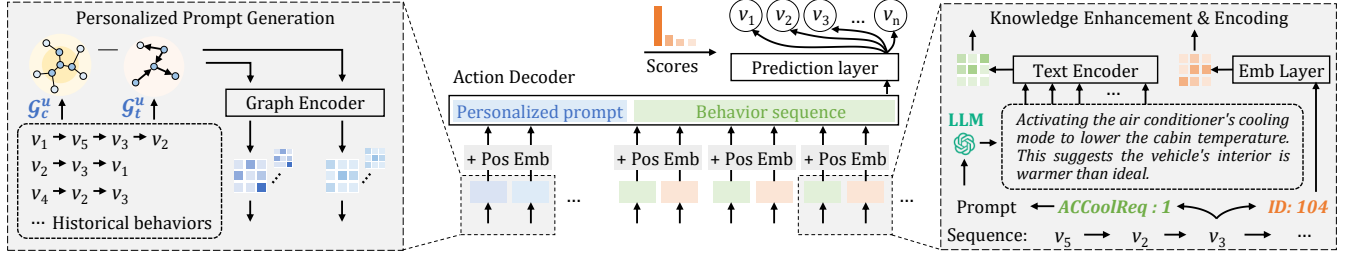
Figure 1: Overview of the proposed method for smart space interaction prediction. Our method constructs graphs from the user's historical sequences to model individual preferences and generate personalized prompts. It also augments event descriptions using an LLM's common knowledge.

Our research targets predicting actions within the context of smart spaces, encompassing a broader range of activities and involving more extended intentions.

Prompt Tuning

Prompt tuning is originally developed in the field of natural language processing, aiming to tailor input to help pretrained language models adapt to different downstream tasks. This approach has recently been employed for recommendation (Wang et al. 2023; Li, Zhang, and Chen 2023; Wu et al. 2024; Yang et al. 2024; Dong et al. 2024). For instance, P5 (Geng et al. 2022) integrated user–item information and corresponding features with personalized hard prompts to form a unified and instruction-based method for varied recommendation tasks. PEPLER (Li, Zhang, and Chen 2023) utilized item features and ID embeddings to generate recommendation explanations. PPR (Wu et al. 2024) built personalized soft prompt via a prompt generator based on user profiles for cold-start recommendation. However, while most existing methods depend on side information, such as user profiles like age and gender, to represent user preferences, our approach leverages historical interaction data to construct graphs capturing personalized user behaviors without relying on user profiles.

Method

Approach Overview

Problem Setup. In this study, we aim to predict a user's next action by analyzing their past interactions, framing this task as a multi-class classification challenge. We introduce the notion of a user set, denoted by $\mathcal{U}$, and an event set, $\mathcal{V}$. For a given user u, one of their interaction sequence is represented as $S_u = \{v_1, v_2, ..., v_{|S_u|}\}$, where $|S_u|$ donates the sequence length. Each event $v_t \in \mathcal{V}$ in this sequence, consisting of both the event name and its value, is identified by a unique ID. The event name, typically a brief, technical descriptor like "FLWdwClsReq" for "front left door close request", records the function of the event. Each event can have different values indicating various user actions and lead to different IDs; typically, a value of 0 signifies deactivation, while 1 indicates activation. Our objective is to leverage such historical sequences to forecast the user's next event, aiming to propose a list of the most probable future events ranked by their likelihood. This prediction is formulated as:

$$v_{|S_u|+1} = \mathrm{argmax}_{v \in \mathcal{V}} P(v|S_u),$$

where $P(v|S_u)$ denotes the probability of event v being the next action given the sequence of past events S_u.

Framework Overview. Smart spaces encompass a variety of functionalities, leading to diverse user behaviors within these settings. Understanding user intentions and predicting their actions to achieve a higher degree of automation is essential yet challenging. One major difficulty is that the log data of user interactions are often too simplistic or incomplete. For instance, entries like "ExCrtLtAutoCfg: 0" provide little insight into the semantics of the actions they represent. And there is significant variability in user behavior preferences; even with a single intent, different users follow varied operational sequences based on personal habits. Another challenge is the long-tail distribution of feature usage. Many functions are rarely used, which complicates the learning process for predictive models, as these underutilized features do not provide enough data to train effectively.

To address these challenges, we propose a data-driven approach that incorporates LLM- enhanced logs and personalized prompts. Initially, our approach constructs two graphs based on the user's historical interactions to capture individual preferences and generate personalized prompts. These prompts are inserted at the beginning of user behavior sequences to guide the model in making differentiated predictions tailored to individual characteristics. Then our approach uses an LLM to enrich event descriptions with detailed natural language explanations. For example, the code "ExCrtLtAutoCfg: 0" is expanded to clarify that it represents "disabling the automatic activation feature for the exterior courtesy lights." This enhancement preserves the semantic coherence among events in the embedding space, which is crucial for forming robust representations, especially for less frequent events. Additionally, a decoder then processes these inputs to predict the user's next likely action. The overall architecture of our framework is depicted in Figure 1.

Personalized Prompt Generator

Prompt tuning is typically used to bridge the gap between a pre-trained model's objectives and the specific requirements of a downstream task (Brown et al. 2020). Here, we employ prompt tuning to craft personalized prompts that inte-

grate individual users' behavioral patterns directly into the model. This customization enables the model to make predictions that truly reflect user-specific characteristics. NLP research has relied on hard prompt designing (Geng et al. 2022), which poses two main limitations for our application (Yang et al. 2024): first, unlike words in NLP, the tokens (event IDs) in our task lack clear, inherent meanings. Second, recommendation requires personalization, meaning prompts must be tailored to individual users. Considering the vast number of users in real-world systems, manually creating custom prompts for each user is infeasible.

To address the challenges, we have adopted a soft prompt based method by building interaction graphs and converting them into continuous prompts to reflect the user-level preferences. To effectively capture the distinct behavioral patterns of different users, we first construct a base graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ using historical event sequences. In this weighted directed graph, $\mathcal{V}$ represents all the events, and $\mathcal{E} = e_{ij}$, where each edge e_{ij} is weighted by the frequency of event v_i being followed by event v_j in the historical data. Building on this foundation, we then create personalized transition and co-occurrence graphs for each user to illustrate the dependencies among events in a user's behavior sequence.

The transition graph, denoted as $\mathcal{G}_t$, is a weighted directed graph designed to capture the specific sequential relationships between events. For user u, adjacent events v_i and v_j in the sequence are connected by an edge e_{ij}^t in $\mathcal{G}_t^u$ if the following conditions are met:

$$w_{ij} > (1 + \lambda)w_{ji}$$

Here, w_{ij} is the weight of the edge from v_i to v_j, and w_{ji} is the weight of the reverse edge. The parameter $\lambda \geq 0$ differentiates between transitional relationships (where one event significantly leads to another) and co-occurrence relationships (where events happen concurrently). In contrast, the co-occurrence graph, $\mathcal{G}_c$, serves a different function. It is a weighted undirected graph that identifies events likely to co-exist within the same sequences but without any implied order. An edge e_{ij}^c exists between v_i and v_j in $\mathcal{G}_c^u$ if

$$\max(w_{ij}, w_{ji}) \leq (1 + \lambda) \min(w_{ij}, w_{ji})$$

After constructing the transition graph $\mathcal{G}_t$ and the co-occurrence graph $\mathcal{G}_o$, we apply a graph convolutional neural network (GCN) to project the representations of events into the prompt embedding space. The GCN utilizes the following message passing method:

$$X_t^{(l+1)} = D_t^{-\frac{1}{2}} A_{\mathcal{G}_t^u} D_t^{-\frac{1}{2}} X_t^{(l)} W_t^{(l)}; \quad X_c^{(l+1)} = D_c^{-\frac{1}{2}} A_{\mathcal{G}_c^u} D_c^{-\frac{1}{2}} X_c^{(l)} W_c^{(l)}$$

Here, $X_t^{(l)}$ and $X_c^{(l)}$ represent the embedding matrices of events over the transition and co-occurrence graphs, respectively, at the l-th graph layer. D_t and D_c are degree matrices that normalize the graphs to ensure stable training and effective feature propagation. To further transform the embeddings into prompts, we utilize a two-layer perceptron $f(\cdot)$ to process X_t^u and X_c^u:

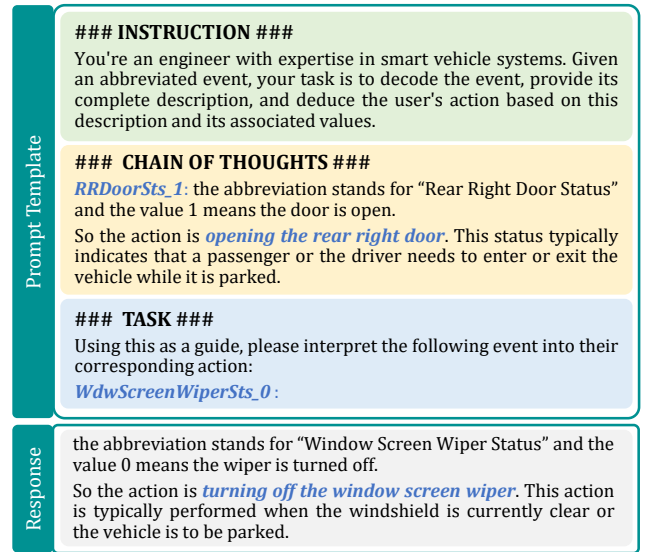$$P_t^u = f(X_t^u) = W_2 \cdot \sigma(W_1 \cdot X_t^u + b_1) + b_2$$

INSTRUCTION
You're an engineer with expertise in smart vehicle systems. Given an abbreviated event, your task is to decode the event, provide its complete description, and deduce the user's action based on this description and its associated values.

CHAIN OF THOUGHTS
RRDoorSts_1: the abbreviation stands for "Rear Right Door Status" and the value 1 means the door is open.
So the action is *opening the rear right door*. This status typically indicates that a passenger or the driver needs to enter or exit the vehicle while it is parked.

TASK
Using this as a guide, please interpret the following event into their corresponding action:
WdwScreenWiperSts_0 :

the abbreviation stands for "Window Screen Wiper Status" and the value 0 means the wiper is turned off.
So the action is *turning off the window screen wiper*. This action is typically performed when the windshield is currently clear or the vehicle is to be parked.

Figure 2: Enhancing an event log with LLM.

where W_1 and W_2 are the learnable weights, b_1 and b_2 are biases, and σ denotes the activation function.

The learned prompts act as prefixes, linked to the user behavior sequence as $S_u = \{p_{t_1}^u, p_{c_1}^u, \ldots, v_1^u, v_2^u, \ldots, v_{|S_u|}^u\}$. This arrangement enhances the input representations by incorporating user-specific context. It is important to note that based on the outputs from the GCN, both P_t^u and P_c^u may include multiple entries.

Enhancing Logs with LLM

Sensor event logs in smart space are commonly recorded for monitoring or debugging purposes (Cook, Augusto, and Jakkula 2009). These logs essentially reflect user behavior and can effectively assist in understanding user intentions. However, these logs are typically concise. For instance, using "FLWwClsReq" to represent "front left window close request" in our user-vehicle interaction dataset. This is because they are often reviewed and processed by professionals from service providers. Traditional methods tend to overlook the incomplete textual descriptions of events, opting instead to assign different IDs to recorded log signals to create event embedding tables for encoding (Tax 2018; Park et al. 2022; Wang et al. 2022b). However, by reconstructing these incomplete textual descriptions, we can more accurately depict the relationships between events from a semantic standpoint, rather than merely relying on their sequence of occurrence. This is particularly useful for long-tail infrequent events. For example, even if the event "RLWwClsReq" appears rarely and is hard to learn for a model, we can deduce its close semantic similarity to "FLWwClsReq" based on their textual information.

We leverage the common sense and reasoning capabilities of a pre-trained LLM to enhance the abbreviated textual descriptions of log events. The training corpora for these models often include a vast array of codes, which feature similar signal abbreviations and annotated descriptions ex-

plaining their meanings. This rich dataset enables LLMs to accurately interpret the semantics of various code abbreviations found in logs. Specifically, we set the LLM's role as an expert engineer in intelligent automotive log analysis. Using a Chain of Thought approach (Brown et al. 2020), we construct prompts that enable the LLM to output possible user actions based on the event logs. Given an event with a description and value, such as "RRDoorSts: 1", the LLM reconstructs a more complete description of the event, like "rear right door status". Subsequently, it infers user actions based on the description and value; for example, "1" indicates the door is open, leading to the inference that the user action is opening the rear right door. The process of prompting is illustrated in Fig. 2. We employed both proprietary Ernie Bot and the open-source Qwen2.5-32B as the LLM to enhance the textual descriptions of all events (description-value pairs) in our datasets, thereby gaining a deeper understanding of the behavioral semantics behind these events.

After the LLM augmentation, each event is associated with a multi-word natural language description. Therefore, we introduce an encoder model (BERT (Devlin et al. 2018)) to learn and utilize these enhanced natural language descriptions for forming representational vectors of events. Given an event with an LLM-enhanced description $\{d_i^1, d_i^2, \cdots, d_i^m\}$ (where m is the number of words in the description), we treat this as a sentence and add a special token [CLS] at the beginning. Consequently, the input to the encoder model is formulated as:

$$T = \{[\text{CLS}], D_i^1, D_i^2, \cdots, D_i^m\}$$

Utilizing BERT's bidirectional transformer architecture, the [CLS] token integrates information from all words in the description and can be used as the textual representation (t_i) of the augmented event.

Sequence Decoding

We treat event prediction as a decoding task with personalized prompts, where given the representations of events in a sequence, a sequence decoder interprets user intent and captures behavioral patterns to predict the next move. In this process, we employ a transformer decoder endowed with self-attention mechanisms to aggregate sequence information (Radford et al. 2019). The decoder comprises several blocks; each block includes a masked multi-headed self-attention layer (MHAttn(·)) and point-wise feed-forward networks (FFN(·)). The use of masks ensures that each prediction is influenced only by prior known outputs. The decoder processes inputs by summing the learned embeddings (v_i) and position embeddings (p_j) at position j. The computation evolves as follows:

$$f_j^0 = v_i + p_j$$

$$F^{l+1} = \text{FFN}(\text{MHAttn}(F^l))$$

Here, $F^l = [f_0^l; \dots; f_n^l]$ represents the l-th layer's position-specific representations. We derive the sequence representation from the final hidden state f_n^L at the n-th position, across a total of L layers in the decoder.

To effectively differentiate between likely and unlikely subsequent events in a fine-grained manner, we employ a pairwise ranking loss (BPR loss) to optimize the decoder (Rendle et al. 2012). Considering a user's historical event sequence $S_u = \{v_1, v_2, \ldots, v_{m-1}\}$, the positive sample i is the target next event, namely v_m. The negative sample j is randomly selected from events not appearing next in the user's interaction data following v_{m-1}. Thus, the ranking loss function is defined as follows:

$$\mathcal{L} = -\sum_{j=1}^{B} \log \sigma(y_m^i - y_m^j)$$

Here, σ denotes the sigmoid activation function, and y_m^i and y_m^j represent the likelihood scores of the user u's next event being i and j, respectively, given S_u. The events with the highest scores will be output as the top-k candidates in the model's prediction list.

Evaluation

In this section, we outline the experimental setup, followed by the presentation of results and analysis in response to the following research questions.
RQ1: Does our method outperform the SOTA methods?
RQ2: What is the impact of varying parts in the framework?
RQ3: Does our method improve rare event performance?
RQ4: How do hyper-parameters affect model performance?

Experimental Setup

Dataset. For our evaluation, we use two industrial datasets of user-smart vehicle interactions and two public datasets of user-smart home interactions. The vehicle datasets contain logs from 100 randomly sampled private smart vehicles of the same model in two local cities. Each log entry includes the vehicle's ID, signal name and value, and timestamp. These datasets comprise 9.5 million events over 90 days, covering 357 sensors with 487 types of events. The sensors are categorized into comfort (129), infotainment (76), monitoring (61), safety (48), and control (43). They track everyday activities like gear changes and air conditioning adjustments. To protect privacy, vehicle IDs are anonymized using random strings, and sensitive information such as locations is omitted. An ethics board from a local institution has approved the dataset. More details can be found in (Song et al. 2024b). The two public smart home datasets Aruba and Milan come from the WSU CASAS project, which are commonly used in the smart home activity prediction (Park et al. 2022). Similar to the vehicle data, these logs capture user-triggered events with information on sensor location, name, readings, and timestamp. The sensors monitor various household items including refrigerators, containers, lights, motion detectors, toilets, ovens, curtains, and doors.

Evaluation Settings. The log data in our datasets exhibit a distinct long-tail distribution, with high-frequency signal events occurring much more frequently than low-frequency ones. Consequently, widely-used metrics in recommendation systems, like *Hit Ratio* and *Normalize Discounted Cumulative Gain*, become less effective as they are heavily influenced by the prediction results of high-frequency signals.

Sources	Datasets	Metrics	CasLSTM	DnnLSTM	CovLMLC	ASGen	BERT4Rec	DuoRec	CL4SRec	RecFormer	Ours
Industrial	City A	H@1	0.1987	0.1794	0.2231	0.3018	0.2715	0.3308	0.3286	0.3191	**0.3450***
		H@3	0.3413	0.3075	0.3670	0.4852	0.4356	0.5114	0.5228	0.5159	**0.5529***
		N@3	0.2817	0.2540	0.3068	0.4065	0.3629	0.4358	0.4427	0.4324	**0.4596***
	City B	H@1	0.2188	0.2043	0.2286	0.3453	0.3044	0.3497	0.3517	0.3230	**0.3743***
		H@3	0.3808	0.3543	0.4041	0.5270	0.4567	0.5315	0.5661	0.5387	**0.5904***
		N@3	0.2974	0.2864	0.3225	0.4640	0.4256	0.4511	0.4589	0.4389	**0.4925***
Public	Home A	H@1	0.2684	0.2478	0.2737	0.3598	0.3508	0.4015	0.3691	0.4103	**0.4293***
		H@3	0.4263	0.4094	0.4383	0.5390	0.5298	0.5634	0.5287	0.5731	**0.5974***
		N@3	0.3349	0.3339	0.3573	0.4685	0.4515	0.4973	0.4688	0.5093	**0.5286***
	Home B	H@1	0.1842	0.1954	0.2032	0.3071	0.2878	0.3026	0.3145	0.3118	**0.3352***
		H@3	0.3497	0.3517	0.3582	0.4715	0.4523	0.4733	0.4842	0.4859	**0.5115***
		N@3	0.2681	0.2764	0.2842	0.4127	0.3909	0.4126	0.4281	0.4314	**0.4487***

Table 1: Comparison across datasets. * denotes our results are significantly better than the best baselines (p < 0.05).

Embedding			PT	H@1		H@3	
ID	Log_Q	Log_E		City A	City B	City A	City B
✓				0.2774	0.3056	0.4835	0.5186
	✓			0.3064	0.3298	0.5125	0.5507
		✓		0.3097	0.3354	0.5172	0.5551
✓	✓			0.3140	0.3396	0.5209	0.5584
✓		✓		0.3176	0.3427	0.5247	0.5623
✓	✓		✓	0.3439	0.3722	0.5495	0.5871
✓		✓	✓	**0.3450**	**0.3743**	**0.5529**	**0.5904**

Table 2: Ablation Study. Log_Q denotes enhanced logs generated by Qwen, and Log_E by Ernie Bot.

Therefore, we adopt class-wise metrics, specifically class-wise Hit Rate@K (abbreviated as $H@K$) and class-wise NDCG@K ($N@K$). $H@K$ calculates the likelihood of the target event appearing in the Top-K recommendations on average for each event class while $N@K$ evaluates the ranking of the target event within the Top-K list. Considering practical applications like providing real-time shortcut recommendations to users, a larger K could significantly increase cognitive load and reduce usability. Hence, we focus on smaller values of K, specifically $K \in \{1, 3\}$. This choice reflects a realistic scenario where a user selects a needed function from a set of three suggested options. Ernie bot is used to enhance the logs in this experiment.

To realistically simulate the scenario of actual deployment and prevent data leakage, we divided our dataset into training, validation, and testing sets based on time. Each dataset is divided into the training set, validation set, and test set in a ratio of 2:1:7 in chronological order. For the validation and testing sets, each user's interaction logs, often comprising tens of thousands of records, were segmented into non-overlapping sequences using a sliding window of length 6. Each sequence was constructed using the first 5 events to predict the 6th event.

Implementation Details. We trained our networks using the Adam optimizer (Kingma and Ba 2014), with hyperparameter optimization conducted on the validation set. The mini-batch size was fixed at 256, and we set the embedding size at 256 for tokens in prompts and behavior sequences. The length of interaction sequences ranged from 1 to 10. During the training phase, the learning rate was adjusted within a range of $[5e^{-3}, 5e^{-4}, 5e^{-5}, 5e^{-6}]$, and weight decay parameters were explored within $[1e^{-3}, 1e^{-4}, 1e^{-5}]$. Additionally, we experimented with different prompt token numbers K, specifically tuning them within the range of $[2, 4, 6, 8, 10]$. The network training extended to a maximum of 100 epochs, a duration typically sufficient to ensure convergence. The model selection was based on the best performance observed on the validation set. All networks were developed using PyTorch and Huggingface frameworks and were run on an RTX 4090 GPU. Each experiment is repeated 20 times with different seeds.

Overall Performance (RQ1)

Baselines. We compare our method against eight related baseline methods using the real-world smart vehicle dataset. These methods are divided into two categories: 1) algorithms for human activity prediction in smart space: CasLSTM (Krishna et al. 2018), DnnLSTM (Park et al. 2022), ConvLMLC (Wang et al. 2022b), and ASGen (Song et al. 2024a); 2) sequential recommendation algorithms applicable to scenarios such as movie watching and online shopping: BERT4Rec (Sun et al. 2019), DuoRec (Qiu et al. 2022), CL4Rec (Xie et al. 2022), and RecFormer (Li et al. 2023). We utilized official implementations for BERT4Rec, DuoRec, CL4Rec, and RecFormer, while for the other models, we utilized PyTorch to replicate these methods as per the specifications provided in their original paper. Our code is included in the supplementary material.

Results. Table 1 shows that our method demonstrated superior performance on all metrics across four datasets compared to existing approaches. We attribute these improvements to the fact that our enhanced representation effectively incorporates semantic information, relationships, and timing of events for prediction, allowing for a more nuanced understanding of user behavior patterns. Furthermore, while both RecFormer and our method utilize LLM-enhanced textual
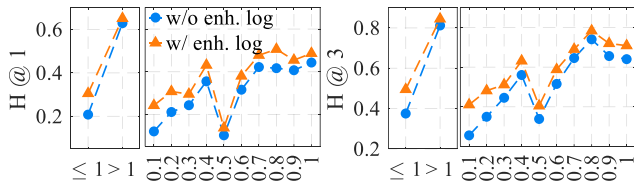
Figure 3: Impact of event frequency on prediction performance. The first and third panels categorize events into two groups based on whether their frequency exceeds 1%. The second and fourth panels further divide events with frequencies below 1% into ten groups using 0.1% intervals.

information, our approach also integrates this information with personalized prompts to guide the model and make predictions tailored for the target user. This integration results in a more effective model than RecFormer.

Ablation Study (RQ2)

Our approach utilizes prompt tuning with two types of embeddings: ID-based and enhanced-log. The enhanced logs are generated using either the Qwen or Ernie bot. We conducted an analysis across seven experimental setups, each featuring different combinations of modules, as shown in Table 2. Our findings reveal that using only event IDs to understand the sequential relationships in log sequences yields a hit rate (H@1) of 0.2774 and 0.3056. Utilizing LLM-enhanced semantics textual description to create text embeddings enhances the accuracy to 0.3176 and 0.3427. Notably, the enhanced logs from both Qwen and Ernie bot exhibit high similarity, leading to comparable performance across both models. Incorporating personalized prompts further boosts performance, reaching a hit rate of 0.3450 and 0.3743 H@1. The ablation study clearly demonstrates the significant effectiveness of our LLM-enhanced descriptions and personalized prompts in enhancing predictive accuracy.

Performance of Events over Frequencies (RQ3)

Generally, for datasets with a significant long-tail distribution in event frequencies. Common events, due to their sufficient samples, tend to be learned more effectively compared to rare events that suffer from insufficient training data. To explore the impact of event frequency on model performance, we divided the events of the vehicle datasets into two main groups based on their occurrence in the log records: events with frequencies above 1% and those below 1%. For the latter, we further segmented the data into ten subgroups at 0.1% intervals, as depicted in Fig. 3. Our analysis focused on comparing the model's predictive accuracy with and without the use of LLM-enhanced descriptions (i.e., text embedding). The results (the 1st and 3rd panels in Fig. 3) clearly show that for events occurring less than 1% of the time, LLM-enhanced descriptions significantly improve the model's predictions, with increases of over 40% in H@1 and 30% in H@3. Conversely, for more frequent events, the improvements, although present, were modest. Further detailed observation of the subdivided frequencies below 1% reveals that events with frequencies under 0.2% saw increases in
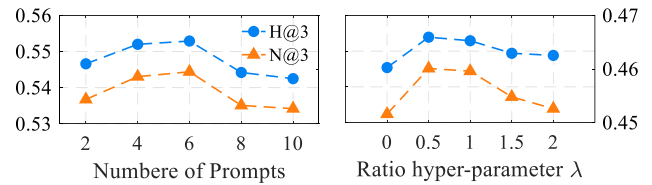

Figure 4: Impact of different hyper-parameters.

H@1 and *H@3* by about 0.1 or more. These findings demonstrate the effectiveness of our LLM-enhanced description for log data, especially for low-frequency events.

Hyper-parameter Analysis (RQ4)

With the smart vehicle datasets, we analyzed the effects of different parameters: the number of prompts and the λ parameter used during graph construction. Fig. 4 illustrates a mild increase in both H@3 and N@3 metrics as the number of prompts rises, achieving peak performance with six prompts. This improvement is likely due to the ability of additional prompts to more accurately capture user preferences in detail. In contrast, the impact of the λ parameter was subtler. Optimal results were attained with λ set to 0.5, as higher λ values tend to overlook important sequential relationships, whereas lower values struggle to effectively differentiate between sequential and co-occurrence relationships. These results indicate that the model is quite robust, showing limited sensitivity to variations in these parameters.

Discussion

Our approach does not impose computational costs associated with LLMs during both training and inference. LLMs are only used in the preprocessing stage to enhance incomplete signals, and the enhanced descriptions are stored. During training and inference, the model simply retrieves these representations, avoiding the need for further LLM calls. This method minimizes LLM usage, limiting it to the number of unique signals, making it well-suited for large-scale log data. Furthermore, in our experiments, the inference time for a single sequence of length 10 is less than 0.1s on an NVIDIA 4090 GPU.

Conclusion

In this paper, we propose a novel approach that incorporates LLM-enhanced logs and personalized prompts. Our approach constructs a graph that captures individual behavior preferences derived from their interaction histories, which effectively transforms into a soft continuous prompt to precede the sequence of user behaviors. our approach then leverages the vast general knowledge and robust reasoning capabilities of a pretrained LLM to enrich the oversimplified and incomplete log records. Our methodology's effectiveness has been validated through rigorous evaluation on large-scale datasets involving user interactions with smart vehicles and smart homes, demonstrating the applicability of our framework in smart environment settings.

Acknowledgments

References

Almeida, A.; and Azkune, G. 2018. Predicting human behaviour with recurrent neural networks. *Applied Sciences*, 8(2): 305.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Cook, D. J.; Augusto, J. C.; and Jakkula, V. R. 2009. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and mobile computing*, 5(4): 277–298.

Desolda, G.; Ardito, C.; and Matera, M. 2017. Empowering end users to customize their smart environments: model, composition paradigms, and domain-specific tools. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24(2): 1–52.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, X.; Song, X.; Liu, T.; and Guan, W. 2024. Prompt-based Multi-interest Learning Method for Sequential Recommendation. *arXiv preprint arXiv:2401.04312*.

Dunne, R.; Morris, T.; and Harper, S. 2023. A semantic blocks model for human activity prediction in smart environments using time-windowed contextual data. *Journal of Reliable Intelligent Environments*, 9(1): 87–95.

Geng, S.; Liu, S.; Fu, Z.; Ge, Y.; and Zhang, Y. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, 299–315.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Hou, Y.; He, Z.; McAuley, J.; and Zhao, W. X. 2023. Learning vector-quantized item representation for transferable sequential recommenders. In *Proceedings of the ACM Web Conference 2023*, 1162–1171.

Hou, Y.; Mu, S.; Zhao, W. X.; Li, Y.; Ding, B.; and Wen, J.-R. 2022. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 585–593.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krishna, K.; Jain, D.; Mehta, S. V.; and Choudhary, S. 2018. An lstm based system for prediction of human activities with durations. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4): 1–31.

Li, J.; Wang, M.; Li, J.; Fu, J.; Shen, X.; Shang, J.; and McAuley, J. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. *arXiv preprint arXiv:2305.13731*.

Li, L.; Zhang, Y.; and Chen, L. 2023. Personalized prompt learning for explainable recommendation. *ACM Transactions on Information Systems*, 41(4): 1–26.

Maharjan, R.; Shiraishi, K.; Yamamoto, T.; Yamamoto, Y.; and Ohshima, H. 2019. Development of IoT monitoring device and prediction of daily life behavior. In *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*, 584–588.

Minor, B.; and Cook, D. J. 2017. Forecasting occurrences of activities. *Pervasive and mobile computing*, 38: 77–91.

Minor, B.; Doppa, J. R.; and Cook, D. J. 2015. Data-driven activity prediction: Algorithms, evaluation methodology, and applications. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 805–814.

Minor, B. D.; Doppa, J. R.; and Cook, D. J. 2017. Learning activity predictors from sensor data: Algorithms, evaluation, and applications. *IEEE transactions on knowledge and data engineering*, 29(12): 2744–2757.

Mohamed, M.; El-Kilany, A.; and El-Tazi, N. 2022. Future Activities Prediction Framework in Smart Homes Environment. *IEEE Access*, 10: 85154–85169.

Park, S.; Lee, H. O.; Hwang, Y. M.; Ko, S.-K.; and Lee, B.-T. 2022. Enhanced Prediction Model for Human Activity Using an End-to-End Approach. *IEEE Internet of Things Journal*.

Qiu, R.; Huang, Z.; Yin, H.; and Wang, Z. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, 813–823.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Rajput, S.; Mehta, N.; Singh, A.; Keshavan, R. H.; Vu, T.; Heldt, L.; Hong, L.; Tay, Y.; Tran, V. Q.; Samost, J.; et al. 2023. Recommender Systems with Generative Retrieval. *arXiv preprint arXiv:2305.05065*.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Shani, G.; Heckerman, D.; Brafman, R. I.; and Boutilier, C. 2005. An MDP-based recommender system. *Journal of Machine Learning Research*, 6(9).

Song, Y.; Bian, Y.; Wang, X.; and Cai, Z. 2024a. Learning from User-driven Events to Generate Automation Sequences. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(4): 1–22.

Song, Y.; Li, J.; Bian, Y.; and Cai, Z. 2024b. Dataset description. https://arxiv.org/abs/2412.12653.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 1441–1450.

Tax, N. 2018. Human activity prediction in smart home environments with LSTM neural networks. In *2018 14th International Conference on Intelligent Environments (IE)*, 40–47. IEEE.

Wang, C.; Wang, Z.; Liu, Y.; Ge, Y.; Ma, W.; Zhang, M.; Liu, Y.; Feng, J.; Deng, C.; and Ma, S. 2022a. Target Interest Distillation for Multi-Interest Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2007–2016.

Wang, W.; Li, J.; Li, Y.; and Dong, X. 2022b. Predicting Activities of Daily Living for the Coming Time Period in Smart Homes. *IEEE Transactions on Human-Machine Systems*.

Wang, Y.; Zhao, X.; Chen, B.; Liu, Q.; Guo, H.; Liu, H.; Wang, Y.; Zhang, R.; and Tang, R. 2023. PLATE: A Prompt-Enhanced Paradigm for Multi-Scenario Recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1498–1507.

Wu, C.; Wu, F.; Qi, T.; Huang, Y.; and Xie, X. 2021. Fastformer: Additive attention can be all you need. *arXiv preprint arXiv:2108.09084*.

Wu, Y.; Xie, R.; Zhu, Y.; Zhuang, F.; Zhang, X.; Lin, L.; and He, Q. 2024. Personalized Prompt for Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.; Ding, B.; and Cui, B. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, 1259–1273. IEEE.

Yang, H.; Gong, S.; Liu, Y.; Lin, Z.; and Qu, Y. 2020. A multi-task learning model for daily activity forecast in smart home. *Sensors*, 20(7): 1933.

Yang, H.; Zhao, X.; Li, Y.; Chen, H.; and Xu, G. 2024. An Empirical Study Towards Prompt-Tuning for Graph Contrastive Pre-Training in Recommendations. *Advances in Neural Information Processing Systems*, 36.