Understanding and Predicting Tourist Behavior through Large Language Models

Anna Dalla Vecchia[1 [0000−0001−7026−5205]], Simone Mattioli[1], Sara Migliorini[1 [0000−0003−3675−7243]], and Elisa Quintarelli[1 [0000−0001−6092−6831]]

Department of Computer Science, University of Verona, Verona, Italy
{name.surname}@univr.it

Abstract. Understanding and predicting how tourists move through a city is a challenging task, as it involves a complex interplay of spatial, temporal, and social factors. Traditional recommender systems often rely on structured data, trying to capture the nature of the problem. However, recent advances in Large Language Models (LLMs) open up new possibilities for reasoning over richer, text-based representations of user context. In this study, we investigate the potential of LLMs in interpreting and predicting tourist movements using a real-world application scenario involving tourist visits to Verona, a municipality in Northern Italy, between 2014 and 2023. We propose an incremental prompt engineering approach that gradually enriches the model input, from spatial features alone to richer behavioral information, including visit histories and user-cluster patterns. The approach is evaluated using six open-source models, enabling us to compare their accuracy and efficiency across various levels of contextual enrichment. Results show that incorporating contextual factors improves predictions, resulting in better overall performance while maintaining computational efficiency. The analysis of the model-generated explanations suggests that LLMs mainly reason through geospatial proximity and the popularity of points of interest. Overall, the study demonstrates the potential of LLMs to integrate multiple contextual dimensions for tourism mobility, highlighting the possibility for a more text-oriented and adaptive recommender system.

Keywords: Tourist Recommender Systems · Large Language Models · Next POI Prediction.

1 Introduction

Tourist Recommender Systems (T-RSs) have gained increased attention in recent years, supported by the availability of a huge amount of information produced by tourists in the form of User Generated Content (UGC), and the rise of sophisticated analysis tools based on machine learning (ML) and deep learning (DL) techniques. Understanding tourists' behavior and predicting their future movements is crucial for producing meaningful suggestions that will be appreciated and accepted by the tourists themselves. However, this is a challenging task, as it involves a complex interplay of spatial, temporal, and social factors, such as

individual user preferences and interactions between different tourists visiting the same area. Moreover, in many real-world situations, touristic applications deal with anonymous or occasional users interacting with a specific application for the first time during each trip or visit. Therefore, in the tourism domain, the development of personalized suggestions becomes even more challenging, often requiring a more flexible form of personalization, such as tailoring recommendations to user clusters or broader user categories rather than to individual users [15].

Several different T-RSs have been proposed in the literature, relying on the collection of structured data to capture the nature of the specific problem at hand, which will, in turn, be used to train more or less sophisticated specialized ML or DL models. These approaches typically fall into the category known as *next-PoI prediction*, which, given the tourist's current position and the sequence of attractions already visited, attempts to predict the next location or place the user will visit [7]. Given this nature, the next-PoI recommendation is usually treated as a sequential recommendation task; therefore, in the past, T-RSs have frequently applied ML and DL techniques typically developed in the context of time-series forecasting, starting from the use of recurrent neural networks [3], passing from reinforcement learning approaches [16] and attention-based methods [14], towards the more recent transformer-based models [28]. However, recent advances in Large Language Models (LLMs) open up new possibilities for reasoning over richer, text-based representations of user context. Moreover, the exploitation of foundational pre-trained models enables the possibility of making meaningful predictions without the need for training a model on specific data, making such approaches applicable even in the absence of historical data or in the presence of a very limited amount of it.

In this paper, we investigate the potential of LLMs in interpreting and forecasting tourist movements in a next-PoI prediction task by comparing six open-source LLM models and experimenting with an incremental prompt engineering approach to incrementally enhance the input provided to the models. In particular, starting from the simplest description of the past user visits, it is then enriched with spatial and temporal features, as well as user-cluster preference patterns. The comparison is performed with reference to both accuracy and efficiency using a real-world application dataset that includes tourist visits to Verona, a municipality in Northern Italy, between 2014 and 2023. Three baselines are considered: random choice, spatial-proximity choice, and popularity-based choice. The obtained results confirm that incorporating contextual factors improves predictions, resulting in better overall performance with respect to the baselines, while maintaining computational efficiency. We also perform an analysis of the explanations provided by the six models about the provided suggestions. This analysis reveals that LLMs primarily reason through geospatial proximity and occasionally consider the popularity of points of interest. Overall, the study demonstrates the potential of LLMs to integrate multiple contextual dimensions for tourism mobility, highlighting the possibility of a more text-oriented and adaptive T-RS.

The remainder of the paper is organized as follows: Sect. 2 summarizes some related work about T-RS and the use of LLM in such context. Sect. 3 formalizes the next-PoI prediction problem, while Sect. 4 introduces the applied methodology, and Sect. 5 presents the experimental results. Finally, Sect. 6 concludes the work and proposes some future extensions.

2 Related Work

Next-PoI Prediction. The next-PoI prediction has been widely studied as a sequential recommendation problem that combines spatial and temporal information. Most of the techniques rely on recurrent neural networks. For instance, the study presented in [3] proposed a model that integrates the location interests of similar users and contextual information, such as time, current location, and friends' preferences. In [14], the authors introduced STAN, a spatio-temporal attention network that explicitly models point-to-point interaction among non-adjacent locations through a bi-layer attention mechanism. By replacing traditional hierarchical gridding and explicit time interval encoding with a linear interpolation, STAN enhances the representation of long-range spatial–temporal dependencies while remaining focused on user-specific patterns. The work [28] further enhanced this line of research with GETNext, which incorporates a global trajectory flow map into a transformer architecture. By combining global transition patterns, users' general preferences, spatio-temporal context, and time-aware category embeddings, the model captures inter-user dependencies and alleviates cold-start issues. A different direction was explored in [16], where the next-PoI task was formulated as a reinforcement-learning problem (QEXP). Their model leverages tourists' past experiences and spatial proximity to recommend diverse and geographically dispersed PoIs, addressing new-user, new-item scenarios, and popularity biases. Overall, these works mark a shift from purely sequential models toward context-aware approaches that integrate spatial, temporal, and behavioral signals, providing the basis for more flexible and interpretable language-based representations of trajectories.

Context-Aware Recommendation. Context-aware recommender systems enhance personalization by integrating contextual factors such as time, location, and weather into the recommendation process [1,19,31]. Specifically in the tourism domain, incorporating spatio-temporal and environmental context has proven particularly effective. For instance, the authors in [30] consider both the time of day and the geographical position of attractions, improving next-PoI prediction accuracy over non-contextual baselines. Similarly, integrating temporal and environmental variables such as weather has been shown to improve both the crowding of PoIs [17] and the sustainability of the suggested itinerary [5]. Recent studies have moved toward dynamic and user-adaptive contexts, where both user preferences and item characteristics evolve over time. Neural network architectures [32] and sentiment-aware models [12,26] have been proposed to refine the prediction of tourist interest when it changes dynamically. Despite

these advances, most systems remain feature-driven, relying on fixed contextual representations and limited reasoning capabilities. Consequently, current context-aware recommender systems struggle to integrate heterogeneous signals or explain their decisions. Overcoming these constraints motivates the exploration of Large Language Models (LLMs), which can flexibly reason over spatial, temporal, and behavioral contexts through natural language understanding.

Large Language Models. Large Language Models (LLMs) are transformer-based, pre-trained models containing billions of parameters, trained on massive amounts of text data. Some of the most popular models are GPT-3 [2], GPT-4 [18], LLaMA [21], and Gemini [20]. While initially developed for language understanding and generation, their emergent capabilities have enabled successful applications in many other domains. A key characteristic of LLMs is their in-context learning (ICL) ability [2]. Instead of requiring fine-tuning, an LLM can adapt to a new task through natural-language prompts that combine task instructions and examples. The prompt engineering plays a crucial role in the model performance. Because of this, the chain-of-thought (CoT) prompt strategy [25] becomes a basis for several prompting extensions [10,24,29], as it encourages the model to reason explicitly through intermediate steps before producing an answer. These advances suggest that LLMs are not limited to linguistic tasks, but can be leveraged for structured reasoning on sequential data, including human mobility. Recent studies, such as UrbanGPT [13] and Traj-LLM [11], demonstrate that LLMs can capture spatial and temporal dependencies and infer movement patterns when provided with well-designed contextual prompts. Nevertheless, the application of LLMs to tourist behavior prediction remains quite unexplored. A recent study, LLM-Mob [23] showed that human mobility can be effectively modeled by treating trajectories as language sequences and leveraging ICL for interpretable predictions. While this marks an important step toward understanding mobility through language, it does not yet consider tourism-specific trajectories or the role of contextual enrichment in improving predictive quality. Building upon these insights, this work explores whether LLMs can both predict and explain tourist behavior when enriched with spatial, temporal, and behavioral context.

3 Problem Statement

This section formalizes the preliminary notions and the problem of interest. First, we need to define the concepts of tourist visits and tourist trajectories.

Definition 1 (Visit). *A tourist visit is a tuple $v = (p, t, \ell)$ where p is a PoI identifier, t is the timestamp of the visit, and ℓ is the location of p in terms of latitude and longitude $\ell = (lat, lon)$.*

In this paper, we assume that a predefined set of tourist attractions or PoIs has been identified, and we collectively denote the set of all available PoIs from which a tourist may choose by the symbol $\mathcal{P}$.

Based on the concept of a tourist visit, we define a tourist trajectory as a sequence of visits made by the same user.

Definition 2 (Trajectory). *A tourist trajectory is a sequence of visits performed by a tourist* $\tau = \langle v_1, \ldots, v_n \rangle$*, where each* v_i *is a tourist visit and the following constraint holds:* $\forall v_i, v_j \in \tau \,.\, i < j \wedge v_i.t < v_j.t$*, where* $v.t$ *denotes the timestamp associated with the visit* v*.*

Given such preliminary definitions, a common challenge in the development of a T-RS is to predict which is the next PoI the tourist will visit. This problem is typically known as next-PoI prediction and can be formalized as follows.

Definition 3 (Next-PoI Prediction). *Given a set of available PoIs* $\mathcal{P}$ *and a partial tourist trajectory* $\tau = \langle v_1, \ldots, v_m \rangle$ *performed by a tourist till the time* $v_m.t$*, the goal is to predict the next PoI* $p \in \mathcal{P}$ *that the tourist will visit.*

In the following, the partial tourist trajectory $\tau = \langle v_1, \ldots, v_m \rangle$ till the current tourist position $v_m.\ell$ will be referred to as the sequence of *historical visits*. These concepts are also represented in Fig. 1, where each historical visit is characterized by a timestamp (represented by a clock), a location (represented by a location mark), and a PoI identifier, as formalized in Def. 1.

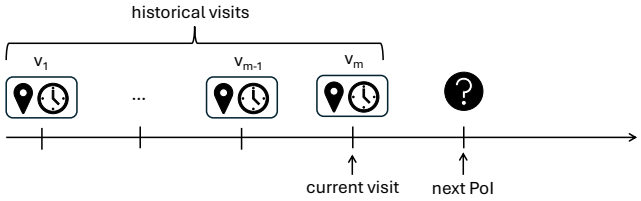

Fig. 1. A partial trajectory for predicting the next PoI from past visits.

The next section outlines the methodology, including the pipeline overview, prompt design, and approach for identifying tourist preferences.

4 Methodology

The overall methodology followed in this paper is depicted in Fig. 2. It begins with the presence of a collection of records about historical visits made by tourists (D1), on which an aggregation and ordering procedure T1 is applied to reconstruct the available set of original tourist trajectories (O1). A filtering is applied to them (T2) in order to discard trajectories that are too short to be relevant for the problem (i.e., they consist of only one or two visits). The set of selected trajectories (O2) is then used as input for the two subsequent tasks: one is the identification of tourist preferences (T3), which will be described in

Sect. 4.1, and the other is the identification of historical visits to use as input for the LLM. In particular, the identification of historical visits involves cutting the original trajectories at a given position, known as the anchor point.

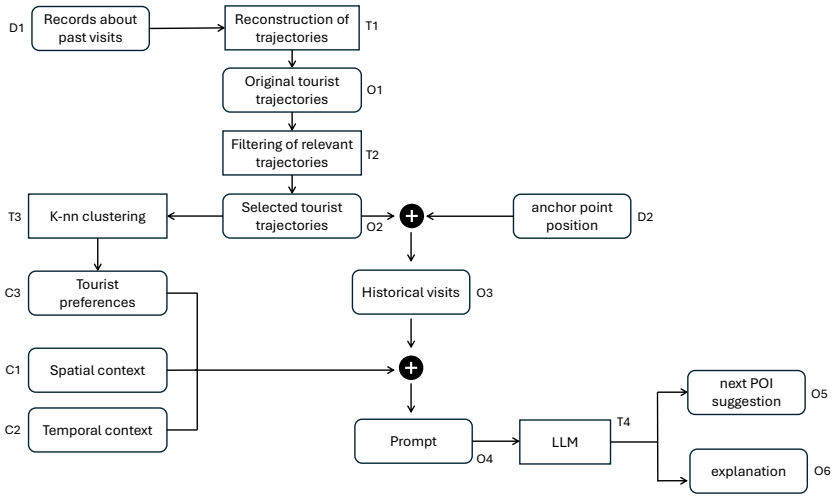

Fig. 2. Overview of the pipeline methodology.

Definition 4 (Anchor point). *Given a complete tourist trajectory* $\tau = \langle v_1, \ldots, v_n \rangle$ *performed by a tourist* u*, an anchor point is an index* $i \in \mathbb{N} \,.\, 1 \leq i < n$ *which identifies an intermediate position inside* τ *in terms of a distance from the end of the sequence. For instance, if* $i = 1$*, it means that the anchor point is located in the penultimate position of the sequence, while* $i = (n-1)$ *denotes the first position in the sequence.*

The anchor point determines the length of the subsequence of visits used to model the user's behaviour before predicting the next PoI. For example, when the anchor point is set to 1, the model predicts the last PoI to be visited using all previously visited PoIs except the last one. The extracted set of historical visits (O3), obtained from the selected tourist trajectories (O2) by considering only the PoIs visited before the specified anchor point (D2), is used to build the prompt for the LLM (O4), which will be enriched with other contextual information. In particular, we consider three incremental contextual prompt information (i.e., C1, C1+C2, or C1+C2+C3), which will be described in Sect. 4.2. The LLM (T4), when queried, will provide two outputs: the next PoI suggestion (O5) and an explanation for that suggestion (O6).

4.1 Identification of Touristic Preferences

To refine the recommendation process, tourist preferences are inferred by clustering the set of visited PoIs (T3). The aim is to identify classes of typical tourists from historical data, where each class reflects distinct preferences for Points of Interest. Once a next-PoI prediction is made for a tourist, the visits completed up to the anchor point can be used to classify such a user and estimate their preferences for the remaining PoIs. To identify the possible classes of tourists, each tourist record in O2 is transformed into a binary vector, where each position corresponds to a PoI and takes the value 1 if the PoI was visited and 0 otherwise. For instance, a user who visited PoIs 2, 3, 4, 7, and 8 would be represented by the vector $\langle 0, 1, 1, 1, 0, 0, 1, 1, 0, 0 \rangle$. These vector representations are then clustered using the k-means algorithm, and as a result, the centroid of each cluster is again a vector where each position indicates the popularity of the corresponding PoI, measured by the number of tourists in that cluster who visited it. In these terms, each centroid identifies distinct tourist behavioral profiles and provides additional information for the LLM prompt, enabling personalized predictions that align with demonstrated behavioral patterns, as described in the following section.

4.2 Prompt Design

This section defines five incremental prompting strategies that progressively enrich the contextual information provided to the model. Each prompt includes at least four main components: (i) *visited PoIs*, listing the attraction already visited in chronological order (i.e., historical visits till the anchor point), (ii) *current location*, corresponding to the most recently visited PoI, (iii) *task instruction*, specifying the expected output, i.e., "Suggest the 5 most likely next PoIs considering typical tourist movement patterns in Verona"; and (iv) *output format* which constrains the model to reply only with a JSON file including the fields `prediction`, representing the identifiers of the recommended PoIs, and `reason` providing a brief explanation of the prediction. Given these four components common to all strategies, each specific strategy can be enriched as summarized in Tab. 1 and described in more detail in the following paragraphs.

(A) Base strategy. The first prompting strategy serves as the foundational LLM baseline, since it does not include any contextual information beyond the chronological sequence of visited PoIs. The model receives the ordered list of previously visited locations (i.e., `chronological_history`) and the current PoI, both represented exclusively by their canonical names. The corresponding prompt template is shown in Tab. 1 denoted by the strategy A.

(B) Spatial strategy – This prompt extends the base strategy by introducing explicit spatial information. In addition to the sequence of visited PoIs, the model is provided with the coordinates of the current location and a list of the ten nearest PoIs, sorted by distance, i.e., `top_10_nearest_with_distances`. These

Table 1. Overview of the four hierarchical prompting strategies (A–E): (A) the base strategy which provides only the sequence of visited PoIs, (B) the inclusion of spatial context, (C) the addition of temporal information, and (D),(E) the integration of tourist preferences derived from clustering analysis in two different way. Each prompt includes structured instructions and output formatting constraints.

Prompt	Strategy
`Cluster typical preference: {the_most_preferred_PoI}`	D
`Cluster typical preferences: {cluster_prefs_with_freqs}`	E
`Visited PoIs: {chronological_history}`	A,B
`Visit history with timestamps: {chronological_history_with_time}`	C,D,E
`Current location: {current_poi}`	A
`Current location: {current_poi} (GPS: {lat}, {lon})`	B,C,D,E
`Current time: {day_of_week}, {hour}:{minute}`	C,D,E
`Nearest PoIs: {top_10_nearest_with_distances}`	B,C,D,E
`Suggest the 5 most likely next PoIs considering typical tourist movement patterns in Verona.`	A

`Suggest the 5 most likely next PoIs considering:`
`- Physical distance from current location`
`- Typical tourist route patterns in Verona`
`- Walking accessibility constraints (2km radius)`
	B

`Suggest the 5 most likely next PoIs considering the current time ({time_period}), the temporal tourist patterns in Verona, suggest 5 most likely next PoIs.`	C

`Suggest the 5 most likely next PoIs considering:`
`- Cluster preferences and typical behavior`
`- Current time ({time_period}) and temporal patterns`
`- Spatial proximity and walking constraints`
`- Historical visit sequence. Suggest 5 most likely next PoIs that align with this tourist's behavioral profile.`
	D,E

`Respond ONLY in JSON format: {"prediction": ["PoI1", "PoI2", "PoI3", "PoI4", "PoI5"], "reason": "brief explanation"}`	A,B,C,D,E

additions enable the model to reason about spatial proximity and typical tourist movement patterns in Verona. As illustrated in Tab. 1 (strategy B), the task instruction is refined to consider physical distance, route patterns, and walking accessibility constraints, while the `reason` field in the output is expected to provide, correspondingly, a brief spatial justification of the prediction.

(C) Spatio-temporal strategy – This strategy builds upon the spatial prompt and integrates temporal information. The history of visited PoIs is enriched with the duration of each visit (i.e., `chronological_history_with_time`), while the current context includes the day of the week, the hour (in 0–23 format), and the minutes of the current moment. Accordingly, the task instruction is refined to require the model to account for temporal dynamics, including time-of-day

classification (i.e., morning, afternoon, or evening). The temporal context is also reflected in the `reason` field of the output, which is expected to provide brief spatio-temporal reasoning. This strategy is denoted as C in Tab. 1.

(D) Spatio-temporal-popularity strategy – This prompt strategy is based on the previous one by integrating behavioral information derived from the k-means clustering analysis introduced in Sect. 4.1. Each tourist trajectory is assigned to a cluster that characterizes the tourist preferences for each PoI. This preference is explicitly embedded in the LLM prompt (strategy D in Tab. 1) by including only the most popular PoI of the cluster. The task instruction and output format are further refined to ensure that the model reasoning reflects the tourist's behavioral profile, combining spatial, temporal, and preference-based information.

(E) Spatio-temporal-preference strategy – The final prompt strategy is based on (C) strategy and is similar to the (D) one. It integrates the behavioral information derived from the k-means clustering analysis introduced in Sect. 4.1 by assigning to each cluster the preference for each PoI. These preferences are explicitly embedded into the LLM prompt, as in the previous case, but by changing the `Cluster typical preference` field, which now provides the preference of each remaining PoI in decreasing order, rather than the single most preferred PoI. The task instruction and output format are equal to the (D) strategy.

4.3 Evaluation Metrics and Quality Assurance

The framework employs a comprehensive evaluation protocol that encompasses multiple metrics commonly used to assess the quality of recommendations. Specifically, the model performances are evaluated using Top-1 Accuracy ($Acc_{@1}$), Top-k Hit Rate ($HR_{@k}$), and Mean Reciprocal Rank (MRR). Each metric is formally defined as follows.

$$Acc_{@1} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{y_i = \hat{y}_i^{(1)}\},$$

where N is the number of test instances, y_i denotes the ground-truth next PoI, $\hat{y}_i^{(j)}$ represents the j-th ranked prediction, in this case the first one. This metric measures the number of cases in which the first system recommendation exactly matches the true next PoI.

$$HR_{@k} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{y_i \in \{\hat{y}_i^{(1)}, \ldots, \hat{y}_i^{(k)}\}\},$$

where $\{\hat{y}_i^{(1)}, \ldots, \hat{y}_i^{(k)}\}\}$ represents the list of the top-k recommendations for instance i. $HR_{@k}$ relaxes the $Acc_{@1}$ metrics by checking if the correct item appears among the top-k, better reflecting tourism scenarios where users consider several suggested PoIs rather than just one.

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank_i},$$

where $rank_i$ is the rank position of the first relevant results for the correct PoI y_i in the ranked list of predictions, if y_i does not appear in the list, $1/rank_i$ is set to 0. MRR quantifies how highly the correct item is ranked on average, assigning higher scores when the correct PoI is at the top of the list.

5 Experiments

All experiments were conducted on Leonardo, the Italian supercomputer operated by CINECA, under the IscrC_LLM-Mob project allocation [4]. The Booster module of Leonardo is equipped with four NVIDIA A100 GPUs (64 GB VRAM each, 256 GB total) interconnected via NVLink, and powered by dual-socket Intel Xeon Sapphire Rapids CPUs with 56 cores per socket (112 cores in total) and 512 GB of DDR5 system memory. This high-performance configuration enabled efficient large-scale parallel processing and multi-instance GPU deployment through the Ollama framework. The complete framework is implemented in Python, featuring comprehensive logging, automatic checkpointing, and support for parallel execution and append mode to enable incremental experiments on large datasets. All experimental code, preprocessing pipelines, and analysis notebooks are available as open-source software at `https://github.com/4nninia/llm_tourist_trajectories`, ensuring full reproducibility.

5.1 Experimental Protocol and Anchor Selection Mechanism

Our evaluation employs a systematic approach, utilizing the VeronaCard dataset, which provides comprehensive tourist mobility trajectories from 2014 to 2023. The dataset contains about 2.7M visits performed by about 570K different tourists, i.e., different VeronaCards, covering 18 PoIs in Verona downtown.

The approach relies on three main components. First, user segmentation is achieved via k-means clustering applied to user-PoI interaction matrices, as discussed in Sect. 4.1, enabling cluster-specific prompting strategies and personalized analysis of mobility patterns. The number of clusters is selected empirically by evaluating the silhouette coefficient. For the dataset at hand, $k = 7$ is the configuration that yields the highest silhouette score. Second, a configurable anchor mechanism determines the reference point for next-PoI prediction. The default configuration utilizes the *penultimate* rule ($i = 1$) while alternative strategies supported by the system (*first*, *middle*, and explicit index) enable analysis of the impact of anchor position on prediction quality. In particular, besides the penultimate strategy, we also consider the middle one in the experiments, which dynamically takes only the initial middle of the trajectory. Finally, distance-based PoI ranking utilizes Haversine distance calculations to rank available PoIs by proximity to the current location, within a 2km walkable radius, with dynamic filtering that excludes already visited PoIs.

5.2 Multi-Model Comparative Framework

The experimental framework is designed for systematic comparative analysis across multiple Large Language Model (LLM) architectures. Building on the described setup, we evaluate the accuracy of different LLMs and anchor strategies. The experimental setup includes six open-source LLM architectures with varying sizes and design principles. *Llama 3.1 8B* [22] is an 8-billion-parameter transformer released by Meta, instruction-tuned, namely fine-trained on data consisting of instructions and corresponding responses, and capable of handling long contexts up to 128k tokens. *Qwen 2.5 7B* and *Qwen 2.5 14B* are models from Alibaba's Qwen 2.5 [27], optimized for reasoning and instruction-following tasks, namely to understand and respond appropriately to natural-language commands and questions. *Mixtral 8×7B* [9], from Mistral AI, employs a sparse mixture-of-experts design, wherein each transformer block contains multiple sub-networks specialized in different types of inputs. Therefore, it can activate a subset of its 47 billion parameters per token to achieve strong performance at reduced computational cost. *Mistral 7B* [8], also from Mistral AI, is an efficient dense model that employs grouped-query and sliding-window attention to make attention faster and more memory-efficient, especially for long contexts. Finally, *DeepSeek Coder 33B* [6] is a large-scale model designed for code understanding and generation, offering strong generalization across various structured prediction tasks.

5.3 Results

The prediction capabilities of the six mentioned LLM models have been evaluated with respect to all the prompt strategies described in Sect. 4.2, by using a real-world dataset of visits in Verona, a city in Northern Italy, from 2014 to 2023. Overall, approximately 554,000 trajectories have been selected as relevant (see T2 in Fig. 2). The obtained results across models, contextual prompt strategies, and anchor point configurations (*middle* and *penultimate*), have also been compared with respect to three baselines: *random*, which randomly chooses the next PoI among the remaining ones, *nearest*, which always selects the nearest available PoI to the current location, and *popular*, which returns the most popular PoI among the remaining ones.

Tab. 2 reports the obtained results under five prompt strategies that differ in context, with the results evaluated using $Acc_{@1}$, $HR_{@5}$, and MRR as evaluation metrics. For each configuration, the average (AVG) and standard deviation (STD) metrics values are reported for all VeronaCard predictions.

Overall, the obtained results indicate that the prompt strategy, which integrates tourist preferences through clustering (strategy E), achieves the best performance across all models, in terms of $Acc_{@1}$, $HR_{@5}$, and MRR. For the *middle anchor point* configuration, the *Mixtral 8x7B* model attains the highest top-1 accuracy ($Acc_{@1} = 34.27$), while the *Qwen2.5 14B* model achieves the best top-5 hit rate and mean reciprocal rank ($HR_{@5} = 73.92$, $MRR = 49.01$). Under the penultimate anchor point configuration, *Mixtral 8x7B* again delivers the best $Acc_{@1}$ (32.15), whereas *Qwen2.5 14B* maintains its lead in $HR_{@5}$

Table 2. Results by model, prompt strategy, anchor point, and metric. BL = baseline, LL8 = Llama3.1 8B, QW7 = Qwen2.5 7B, QW14 = Qwen2.5 14B, MX8 = Mixtral 8x7B, MS7 = Mistral 7B, DS33 = DeepSeek Coder 33B. For each configuration, the average (AVG) and standard deviation (STD) of Top-1 accuracy ($Acc_{@1}$), Top-5 hit rate ($HR_{@5}$), and Mean Reciprocal Rank (MRR) are reported. The highest values for each metric are highlighted in **bold**.

		Middle						**Penultimate**					
		$Acc_{@1}$		$HR_{@5}$		MRR		$Acc_{@1}$		$HR_{@5}$		MRR	
Model	**Context**	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
BL	Random	5.22	22.24	26.05	43.89	11.9	24.93	5.65	23.09	28.19	44.99	12.88	25.69
	Nearest	2.06	14.2	35.61	47.89	11.82	19.21	3.3	17.87	29.87	45.77	11.0	21.15
	Popular	29.63	45.66	29.63	45.66	29.63	45.66	**32.39**	46.8	32.39	46.8	32.39	46.8
LL8	A	8.11	27.3	25.5	43.58	13.54	28.49	11.27	31.62	27.39	44.6	17.2	32.74
	B	13.83	34.52	50.19	50.0	24.53	33.57	4.9	21.6	41.88	49.34	15.83	24.3
	C	14.98	35.68	49.86	50.0	25.74	34.66	14.29	35.0	45.39	49.79	24.2	34.48
	D	13.89	34.59	48.86	49.99	25.14	34.11	10.35	30.46	42.43	49.42	20.74	31.3
	E	31.11	46.29	67.6	46.8	44.53	40.86	27.47	44.64	57.99	49.36	38.5	41.29
QW7	A	0.01	1.03	0.88	9.33	0.34	3.82	0.0	0.71	0.41	6.39	0.17	2.72
	B	19.98	39.99	50.79	49.99	29.34	38.09	14.41	35.12	43.03	49.51	23.16	34.66
	C	18.6	38.91	50.53	50.0	29.43	37.47	13.59	34.27	42.88	49.49	23.59	34.44
	D	10.9	31.16	45.91	49.83	22.78	31.99	8.88	28.45	37.03	48.29	18.34	30.16
	E	31.57	46.48	71.6	45.09	47.02	39.94	25.82	43.77	62.94	48.3	40.0	39.68
QW14	A	0.86	9.23	20.83	40.61	6.21	14.35	2.91	16.82	30.66	46.11	11.25	21.19
	B	26.29	44.02	64.82	47.75	39.66	39.71	20.02	40.02	54.16	49.83	31.9	38.01
	C	25.54	43.61	62.25	48.48	37.99	39.79	19.09	39.3	52.13	49.95	30.36	37.62
	D	14.96	35.66	61.22	48.73	30.59	33.92	12.21	32.74	50.28	50.0	25.02	32.84
	E	34.03	47.38	**73.92**	43.91	**49.01**	40.25	31.02	46.26	**65.49**	47.54	**43.82**	41.31
MX8	A	1.35	11.53	34.27	47.46	14.2	21.69	4.13	19.89	35.42	47.83	16.78	26.09
	B	13.14	33.78	57.41	49.45	28.55	33.12	10.16	30.21	48.29	49.97	23.31	31.25
	C	5.99	23.73	52.98	49.91	20.91	26.11	6.93	25.39	45.74	49.82	19.3	27.49
	D	13.28	33.93	57.13	49.49	28.23	33.19	11.33	31.7	46.94	49.91	23.76	32.44
	E	**34.27**	47.46	71.56	45.11	48.82	40.77	**32.15**	46.71	59.93	49.0	42.88	42.77
MS7	A	0.17	4.17	13.72	34.4	5.03	13.44	0.14	3.67	22.5	41.76	8.49	16.72
	B	25.49	43.58	58.0	49.36	36.68	40.38	22.32	41.64	46.37	49.87	30.35	40.12
	C	17.83	38.28	53.36	49.89	28.83	36.46	17.46	37.96	41.46	49.27	24.82	37.17
	D	17.67	38.14	53.5	49.88	30.14	36.63	12.64	33.23	43.71	49.6	23.56	33.68
	E	29.85	45.76	69.85	45.89	45.17	39.79	30.88	46.2	65.37	47.58	43.75	41.28
DS33	A	0.01	1.14	0.52	7.18	0.18	2.56	2.33	15.09	25.51	43.59	9.27	19.56
	B	4.98	21.75	40.15	49.02	15.59	24.6	4.32	20.33	35.04	47.71	13.93	23.87
	C	4.72	21.22	40.57	49.1	15.6	24.34	4.02	19.65	35.46	47.84	13.8	23.44
	D	5.49	22.79	43.24	49.54	17.25	25.82	4.87	21.53	37.17	48.33	15.27	25.14
	E	28.24	45.02	60.21	48.95	39.66	41.24	25.11	43.36	51.47	49.98	34.6	41.14

and MRR (65.49 and 43.82, respectively). In general, comparing the two anchor point strategies, the *middle* configuration performs slightly better on average than the *penultimate* one, showing an average improvement of approximately 6.6% in $Acc_{@1}$, 12.9% in $HR_{@5}$, and 11.8% in MRR, suggesting that aligning the prompt with the *middle anchor point* enhances model adaptability to user preference patterns. The only exception is observed for $Acc_{@1}$ under the *penultimate anchor point* configuration, which marginally outperforms the popularity-based baseline. Nevertheless, the difference is marginal, indicating that employing LLMs with the clustering-based prompt strategy (E) remains

generally preferable. Compared to baselines that rely only on the popularity of the PoIs, the nearest PoI, or random selection, all LLM-based models consistently achieve higher performance, particularly in terms of $HR_{@5}$ and MRR, highlighting the capability of LLM to better incorporate contextual and user-specific information in the next-PoI prediction task.

Table 3. Average, minimum, and maximum computation time in seconds for each LLM and prompt strategy. LL8 = Llama3.1 8B, QW7 = Qwen2.5 7B, QW14 = Qwen2.5 14B, MX8 = Mixtral 8x7B, MS7 = Mistral 7B, DS33 = DeepSeek Coder 33B.

Context	LL8	QW7	QW14	MX8	MS7	DS33
Min (s)						
A	0.762	0.299	0.927	0.844	0.532	0.545
B	0.685	0.128	0.762	0.249	0.505	1.196
C	0.550	0.567	0.831	0.969	0.713	1.516
D	0.173	0.778	0.200	0.179	0.179	1.472
E	0.553	0.711	0.981	0.977	0.724	0.183
Mean (s)						
A	1.349	1.548	2.400	2.881	1.573	3.850
B	1.259	1.150	1.669	3.134	1.285	3.258
C	1.610	1.327	1.624	3.963	1.531	3.167
D	1.356	1.727	2.479	4.270	1.876	4.306
E	1.478	1.769	2.632	4.576	2.184	5.401
Max (s)						
A	8.801	28.308	12.974	179.685	12.317	174.371
B	8.509	73.986	25.250	586.100	11.672	279.573
C	32.028	76.691	13.608	504.931	11.135	103.022
D	60.572	9.017	164.247	245.922	77.656	260.507
E	9.107	9.363	39.187	265.486	13.146	25.463

Another important factor is the model response time under different prompt strategies. Tab. 3 reports the minimum, mean, and maximum execution time in seconds per prediction. The results show that, in general, as prompt complexity and contextual information increase, average time rises slightly but remains acceptable. Larger models, such as *Mixtral 8x7B* and *DeepSeek Coder 33B*, exhibit higher latency in worst-case scenarios, while richer contextual prompts boost reasoning quality with only a modest rise in computation time.

Finally, to evaluate the decision-making processes and the argumentative quality of the LLMs, a textual analysis was performed on the `reason` field. This analysis focuses exclusively on predictions marked as `success` and containing a non-null `reason`. Argumentative patterns were identified through heuristic keyword matching, allowing the reasoning content to be classified into four main semantic categories: *geospatial reasoning* (e.g., near, route, walk, meters), *popularity reasoning* (e.g., popular, famous, highlight, guidebook, important), *time reasoning* (e.g., hour, before, when, late), and *other*, which includes the category and logical reasoning. This classification provides a structured overview of the reasoning strategies employed by each model, serving as the basis for the subsequent comparative analysis. Fig. 3 illustrates the percentage distribution of reasoning types for the clustering prompt strategy (E), which achieves the best

performance according to Tab. 2. Geospatial reasoning dominates most models, except *DeepSeek Coder 33B*, which favors time reasoning.

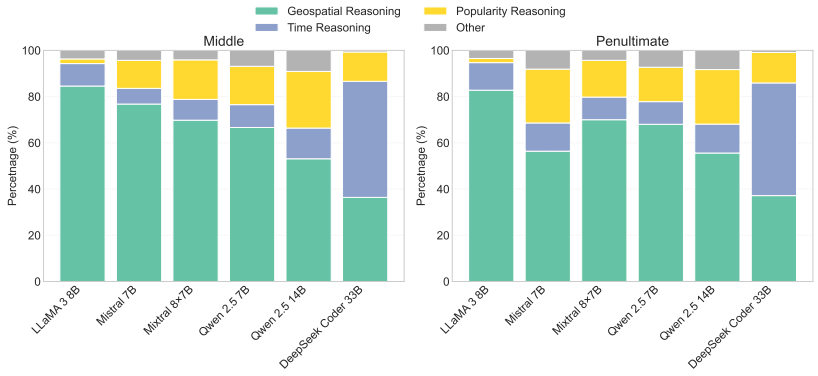

Fig. 3. Percentage distribution of reasoning categories for the richer prompt strategy (E) across models and anchor selection mechanism.

6 Conclusion

This work explored the potential of LLMs to understand and predict tourist movements in a next-PoI prediction task through an incremental prompt strategy, using a real-world tourist dataset from the municipality of Verona, Italy.

Experiments conducted across six different LLM models demonstrated that progressively enriching the prompt with spatial, temporal, and preference information can significantly improve prediction accuracy compared to traditional baselines. *Qwen2.5 14B* and *Mixtral 8x7B* achieved the best overall results when integrating the list of clustering preferences, suggesting that the middle anchor point strategy better captures user behavior patterns. At the same time, the reasoning analysis revealed that most LLMs primarily rely on geospatial reasoning, while temporal and popular reasoning play a secondary role, except for the *DeepSeek Coder 33B*, which exhibits stronger temporal awareness.

Overall, our findings indicate that LLMs can serve as flexible, data-efficient mobility interpreters capable of integrating heterogeneous contextual dimensions. Future research will explore the integration of additional context sources, such as weather conditions and real-time PoI crowding, to enhance personalization, interpretability, and scalability in real-world tourist recommender systems.

Acknowledgments. We acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CINECA (Italy).

References

1. Adomavicius, G., Bauman, K., Tuzhilin, A., Unger, M.: Context-Aware Recommender Systems: From Foundations to Recent DevelopmentsContext-aware recommender systems, pp. 211–250. Springer US, New York, NY (2022). https://doi.org/10.1007/978-1-0716-2197-4_6
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
3. Chen, M., Li, W.Z., Qian, L., Lu, S.L., Chen, D.X.: Next POI Recommendation Based on Location Interest Mining with Recurrent Neural Networks. Journal of Computer Science and Technology **35**(3), 603–616 (2020). https://doi.org/10.1007/s11390-020-9107-3
4. CINECA Supercomputing Centre, S.A., Department, I.: Leonardo: A pan-european pre-exascale supercomputer for hpc and ai applications. Journal of large-scale research facilities JLSRF **9**(1) (2024). https://doi.org/https://doi.org/10.17815/jlsrf-8-186
5. Dalla Vecchia, A., Migliorini, S., Quintarelli, E., Gambini, M., Belussi, A.: Promoting sustainable tourism by recommending sequences of attractions with deep reinforcement learning. Information Technology & Tourism **26**(3), 449–484 (2024)
6. Guo, D., Zhu, Q., Yang, D., Xie, Z., Dong, K., Zhang, W., Chen, G., Bi, X., Wu, Y., Li, Y.K., Luo, F., Xiong, Y., Liang, W.: Deepseek-coder: When the large language model meets programming – the rise of code intelligence. arXiv preprint arXiv:2401.14196 (2024), https://arxiv.org/abs/2401.14196
7. Islam, M.A., Mohammad, M.M., Das, S.S.S., Ali, M.E.: A Survey on Deep Learning Based Point-Of-Interest (POI) Recommendations. CoRR **abs/2011.10187** (2020), https://arxiv.org/abs/2011.10187
8. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.A., Stock, P., Scao, T.L., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mistral 7b: A 7-billion-parameter foundation language model with grouped-query and sliding-window attention. arXiv preprint arXiv:2310.06825 (2023), https://arxiv.org/abs/2310.06825
9. Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D.S., de las Casas, D., Hanna, E.B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L.R., Saulnier, L., Lachaux, M.A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T.L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mixtral 8×7b: A sparse mixture-of-experts language model. arXiv preprint arXiv:2401.04088 (2024), https://arxiv.org/abs/2401.04088
10. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22, Curran Associates Inc., Red Hook, NY, USA (2022)
11. Lan, Z., Liu, L., Fan, B., Lv, Y., Ren, Y., Cui, Z.: Traj-llm: A new exploration for empowering trajectory prediction with pre-trained large language models. IEEE

Transactions on Intelligent Vehicles **10**(2), 794–807 (2025). https://doi.org/10.1109/TIV.2024.3418522
12. Li, G., Chen, Q., Zheng, B., Yin, H., Nguyen, Q.V.H., Zhou, X.: Group-based recurrent neural networks for POI recommendation. ACM/IMS Trans. Data Sci. **1**(1) (2020). https://doi.org/10.1145/3343037
13. Li, Z., Xia, L., Tang, J., Xu, Y., Shi, L., Xia, L., Yin, D., Huang, C.: Urbangpt: Spatio-temporal large language models. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. p. 5351–5362. KDD '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3637528.3671578, https://doi.org/10.1145/3637528.3671578
14. Luo, Y., Liu, Q., Liu, Z.: Stan: Spatio-temporal attention network for next location recommendation. In: Proceedings of the Web Conference 2021. p. 2177–2185. WWW '21 (2021). https://doi.org/10.1145/3442381.3449998
15. Massimo, D., Ricci, F.: Building effective recommender systems for tourists. AI Magazine **43**(2), 209–224 (2022). https://doi.org/https://doi.org/10.1002/aaai.12057, https://onlinelibrary.wiley.com/doi/abs/10.1002/aaai.12057
16. Massimo, D., Ricci, F.: Combining reinforcement learning and spatial proximity exploration for new user and new poi recommendations. In: Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization. p. 164–174. UMAP '23 (2023). https://doi.org/10.1145/3565472.3592966
17. Migliorini, S., Dalla Vecchia, A., Belussi, A., Quintarelli, E.: Artemis: a context-aware recommendation system with crowding forecaster for the touristic domain. Information Systems Frontiers pp. 1–27 (2024)
18. OpenAI: Gpt-4 technical report (2024), https://arxiv.org/abs/2303.08774
19. Rahmani, H.A., Deldjoo, Y., Noia, T.D.: The role of context fusion on accuracy, beyond-accuracy, and fairness of point-of-interest recommendation systems. Expert Syst. Appl. **205**, 117700 (2022). https://doi.org/10.1016/j.eswa.2022.117700
20. Team, G., Anil, R., Borgeaud, S., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., Millican, K., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
21. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models (2023), https://arxiv.org/abs/2302.13971
22. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023), https://arxiv.org/abs/2302.13971
23. Wang, X., Fang, M., Zeng, Z., Cheng, T.: Where would i go next? large language models as human mobility predictors (2024), https://arxiv.org/abs/2308.15197
24. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., Zhou, D.: Self-consistency improves chain of thought reasoning in language models (2023), https://arxiv.org/abs/2203.11171
25. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22, Curran Associates Inc., Red Hook, NY, USA (2022)
26. Xing, S., Liu, F., Wang, Q., Zhao, X., Li, T.: Content-aware point-of-interest recommendation based on convolutional neural network. Applied Intelligence **49**(3), 858–871 (2019). https://doi.org/10.1007/s10489-018-1276-1

27. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z.: Qwen technical report. arXiv preprint arXiv:2309.16609 (2023), https://arxiv.org/abs/2309.16609
28. Yang, S., Liu, J., Zhao, K.: 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. p. 1144–1153. SIGIR '22 (2022). https://doi.org/10.1145/3477495.3531983
29. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T.L., Cao, Y., Narasimhan, K.: Tree of thoughts: deliberate problem solving with large language models. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA (2023)
30. Yuan, Q., Cong, G., Ma, Z., Sun, A., Thalmann, N.M.: Time-Aware Point-of-Interest Recommendation. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 363–372. SIGIR'13 (2013). https://doi.org/10.1145/2484028.2484030
31. Zheng, W., Huang, L., Lin, Z.: Multi-attraction, hourly tourism demand forecasting. Annals of Tourism Research **90**, 103271 (2021). https://doi.org/https://doi.org/10.1016/j.annals.2021.103271, https://www.sciencedirect.com/science/article/pii/S0160738321001493
32. Zhou, Y.: A Dynamically Adding Information Recommendation System based on Deep Neural Networks. In: 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS). pp. 1–4 (2020). https://doi.org/10.1109/ICAIIS49377.2020.9194792