LIMP: Large Language Model Enhanced Intent-aware Mobility Prediction

Songwei Li, Jie Feng, Jiawei Chi, Xinyuan Hu*, Xiaomeng Zhao, Fengli Xu
Department of Electronic Engineering, Tsinghua University, Beijing, China,
***Quantitative Method and Theory, Emory University, USA,**
{fengjie,fenglixu}@tsinghua.edu.cn

Abstract

Human mobility prediction is essential for applications like urban planning and transportation management, yet it remains challenging due to the complex, often implicit, intentions behind human behavior. Existing models predominantly focus on spatiotemporal patterns, paying less attention to the underlying intentions that govern movements. Recent advancements in large language models (LLMs) offer a promising alternative research angle for integrating commonsense reasoning into mobility prediction. However, it is a non-trivial problem because LLMs are not natively built for mobility intention inference, and they also face scalability issues and integration difficulties with spatiotemporal models. To address these challenges, we propose a novel LIMP (LLMs for Intent-ware Mobility Prediction) framework. Specifically, LIMP introduces an "Analyze-Abstract-Infer" (A2I) agentic workflow to unleash LLM's commonsense reasoning power for mobility intention inference. Besides, we design an efficient fine-tuning scheme to transfer reasoning power from commercial LLM to smaller-scale, open-source language model, ensuring LIMP's scalability to millions of mobility records. Moreover, we propose a transformer-based intention-aware mobility prediction model to effectively harness the intention inference ability of LLM. Evaluated on two real-world datasets, LIMP significantly outperforms baseline models, demonstrating improved accuracy in next-location prediction and effective intention inference. The interpretability of intention-aware mobility prediction highlights our LIMP framework's potential for real-world applications.

Introduction

Predicting human mobility behavior is a crucial task with significant implications for various domains, including urban planning, transportation management, and public safety. However, the inherent complexity of human mobility poses substantial challenges, especially the implicit intentions that are often not directly observable. Previous studies have shown that human researchers can infer the intention of human movements with high accuracy by examining their spatiotemporal trajectory (Jiang et al. 2016; Liccardi, Abdul-Rahman, and Chen 2016). However, it is not scalable to ask human researchers to manually label mobility data. Thus, most of existing mobility prediction models (Liu et al. 2016; Feng et al. 2018; Sun et al. 2020; Luo, Liu, and Liu 2021; Yang, Liu, and Zhao 2022) focus on capturing spatiotemporal patterns using advanced recurrent network and attention models. While these methods have shown promise, they often fail to effectively model the underlying intentions that drive each movement. This limitation highlights the need for new approaches that can incorporate a deeper understanding of human behavior.

Recent advancements in large language models (LLMs) have demonstrated emergent capabilities in commonsense reasoning (Wei et al. 2022a,c), offering a novel research angle for intention-aware mobility prediction. Despite this promise, several challenges remain in leveraging LLMs for mobility prediction. First, LLMs are not inherently optimized for inferring behavioral intentions from spatiotemporal data. Directly prompting LLMs to interpret human movements often yields inaccurate or overly generalized results, limiting their utility in mobility prediction. Second, the massive size and proprietary nature of state-of-the-art LLMs, such as GPT-4 (Achiam et al. 2023), present practical challenges, including high API costs and the inability to deploy these models locally. These factors hinder the scalability of LLM-based mobility prediction in real-world applications. Third, the domain-specific nature of spatiotemporal deep learning models and LLMs creates a disconnect, making it unclear how to effectively integrate the two to enhance prediction accuracy.

In response to these challenges, we propose a novel framework, LIMP (LLMs for Intent-ware Mobility Prediction), designed to harness the commonsense reasoning abilities of LLMs for intention-aware mobility prediction. The framework comprises three key components. First, we introduce an "Analyze-Abstract-Infer" (A2I) agentic workflow that guides LLMs through the process of mobility intention inference in a principal manner, which emulates the methodology of human expert annotators. The A2I workflow enables LLMs to reason through the intentions behind movements step-by-step: analyzing notable features, abstracting high-level insights, and inferring the most likely intention with a comprehensive assessment. Second, we propose an efficient fine-tuning scheme that transfers the reasoning power of large commercial LLMs, such as GPT-4, to smaller, locally deployable models like Llama 3 (Dubey et al. 2024). This approach ensures that our framework can scale to handle millions of mobility records at low cost. Finally, we design a transformer-based intention-aware mobility prediction model that effectively integrates the inferred intentions from LLMs to enhance next-location prediction.

Our framework is evaluated on two real-world datasets, demonstrating significant improvements over baseline models. Specifically, LIMP achieves a 6.64% to 9.52% increase in top-1 accuracy for next-location prediction compared to state-of-the-art baselines. Additionally, the A2I workflow boosts LLMs' mobility intention inference accuracy by 16.28%. The fine-tuning scheme successfully transfers the reasoning abilities of GPT-4 to Llama 3 without any statistically significant performance loss. Ablation studies further validate the effectiveness of our design choices.

In summary, our contributions are threefold.

- We introduce a novel framework, LIMP, that leverages the commonsense reasoning power of LLMs for enhanced mobility prediction, incorporating intention inference to improve both performance and interpretability.
- We develop a fine-tuning strategy that enables the deployment of high-performing, cost-effective, and scalable models by transferring capabilities from large, proprietary LLMs to smaller, open-source alternatives.
- We provide extensive empirical validation on real-world datasets, demonstrating the robustness and applicability of our approach to practical scenarios.

Method

Overview

As Figure 1 shows, LIMP contains three main steps ranging from GPT-4o based A2I agentic workflow for zero-shot intent annotation for mobility trajectory, agentic workflow fine-tuning framework for distilling the capabilities from GPT-4o to smaller LLMs (e.g., Llama3-8B), and intent enhanced mobility prediction model. Detailed design of three steps are introduce in the following sections.

Intent Annotation via A2I Agentic Workflow

In this section, we present a novel agentic workflow specifically developed to leverage the commonsense reasoning capabilities of LLMs, such as GPT-4o, for inferring user intentions from mobility data. Recognizing that LLMs inherently lack specialized capabilities for mobility intention inference, we proposed the **Analyze-Abstract-Infer (A2I) Agentic Workflow**, a systematic approach inspired by methodologies employed by human researchers for annotating trajectories and designing algorithmic processes (Zeng et al. 2017; Chen and Poorthuis 2021; Liccardi, Abdul-Rahman, and Chen 2016; Jiang et al. 2016). We designed A2I with three sequential steps following the idea of Chain-of-Thought (CoT) (Wei et al. 2022b), intent feature analysis, summative insights generation and user intentions inference. Detailed designs of them are introduce below.

Expert-Knowledge Guided Intent Feature Analysis Our approach began with analyzing the expert-knowledge statistics derived from a small, human-labeled dataset. Inspired by the TimeGeo modeling framework for urban mobility (Jiang et al. 2016), we identified features that are indicative of user behavior patterns and essential for guiding LLMs in the task of mobility intention inference by providing necessary expert knowledge. The features analyzed include:

- Percentage Distribution: We computed the relative frequency of specific intents—*'At Home', 'Working', 'Running Errands'*—to capture the overall distribution of user behaviors in the dataset.
- Average Visit Frequency: This metric quantified the frequency with which users visited specific Points of Interest (POIs), providing insights into habitual behaviors and their associated intents.
- Time Distribution: For each of the six intents—*'At Home', 'Working', 'Running Errands', 'Eating Out', 'Leisure and Entertainment', 'Shopping'*—we calculated the temporal distribution of visits, allowing us to model typical user behavior patterns across different times of the day.

These knowledge served as the foundation for generating summative insights.

Summative Insights Generation Human researchers are adept at synthesizing statistical data into high-level insights that can be generalized across different scenarios and tasks (Zhao et al. 2024). Emulating this capability, we leveraged the commonsense reasoning abilities of LLMs to generate n summative insights from the previously identified features that could aid LLMs in differentiating user intents in the subsequent process. For instance, the LLM was tasked with identifying key patterns, such as the peak activity times for different intents or the typical working hours for users. Then, we transformed the larger dataset's raw mobility data into a structured format suitable for intent prediction, which: first divided the trajectory data into daily segments for each user, capturing the sequence of POIs visited along with key metadata; and then computed the visit frequency and time distribution for each POI, establishing a statistical basis for context-aware prediction.

User Intentions Inference Finally, we used in-context learning (ICL) to synthesize the insights derived from feature extraction and statistical analysis into a cohesive intent inference process:

- Home and Workplace Identification (HWI): Leveraging the high-level insights generated from *intent feature analysis*, combined with the POI statistics from *summative insights generation* into the prompt, we applied ICL to guide GPT-4o in predicting the locations most likely corresponding to the user's home and workplace. This initial prediction step was conducted once per user at the outset of the prediction process.
- Daily Intent Prediction: For each stay within the user's daily trajectory, GPT-4o predicted the user's intent by considering the identified home and workplace locations, the sequence of POIs, and the time of day. By using ICL, the model could apply its understanding of typical behaviors to infer intents in novel situations.

By first identifying the home and workplace, and instructing the LLM to maintain consistency with these predicted loca-

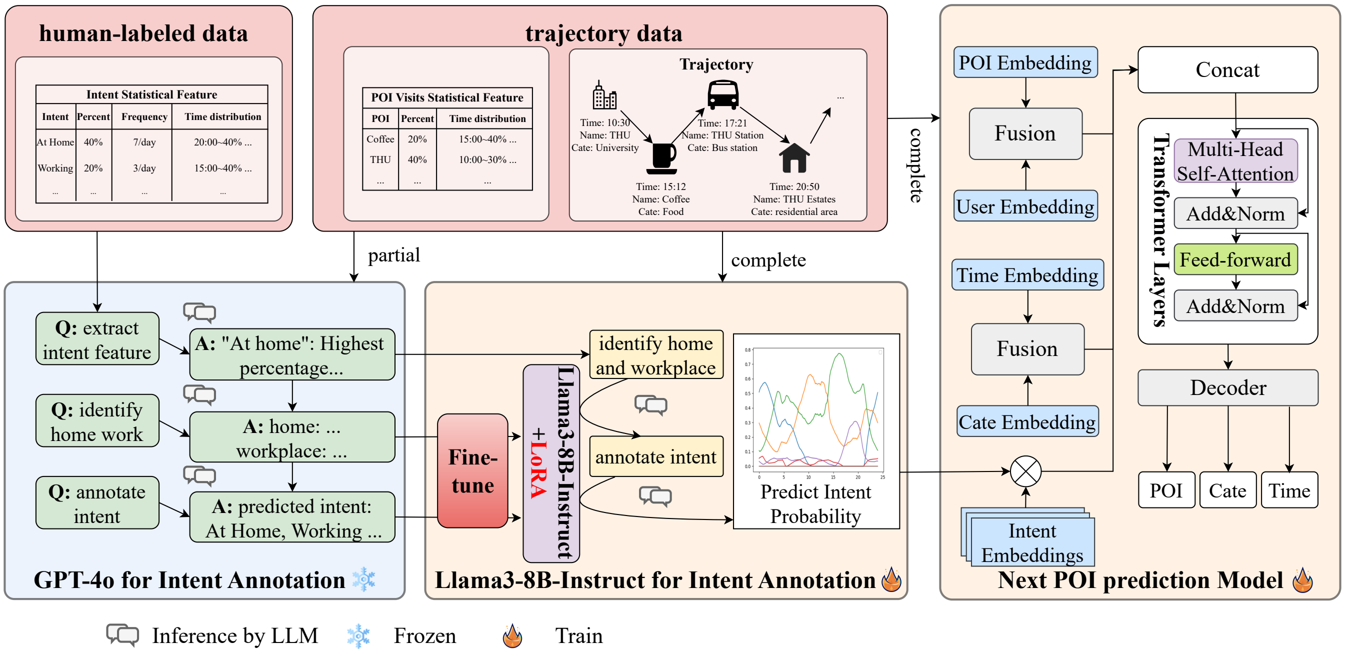

Figure 1: The framework of LIMP, including Analyze-Abstract-Infer (A2I) agentic workflow for intent annotation, agentic workflow fine-tuning schema, and transformer based mobility prediction model.

tions, we streamlined the analysis process and improved the accuracy of subsequent predictions.

Agentic Workflow Fine-tuning

In this section, we annotate the entire dataset with intents and predict the probability distribution of the next movement's intent based on historical data. For large datasets, using high-parameter models such as GPT-4o for intentions inference incurs excessive costs, making it inefficient to annotate the entire dataset directly. On the other hand, smaller models like Llama3-8B perform poorly on related tasks and fail to meet the expected performance. To address this issue, we employed a smaller model, Llama3-8B-Instruct, and fine-tuned it with data generated by GPT-4o. This approach allows the model to achieve performance on user intentions inference tasks that is close to that of GPT-4o. The prompts of fine-tune are similar to the user intentions inference part of A2I workflow. We only modified some parts to better fine-tune it. The fine-tuning process consists of two tasks.

- **Task 1**: Based on the distribution of visit times to POIs by specific users and the summative insights of the human-labeled dataset extracted by GPT-4o, determine the POIs of the users' homes and workplaces.
- **Task 2**: For a specific user, given the POIs of their home and workplace, annotate the intent for each movement in a segment of their trajectory.

We extracted 100 users from training sets in both Beijing and Shanghai, using 20% of each user's trajectory information. This data was annotated by GPT-4o and then used to fine-tune the *llama3-8B-Instruct* model, enabling the *llama3-8B* model to perform both tasks simultaneously. For detailed fine-tuning prompts and answers, please see the appendix.

Through experiments, we found that providing the probability distribution of the next movement's intent based on historical data is more effective than directly giving the intent information. We developed a method based on expert knowledge to calculate the intent probability distribution.

We believe that people tend to have the same behavioral patterns at the same time each day, and thus the historical intent distribution can be used to infer future intent distribution. Specifically, for the intent sequence of a user u corresponding to all of his movements in the train dataset, $(Q_1, Q_2, Q_3, \ldots Q_N)$, and their corresponding time sequence $(t_1, t_2, t_3, \ldots t_N)$, we consider that an intent Q_i recorded over a period provides a likelihood of this intent occurring. The influence period is defined as $t_{begin,i} = \max(t_{i-1}, t_i - T)$ and $t_{end,i} = \min(t_{i+1}, t_i + T)$, where T is a parameter representing the maximum influence time range.

For each intent I_j, we construct the function $f_{I_j}(t)$ as follows to represent the effect of the intent:

$$f_{I_j}(t) = \sum_{Q_i = I_j} \max\left(0, \min\left(\frac{t - t_{begin,i}}{t_i - t_{begin,i}}, \frac{t_{end,i} - t}{t_{end,i} - t_i}\right)\right)$$

For a specific time t_0 within a day, the predicted probability of intent I_j occurring is calculated as:

$$P(I_j \mid t_0, u) = \frac{\sum_k f_{I_j}(t_0 + k\Delta t)}{\sum_{k,j} f_{I_j}(t_0 + k\Delta t)}$$

where Δt represents a time interval of a day.

Intent Enhanced Mobility Prediction

In this section, we present a mobility prediction model that effectively utilizes intent information along with other trajectory data to achieve high prediction accuracy. Among the numerous previous approaches, models employing the transformer architecture have demonstrated outstanding performance. Therefore, our model considers using the transformer architecture. Specifically, we draw inspiration from the GETNext (Yang, Liu, and Zhao 2022) model to develop our own model. GETNext is an excellent mobility prediction model that leverages the transformer architecture to fully utilize temporal, user, POI, POI category and map information. Building on this model, we incorporate intent information and simplify the original model structure, achieving promising results.

Our model's input sequence unit is defined as $(u, p, c, t, P(I|t, u))$, where u is the user ID, p is the current POI of the user, c is the category of the POI, t is the time of day of the next movement, and $P(I|t, u)$ is the predicted probability distribution of the next movement's intent. To effectively handle the predicted probability information, we used a method of embedding intents weighted by their probabilities. This means that the higher the probability of a certain intent, the closer the overall intent embedding is to the individual embedding of that intent. Compared to providing only the single most likely intent information, this method offers richer intent probability information. This makes it easier for the model to determine the extent to which it should rely on the intent information, resulting in a more effective combination of intent information and trajectory information. Specifically, for each intent I_i, we map it to a high-dimensional vector

$$e_{I_i} = f_{embed}(I_i) \in \mathbb{R}^{\Gamma}$$

The embedding of the total intent at time t is defined as

$$e_I = \sum_i e_{I_i} P(I_i|t, u)$$

We simplified the GETNext model by omitting the generic movement learning part and directly embedding p:

$$e_p = f_{embed}(p) \in \mathbb{R}^{\Omega}$$

The definitions of the embeddings for u, c, and t, denoted as e_u, e_c, and e_t, and the fused embeddings $e_{p,u}$, $e_{c,t}$ are consistent with GETNext. The final embedding representation for each movement is $e_q = [e_{p,u}, e_{c,t}, e_I]$. For a trajectory $(q_1, q_2, \ldots, q_N)$, we use $(e_{q_1}, e_{q_2}, \ldots, e_{q_N})$ as the input to the transformer Encoder. We adopted the same Transformer Encoder, MLP Decoders structure and loss function as GETNext.

Experiment

Settings

Dataset To evaluate our model, we collected two representative datasets: one comprising mobile application location data from a popular social network vendor, referred to as the Beijing dataset, and the other consisting of call detail records (CDR) data from a major cellular network operator, referred to as the Shanghai dataset. These datasets are named after the cities from which the data was sourced: Beijing and Shanghai. The data generation mechanisms differ significantly: mobile application data with location records is generated on application servers when users request location-based services within the app, such as searches, check-ins, and similar activities, while call detail records data with location information is generated at cellular network base stations when users access the network for communication or data services.

The Beijing dataset includes check-ins from late September 2019 to late November 2019, spanning approximately three months, while Shanghai dataset primarily covers January 2016. The number of users, locations, and check-ins for each dataset are detailed in Table 1. Following the common practice of mobility prediction (Feng et al. 2018; Sun et al. 2020; Yang, Liu, and Zhao 2022), we segmented trajectories into fixed-length sessions and applied a sliding window over the dataset to make full use of the data during training. Specifically, for a user with m check-ins, and a fixed length n, the processed trajectories will contain $(m-n+1)n$ check-ins, where each sequence of n check-ins forms a trajectory, and consecutive trajectories overlap by $n - 1$ point. This strategy is also applied in all the baselines to ensure the fair comparison.

	Duration	**Users**	**POIs**	**Records**
Beijing	3 months	1566	5919	744813
Shanghai	1 month	841	6955	215379

Table 1: Basic statistics of two mobility datasets.

Manual Intent Annotation on Real Data We hire 10 undergraduates to manually annotated intent labels for a small dataset. The annotation process was conducted as follows:

Identification of Home and Work Locations: The primary step involves identifying the home and work locations as anchor points. This is determined based on the frequency, time, and location of Points of Interest (POIs) within a user's trajectory. Specifically, if a user frequently visits a location during the night until the next morning, and this location is in a residential area, school, or other dwelling place, we consider the location as home. Conversely, if a user consistently visits a location during typical work hours, the location will be considered as working place. The intents for these anchor points are accordingly set as "At Home" and "Working".

Determination of Other Trajectory Intents: For other trajectories, intent is assigned based on the POI type and the timing of the visit. If a POI is a dining location, such as a restaurant or food street, and the visit occurs during meal times, the intent is labeled as "Eating Out." If the POI corresponds to recreational venues like bars, game centers, or sports grounds, the intent is marked as "Leisure and Entertainment". If the POI is located at a store or shopping mall, the intent is labeled as "Shopping". Trajectories that do not fit into the aforementioned categories, such as medical visits, or those where the POI cannot be precisely identified on the map, are labeled with the intent ”Running Errands”.

Baselines We consider the following methods as baselines to benchmark the performance of our model.

- **RNN (Graves and Graves 2012)**: a classical model for processing sequential data, capturing temporal dependencies through recurrent connections.
- **DeepMove (Feng et al. 2018)**: a model that combines recurrent networks and attention layer to capture multi-scale temporal periodicity of human mobility.
- **STAN (Luo, Liu, and Liu 2021)**: a model that utilizes a dual-attention structure to enhance next-location recommendation by aggregating spatio-temporal correlations and incorporating personalized item frequency (PIF).
- **LSTPM (Sun et al. 2020)**: it integrates a non-local network for capturing long-term preferences and a geo-dilated recurrent neural network for short-term preferences modelling.
- **Graph-Flashback (Rao et al. 2022)**: it enhances point-of-interest (POI) representation by combining a graph convolution network (GCN) with a POI transition graph.
- **GETNext (Yang, Liu, and Zhao 2022)**: a state-of-art model that introduces a user-agnostic global trajectory flow map and a novel graph enhanced transformer to improve next POI recommendation.

Training Settings For baselines, we use official codes from authors to implement RNN, DeepMove [1], STAN [2], LSTPM [3], Graph-Flashback [4], GETNext [5]. The hyperparameters for the models were adjusted according to the dataset, beginning with the default settings provided in the original codebases. For both RNN and DeepMove, the embedding dimensions for location and user were set to 128, with the Adam optimizer applied using a learning rate of 1e-3. The maximum number of training epochs was set to 20. For LSTPM, the embedding dimensions for location and user were set to 50, with the Adam optimizer configured with a learning rate of 1e-4, and a maximum of 20 training epochs. Similarly, STAN utilized embedding dimensions of 50 for both location and user, an Adam optimizer with a learning rate of 3e-3, and a maximum of 20 training epochs. Graph-Flashback employed embedding dimensions of 100 for both location and user, with the Adam optimizer set to a learning rate of 1e-2, and a maximum of 100 training epochs. Lastly, GETNext was configured with embedding dimensions of 128 for both location and user, an Adam optimizer with a learning rate of 1e-3, and a maximum of 200 training epochs. The length of all trajectories was standardized to 12. All other unique parameters for each model, as well as those not explicitly mentioned, adhered to the default settings of the respective codebases.

We fine-tuned the llama3-8B-Instruct model using LLaMA-Factory (Zheng et al. 2024). Our model is deployed on Google Colab, utilizing cloud computing resources for fine-tuning. The original model was a 4-bit quantized version of llama3-8B-Instruct. We utilized LoRA adapters and performed fine-tuning using 4-bit QLoRA. The training was conducted for 3 epochs and a batch size of 2. The parameter of learning rate is set to 5e-5. After fine-tuning, we merged the LoRA adapter weights with the non-quantized llama3-8B-Instruct to obtain the final model. We used the PyTorch framework for training the transformer based mobility prediciton model on the following hardware platform (CPU: Intel Xeon Platinum 8358, GPU: NVIDIA GeForce RTX 4090). The embedding dimensions for POI and user are $\Omega = 128$, while the embedding dimensions for time, POI category Ψ, and intent Γ are all 32. The transformer architecture includes two encoder layers. In the encoder layers, the feed-forward network dimension is 1024, and the multi-head attention module uses two attention heads. The transformer architecture employs a dropout with a probability of 0.3. We used the Adam optimizer with a learning rate of 1e-3, a weight decay rate of 5e-4 and a maximum of 200 training epochs.

Metrics We used two common metrics in existing works (Sun et al. 2020; Luo, Liu, and Liu 2021) to evaluate the performance of our model: Accuracy at k (Acc@k) and Mean Reciprocal Rank at 5 (MRR@5). Acc@k measures the proportion of the true next POI is found among the top k recommendations. MRR@5 evaluates the average of the reciprocal ranks of the true next POI in the top 5 recommendations. Define

$$\text{Acc@k} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(r \leq k)$$

$$\text{MRR@5} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{R}$$

where N is the total number of queries, $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if the condition is true and 0 otherwise, r is the rank position of the true next POI. R is equal to r only when $r \leq 5$; otherwise, it tends to infinity.

Main Results

The main results on the Beijing and Shanghai datasets are shown in Table 2 and Table 3. Our method outperforms all baseline methods across various metrics on both datasets. For instance, in terms of Acc@1 on the Beijing dataset, the best baseline achieves 0.4547, while our method reaches 0.4980, representing an approximate 10% relative improvement. Similarly, our method also shows nearly 10% relative improvements over the best baseline method in terms of Acc@10 and MRR@5 on the Shanghai dataset. Our method performed worse on the Shanghai dataset compared to the Beijing dataset. This could be because the model architecture is less effective with the smaller data volume and higher

[1]https://github.com/vonfeng/DeepMove

[2]https://github.com/yingtaoluo/Spatial-Temporal-Attention-Network-for-POI-Recommendation

[3]https://github.com/NLPWM-WHU/LSTPM

[4]https://github.com/kevin-xuan/Graph-Flashback

[5]https://github.com/songyangme/GETNext

total number of POIs in the Shanghai dataset. As a comparison, the GETNext model, which has a very similar architecture to ours, also showed a significant performance drop on the Shanghai dataset compared to the Beijing dataset.

Models	Acc@1	Acc@5	Acc@10	MRR@5
RNN	0.2290	0.3667	0.3941	0.2846
DeepMove	0.3129	0.5202	0.5482	0.3999
STAN	0.3270	0.6532	0.7419	0.4548
LSTPM	0.4291	0.7910	0.8202	0.5826
Graph-Flashback	0.4387	0.8103	0.8508	0.5923
GETNext	0.4547	0.8175	0.8596	0.6126
Ours	**0.4980**	**0.8337**	**0.8718**	**0.6444**
δ	9.52%	1.98%	1.42%	5.19%

Table 2: Intent-aware mobility prediction results on Beijing data.

Models	Acc@1	Acc@5	Acc@10	MRR@5
RNN	0.2530	0.3899	0.4232	0.3084
DeepMove	0.2713	0.3893	0.4123	0.3199
STAN	0.2566	0.5411	0.6544	0.3641
LSTPM	0.4489	0.7018	0.7422	0.5518
Graph-Flashback	0.4351	0.6966	0.7445	0.5416
GETNext	0.4177	0.6782	0.7363	0.5348
Ours	**0.4787**	**0.7517**	**0.8154**	**0.6025**
δ	6.64%	7.11%	9.52%	9.19%

Table 3: Intent-aware mobility prediction results on Shanghai dataset.

Intent Prediction Analysis

We evaluated our intent prediction workflow with multiple experimental setups on the small, human-labeled dataset to explore how each component in the workflow affects the performance. Results in Table 4 show that our full workflow achieved the highest accuracy, outperforming both zero-shot prediction and workflow without generating features or home and work identification.

The results indicate that the step of home and workplace identification is a critical factor in improving prediction accuracy. Specifically, the workflow with home and workplace identification demonstrated a substantial increase in accuracy compared to the workflow without this step. In contrast, the generation of additional features did not significantly enhance performance when compared to the base workflow. We analyzed the annotation of specific intents by compiling confusion matrices for six types of intents on human-labeled dataset, as shown in Figure 2. We found that the model's predictive ability is excellent for the intents of At home, Working, Eating Out, and Leisure and Entertainment. Among the incorrect annotations, the most common error is labeling Running errands as Working. This is because the POIs associated with these two intents have similar attributes, and the timing of these activities often overlaps, making it challenging to perfectly distinguish between them. Additionally,

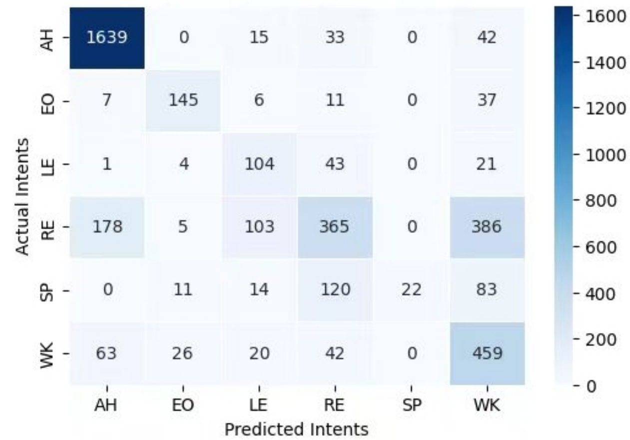

Figure 2: Confusion matrices of intents. AH, EO, LE, RE, SP and WK refer to ”At Home”, ”Eating Out”, ”Leisure and Entertainment”, ”Running Errands”, ”Shopping” and ”Working” respectively.

Shopping is frequently mislabeled as other intents because the POIs where shopping occurs often serve multiple functions beyond just shopping.

Experiments	Accuracy	Precision	Recall	F1 Score
A2I Workflow	**67.9%**	**0.715**	**0.679**	**0.657**
NFE	67.2%	0.707	0.672	0.651
NHWI	60.0%	0.627	0.600	0.575
ZS	59.0%	0.609	0.590	0.565

Table 4: Ablation study of the prompt designs in A2I workflow, where ‘NFE’ refers to the model run without feature engineering, and ‘NHWI’ refers to the model run without home-work identification, ‘ZS’ denotes the basic prompt settings.

Ablation Study

We conducted ablation experiments on the Beijing dataset to demonstrate the effectiveness of our designed architecture. There are four experiments:

- **LIMP**: The experiment with the complete model architecture.
- **Max-Prob**: This experiment does not use probabilities to weight the embeddings; instead, it only uses the embedding of the intent with the highest probability, aiming to verify the effect of probability weighting on intent embeddings.
- **Train-Real**: This experiment does not predict the intent for the training set but uses the next intent generated by LLM as the predicted intent to generate embeddings. For the test and validation sets, the embeddings of the intent with the highest probability are used. This experiment compares the effects of training the model with real intent and predicted intent.
- **w/o intent**: The model without the intent input.

The experimental results are shown in Table 5. Overall, the Full Model achieved the best performance. The Max-Prob model, which does not provide the probability distribution information of the intent, and the w/o intent model, which does not provide any intent information, both showed performance degradation compared to the Full Model. The Train-Real experiment had a relatively high Acc@1 indicator but performed significantly worse than the other three models in terms of Acc@5, Acc@10, and MRR@5 indicators, making its overall performance unsatisfactory. This is because the training set directly input the real intent, bypassing the intent prediction step, causing the model to rely too heavily on the intent and become insensitive to other input information. During testing, the intent can only be predicted based on historical data, reducing the success rate of direct intent prediction and thus leading to performance degradation.

Experiments	Acc@1	Acc@5	Acc@10	MRR@5
Full Model	0.4980	0.8337	0.8718	0.6444
Max-Prob	0.4835	0.8328	0.8726	0.6356
Train-Real	0.4989	0.7623	0.7916	0.6112
w/o intent	0.4670	0.8336	0.8729	0.6271

Table 5: Ablation study of intent-aware mobility prediction model. 'Full Model' refers to the complete structure, 'Max-Prob' refers to uses the embedding of the most possible as the embedding, 'Train-Real' refers to use the intent generated by LLM to train directly, 'w/o intent' refers to structure without intent information

Related Work

Mobility Prediction

Mobility prediction involves anticipating future locations or visits of individuals based on historical mobility data. Before the population of deep learning based models, researchers employ Markov models (Rendle, Freudenthaler, and Schmidt-Thieme 2010; Cheng et al. 2013) to predict the next location by learning the transition relations between consecutive POIs. However, they fail to capture the high-order transitions among trajectories and also struggling in considering the effects of urban environment and user preference. Thus, research introduce the deep learning methods (Liu et al. 2016; Feng et al. 2018; Sun et al. 2020; Luo, Liu, and Liu 2021; Yang, Liu, and Zhao 2022) to address these issues and achieve promising results. Liu et al. (Liu et al. 2016) first introduce RNN to predict the next location by proposing the spatial-temporal interpolation based ST-RNN. Feng et al. (Feng et al. 2018) apply attention mechanism to capture the multi-scale temporal patterns of human mobility. Luo et al. (Luo, Liu, and Liu 2021) further introduce the dual attention module to capture the spatial-temporal correlations among trajectories. While these methods succeed in modelling the sequential patterns in the trajectory, they cannot capture the shared mobility patterns between users effectively. To solve this problem, graph neural networks are introduced into the mobility prediction modelling. By representing trajectories as graph or hypergraph (Lim et al. 2020; Yan et al. 2023), these approaches enable a comprehensive understanding of transition patterns at different locations (Wang et al. 2023b; Yin et al. 2023), thus enriching the representations of users and POIs. However, due to the absence of large scale mobility intent dataset, these works ignore the intent modelling behind the mobility. In this work, we employ the agentic workflow to enable the large-scale automatic annotation of mobility intent and further propose an effective intent-aware mobility prediction model.

Large Language Models

Large language models (LLMs) (Achiam et al. 2023; Dubey et al. 2024) have achieved rapid development in the past few years, making significant progress in reasoning (Wei et al. 2022b), planning, coding (Roziere et al. 2023), mathematics (Luo et al. 2023), and other fields. Some researchers also try to directly apply LLMs in the mobility modelling (Wang et al. 2023a; Beneduce, Lepri, and Luca 2024). Different from these works, we use LLMs as the mobility data annotator to augment mobility data for training a stronger small domain-specific model. While LLMs are applied into many fields and show promising results to some extent, the lack of domain knowledge of LLMs on specific application limit their further improvement. Recently, LLM based agent framework (Wang et al. 2024; Xi et al. 2023) are proposed to complement the deficiencies of LLMs on specific domain knowledge and unleash the power of LLMs in real-world tasks, e.g., ChatDev (Qian et al. 2023) for project programming and WebAgent (Gur et al. 2023) for autonomous web tasks. Different from these works, we focus on the mobility prediction task and design the first agentic workflow for challenging intent annotation.

Conclusion

In this paper, we investigate the problem of intent enhanced mobility prediction problem. We propose LIMP, an agentic workflow based framework to harness the commonsense reasoning abilities of LLMs for intention-aware mobility prediction. We first design the "Analyze-Abstract-Infer" (A2I) framework to help the powerful LLM, GPT-4o in the paper, infer the mobility intent precisely. Then, we design agentic workflow fine-tuning mechanism to distilling common sense reasoning power of powerful LLM to smaller LLM, e.g., Llama3-8B. With the support of smaller LLM for large scale mobility data annotation, we finally design a transformer based model to complete the intent-aware mobility prediction task. Extensive experiments on two real-world datasets demonstrate the effectiveness of proposed framework. In the future, we plan to extend the framework to other spatial-temporal applications.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Beneduce, C.; Lepri, B.; and Luca, M. 2024. Large Language Models are Zero-Shot Next Location Predictors. *arXiv preprint arXiv:2405.20962*.

Chen, Q.; and Poorthuis, A. 2021. Identifying home locations in human mobility data: an open-source R package for comparison and reproducibility. *International Journal of Geographical Information Science*, 35(7): 1425–1448.

Cheng, C.; Yang, H.; Lyu, M. R.; and King, I. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Twenty-Third international joint conference on Artificial Intelligence*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; and Jin, D. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *Proceedings of the 2018 World Wide Web Conference*.

Graves, A.; and Graves, A. 2012. *Supervised sequence labelling*. Springer.

Gur, I.; Furuta, H.; Huang, A.; Safdari, M.; Matsuo, Y.; Eck, D.; and Faust, A. 2023. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*.

Jiang, S.; Yang, Y.; Gupta, S.; Veneziano, D.; Athavale, S.; and González, M. C. 2016. The TimeGeo modeling framework for urban mobility without travel surveys. *Proceedings of the National Academy of Sciences*, 113(37): E5370–E5378.

Liccardi, I.; Abdul-Rahman, A.; and Chen, M. 2016. I know where you live: Inferring details of people's lives by visualizing publicly shared location data. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 1–12.

Lim, N.; Hooi, B.; Ng, S.-K.; Wang, X.; Goh, Y. L.; Weng, R.; and Varadarajan, J. 2020. STP-UDGAT: Spatial-temporal-preference user dimensional graph attention network for next POI recommendation. In *Proceedings of the 29th ACM International conference on information & knowledge management*, 845–854.

Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*.

Luo, H.; Sun, Q.; Xu, C.; Zhao, P.; Lou, J.; Tao, C.; Geng, X.; Lin, Q.; Chen, S.; and Zhang, D. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Luo, Y.; Liu, Q.; and Liu, Z. 2021. Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the web conference 2021*, 2177–2185.

Qian, C.; Cong, X.; Yang, C.; Chen, W.; Su, Y.; Xu, J.; Liu, Z.; and Sun, M. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 6.

Rao, X.; Chen, L.; Liu, Y.; Shang, S.; Yao, B.; and Han, P. 2022. Graph-flashback network for next location recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1463–1471.

Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, 811–820.

Roziere, B.; Gehring, J.; Gloeckle, F.; Sootla, S.; Gat, I.; Tan, X. E.; Adi, Y.; Liu, J.; Remez, T.; Rapin, J.; et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Sun, K.; Qian, T.; Chen, T.; Liang, Y.; Nguyen, Q. V. H.; and Yin, H. 2020. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 214–221.

Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6): 186345.

Wang, X.; Fang, M.; Zeng, Z.; and Cheng, T. 2023a. Where would i go next? large language models as human mobility predictors. *arXiv preprint arXiv:2308.15197*.

Wang, Z.; Zhu, Y.; Wang, C.; Ma, W.; Li, B.; and Yu, J. 2023b. Adaptive Graph Representation Learning for Next POI Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 393–402.

Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Chi, E. H.; Le, Q.; and Zhou, D. 2022b. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *CoRR*, abs/2201.11903.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022c. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; Zhang, M.; Wang, J.; Jin, S.; Zhou, E.; et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Yan, X.; Song, T.; Jiao, Y.; He, J.; Wang, J.; Li, R.; and Chu, W. 2023. Spatio-temporal hypergraph learning for next POI recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, 403–412.

Yang, S.; Liu, J.; and Zhao, K. 2022. GETNext: trajectory flow map enhanced transformer for next POI recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on research and development in information retrieval*, 1144–1153.

Yin, F.; Liu, Y.; Shen, Z.; Chen, L.; Shang, S.; and Han, P. 2023. Next POI recommendation with dynamic graph and explicit dependency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4827–4834.

Zeng, W.; Fu, C.-W.; Arisona, S. M.; Schubiger, S.; Burkhard, R.; and Ma, K.-L. 2017. Visualizing the relationship between human mobility and points of interest. *IEEE Transactions on Intelligent Transportation Systems*, 18(8): 2271–2284.

Zhao, A.; Huang, D.; Xu, Q.; Lin, M.; Liu, Y.-J.; and Huang, G. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19632–19642.

Zheng, Y.; Zhang, R.; Zhang, J.; Ye, Y.; Luo, Z.; and Ma, Y. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. *ArXiv*, abs/2403.13372.

Appendix

Prompt example for generating high-level patterns of intent

```
Your task is to extract the features of intent \'At Home\', \'Working\', \'Running errands\' from the statistical data. Please think step by step.

Here's the statistical data of the user's intent distribution:{{\'Intent 1\': {\'percentage_distribution\': 43.17, \'average_visit\': 2.5426470588235293}, \'Intent 2\': {\'percentage_distribution\': 15.23, \'average_visit\': 2.489795918367347}, \'Intent 3\': {\'percentage_distribution\': 25.89, \'average_visit\': 1.4730113636363635}, \'Time Distribution of Intents\': {\'Intent 1\': {0: 75.19, 1: 80.0, 2: 85.19, 3: 85.71, 4: 72.0, 5: 66.67, 6: 71.25, 7: 58.17, 8: 29.95, 9: 25.93, 10: 29.13, 11: 30.18, 12: 34.78, 13: 27.27, 14: 32.69, 15: 22.51, 16: 26.2, 17: 31.98, 18: 42.19, 19: 43.43, 20: 52.36, 21: 68.69, 22: 74.11, 23: 70.5}, \'Intent 2\': {0: 1.55, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0, 5: 0.0, 6: 3.75, 7: 0.65, 8: 3.69, 9: 4.04, 10: 7.39, 11: 6.76, 12: 5.53, 13: 6.06, 14: 4.81, 15: 3.66, 16: 7.49, 17: 8.11, 18: 8.02, 19: 7.17, 20: 6.44, 21: 4.21, 22: 3.05, 23: 4.32}, \'Intent 3\': {0: 2.33, 1: 0.0, 2: 0.0, 3: 0.0, 4: 12.0, 5: 0.0, 6: 5.0, 7: 3.27, 8: 2.3, 9: 5.05, 10: 6.96, 11: 4.5, 12: 4.74, 13: 8.08, 14: 4.33, 15: 2.09, 16: 3.74, 17: 8.11, 18: 5.06, 19: 2.79, 20: 3.0, 21: 5.14, 22: 2.54, 23: 2.88}, \'Intent 4\': {0: 13.95, 1: 11.67, 2: 14.81, 3: 14.29, 4: 8.0, 5: 19.05, 6: 12.5, 7: 18.95, 8: 24.88, 9: 30.3, 10: 27.83, 11: 35.14, 12: 32.41, 13: 30.3, 14: 30.77, 15: 39.27, 16: 34.22, 17: 28.38, 18: 27.0, 19: 27.89, 20: 28.33, 21: 10.75, 22: 13.2, 23: 12.95}, \'Intent 5\': {0: 3.1, 1: 5.0, 2: 0.0, 3: 0.0, 4: 0.0, 5: 0.0, 6: 0.0, 7: 5.88, 8: 11.06, 9: 5.72, 10: 6.96, 11: 7.21, 12: 9.49, 13: 7.58, 14: 6.25, 15: 8.9, 16: 10.7, 17: 7.66, 18: 6.75, 19: 5.58, 20: 3.43, 21: 4.21, 22: 1.02, 23: 4.32}, \'Intent 6\': {0: 3.88, 1: 3.33, 2: 0.0, 3: 0.0, 4: 8.0, 5: 14.29, 6: 7.5, 7: 13.07, 8: 28.11, 9: 28.96, 10: 21.74, 11: 16.22, 12: 13.04, 13: 20.71, 14: 21.15, 15: 23.56, 16: 17.65, 17: 15.77, 18: 10.97, 19: 13.15, 20: 6.44, 21: 7.01, 22: 6.09, 23: 5.04}}}}

The meanings of statistical data are as follows:
- Percentage Distribution: The percent of the intent in the whole dataset.
- Time Distribution: The start_time distribution of visits to the POI with the intent, in the format of (start hour: percentage).

There are 6 intents in total: ['At Home', 'Working', 'Running errands', 'Eating Out', 'Leisure and entertainment', 'Shopping'], each intent has a percentage distribution and a time distribution.

Instruction:
- You need to extract the unique and prominent features of intent 'At Home', 'Working', 'Running errands' which can distinguish them from other intents.
- Each intent should have about 6-8 features.
- Should be based on the percentage distribution, and time distribution of the intent.
- Should be able to help identify the user's home, work place, and running errands place.
- Some features need to be specificity to the intent, such as the time distribution of the intent.

Answer using the following JSON format:
{{
"features": ["features of 'intents'"],
}}
```

Prompt example for home and work identification.

```
Your objective is to identify the potential 'home,' and 'work' places of a user's intent based on their trajectory data and the features associated with the intents 'At Home' and 'Working'. Please think step by step.

```

```
The trajectory data under analysis is as follows: {[{\'Name\': \'poi1\', \'Percent\': \'8.5%\', \'Time Distribution\': [(\'10:00\', \'40.0%\'), (\'12:00\', \'20.0%\'), (\'13:00\', \'20.0%\'), (\'14:00\', \'20.0%\')]}, {\'Name\': \'poi2\', \'Percent\': \'8.5%\', \'Time Distribution\': [(\'15:00\', \'20.0%\'), (\'16:00\', \'80.0%\')]}, {\'Name\': \'poi3\', \'Percent\': \'1.7%\', \'Time Distribution\': [(\'17:00\', \'100.0%\')]}, {\'Name\': \'poi4\', \'Percent\': \'3.4%\', \'Time Distribution\': [(\'15:00\', \'50.0%\'), (\'16:00\', \'50.0%\')]}, {\'Name\': \'poi5\', \'Percent\': \'50.8%\', \'Time Distribution\': [(\'9:00\', \'26.7%\'), (\'10:00\', \'3.3%\'), (\'11:00\', \'3.3%\'), (\'12:00\', \'6.7%\'), (\'13:00\', \'23.3%\'), (\'14:00\', \'13.3%\'), (\'15:00\', \'16.7%\'), (\'17:00\', \'6.7%\')]}]}.

The meaning of each element in the trajectory data is as follows:
    - Name: the POI the user visited.
    - Percent: The percentage of times the behavior pattern occurred
    - Time Distribution: the start time distribution of the number of visits to the POI, in the format of (start hour: percentage).

Here are the general and unique features of intent 'At Home' , 'Working' , 'Running errands':{            "intent": "At Home",
            "features": [
                "High percentage distribution: 43.17%",
                "Average visit: 2.54 times",
                "Peak time distribution: Early morning (0-3 AM) and late evening (8-11 PM)",
                "Consistent presence throughout the day with notable dips during typical working hours (8 AM - 5 PM)",
                "Significant presence during late night hours (12 AM - 3 AM)",
                "Gradual increase in presence from 4 PM onwards, peaking at 9 PM",
                "High presence during night hours (10 PM - 12 AM)"
            ]
        },
        {
            "intent": "Working",
            "features": [
                "Moderate percentage distribution: 15.23%",
                "Average visit: 2.49 times",
                "Peak time distribution: Morning (8-11 AM) and early afternoon (12-3 PM)",
                "Significant presence during typical working hours (8 AM - 5 PM)",
                "Notable drop in presence during early morning (0-6 AM) and late evening (6 PM onwards)",
                "Presence peaks at 9 AM and 10 AM, indicating start of workday",
                "Gradual decrease in presence after 3 PM"
            ]
        },
        {
            "intent": "Running errands",
            "features": [
                "Moderate percentage distribution: 25.89%",
                "Average visit: 1.47 times",
                "Peak time distribution: Late morning (9-11 AM) and early afternoon (12-3 PM)",
                "Significant presence during mid-morning to early afternoon (9 AM - 3 PM)",
                "Notable drop in presence during early morning (0-6 AM) and late evening (6 PM onwards)",
                "Presence peaks at 11 AM, indicating common time for running errands",
                "High presence during mid-day hours (10 AM - 2 PM)"
            ]
        }
    ]}

Respond using the following JSON format:
{{
    "home": "home place",
    "work": "work place"
    "reason": "reason for prediction"
}}
```

Prompt example for intent prediction.

```
Your task is to give intent prediction using trajectory data. Let's think step by step.

1. Analyze the user's behavior pattern
```

```
based on the trajectory data.
2. Consider and think about the name of the POI and the time distribution of visits to the POI with the intent. (This is the trajectory of one person, so thinking about the user's daily routine is important.)
3. Based on the user's behavior pattern and please consider the features of intent 'At Home', 'Working', 'Running errands', predict the intent of each stay in the trajectory data.

The trajectory data under analysis is as follows: {(poi name1, 2019-10-11 00:30:00, 2019-10-11 07:30:00)(poi name2, 2019-10-11 08:15:00, 2019-10-11 15:30:00)(poi name3, 2019-10-11 15:45:00, 2019-10-11 17:00:00)}.

Each stay in trajectory data is represented as (poi, start time).

Here's what each element means:
- poi: the POI the user visited.
- start time: the time the user arrived at the POI.

Please judge the function of POI based on its name, time distribution, and features provided. You should take the meaning of each intent as reference, but the final judgment shouldn't be fully rely on that.

Intent you can choose:['At Home', 'Working', 'Running errands', 'Eating Out', 'Leisure and entertainment', 'Shopping']

Here's what each intent means:
- At Home: When the user is at {poi name1}, it is mostly considered as being at home. And Other places are NOT considered as home!
- Working: When the user is at {poi name2}, it is mostly considered as working. And Other places are NOT considered as working!
But, you should still consider the user\'s behavior pattern, POI_name, and the time the user visited the POI.

Note: If multiple conditions are met, priority should be given to 'At Home' and 'Running Errands'.

There are {3} stays in the trajectory data. So, the output should have {3} predicted intents.

Consider step by step, finally respond using the following JSON format (Make sure to have one predicted intent for each stay in the trajectory data, And you have to assign one of the intents to each stay in the trajectory data):
{{
"predicted_intent": ["adjusted predicted intents"],
}}
```

Prompt example for zero-shot intent prediction.

```
Your task is to give intent prediction using trajectory data.

The trajectory data under analysis is as follows: {(poi name1, 2019-10-11 00:30:00, 2019-10-11 07:30:00)(poi name2, 2019-10-11 08:15:00, 2019-10-11 15:30:00)(poi name3, 2019-10-11 15:45:00, 2019-10-11 17:00:00)}.

Each stay in trajectory data is represented as (poi, start time).

Here's what each element means:
    - POI: the POI the user visited.
    - Start Time: the time the user arrived at the POI.

Intent you can choose:['At Home', 'Working', 'Running errands', 'Eating Out', 'Leisure and entertainment', 'Shopping']

There are {3} stays in the trajectory data. So make sure the output should only have {3} predicted intents.

Respond using the following JSON format to provide the predicted intents:
{{
"predicted_intent": ["adjusted predicted intents"],
}}
```

Prompt example of home and work identification with feature ablation

```
Your objective is to identify the potential 'home,' and 'work' places of a user's intent based on their trajectory data and the features associated with the intents 'At Home' and 'Working'. Please think step by step.

The trajectory data under analysis is as follows: {[{\'Name\': \'poi1\', \'Percent\': \'8.5%\', \'Time Distribution\': [(\'10:00\', \'40.0%\'), (\'12:00\', \'20.0%\'), (\'13:00\', \'20.0%\'), (\'14:00\', \'20.0%\')]}, {\'Name\': \'poi2\', \'Percent\': \'8.5%\', \'Time Distribution\': [(\'15:00\', \'20.0%\'), (\'16:00\', \'80.0%\')]}, {\'Name\': \'poi3\', \'Percent\':
```

```
    \'1.7%\', \'Time Distribution\':
    [(\'17:00\', \'100.0%\')]}, {\'Name\':
    \'poi4\', \'Percent\': \'3.4%\', \'Time
    Distribution\': [(\'15:00\',
    \'50.0%\'), (\'16:00\', \'50.0%\')]},
    {\'Name\': \'poi5\', \'Percent\':
    \'50.8%\', \'Time Distribution\':
    [(\'9:00\', \'26.7%\'), (\'10:00\',
    \'3.3%\'), (\'11:00\', \'3.3%\'),
    (\'12:00\', \'6.7%\'), (\'13:00\',
    \'23.3%\'), (\'14:00\', \'13.3%\'),
    (\'15:00\', \'16.7%\'), (\'17:00\',
    \'6.7%\')]}]}.

The meaning of each element in the
    trajectory data is as follows:
- Name: the POI the user visited.
- Percent: The percentage of times the
    behavior pattern occurred
- Time Distribution: the start time
    distribution of number of visits to the
    POI, in the format of (start hour:
    percentage).

Respond using the following JSON format:
{{
"home": "home place",
"work": "work place",
"reason": "reason for prediction"
}}
```

Prompt example for intent prediction with features ablation

```
Your task is to give intent prediction
    using trajectory data. Let's think step
    by step.

1. Analyze the user's behavior pattern
    based on the trajectory data.
2. Consider and think about the name of the
    POI and the time distribution of visits
    to the POI with the intent. (This is
    the trajectory of one person, so
    thinking about the user's daily routine
    is important.)
3. Based on the user's behavior pattern and
    please consider the features of intent
    'At Home', 'Working', 'Running
    errands', predict the intent of each
    stay in the trajectory data.

The trajectory data under analysis is as
    follows: {(poi name1, 2019-10-11
    00:30:00, 2019-10-11 07:30:00)(poi
    name2, 2019-10-11 08:15:00, 2019-10-11
    15:30:00)(poi name3, 2019-10-11
    15:45:00, 2019-10-11 17:00:00)}.

Each stay in trajectory data is represented
    as (poi, start time).

Here's what each element means:
- poi: the POI the user visited.
- start time: the time the user arrived at
    the POI.

Please judge the function of POI based on
    its name, time distribution, and
    features provided. You should take the
    meaning of each intent as reference,
    but the final judgment shouldn't be
    fully rely on that.

Intent you can choose:['At Home',
    'Working', 'Running errands', 'Eating
    Out', 'Leisure and entertainment',
    'Shopping']

Here's what each intent means:
- At Home: When the user is at {poi name1},
    it is mostly considered as being at
    home. And Other places are NOT
    considered as home!
- Working: When the user is at {poi name2},
    it is mostly considered as working. And
    Other places are NOT considered as
    working!
But, you should still consider the user\'s
    behavior pattern, POI_name, and the
    time the user visited the POI.

Note: If multiple conditions are met,
    priority should be given to 'At Home'
    and 'Running Errands'.

There are {} stays in the trajectory data.
    So, the output should have {} predicted
    intents.

Consider step by step, finally respond
    using the following JSON format (Make
    sure to have one predicted intent for
    each stay in the trajectory data, And
    you have to assign one of the intents
    to each stay in the trajectory data):
{{
"predicted_intent": ["adjusted predicted
    intents"],
}}
```

Prompt and answer example for fine-tune task 1.

```
Your task is to identify the user's home
    and work place based on the trajectory
    data and the features of intent 'At
    Home' and 'Working'.
The trajectory data under analysis is as
    follows: [{'Name': 'poi name1',
    'Percent': '79.2%', 'Time
    Distribution': [('0:00', '2.4%'),
    ('1:00', '9.5%'), ('2:00', '9.5%'),
    ('3:00', '7.1%'), ('4:00', '7.1%'),
    ('6:00', '21.4%'), ('7:00', '7.1%'),
    ('8:00', '7.1%'), ('9:00', '7.1%'),
    ('10:00', '7.1%'), ('11:00', '7.1%'),
    ('12:00', '2.4%'), ('23:00', '4.8%')]},
    {'Name': 'poi name2', 'Percent':
```

```
    '7.5%', 'Time Distribution': [('1:00',
    '25.0%'), ('10:00', '25.0%'), ('11:00',
    '25.0%'), ('14:00', '25.0%')]},
    {'Name': 'poi name3', 'Percent':
    '5.7%', 'Time Distribution': [('7:00',
    '33.3%'), ('9:00', '33.3%'), ('11:00',
    '33.3%')]}, {'Name': 'poi name4',
    'Percent': '5.7%', 'Time Distribution':
    [('3:00', '33.3%'), ('4:00', '33.3%'),
    ('5:00', '33.3%')]}, {'Name': 'poi
    name5', 'Percent': '1.9%', 'Time
    Distribution': [('13:00', '100.0%')]}].
Each entry represents a POI-intent pair
    that the user has visited.\n    The
    meanings of each feature are as follows:
- Name: POI name
- Percent: The percentage of times the
    behavior pattern occurred
- Time Distribution: The time distribution
    of visits to the POI with the intent,
    in the format of (hour, percentage).
Here are the features of intent 'At Home'
    and 'Working':{'features': [{'intent':
    'At Home', 'features': ['Highest
    overall percentage distribution at
    43.2%', 'Significant activity during
    late evening and night hours (20:00 -
    23:00) with peaks at 21:00 (8.5%) and
    22:00 (8.4%)', 'Consistent activity
    throughout the day with notable peaks
    at 7:00 (5.1%), 12:00 (5.1%), and 19:00
    (6.3%)', 'Low activity during early
    morning hours (1:00 - 5:00) with
    percentages below 3%', 'Evening
    activity starts to increase
    significantly from 18:00 (5.8%)',
    'Activity remains relatively high and
    stable from 17:00 to 23:00']},
    {'intent': 'Working', 'features':
    ['Second highest overall percentage
    distribution at 15.2%', 'Peak activity
    during typical working hours (8:00 -
    17:00) with the highest at 9:00
    (14.1%)', 'Significant drop in activity
    after 18:00, with percentages below 5%
    from 18:00 onwards', 'Low activity
    during early morning hours (0:00 -
    7:00) with percentages below 4%',
    'Notable activity during mid-morning to
    early afternoon (8:00 - 15:00) with
    percentages above 5%', 'Activity starts
    to decline significantly after
    17:00']}, {'intent': 'Running errands',
    'features': ['Third highest overall
    percentage distribution at 25.9%',
    'Peak activity during late morning to
    early afternoon (9:00 - 14:00) with the
    highest at 9:00 (8.7%)', 'Consistent
    activity throughout the day with
    notable peaks at 11:00 (7.5%) and 12:00
    (7.9%)', 'Low activity during late
    night and early morning hours (0:00 -
    6:00) with percentages below 2%',
    'Activity remains relatively high and
    stable from 8:00 to 19:00',
    'Significant drop in activity after
    20:00, with percentages below 3% from
    21:00 onwards']}]}\n    Respond using
    the following JSON format:
{\"home\": \"home place\",\"work\": \"work
    place\"}

answer:
{"home": "poi name1", "work": "poi name2"}
```

Prompt and answer example for fine-tune task 2.

```
"Your task is to give intent prediction
    using trajectory data. Stay in
    trajectory data corresponds one by one
    to intent.
The trajectory data under analysis is as
    follows: (poi name1, High School,
    2019-11-18 01:00:00)(poi name1, High
    School, 2019-11-18 13:15:00)(poi name2,
    Educational Facilities, 2019-11-19
    00:00:00).
Each stay in trajectory data is represented
    as (poi, category of poi, start time).
Here's what each element means:\n    - poi:
    the POI the user visited.
- category of poi: category the POI belongs
    to.
- start time: the time the user arrived at
    the POI.
Please mainly judge the function of POI
    based on its name. The POI category can
    be used to assist in judgment.
Intent you can choose:['At Home',
    'Working', 'Running errands', 'Eating
    Out', 'Leisure and entertainment',
    'Shopping']
Here's what each intent means:
- At Home: When the user is at poi name2,
    it is always considered as being at
    home, regardless of the time and POI
    category. When the user is at other
    places, it is not considered as being
    at home.
- Working: When the user is at poi name1,
    it is always considered as working,
    regardless of the time and POI
    category. When the user is at other
    places, it is not considered as working.
- Running errands: When the user is not at
    poi name2 or poi name1, and the POI is
    unlikely to be a place for shopping,
    entertainment or eating, it is
    considered as running errands.
Note: If multiple conditions are met,
    priority should be given to 'At Home'
    and 'Working'.\n    There are 3 stays
    in the trajectory data. So, the output
    should have 3 predicted intents.
Respond using a list: ["intent1", "intent2"
    ...]"

answer:
['Working','Working','At Home']
```