Beyond Shortest Path: Agentic Vehicular Routing with Semantic Context

Carnot Braun* **Rafael O. Jarczewski*** **Gabriel U. Talasso***
Leandro A. Villas **Allan M. de Souza**
Institute of Computing (IC)
Hub de Inteligência Artificial e Arquiteturas Cognitivas (H.IAAC)
Universidade Estadual de Campinas (UNICAMP)
{c255785,r200219,g235078}@dac.unicamp.br
{lvillas,allanms}@unicamp.br

Abstract

Traditional vehicle routing systems efficiently optimize singular metrics like time or distance, and when considering multiple metrics, they need more processes to optimize . However, they lack the capability to interpret and integrate the complex, semantic, and dynamic contexts of human drivers, such as multi-step tasks, situational constraints, or urgent needs. This paper introduces and evaluates PAVe (Personalized Agentic Vehicular Routing), a hybrid agentic assistant designed to augment classical pathfinding algorithms with contextual reasoning. Our approach employs a Large Language Model (LLM) agent that operates on a candidate set of routes generated by a multi-objective (time, CO_2) Dijkstra algorithm. The agent evaluates these options against user-provided tasks, preferences, and avoidance rules by leveraging a pre-processed geospatial cache of urban Points of Interest (POIs). In a benchmark of realistic urban scenarios, PAVe successfully used complex user intent into appropriate route modifications, achieving over 88% accuracy in its initial route selections with a local model. We conclude that combining classical routing algorithms with an LLM-based semantic reasoning layer is a robust and effective approach for creating personalized, adaptive, and scalable solutions for urban mobility optimization.

1 Introduction

With the advance of Artificial Intelligent in urban environments, the best route recommendation problem moves from the chosen of fastest route between origin and destination to personalized paths based on different user's contexts and requirements that dynamically vary with safety concerns, environmental conditions, or personal preferences changing over time [1–3]. Furthermore, route personalization enables intelligent traffic management and the optimization of urban resources, such as reducing congestion and pollutant emissions. However, much of this contextual and user preference information is descriptive, complex, and dynamic, making it difficult to model classical algorithmic solutions that does not deal with semantic contextual information. In this regard, Large Language Model (LLM) emerges as an alternative for translating the semantic user's needs involved in describing preferences, taking into account the context and offering the best recommendations.

Recent works demonstrate the potential of LLM and agent-based frameworks for diverse applications in the urban domain [4, 5]. Applications vary from the construction of urban knowledge graphs and the simulation of multi-agent participatory planning to the analysis of mobility data to understand

*Equal Contribution

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: UrbanAI: Harnessing Artificial Intelligence for Smart Cities.

visit intent and the generation of personal mobility trajectories. These approaches open new avenues for the analysis and planning of smart cities [6–8]. Otherwise, in these works, it was observed the difficulty in making agents adapt to the different characteristics of the vehicular scenario, especially when trying to adjust to various types of Points of Interest (POI), and since vehicle mobility data (routes, destinations, POIs related to vehicular services) are intrinsically linked to specific regions and their characteristics, making it challenging to apply a model trained in one domain to another without significant adaptations.

On the other hand, other solutions [9–11] use the reasoning capabilities of LLM to predict an individual's next destination on public transport, solving vehicle routing problems from natural language descriptions, and predict general patterns of human mobility. Such methods transform spatiotemporal data and problems into formats that LLM handle, allowing flexible solutions. However, these predictions are based on historical visits and may not accurately reflect the user's current desires or preferences in these contexts. This highlights the need for methods that consider both the user's context and the vehicular scenario.

As previous route recommendation systems do not take into account users' individual semantic intentions when suggesting routes [7], in this article, we explore the potential of LLM agents for route personalization based on context and individual user needs. For this, we propose PAVe (Personalized Agentic Vehicular Routing), an agent that plans and modifies routes prioritizing locations of interest and needs, taking into account characteristics such as urgency, preferences, time of day, and traffic. We opted for a hybrid approach to leverage the proven optimality and computational efficiency of classical algorithms for pathfinding, while dedicating the LLM's resources to its core strength: semantic reasoning and contextual understanding. The main contributions of this paper can be summarized as follows:

- We develop PAVe, an agentic assistant for vehicular route planning and route selection, that considers contextual information, such as user route preferences and requirements, POI and destination urgency and importance.
- We evaluate PAVe into four realistic scenarios with real-world urban datasets with contextual street information and several user personalized requirements to fully assess the framework's capability and flexibility.
- We made all prompts from different evaluation scenarios and user requests, as well as the code implementation of PAVe publicly available for future research and developments [2].

2 Personalized Agentic Vehicular Routing

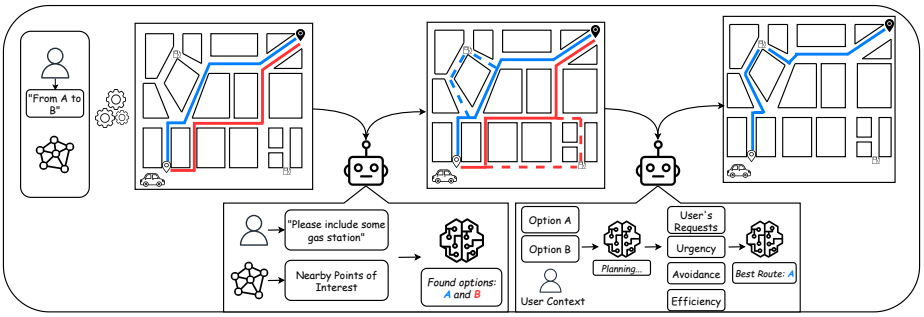

Figure 1: The overall of our PAVe framework. **a)** Where the User chooses the desired trajectory **b)** With the information of the map, the path algorithm chooses the best two options. **c)** The LLM Agentic Assistant considers the context of the User and the Environment to include the POIs of the desire in the two options. **d)** The LLM translates all this for the path algorithm to recalculate the routes. **e)** The LLM chooses the best option for the User, considering the context, and returns the output.

[2] `https://github.com/carnotbraun/BsP_PAVe`.

PAVe is a ReAct Agentic Assistant that integrates classical location and graph tools Figure 1. The methodology is designed to translate complex, contextual, multi-objective user requests into optimized, context-aware routing and decisions, focusing on efficiency, sustainability, personalization, and task completion.

First, user input is injected into the LLM, which identifies and classifies the priority of each location within the user's request. We use Chain-of-Thought (CoT) for this, [12]. This way, it can differentiate between an urgent and casual need in a request. Furthermore, our system is composed of three tools designed for robust decision-making based on users' preferences, described below.

Rounting Engine Tool (RET). Responsible for discovering the tuple of possible paths $\rho = (s, d)$, that is, user's origin s and destination d. It operates on a graph-based representation of the urban road network $G = (V, E)$ where E and V represent, respectively, the sets of corners and streets on a city map. Generating a candidate set of paths $\mathcal{P} = \{\rho_0, \rho_1, \ldots, \rho_k\}$ where k is a set the routes from s to d. The agent interacts with the map through this tool, finding the shortest paths to points of interest as well as extracting your characteristics.

Geospatial Context Tool (GCT). This tool serves to ground the graph G in the semantic context of the real world. The agent uses it to extract geospatial information, mainly POI, such as supermarkets, hospitals, and parks along the routes encountered. For our experiments, we used information obtained from crowdsourced geographic data (**OpenStreetMap**). Its goal is to create a persistent link between these locations and their corresponding nodes in G, allowing the assistant agent to understand the environmental context of any route.

Contextual Route Assessment Tool (CRAT). In this tool, the agent uses tools to identify the time of day, seasonal events, or periods of the year, such as school seasons, and, by cross-referencing this information, modify routes.

Thus, LLM agent is responsible for making decisions and translating the semantic context of the user's goals, and building a route recommendation that meets the user's expectations.

2.1 Agentic Operational Pipeline and Hierarchical Decision Making

The pipeline consists of five steps: **i) Task Classification**: The system receives the user's request in natural language. The LLM classifies the task, assigning a priority (`URGENT` or `NORMAL`) and extracting relevant OpenStreatMap (OSM) features (e.g., `'amenity': 'fuel'`). **ii) Candidate Route Generation**: The RET is called and calculates a set of routes P between the s and d. Each candidate route is then annotated with metrics like total travel time and CO_2 emissions. **iii) Contextual Enrichment**: For each route, the system adds valuable layers of information using OSM. Then, if the task is classified as `URGENT`, an additional metric is calculated: the time needed to reach the nearest urgent POI. **iv) Agent Evaluation**: The LLM receives a complete "dossier." This dossier contains detailed information about the candidate routes, including total time, CO_2 emissions, nearby POI, and, if applicable, the time to reach the urgent POI, along with the user's context and preferences using the CRAT. **v) Feedback Loop and Recalculation**: Based on the evaluation, the LLM may request a new action, such as `ADD_WAYPOINT`. When this occurs, the system initiates a feedback loop: a new route calculation is performed (Origin $\rightarrow$ Waypoint $\rightarrow$ Destination), dynamically adapting the final path.

3 Experiments

To evaluate the effectiveness of PAVe , in this section, we propose four evaluation scenarios based on real user requirements and public urban maps of Luxembourg [13]: Simple Task, Urgency, Avoidance, and Preferences. All scenarios were chosen after a search for routes that presented the characteristics we wanted to evaluate, varying route length and complexity. The evaluation showed that PAVe not only succeeds in choosing the best routes based on context and user preferences, but also prioritizes points of interest based on semantic information, such as urgency. In this paper we consider "best route" like the shortest path from s to the POI and the shortest path from POI to d. In addition to a comprehensive qualitative analysis of the results, we performed quantitative metrics, such as the accuracy of best route selection and task completion rate.

3.1 Scenarios

To better evaluate PAVe 's capabilities, we present three scenarios: **(a) The simple scenario**, which replicates more common and routine situations, such as going to the supermarket or passing through a park; **(b) Urgency**, which attempts to reflect scenarios in which passing through a given location is not optional, but rather necessary, such as stopping at a gas station with your car almost empty, since, even with two valid options, the best option would be to get to the point of interest faster; **(c) Avoidance**, which demonstrates scenarios in which the user's context and preference is to avoid a specific area; **(d) Efficiency**, which attempts to evaluate the choice of the route based on the distance, time and eco routing.

The experiments were conducted within a simulated urban environment, that is based on the **Luxembourg SUMO Traffic (LuST)** scenario, which provides a realistic road network forming the multi-weighted directed graph G, where the edge E weights represent both estimated travel time and a CO_2 emission heuristic, which aim to suggest routes that are based on both the shortest distance and the most eco-friendly. To incorporate real-world context, POI were sourced from **OSM** using the `osmnx` library. This geospatial data was subsequently pre-processed into a local `GeoPackage` cache via `geopandas`, with each POI pre-linked to its nearest SUMO network node to enable efficient runtime queries. The decision-making agent is powered by a locally-hosted Qwen-series LLM from the **Hugging Face Hub** and also the API model(GPT-o3), executed via the `transformers` library.

To understand the metrics we use, we first need to understand the difference between preferences and requirements. In this work, we use preferences as something that is not a priority, but rather desired (e.g., "...I want to pass through a park on the way to the grocery store..."), and requirements as something we consider necessary to do, e.g., "...I need to go to the gas station before going to the supermarket...". It is important to note that POI is not necessarily a requirement, depending on POI (`UGERNT` or `NORMAL`). That way, **(i) Accuracy**: represents how many times the agent chose the best route (top-k) that best matches the user's preferences and requirements, and **(ii) Completeness**: represents how many times the agent successfully edited the route correctly to add POI based on user requests, i.e. even if POI is not `URGENT` the agent chose the path that passes through it.

3.2 Main Results

Table 1: Performance analysis of PAVe across all simulated scenarios with different models using accuracy and completeness metrics (**Bold** are the best between them).

Scenarios	**Local** - *Qwen 3 - 4B*		**API** - *gpt-o3*	
	Acc. (%)	Complet. (%)	Acc.(%)	Complet.(%)
Simple	**83.33**	66.66	77.76	66.66
Urgency	91.66	58.33	**95.83**	66.66
Avoidance	**75.0**	91.66	25.00	50.00
Efficiency	**93.30**	95.66	88.88	88.88
Overall	**88.24**	76.47	76.27	73.33

The primary objective of this work was to explore the potential of a LLM agent not as a replacement for, but as an intelligent assistant to, state-of-the-art routing algorithms. Our proposed PAVe agent is designed to bridge the gap between human semantic intent and algorithmic execution by translating complex, context-dependent user tasks into structured data. To evaluate this methodology, agent performance was compared in four scenarios with four different paths each executed three times, comparing a smaller, locally-hosted model (*Qwen 3 - 4B*) and also a larger, general-purpose API-based model (*GPT-o3*). The results are summarized in Table 1. Furthermore, we conducted some more experiments varying the number of routes considered (top-k), which can be seen in the appendix.

Overall, the local **Qwen model demonstrated superior performance**, achieving higher Accuracy (88.24%) and Completeness (76.47%) than its larger API-based counterpart. The most significant divergence occurred in the **Avoidance** scenarios, where the local model's ability to adhere to strict negative constraints was substantially better (75.0% Acc. vs. 25.0% Acc.). The results, particularly the consistently lower **Completeness** metric, also highlight two primary limitations of the current

framework: an inconsistency in the agent's generation of the correct action schema, and a functional limitation wherein the feedback loop can only process a single waypoint addition.

These limitations provide a clear roadmap for future research. Future work will focus on enhancing action reliability through advanced prompting techniques like CoT and fine-tuning the local model on a curated dataset of '(context, action)' pairs. Furthermore, a significant functional enhancement will be the integration of a multi-stop route planner. This would evolve the LLM's role from identifying a single waypoint to generating a *list* of required waypoints, with the orchestrator then solving the resulting Traveling Salesperson Problem (TSP) variant to determine the optimal stop order. To make the agent truly personalized, we also plan to develop a user profile module that learns preferences and avoidance patterns over time, potentially using **Distributed Learning** to preserve user privacy. Finally, the system's sustainability metrics can be improved by integrating high-fidelity context from real-time traffic and vehicle-specific emission APIs, further grounding the agent's decisions in measurable, real-world impact.

Acknowledgments

This project was supported by the brazilian Ministry of Science, Technology and Innovations, with resources from Law nº 8,248, of October 23, 1991, within the scope of PPI-SOFTEX, coordinated by Softex and published Arquitetura Cognitiva (Phase 3), DOU 01245.003479/2024-10.

References

[1] Ren Wang, Mengchu Zhou, Kaizhou Gao, Ahmed Alabdulwahab, and Muhyaddin J. Rawa. Personalized route planning system based on driver preference. *Sensors*, 22(1), 2022. ISSN 1424-8220. doi: 10.3390/s22010011. URL `https://www.mdpi.com/1424-8220/22/1/11`.

[2] Shiming Zhang, Zhipeng Luo, Li Yang, Fei Teng, and Tianrui Li. A survey of route recommendations: Methods, applications, and opportunities, 2024. URL `https://arxiv.org/abs/2403.00284`.

[3] Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. A survey on deep learning for human mobility, 2021. URL `https://arxiv.org/abs/2012.02825`.

[4] Zengqing Wu, Run Peng, Xu Han, Shuyuan Zheng, Yixin Zhang, and Chuan Xiao. Smart agent-based modeling: On the use of large language models in computer simulations, 2023. URL `https://arxiv.org/abs/2311.06330`.

[5] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. The rise and potential of large language model based agents: A survey, 2023. URL `https://arxiv.org/abs/2309.07864`.

[6] Letian Gong, Yan Lin, Xinyue Zhang, Yiwen Lu, Xuedi Han, Yichen Liu, Shengnan Guo, Youfang Lin, and Huaiyu Wan. Mobility-llm: Learning visiting intentions and travel preference from human mobility data with large language models. *Advances in Neural Information Processing Systems*, 37:36185–36217, 2024.

[7] Jiawei Wang, Renhe Jiang, Chuang Yang, Zengqing Wu, Makoto Onizuka, Ryosuke Shibasaki, Noboru Koshizuka, and Chuan Xiao. Large language models as urban residents: An llm agent framework for personal mobility generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 124547–124574. Curran Associates, Inc., 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/e142fd2b70f10db2543c64bca1417de8-Paper-Conference.pdf`.

[8] Yansong Ning and Hao Liu. Urbankgent: A unified large language model agent framework for urban knowledge graph construction. *Advances in Neural Information Processing Systems*, 37: 123127–123154, 2024.

[9] Zhenlin Qin, Pengfei Zhang, Leizhen Wang, and Zhenliang Ma. Lingotrip: Spatiotemporal context prompt driven large language model for individual trip prediction. *Journal of Public Transportation*, 27:100117, 2025.

[10] Zhehui Huang, Guangyao Shi, and Gaurav S. Sukhatme. From words to routes: Applying large language models to vehicle routing. *CoRR*, abs/2403.10795, 2024. doi: 10.48550/ARXIV.2403.10795. URL `https://doi.org/10.48550/arXiv.2403.10795`.

[11] Xinglei Wang, Meng Fang, Zichao Zeng, and Tao Cheng. Where would I go next? large language models as human mobility predictors. *CoRR*, abs/2308.15197, 2023. doi: 10.48550/ARXIV.2308.15197. URL `https://doi.org/10.48550/arXiv.2308.15197`.

[12] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[13] Lara Codecá, Raphaël Frank, Sébastien Faye, and Thomas Engel. Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation. *IEEE Intelligent Transportation Systems Magazine*, 9(2):52–63, 2017.

[14] Zhilun Zhou, Yuming Lin, Depeng Jin, and Yong Li. Large language model for participatory urban planning, 2024. URL `https://arxiv.org/abs/2402.17161`.

Appendix

A Related Works

In this subsection, we gonna enter into more details about the works briefly discussed in the first section of this paper.

A.1 LLMs for Urban Knowledge and Planning

Recent works have explored the application of LLMs (LLMs) to various urban computing tasks. Some works present [8] a unified LLM agent framework designed to automate Urban Knowledge Graph Construction (UrbanKGC), addressing the significant manual effort traditionally required. Their implementation involves fine-tuning Llama 2 and Llama 3 models on knowledgeable instruction sets derived from GPT-4. A key component is a tool-augmented iterative refinement module that invokes external geospatial tools to enhance the model's geospatial reasoning. The framework was evaluated on datasets from New York City and Chicago, focusing on Relational Triplet Extraction (RTE) and Knowledge Graph Completion (KGC) tasks. Performance was assessed using accuracy, measured through both human evaluation and GPT-4 self-evaluation. In a related domain [14], a multi-agent framework using LLMs to simulate participatory urban planning, aiming to generate land-use plans that reflect diverse resident needs efficiently. Their implementation uses 'gpt-4-vision-preview' for a planner agent that interprets maps and 'gpt-3.5-turbo-1106' for resident agents. To manage collaboration among a large number of agents, they employ a fishbowl discussion mechanism. The system was experimentally deployed in two Beijing urban regions, Huilongguan (HLG) and Dahongmen (DHM). Its effectiveness was measured against human experts and a Deep Reinforcement Learning baseline, using need-agnostic metrics like Service and Ecology, alongside novel need-aware metrics, Satisfaction and Inclusion.

A.2 LLMs for Mobility and Trajectory Analysis

The domain of human mobility has also seen an influx of LLM-based approaches. Recently, a team developed some very promising work that proposes [7] an agent framework for generating personal mobility trajectories by leveraging the semantic understanding of LLMs. The framework operates in two phases: first, it identifies habitual activity patterns using a self-consistency evaluation, and second, it performs motivation-driven activity generation through retrieval-augmented strategies. The implementation relies on GPT-o3 APIs and was validated on a real-world personal activity trajectory dataset from Tokyo. The evaluation compared the generated data distributions to real-world data using Jensen-Shannon Divergence (JSD) across four metrics: Step Distance (SD), Step Interval (SI), Daily Activity Routine Distribution (DARD), and Spatio-temporal Visits Distribution (STVD). Similarly, a unified framework for analyzing check-in sequences across multiple downstream tasks [6]. It aims to achieve a deeper semantic understanding of user behavior by capturing both visiting intentions and travel preferences. The implementation features a Visiting Intention Memory Network (VIMN) and a Human Travel Preference Prompt (HTPP) pool, with 'TinyLlama-1B' as the backbone model, fine-tuned using LoRA. The framework was tested on four benchmark datasets, including Gowalla and Foursquare, for Next Location Prediction (LP), Trajectory User Link (TUL), and Time Prediction (TP) tasks. Evaluation metrics included Accuracy@k, Mean Reciprocal Rank (MRR), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

A.3 LLMs for Routing and Trip Prediction

Beyond analysis and generation, LLMs are being applied to complex routing and prediction problems. A model was developed that uses 'ChatGPT-3.5-Turbo' in a zero-shot, in-context learning setting for predicting an individual's next trip [9]. Their implementation addresses the limitations of data-driven models by formatting historical mobility data into long-term, mid-term, and short-term contexts, which are fed to the LLM via a structured prompt designed to guide its reasoning. The model was evaluated on an Automated Fare Collection (AFC) dataset from the Hong Kong Mass Transit Railway (MTR). Its performance was benchmarked against traditional models using Accuracy and Weighted F1 score, particularly showing strength in few-shot learning scenarios. In the broader context of routing, the ability of LLMs to solve Vehicle Routing Problems (VRPs) directly from natural language descriptions [10]. They constructed a benchmark dataset of 21 VRP variants and

evaluated the performance of GPT-4 and Gemini 1.0 Pro. Their proposed framework improves performance through a self-reflection mechanism that includes self-debugging of generated code and self-verification using LLM-generated unit tests. The experiments were evaluated on three primary metrics: feasibility (ratio of valid solutions), optimality (ratio of optimal solutions), and efficiency (normalized solution quality).

B PAVe Overview

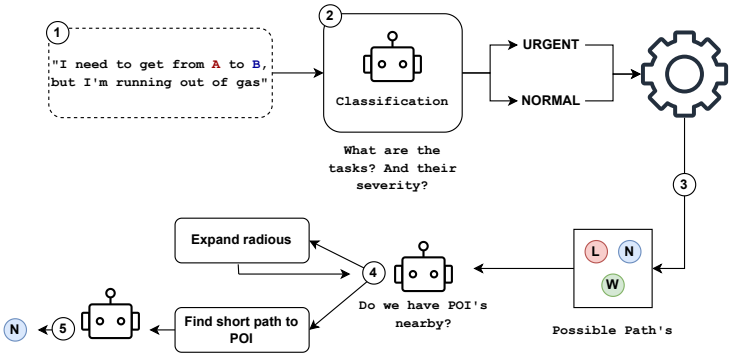

Figure 2: PAVe System Overview

The agent's decision-making process follows a structured, multi-stage pipeline, as illustrated in the main paper. This section provides a detailed technical description of each stage, as we can see in the image of the overview Figure 2.

1. **User Request Input and Parsing:** The pipeline is initiated with a user's natural language request, which includes an origin (A), a destination (B), and a set of tasks (e.g., *"I'm running out of gas"*). This unstructured input is parsed by the central orchestrator to initialize the routing query.

2. **LLM-based Task Classification:** The raw tasks are passed to the LLM agent for classification. Using a specifically engineered prompt, the agent analyzes the semantic content to determine the severity (`URGENT` or `NORMAL`) and extracts the corresponding OpenStreetMap (OSM) tags for each sub-task (e.g., `{'amenity': 'fuel'}`). This step translates unstructured human intent into a machine-readable set of objectives and constraints.

3. **Multi-Objective Candidate Path Generation:** The routing engine computes a set of k diverse, candidate paths between the origin and the final destination. Each path is quantitatively annotated with a vector of objective metrics, including estimated travel time and a CO_2 emission heuristic, forming the basis for the subsequent evaluation.

4. **Geospatial Contextual Enrichment:** Each candidate path undergoes an enrichment process. If an `URGENT` task was identified, an additional metric representing the time to reach the nearest relevant POI is calculated for each path. Concurrently, the `POIFinder` module performs a spatial query against a pre-processed geospatial cache to identify and attach a list of all normal-priority POIs that lie within a predefined radius of each route's geometry.

5. **Hierarchical Evaluation and Action:** The enriched candidate routes, along with the full user and scenario context, are compiled into a final "dossier" for the LLM agent. The agent executes a hierarchical reasoning process to evaluate the options and select the optimal route that best satisfies the complete set of user needs. If this decision requires a route modification (e.g., adding a non-urgent waypoint), the agent's structured output triggers a feedback loop, where the orchestrator recalculates the final trajectory to include the new point of interest.

C Other Results

To evaluate the sensitivity of the PAVe agent to the number of initial choices, we conducted an experiment by varying the number of candidate paths (k) generated by the Routing Engine. We compared the performance of three distinct LLM for k values of 1, 3, 5, 10, and 20. The results, summarized in Table 2, reveal significant differences in their decision-making capabilities and robustness as task complexity increases, considering that all the results consist of the overall scenario(The mean between all other Section 3.1).

The results highlight a critical trade-off between peak performance and stability. The `Qwen3-4B` model achieved the highest accuracy (88.24%) for low k values (1 and 3), but its performance degraded sharply as more options were introduced, with accuracy dropping to 54.90% for k=20. In contrast, the OpenAI API model, while not reaching the same initial peak, demonstrated remarkable stability. Its accuracy remained constant at 76.47% for k up to 10, with only a marginal decrease at k=20. A similar pattern of high consistency was observed for its completeness metric. The `Phi-3.5` model consistently underperformed across all configurations.

This performance degradation, most prominent in the local models, is likely due to two factors. First, a larger k increases the cognitive load on the LLM. Second, as k increases, the marginal differences between candidate paths diminish, making the choice more ambiguous. This suggests that the OpenAI API model is more robust in handling such ambiguity.

Table 2: Performance analysis of PAVe on different top-k paths, and different models, using accuracy and completeness metrics.

Routes	**Qwen3 4B**		**Phi-3.5 3.82B**		**OpenAi - API o3**	
	Acc. (%)	**Complet. (%)**	**Acc. (%)**	**Complet. (%)**	**Acc. (%)**	**Complet. (%)**
k=1	**88.24**	**76.47**	35.57	71.88	76.47	73.33
k=3	**88.24**	**76.47**	33.33	68.75	76.47	73.33
k=5	**76.47**	**73.33**	31.37	70.59	**76.47**	**73.33**
k=10	66.67	**73.33**	25.49	71.22	**76.47**	**73.33**
k=20	54.90	64.70	15.69	67.65	**74.51**	**72.73**

The main conclusion is that for an agent like PAVe , simply increasing the number of options is detrimental. Our findings suggest that future work should focus on a dynamic k, potentially adapted to the number of points of interest (POIs) or the semantic diversity of the routes. Furthermore, integrating tools to pre-filter candidate routes could empower the agent to make more informed decisions from a high-quality, curated set of options.

D Prompts

The PAVe agent's cognitive abilities are orchestrated through a series of structured prompts designed to elicit specific, reliable, and machine-parsable responses. We developed three distinct prompts for the primary stages of the agent's operational pipeline.

Task Classification Prompt. This initial prompt Figure 3 functions as the primary Natural Language Understanding (NLU) component of the system. It is responsible for deconstructing a user's free-form request into a structured set of objectives. *Input:* The prompt receives a list of user tasks as raw strings. *Output:* It returns a JSON list of objects, where each object represents a distinct sub-task and is annotated with a classified priority (`URGENT` or `NORMAL`) and a dictionary of the corresponding OpenStreetMap (OSM) tags required to locate a relevant Point of Interest (POI). This structured output serves as the foundational input for all subsequent planning and routing stages.

Classify Prompt

```
SYSTEM PROMPT: "You are a task analysis expert. Your job is to analyze a list of user
tasks and classify them by priority and required Point of Interest (POI) type.
        Priorities can be 'URGENT' or 'NORMAL'.
        A task is 'URGENT' if it implies immediate need, danger, or a critical system
state.
        Examples:
        - ["I need to get to the hospital"] -> [{{"task": "I need to get to the
hospital", "priority": "URGENT", "poi_tags": {{"amenity": "hospital"}}}}]
        - ["my car is running on fumes I need gas"] -> [{{"task": "need gas",
"priority": "URGENT", "poi_tags": {{"amenity": "fuel"}}}}]
        - ["pick up groceries", "go to the park"] -> [{{"task": "pick up groceries",
"priority": "NORMAL", "poi_tags": {{"shop": "supermarket"}}}}, {{"task": "go to the
park", "priority": "NORMAL", "poi_tags": {{"leisure": "park"}}}}]
        - ["I want to buy some meat to cook"] -> [{{"task": "I want to buy some meat to
cook", "priority": "NORMAL", "poi_tags": {{"shop": "supermarket"}}}}]
        Analyze the following user tasks and return a JSON list of objects.
        Tasks: {json.dumps(tasks)}
        Respond ONLY with the final JSON list."
```

Figure 3: Classification Prompt

Tag Aggregation Prompt. To prepare for geospatial queries, this utility prompt Figure 4 is used to consolidate the objectives from all classified tasks into a single, unified search query. *Input:* It takes a list of user tasks as strings. *Output:* The prompt yields a single JSON object that aggregates all necessary OSM tags from the various user tasks, creating a master list of all POI types that need to be searched for within the given scenario.

Build Tags

```
        SYSTEM PROMPT: "You are an expert in OpenStreetMap (OSM) data. Your task
is to analyze a list of user tasks and convert them into a JSON object of OSM
tags suitable for a search query.
        Examples:
        - ["go to the pharmacy"] -> {{"amenity": "pharmacy"}}
        - ["pick up groceries", "go to the supermarket"] -> {{"shop":
"supermarket"}}
        - ["take the kids to the park"] -> {{"leisure": "park"}}
        - ["go to the hospital"] -> {{"amenity": "hospital"}}
        - ["I need to buy some food"] -> {{"shop": "supermarket"}}
        Analyze the following user tasks and provide the corresponding JSON
object of OSM tags.
        User Tasks: {json.dumps(tasks)}
        Respond ONLY with the final JSON object."
```

Figure 4: Builds Prompt

Route Evaluation Prompt. This is the core reasoning prompt, where the agent performs its final, multi-objective decision-making Figure 5. It is explicitly instructed to follow a strict decision hierarchy to ensure predictable and logical behavior. *Input:* The prompt is provided with a comprehensive "dossier" in JSON format, containing the full user context (preferences, avoidance rules), the scenario context (time, traffic), and a list of candidate routes, each annotated with its respective metrics (e.g., travel time, CO_2 emissions, nearby POIs, and time to an urgent POI). *Output:* The agent returns a single, structured JSON object containing the ID of the chosen route, a detailed natural language justification for its decision, and a `required_action` object that instructs the orchestrator on the next step, such as initiating a feedback loop to add a waypoint.

Evaluation Prompt

```
SYSTEM PROMPT: "You are an intelligent vehicle navigation assistant. Your task is to choose the best route from
the given options.

        Decision Hierarchy:
        1.  URGENT TASK: If the metric 'time_to_urgent_poi_minutes' is present, you MUST choose the route with the
lowest value for this metric. This is the most important factor.
        2.  AVOIDANCE RULES: If no urgent task is present, check if any route violates the `avoidance_rules` based
on the current time and its POIs. Penalize those routes, because the user doesn't want to pass in there.
        3.  NORMAL TASKS: Prioritize routes that fulfill the normal `tasks_for_the_day`.
        4.  EFFICIENCY:Finally, choose the best balance of low travel time and low CO2 emissions, not necessary
the best option its the route with lowest time, but the one who consider the context.

        Your Response:
        Analyze all data and respond ONLY with a single, valid JSON object. Do not add any text before or after
it.
        The JSON object MUST strictly follow this structure. The value for "required_action" MUST be an object
with "type" and "description" keys.

        Example of a valid response for a non-urgent task:
        {{
          "chosen_route_id": 2,
          "justification": "Route 2 is chosen because it passes a supermarket, which fulfills the user's primary
task. Even though it is slightly slower, completing the task is the priority.",
          "required_action": {{
            "type": "ADD_WAYPOINT",
            "description": "supermarket"
          }}
        }}
```

Figure 5: Evaluation Prompt

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the last sentences, we prioritized making clear the application of our work, as well as our contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation, as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have had a discussion during the last two paragraphs about the limitations and the work we can do to address them.

   Guidelines:

   - The answer NA means that the paper has no limitation, while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or when images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed not to penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: There are no theoretical assumptions in this paper.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in the appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We will provide the source codes of this study. The dataset of the scenario can be found online, in the reference paper of LuST Scenario, as well as the data from the POI's in OpenStreetMaps. Researchers can reproduce the results by running our source code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same

dataset or provide access to the model. In general. Releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in the supplemental material?

   Answer: [Yes]

   Justification: We will provide the source code for this study. The dataset is already available online.

   Guidelines:

   - The answer NA means that the paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

Justification: We clarified the characteristics that were being evaluated, and how we configured each experiment in general.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined, or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our results are not accompanied by error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar rather than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We do not clarify in which environment the experiments were conducted.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers, CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs, as well as estimate the total compute.
   - The paper should disclose whether the full research project required more computing than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We oblige to the NeurIPS Code of Ethics. The authors are responsible for all the materials presented in this paper.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The social impacts are presented in the initial discussion of the introduction, when we demonstrate how the application can be viable in a smarter and more integrated urban scenario.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for the responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We only use the Qwen3 model and GPT-o3 API, but we do not release any models.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example, by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best-faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited, and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We clarify how the dataset was obtained for the experiment, as well as clarifying that they are all open source.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented, and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Neither crowdsourcing nor human subjects are involved in this research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Neither crowdsourcing nor human subjects are involved in this research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, a declaration is not required.

    Answer: [NA]

    Justification: We do not use LLM for methodology, scientific rigor, or originality of the research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.