big data and cognitive computing

MDPI

Article

Understanding and Predicting Tourist Behavior Through Large Language Models

Anna Dalla Vecchia, Simone Mattioli, Sara Migliorini * and Elisa Quintarelli

Department of Computer Science, University of Verona, 37134 Verona, Italy; anna.dallavecchia@univr.it (A.D.V.); simone.mattioli@studenti.univr.it (S.M.); elisa.quintarelli@univr.it (E.Q.)
* Correspondence: sara.migliorini@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 new possibilities for reasoning over richer, text-based representations of user context, even without a dedicated pre-training phase. In this study, we investigate the potential of LLMs to interpret and predict tourist movements in 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, time information, 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. The results provide a first insight about the abilities of LLMs to incorporate spatio-temporal contextual factors, thus improving predictions, while maintaining computational efficiency. The analysis of the model-generated explanations completes the picture by adding an interpretability dimension that most existing next-PoI prediction solutions lack. Overall, the study demonstrates the potential of LLMs to integrate multiple contextual dimensions in tourism mobility, highlighting the possibility of a more text-oriented, adaptive, and explainable T-RS.

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 tourist behaviors and predicting their future movements is crucial for producing meaningful suggestions that are appreciated and accepted by tourists themselves. However, this is a challenging task, as it involves the complex interplay of spatial, temporal, and social factors, including individual user preferences and interactions among different tourists visiting the same area at the same time. For this reason, several different T-RSs have been proposed in the literature, all 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

Academic Editor: Francesco Archetti

Received: 3 March 2026
Revised: 1 April 2026
Accepted: 9 April 2026
Published: 11 April 2026

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 [1]. Given this nature, a 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 [2], passing from reinforcement learning approaches [3] and attention-based methods [4], towards the more recent transformer-based models [5]. 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 pre-trained foundational models breaks barriers towards the construction of meaningful predictions without the need for specific training, making such approaches applicable even in the absence of, or with very limited amounts of, historical data. This is particularly useful in many real-world tourist applications that deal with anonymous or occasional users interacting with a specific application for the first time during each trip or visit. Indeed, in the tourism domain, developing personalized suggestions is very 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 [6].

Based on these premises, this paper investigates the potential of LLMs for interpreting and forecasting tourist movements in a next-PoI prediction task by comparing six open-source LLMs and experimenting with an incremental prompt engineering approach to enhance the input provided to the models. The aim is to evaluate an LLM's ability to provide meaningful suggestions in the tourism domain without any prior specific training, making the technique particularly suitable in overcoming the aforementioned bootstrapping problem of AI-based T-RSs. Indeed, as mentioned above, traditional T-RS typically suffer from genuine cold-start limitations that can only be partially addressed by transfer learning techniques [7]. As highlighted in the literature, the tourism domain is characterized not only by limited data per tourist, since they typically visit a city only once in a lifetime, but also by difficulties in relying on similarities in tourist preferences due to privacy concerns that prevent a complete picture of individual users. Moreover, when new attractions are introduced or there is a need to promote less popular POIs, this problem can be exacerbated. At the same time, when a large amount of historical data is available, the training process can be very intensive, imposing practical limitations [1,8]. Conversely, the proposed methodology starts from the simplest description of the past anonymous user visits, progressively enriching this information with spatial and temporal features, as well as user cluster preference patterns. In this regard, we can observe that LLMs are typically designed to process and understand textual data, while their capabilities for managing and understanding temporal and spatial components have not been investigated in depth. This paper also provides a first insight into LLMs' abilities to understand concepts such as spatial distance and temporal duration.

The comparative evaluation of six pre-trained LLM models has been undertaken with respect to both accuracy and efficiency by using a real-world dataset of tourists' visits to Verona, a municipality in Northern Italy, between 2014 and 2023. Four baselines are considered, besides the naïve random choice, spatial proximity choice, and popularity-based choice, we compare the methodology with respect to the GETNext [5] approach, an innovative solution based on the use of a transformer-based architecture. The results confirm that incorporating contextual factors improves predictions, resulting in better overall performance than the baselines while maintaining computational efficiency. Moreover, the use of incremental prompts inside the experiments confirms that all the considered LLM models exhibit some geospatial and temporal reasoning capabilities, while the analysis of the reason provided by the suggestion allows us to complete the picture and leave room

for future improvements in this regard. In particular, even if each model gives a different importance to each contextual aspect, all the models demonstrate their potential to integrate multiple contextual dimensions in tourism mobility and highlight the possibility of a more text-oriented, adaptive T-RS.

Even if the paper does not propose an innovative LLM model, its extensive exploration of the capabilities of foundational LLM models in the tourist domain is very useful in practical contexts. In summary, the provided contribution could be considered threefold: (i) it provides a first insight into LLMs' capabilities to understand concepts such as spatial distance and temporal duration, for which they are not originally trained, (ii) it contributes in overcoming the typical cold bootstrap of any AI-based solutions, and (iii) it completes the picture by analyzing the reasons provided for the given suggestions, adding an interpretability dimension that most existing next-PoI prediction solutions lack.

The remainder of the paper is organized as follows: Section 2 summarizes some related work about T-RS and the use of LLM in such a context. Section 3 formalizes the next-PoI prediction problem, while Section 4 introduces the applied methodology, and Section 5 presents the experimental results. Section 6 discusses the obtained results and summarizes the general findings and limitations of the approach. Finally, Section 7 concludes the work and proposes some future extensions.

2. Related Work

This paper provides a picture of the most recent literature findings about AI-techniques and the tourist domain. Given the vast literature in this field, we focus on next-PoI predictions, context-aware recommendations, and LLMs for tourism, which are the most relevant contributions to the present work.

2.1. 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 [2] proposed a model that integrates the location interests of similar users and contextual information, such as time, current location, and friends' preferences. In [4], 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 linear interpolation, STAN enhances the representation of long-range spatial–temporal dependencies while remaining focused on user-specific patterns. The work [5] 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, even though it does not completely solve them. A different direction was explored in [3], where the next-PoI task was formulated as a reinforcement learning problem (QEXP). The model leverages tourists' past experiences and spatial proximity to recommend diverse, geographically dispersed PoIs, addressing new user and 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. Anyway, all these proposals rely on historical data to perform the preliminary dedicated pre-training of the models, which introduces a bootstrap problem in new applications or when there is a contextual shift.

2.2. Context-Aware Recommendation

Context-aware recommender systems enhance personalization by integrating contextual factors such as time, location, and weather into the recommendation process [9–11]. Specifically in the tourism domain, incorporating spatio-temporal and environmental context has proven particularly effective. For instance, the authors in [12] 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 [13] and the sustainability of the suggested itinerary [14]. Recent studies have moved toward dynamic and user-adaptive contexts, where both user preferences and item characteristics evolve over time. Neural network architectures [15] and sentiment-aware models [16,17] 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.

2.3. 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 [18], GPT-4 [19], LLaMA [20], and Gemini [21]. 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 [18]. Instead of requiring fine-tuning, an LLM can adapt to a new task through natural language prompts that combine task instructions and examples. Prompt engineering plays a crucial role in model performance. Because of this, the chain-of-thought (CoT) prompt strategy [22] becomes a basis for several prompting extensions [23–25], as it encourages the model to reason explicitly through intermediate steps before producing an answer. Extensions such as self-consistency [24] and tool-augmented prompting [26] further improve robustness and factual grounding. These advances suggest that LLMs are not limited to linguistic tasks, but can be leveraged for structured reasoning on sequential data, including human mobility. In particular, recent works have explored representing non-textual data (e.g., time series, user behavior, and trajectories) as token sequences, allowing LLMs to generalize to domains beyond natural language [27,28], even without providing evidence about their capability to provide integrated contextual knowledge for context-aware recommendations. Recent studies, such as UrbanGPT [29] and Traj-LLM [30], 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 predicting tourist behavior remains largely unexplored. A recent study, LLM-Mob [31] 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.

In general, LLMs are increasingly being integrated into recommender systems and are now a well-established research direction [32,33]. However, T-RSs constitute a distinct, context-rich application domain, where general-purpose LLM knowledge needs to be driven by contextual relevance, reliability, and factual accuracy [34,35]. 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 Formulation

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 these preliminary definitions, a common challenge in developing a T-RS is predicting which PoI the tourist will visit next. This problem, known as next-PoI prediction, 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 Figure 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 Definition 1.

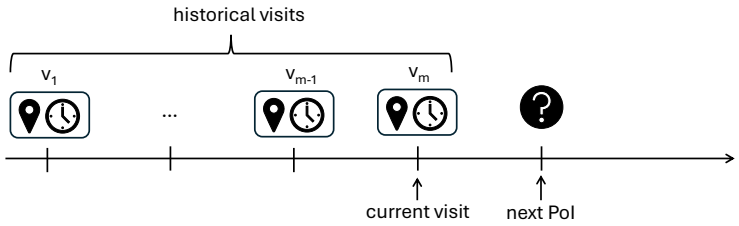

Figure 1. A partial trajectory for predicting the next PoI from past visits. Symbols $v_1, \ldots, v_m$ denote historical visits, including the current one, while the question mark indicates the next PoI that will be visited by the tourist.

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 Figure 2. It begins with a collection of records about historical visits performed by tourists (D1), to which an aggregation and ordering procedure T1 is applied to reconstruct the available set of original tourist trajectories (R1). A filtering is applied to them (T2) to discard trajectories that are too short to be relevant to the problem (i.e., consisting of only one or two visits). The set of selected trajectories (R2) is then used as input for the identification of tourist preferences (T3),

as described in Section 4.1, as well as the identification of historical visits (R3) obtained by cutting the original trajectories at a given position, known as the anchor point (D4).

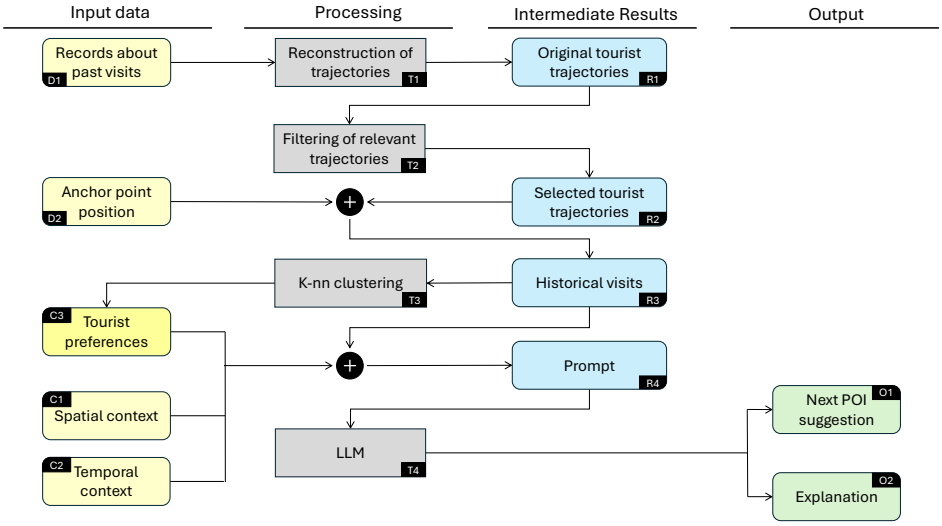

Figure 2. Overview of the pipeline methodology. Rounded boxes represent data, while squared boxes indicate processing steps. The black-and-white label, together with the different colors of the rounded boxes, suggests their roles: a darker yellow box indicates input data produced by previous processing steps rather than from external sources. The white-and-black cross symbol denotes a combination of multiple data sources.

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 used to model the user's behavior before predicting the next PoI. For example, when the anchor point is set to one, the model predicts the last PoI to be visited using all previously visited PoIs except the last one.

The set of historical visits is then used to build the input for the LLM (T4). In particular, the extracted set of historical visits (R3), obtained from the selected tourist trajectories (R2) by considering only the PoIs visited before the specified anchor point (D2), is enriched with other contextual information to build the prompt for the LLM (R4). In particular, we consider three incremental contextual prompt information (i.e., C1, C1 + C2, or C1 + C2 + C3), which will be described in Section 4.2. The LLM (T4), when queried, will provide two outputs: the next PoI suggestion (O1) and an explanation for that suggestion (O2).

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, each reflecting distinct preferences for attractions. Once a next-PoI prediction is made for a tourist, visits completed up to the anchor point can be used to classify the user and estimate their preferences for the remaining PoIs. This classification methodology could be considered quite simple; however, it facilitates providing recommendations without the existence of long-term preference profiles, relying only on what are called in the literature *session-based* preferences [36]. The usefulness of clustering techniques such as this for identifying user classes has also been recognized in [6].

To identify possible tourist classes, 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 the resulting centroids are again vectors, 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 about contextualized popularity, 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 model's contextual information. 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 five most likely next PoIs considering typical tourist movement patterns in Verona", and (iv) *the 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 Table 1 and described in the following paragraphs.

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) and (E) the integration of tourist preferences derived from clustering analysis in two different ways. Each prompt includes structured instructions and constraints on output formatting.

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

(A) Base strategy

The first prompting strategy serves as the elementary LLM prompt 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 Table 1 as 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 additions enable the model to reason in relation to spatial proximity and typical tourist movement patterns. As illustrated in Table 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 Table 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 Section 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 Table 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 the (C) strategy and is similar to the (D) one. It integrates the behavioral information derived from the *k*-means clustering analysis introduced in Section 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 encompassing multiple metrics commonly used to assess recommendation quality. 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.

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

where N is the number of test instances, y_i denotes the ground truth next PoI, and $\hat{y}_i^{(j)}$ 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.

$$HR_{@k} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{y_i \in \{\hat{y}_i^{(1)}, \ldots, \hat{y}_i^{(k)}\}\} \tag{2}$$

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 $Acc_{@1}$ metric by checking whether the correct item appears among the top-k, better reflecting tourism scenarios in which users consider several suggested PoIs rather than just one.

$$MRR = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{rank_i} \tag{3}$$

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 zero. 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

Our evaluation employs a systematic approach and utilizes a real-world dataset containing the tourist mobility trajectories from 2014 to 2023 made by tourists of Verona, a city in Northern Italy, through the use of a city pass, called VeronaCard. The dataset contains about 2.7M visits performed by about 570K different anonymous tourists covering 18 PoIs in Verona downtown, including both well-known attractions and less popular PoIs.

All the experiments were conducted on Leonardo, the Italian supercomputer operated by CINECA, under the IscrC_LLM-Mob project allocation [37]. The Booster module of Leonardo is equipped with four NVIDIA A100 GPUs (NVIDIA, Santa Clara, CA, USA) (64 GB VRAM each; 256 GB total) interconnected via NVLink, and powered by dual-socket Intel Xeon Sapphire Rapids CPUs (Intel, Santa Clara, CA, USA) 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 3.10, featuring comprehensive logging, automatic checkpointing, and support for parallel execution and append mode to enable incremental experiments on large datasets. All the experimental code, preprocessing pipelines, and analysis notebooks are available as open-source software (version 1.0) at https://github.com/4nnina/llm_tourist_trajectories (accessed on 10 March 2026), ensuring full reproducibility.

5.1. Experimental Protocol and Anchor Selection Mechanism

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 Section 4.1, enabling cluster-specific prompting strategies and the personalized analysis of mobility patterns. The number of clusters is selected empirically by evaluating the silhouette coefficient. For the dataset at hand, $k = 7$ 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 the analysis of the impact of anchor position on prediction quality. In particular, in addition to the penultimate

strategy, we also consider the middle strategy in the experiments, which dynamically uses only the initial middle of the trajectory. Finally, distance-based PoI ranking uses the Haversine distance to rank the available PoIs by proximity to the current location within a 2 km walkable radius, with dynamic filtering that excludes already visited PoIs.

5.2. Multi-Model Comparative Framework

The experimental framework is designed for performing a systematic comparative analysis across multiple LLM architectures and anchor strategies. In particular, we consider six open-source LLM architectures with varying sizes and design principles. *LLaMA 3.1 8B* [20] is an 8-billion-parameter transformer released by Meta, instruction-tuned and 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 [38], optimized for reasoning and instruction-following tasks, namely to understand and respond appropriately to natural language commands and questions. *Mixtral 8×7B* [39], 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* [40], also from Mistral AI, is an efficient dense model that employs grouped queries and sliding-window attention to accelerate and make attention more memory-efficient, especially for long contexts. Finally, *DeepSeek Coder 33B* [41] is a large-scale model for code understanding and generation, offering strong generalization across a range of structured prediction tasks.

5.3. Results

The prediction capabilities of the six LLMs mentioned have been evaluated across all the prompt strategies described in Section 4.2 using the real-world dataset of visits to Verona. Overall, approximately 554,000 trajectories have been selected as relevant (see T2 in Figure 2). The obtained results across models, contextual prompt strategies, and anchor point configurations (*middle* and *penultimate*), have also been first compared with respect to three heuristic 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. In addition, we consider a learning-based baseline, namely GETNext [5], a recently proposed transformer-based model for next-PoI recommendation that captures users' future movements by jointly modeling spatial, temporal, and preference-related signals. It is evaluated in two variants: a simplified version without the attention mechanism and the full attention-based model.

Since the GETNext model requires the use of some data for training and some for testing, while the LLM models and the heuristic baselines do not require a dedicated pre-training phase, we split the evaluation into two to ensure fairness. First, we evaluate all six models against the heuristic baselines using all selected trajectories. Second, we use some of them to train the GETNext model, and we again evaluate the six LLM models against it on the same test set. Table 2 reports the results obtained for the first set of experiments by applying the introduced five prompt strategies that differ in context, evaluated using $Acc_{@1}$, $HR_{@5}$, and MRR. For each configuration, the average (AVG) and standard deviation (STD) metrics values are reported for all VeronaCard predictions. Overall, the results indicate that the prompt strategy that 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 8×7B* model attains the highest top-1 accuracy ($Acc_{@1} = 34.27$), while the *Qwen 2.5 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 8×7B* again delivers the best $Acc_{@1}$ (32.15), whereas *Qwen 2.5 14B* maintains its lead in $HR_{@5}$ 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 the heuristic 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.

Table 2. Results by model, prompt strategy, anchor point, and metric. BL = baseline, LL8 = *LLaMA 3.1 8B*, QW7 = *Qwen 2.5 7B*, QW14 = *Qwen 2.5 14B*, MX8 = *Mixtral 8×7B*, MS7 = *Mistral 7B*, and 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

As a second set of experiments, we isolate 80% of trajectories for training the GETNext model, and then we use the same 20% of the data to compare its performance with the LLMs one. The corresponding results are reported in Table 3, allowing for a consistent and comparable evaluation across the methods. The results on this subset confirm the trends observed in the full dataset, showing consistent relative performance across models and prompt strategies, thereby reinforcing the validity of the conclusions drawn from Table 2. In particular, LLM-based models already achieve strong performance under the *middle anchor point* configuration relative to GETNExt, whereas the penultimate configuration yields results that are slightly similar. We can also observe that the performance of GETNext with and without attention is very similar for *middle anchor point* configuration, while the presence of attention shows a significant role in the presence of the *penultimate anchor point* configuration, suggesting that the good LLM results are also due to the use of a self-attention mechanism. Overall, these results provide a first insight into LLMs' capabilities to achieve the same performance without a dedicated pre-training phase, overcoming the bootstrap problem of more traditional AI approaches, such as GETNext. Despite the need for historical data that are not always available, we also find that GETNext achieves $HR_{@5}$ values comparable to those of *Qwen 2.5 14B*, but at the cost of approximately 25 h of training, resulting in a significantly higher computational cost than LLM-based approaches.

Table 3. Results by model, prompt strategy, anchor point, and metric on the validation subset used for GETNext (20% of the data). BL = baseline, GN = GETNext with and without attention, LL8 = *LLaMA3.1 8B*, QW7 = *Qwen 2.5 7B*, QW14 = *Qwen 2.5 14B*, MX8 = *Mixtral 8×7B*, MS7 = *Mistral 7B*, and 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	5.14	22.08	25.68	43.69	11.72	24.78	5.73	23.23	27.96	44.88	12.85	25.78
	Nearest	2.04	14.15	34.97	47.69	11.6	19.1	3.2	17.59	30.26	45.94	11.07	21.03
	Popular	30.3	45.96	30.3	45.96	30.3	45.96	32.91	46.99	32.91	46.99	32.91	46.99
GN	No Att	1.73	2.96	35.78	10.87	18.34	4.05	1.74	2.88	35.80	10.63	18.34	4.08
	Att	1.73	2.36	35.78	9.00	18.33	3.40	23.99	7.60	66.41	8.40	43.53	5.99
LL8	A	8.46	27.83	25.8	43.76	13.85	28.91	11.49	31.89	27.37	44.58	17.3	32.92
	B	13.31	33.97	49.32	50.0	23.85	33.18	4.78	21.32	42.3	49.4	15.88	24.13
	C	14.31	35.02	49.05	49.99	24.98	34.19	14.35	35.06	45.49	49.8	24.34	34.55
	D	16.9	37.47	47.5	49.94	26.87	36.5	13.46	34.13	43.48	49.57	23.59	34.19
	E	32.25	46.75	68.43	46.48	46.2	40.95	30.29	45.95	60.03	48.98	41.6	42.06
QW7	A	0.01	0.99	1.14	10.63	0.45	4.33	0.01	0.72	0.44	6.64	0.18	2.78
	B	19.82	39.86	50.12	50.0	28.92	38.03	15.02	35.72	43.21	49.54	23.6	35.14
	C	18.4	38.75	49.73	50.0	28.89	37.4	14.16	34.86	42.94	49.5	23.96	34.88
	D	12.43	32.99	44.33	49.68	23.32	33.41	9.75	29.67	36.69	48.2	18.96	31.15
	E	31.49	46.45	71.62	45.08	46.81	39.95	28.67	45.22	65.93	47.4	42.81	40.28
QW14	A	0.86	9.26	21.06	40.78	6.29	14.44	2.77	16.41	30.71	46.13	11.1	20.88
	B	25.72	43.71	64.53	47.84	39.06	39.53	20.52	40.38	55.09	49.74	32.53	38.19
	C	24.88	43.23	61.84	48.58	37.41	39.55	19.5	39.62	52.78	49.92	30.9	37.81
	D	21.19	40.87	63.83	48.05	36.73	37.38	16.51	37.13	53.36	49.89	29.95	36.03
	E	33.36	47.15	**73.95**	43.89	**48.89**	39.95	31.89	46.61	**67.04**	47.01	**45.51**	41.16
MX8	A	1.37	11.61	34.67	47.59	14.34	21.73	4.04	19.68	35.73	47.92	16.8	25.95
	B	13.8	34.49	57.5	49.44	29.05	33.59	10.61	30.8	49.22	49.99	24.03	31.64
	C	6.75	25.09	53.42	49.88	21.54	26.93	7.24	25.92	46.52	49.88	19.79	27.83
	D	19.84	39.88	60.06	48.98	34.39	37.23	15.67	36.35	50.08	50.0	28.51	35.82
	E	**33.37**	47.15	73.33	44.22	48.85	40.05	**34.55**	47.55	63.76	48.07	45.99	42.74

Table 3. *Cont.*

		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**
MS7	A	0.15	3.87	13.85	34.55	5.12	13.55	0.1	3.2	21.68	41.21	8.14	16.39
	B	24.6	43.07	57.43	49.45	35.85	40.03	21.96	41.4	46.85	49.9	30.27	39.86
	C	16.94	37.51	52.75	49.92	27.9	35.88	17.12	37.67	42.07	49.37	24.77	36.88
	D	19.32	39.48	54.07	49.83	32.2	37.74	15.35	36.05	45.98	49.84	26.89	35.96
	E	29.71	45.7	71.24	45.27	45.39	39.45	31.72	46.54	66.98	47.03	45.37	41.12
DS33	A	0.02	1.39	0.54	7.34	0.19	2.73	2.77	16.41	30.61	46.09	11.08	20.88
	B	4.86	21.5	38.87	48.75	15.03	24.35	4.34	20.37	35.45	47.84	14.03	23.89
	C	4.75	21.27	39.61	48.91	15.26	24.3	4.09	19.82	35.96	47.99	14.0	23.57
	D	6.89	25.32	40.98	49.18	17.43	27.48	5.71	23.2	36.99	48.28	15.97	26.41
	E	29.27	45.5	62.3	48.47	41.05	41.29	29.49	45.6	55.49	49.7	38.8	42.58

Another important factor to evaluate is the model response time under different prompt strategies. Tables 4 and 5 report the minimum, mean, and maximum execution time in seconds and milliseconds per prediction. The results show that, in general, as prompt complexity and contextual information increase, the average time rises slightly but remains acceptable. Larger models, such as *Mixtral 8×7B* 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. From the other side, GETNext requires substantially lower computation time to get the PoI prediction than LLMs, as reported in Table 5. This advantage comes at the cost of a considerably longer training phase, which, as mentioned before, exceeds 25 h.

Table 4. Average, minimum, and maximum computation time in seconds for each LLM and prompt strategy. LL8 = *LLaMA3.1 8B*, QW7 = *Qwen 2.5 7B*, QW14 = *Qwen 2.5 14B*, MX8 = *Mixtral 8×7B*, MS7 = *Mistral 7B*, and 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

Table 5. Average, minimum, and maximum response time in milliseconds for the GETNext strategy after the expensive training phase.

Confirugration	**Min (ms)**	**Mean (ms)**	**Max (ms)**
No Attention	0.827	0.967	63.004
With Attention	0.827	0.968	63.786

5.4. Reasoning Analysis

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, and meters), popularity reasoning (e.g., popular, famous, highlight, guidebook, and important), time reasoning (e.g., hour, before, when, and late), and others, which include the categories 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. Figure 3 illustrates the percentage distribution of reasoning types for the clustering prompt strategy (E), which achieves the best performance according to Table 2. Geospatial reasoning dominates most models, except *DeepSeek Coder 33B*, which favors time reasoning. The role of attraction popularity is fluctuating, with some models like *LLaMA 3.1 8B*, which essentially does not consider it. This can make such models more suitable for dealing with less popular cities or attractions, or with limited historical data.

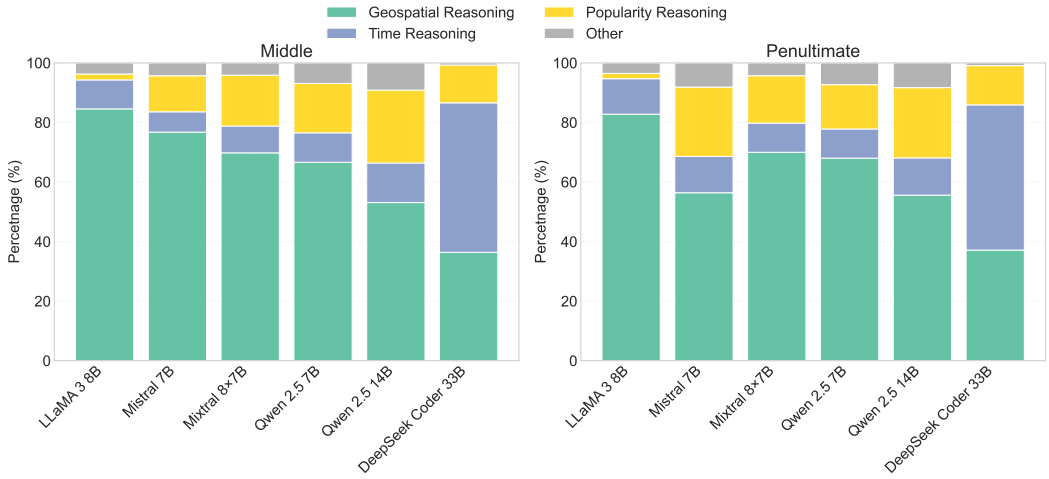

Figure 3. Percentage distribution of reasoning categories for the richer prompt strategy (E) across models and anchor selection mechanism.

6. Discussion

The previous section illustrates the main contributions of the paper. The incremental prompt engineering approach, tested across six different LLM architectures, enables the clear identification of LLM capabilities for providing meaningful tourist suggestions. Related to this first point, we also demonstrate, through comparison with the GETNext approach, that they can address the genuine cold start limitations of traditional AI-based recommender systems, yielding practical advantages even when historical data are not available. In particular, while GETNext achieves competitive performance with the penultimate anchor point, comparable to the best-performing LLMs, it requires a dedicated training phase and relies on historical interaction data, whereas LLM-based approaches can generate meaningful recommendations without task-specific training. The reasoning analysis, examining how LLMs justify their predictions, adds an interpretability dimension that most next-PoI prediction papers lack.

The reasoning analysis reveals the great capability of all LLMs in understanding and using geospatial concepts, like the notion of proximity and distance, which is used as a preferred criterion in almost all cases. The *DeepSeek Coder 33B* model exhibits different behavior in this regard, as its role for geospatial information is less prominent, while temporal information plays a greater role. Conversely, for all the other models, temporal

information, such as the remaining available time, constitutes the second or third guiding principle, with a proportion consistently smaller than the geospatial one. The popularity reasoning also plays a controversial role in the models, since for some, such as *LLaMA 3.1 8B*, it is quite negligible, while for others it outweighs the role of temporal aspects. The marginal role of preferences in some models makes them more suitable for lesser-known cities and attractions, further reducing the amount of historical information needed.

Even if the validation is restricted to one city, due to the availability of real data in this sense, we can observe the following. (i) The information used by the proposed methodology and investigation are only the ones provided by a tourist city pass, an instrument increasingly offered by many tourist cities around the world; therefore, the practical applicability of the approach is straightforward, since we do not rely on particular or personal data about users. This is, in some sense, in contrast with many other approaches in the literature, which instead rely on a large amount of tourist information, such as posts on social networks or personal information like age or tastes. (ii) The city pass used in the experiments includes both very popular places, like the "Arena Amphitheater", which could already be known by a pre-trained LLM, but also less popular attractions, like "Museo della Fotografia", for which it is quite certain that a pre-trained LLM has very little knowledge. Since the approach provides suggestions for both in a reasonable, spatio-temporal way, we can assume that the method can be applied to other tourist cities with the same results. Thanks to the increasingly wide adoption of a city pass in many tourist cities, we can assume that the method can be applied to them in a straightforward manner and with the same results.

7. Conclusions

This work explored the potential of LLMs to understand and predict tourist movements in a next-PoI prediction task using an incremental prompt strategy, drawing on a real-world tourist dataset from the municipality of Verona, Italy. Experiments across six LLMs demonstrated that progressively enriching the prompt with spatial, temporal, and preference information can significantly improve prediction accuracy compared to traditional baselines. *Qwen 2.5 14B* and *Mixtral 8×7B* 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 at the expense of geospatial reasoning.

Overall, our findings indicate that LLMs can serve as flexible, data-efficient mobility interpreters that can integrate heterogeneous contextual dimensions, which can be applied also in practical contexts where the available historical information is very limited or absent. This addresses the classical bootstrap problem in many real-world scenarios and extends the applicability of T-RSs in new settings, thereby reducing initial costs and implementation time. The observation that using a richer prompt yields better suggestions is almost intuitive. However, the use of an incremental prompt strategy applied here, together with the final analysis of the motivations underlying the suggestions, allows for a better understanding of the reasoning capabilities and differences among the considered LLM architectures, also with respect to geospatial and temporal concepts, which present specific characteristics compared to simple textual information. Future research will explore integrating additional context sources, such as weather conditions and real-time PoI crowding, to enhance personalization, interpretability, and scalability in real-world tourist recommender systems.

Author Contributions: Conceptualization, S.M. (Sara Migliorini); methodology, S.M. (Sara Migliorini); software, S.M. (Simone Mattioli); validation, A.D.V.; data curation, A.D.V.; writing—original draft preparation, A.D.V. and S.M. (Simone Mattioli); writing—review and editing, S.M. (Sara Migliorini) and E.Q.; and supervision, S.M. (Sara Migliorini) and E.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the experimental code, preprocessing pipelines, and analysis notebooks are available at https://github.com/4nnina/llm_tourist_trajectories (accessed on 10 March 2026).

Acknowledgments: We acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CINECA (Italy). During the preparation of this work, the authors used Grammarly's generative AI (v14.1282.0) writing feature in order to check the grammar, improve the writing, and change passive verbs to active verbs. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Islam, M.A.; Mohammad, M.M.; Sarathi Das, S.S.; Ali, M.E. A survey on deep learning based Point-of-Interest (POI) recommendations. *Neurocomputing* **2022**, *472*, 306–325. [CrossRef]
2. 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. *J. Comput. Sci. Technol.* **2020**, *35*, 603–616. [CrossRef]
3. 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, UMAP '23, Limassol, Cyprus, 26–29 June 2023 ; pp. 164–174. [CrossRef]
4. Luo, Y.; Liu, Q.; Liu, Z. STAN: Spatio-Temporal Attention Network for Next Location Recommendation. In Proceedings of the Web Conference 2021, WWW '21, Ljubljana, Slovenia, 19–23 April 2021; pp. 2177–2185. [CrossRef]
5. 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, SIGIR '22, Madrid, Spain, 11–15 July 2022; pp. 1144–1153. [CrossRef]
6. Massimo, D.; Ricci, F. Building effective recommender systems for tourists. *AI Mag.* **2022**, *43*, 209–224. [CrossRef]
7. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
8. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* **2019**, *52*, 1–38. [CrossRef]
9. Adomavicius, G.; Bauman, K.; Tuzhilin, A.; Unger, M. Context-Aware Recommender Systems: From Foundations to Recent DevelopmentsContext-aware recommender systems. In *Recommender Systems Handbook*; Ricci, F., Rokach, L., Shapira, B., Eds.; Springer: New York, NY, USA, 2022; pp. 211–250. [CrossRef]
10. 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.* **2022**, *205*, 117700. [CrossRef]
11. Zheng, W.; Huang, L.; Lin, Z. Multi-attraction, hourly tourism demand forecasting. *Ann. Tour. Res.* **2021**, *90*, 103271. [CrossRef]
12. 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, SIGIR'13, Dublin, Ireland, 28 July–1 August 2013; pp. 363–372. [CrossRef]
13. Migliorini, S.; Dalla Vecchia, A.; Belussi, A.; Quintarelli, E. ARTEMIS: A context-aware recommendation system with crowding forecaster for the touristic domain. *Inf. Syst. Front.* **2024**, 1–27. [CrossRef]
14. Dalla Vecchia, A.; Migliorini, S.; Quintarelli, E.; Gambini, M.; Belussi, A. Promoting sustainable tourism by recommending sequences of attractions with deep reinforcement learning. *Inf. Technol. Tour.* **2024**, *26*, 449–484. [CrossRef]

15. Zhou, Y. A Dynamically Adding Information Recommendation System based on Deep Neural Networks. In Proceedings of the 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS), Dalian, China, 20–22 March 2020; pp. 1–4. [CrossRef]
16. 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.* **2020**, *1*, 1–18. [CrossRef]
17. Xing, S.; Liu, F.; Wang, Q.; Zhao, X.; Li, T. Content-aware point-of-interest recommendation based on convolutional neural network. *Appl. Intell.* **2019**, *49*, 858–871. [CrossRef]
18. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Red Hook, NY, USA, 6–12 December 2020.
19. OpenAI. GPT-4 Technical Report. *arXiv* **2024**, arXiv:2303.0877. [CrossRef]
20. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv* **2023**, arXiv:2302.13971. [CrossRef]
21. 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* **2023**, arXiv:2312.11805. [CrossRef]
22. 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, Red Hook, NY, USA, 28 November–9 December 2022.
23. 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, Red Hook, NY, USA, 28 November–9 December 2022.
24. Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.V.; Chi, E.H.; Narang, S.; Chowdhery, A.; Zhou, D. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In Proceedings of the The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, 1–5 May 2023.
25. 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, Red Hook, NY, USA, 10–16 December 2023.
26. Schick, T.; Dwivedi-Yu, J.; Dessí, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language models can teach themselves to use tools. In Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23, Red Hook, NY, USA, 10–16 December 2023.
27. Min, S.; Lyu, X.; Holtzman, A.; Artetxe, M.; Lewis, M.; Hajishirzi, H.; Zettlemoyer, L. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022*; Goldberg, Y., Kozareva, Z., Zhang, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 11048–11064. [CrossRef]
28. Jiang, Y.; Pan, Z.; Zhang, X.; Garg, S.; Schneider, A.; Nevmyvaka, Y.; Song, D. Empowering time series analysis with large language models: A survey. In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI '24, Jeju, Republic of Korea, 3–9 August 2024. [CrossRef]
29. 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, KDD '24, Barcelona, Spain, 25–29 August 2024; pp. 5351–5362. [CrossRef]
30. 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 Trans. Intell. Veh.* **2025**, *10*, 794–807. [CrossRef]
31. Wang, X.; Fang, M.; Zeng, Z.; Cheng, T. Where Would I Go Next? Large Language Models as Human Mobility Predictors. *arXiv* **2024**, arXiv:2308.15197. [CrossRef]
32. Wu, L.; Zheng, Z.; Qiu, Z.; Wang, H.; Gu, H.; Shen, T.; Qin, C.; Zhu, C.; Zhu, H.; Liu, Q.; et al. A Survey on Large Language Models for Recommendation. *World Wide Web* **2024**, *27*, 60. [CrossRef]
33. Munson, J.; Cuezze, T.; Nesar, S.; Zosso, D. A Review of Large Language Models and the Recommendation Task. *Discov. Artif. Intell.* **2025**, *5*, 203. [CrossRef]
34. Karlović, R.; Rovis, M.; Smajić, A.; Sever, L.; Lorencin, I. Context-Aware Tourism Recommendations Using Retrieval-Augmented Large Language Models and Semantic Re-Ranking. *Electronics* **2025**, *14*, 4448. [CrossRef]
35. Borràs, J.; Moreno, A.; Valls, A. Intelligent Tourism Recommender Systems: A Survey. *Expert Syst. Appl.* **2014**, *41*, 7370–7389. [CrossRef]
36. Ludewig, M.; Mauro, N.; Latifi, S.; Jannach, D. Empirical analysis of session-based recommendation algorithms. *User Model. User-Adapt. Interact.* **2021**, *31*, 149–181. [CrossRef]

37. Turisini, M.; Amati, G.; Cestari, M. LEONARDO: A Pan-European Pre-Exascale Supercomputer for HPC and AI applications. *J. Large-Scale Res. Facil. JLSRF* **2024**, *9*, A186. [CrossRef]
38. Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. Qwen Technical Report. *arXiv* **2023**, arXiv:2309.16609. [CrossRef]
39. 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.; et al. Mixtral 8×7B: A Sparse Mixture-of-Experts Language Model. *arXiv* **2024**, arXiv:2401.04088.
40. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B: A 7-Billion-Parameter Foundation Language Model with Grouped-Query and Sliding-Window Attention. *arXiv* **2023**, arXiv:2310.06825.
41. Guo, D.; Zhu, Q.; Yang, D.; Xie, Z.; Dong, K.; Zhang, W.; Chen, G.; Bi, X.; Wu, Y.; Li, Y.K.; et al. DeepSeek-Coder: When the Large Language Model Meets Programming—The Rise of Code Intelligence. *arXiv* **2024**, arXiv:2401.14196.