AgentTravel: Knowledge-Augmented LLM Agent Framework for Urban Travel Planning

Jie Zhao, Jie Feng and Yong Li*

Department of Electronic Engineering, Tsinghua University, Beijing, China
Beijing National Research Center for Information Science and Technology (BNRist), China

Abstract

Large language models are opening new opportunities for intelligent decision support, with urban travel planning as a challenging and high-impact use case. Effective planning requires integrating real-time, multi-source data (e.g., such as points of interest, transportation, and user preferences), while reasoning spatially to generate feasible itineraries. This paper proposes AgentTravel, a unified framework that combines knowledge-grounded modeling, agentic reasoning, and multi-perspective evaluation. It includes: (1) TravelLLM, a domain-adapted model enriched with urban and spatial knowledge, (2) TravelAgent, an agentic planner with structured itinerary memory and real-time data retrieval, and (3) TravelBench, a benchmark assessing both knowledge grounding and plan quality. Experiments on five Chinese cities show that AgentTravel outperforms strong baselines in factual reasoning and itinerary feasibility in the majority of cases, offering a promising step toward grounded and adaptive LLMs for urban intelligence. Source code and datasets are available at https://github.com/csjiezhao/AgentTravel.

Keywords

Urban Travel Planning, Knowledge-Grounded Agents, Benchmarking and Evaluation

1. Introduction

The rapid advancement of large language models (LLMs) has opened new opportunities for building agentic intelligent systems in real-world decision-making tasks. Among these, urban travel planning has emerged as a particularly promising and impactful application domain [1, 2]. As a representative case of urban intelligence, travel planning inherently integrates multiple subtasks: retrieving up-to-date information about points of interest (POIs), reasoning over spatial relationships, selecting transportation options, and organizing itineraries that satisfy diverse user preferences and constraints. Such complexity requires LLM-driven systems not only access and integrate heterogeneous knowledge sources, but also demonstrate spatial reasoning and multi-step decision-making capabilities to operate effectively in dynamic urban environments.

Despite recent advances in benchmarking [3], agent architectures [4], and iterative plan refinement [5], several fundamental challenges remain unresolved. First, current LLMs exhibit limited spatial reasoning capabilities, they often fail to accurately account for geographic distances, travel times, or accessibility constraints when generating feasible itineraries [6, 7]. Second, integrating heterogeneous and real-time information from open APIs, transportation platforms, and local knowledge bases remains non-trivial: most existing systems either ignore dynamic contextual factors or depend on narrow, domain-specific data sources. Third, while prior work such as TravelPlanner [1] has proposed evaluation frameworks based on commonsense and hard constraints, there is still a lack of scalable, multi-perspective benchmarks that jointly assess knowledge grounding, contextual reasoning, and the practical quality of generated travel plans.

To address these challenges, we propose **AgentTravel**, a unified framework designed to advance urban travel planning through knowledge-augmented LLM agent. The framework integrates three complementary components designed for reasoning, planning, and evaluation: (1) **TravelLLM**, a

CEUR-WS.org/Vol-4162/paper5.pdf

NORA'25: 1st Workshop on Knowledge Graphs & Agentic Systems Interplay co-located with NeurIPS, Dec.1, 2025, Mexico City, Mexico

*Corresponding author.

csjiezhao@gmail.com (J. Zhao); fengjie@tsinghua.edu.cn (J. Feng); liyong07@tsinghua.edu.cn (Y. Li)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-02-07

domain-adapted base model fine-tuned with curated knowledge about cities and POIs. This component enhances the model's spatial reasoning and domain adaptability for diverse urban contexts; (2) **TravelAgent**, an online agentic planner built upon TravelLLM that leverages open Web APIs for real-time information retrieval, maintains structured itinerary memory, and employs adaptive planning strategies to meet user preferences and contextual constraints; (3) **TravelBench**, a scalable benchmark suite with two complementary modules: *KnowEval*, which evaluates factual and spatial knowledge integration using curated urban datasets, and *TripEval*, which measures plan feasibility, personalization, and constraint satisfaction across realistic travel scenarios.

The contributions of this paper are threefold: (1) We release a multi-source urban knowledge dataset covering five representative Chinese cities, encompassing road networks, POIs, attractions, accommodations, and restaurants. The dataset supports both LLM fine-tuning and knowledge-grounded evaluation for urban planning tasks. (2) We develop an online agentic framework that integrates real-time information retrieval, spatially aware planning strategies, and persistent itinerary memory to generate user-centered travel plans. (3) We introduce a comprehensive evaluation suite that jointly assesses knowledge grounding and multi-criteria plan quality, enabling a holistic assessment of knowledge-augmented LLM agents for urban travel planning.

2. Related Work

Recent research on LLM-based travel planning [5, 8] can be broadly categorized into two paradigms: **LLM as Planner** and **LLM as Translator**. The former treats the LLM as the central reasoning and generation engine that directly produces travel itineraries, often enhanced with tool use, agent-based strategies, or prompt optimization. The latter leverages the LLM primarily as a natural language interface, translating user requirements into formal or symbolic representations that external solvers can optimize.

LLM as Planner. Planner-based approaches focus on empowering LLMs to handle the end-to-end travel planning pipeline, from understanding user constraints to generating detailed itineraries. Early efforts such as TravelPlanner [1] established a benchmark for evaluating an LLM agent's ability to use tools and satisfy commonsense and hard constraints. TravelPlanner+ [9] extended this with personalized user models, highlighting the impact of tailoring itineraries to user preferences. Flex-TravelPlanner [10] examined the robustness of planning under dynamic and uncertain conditions, while NATURAL PLAN [3] revealed persistent challenges in multi-city, long-duration scenarios despite providing full task information. Beyond benchmarking, multi-phase planning frameworks [11] such as TDAG [12] and HyperTree Planning [13] decomposed complex trips into manageable sub-tasks, improving scalability. Additional work has targeted prompt optimization [14, 15], multi-module agent designs such as TravelAgent [4], and dialogue-driven multi-agent planning [16]. Collectively, these studies advance the ability of LLMs to operate as autonomous planners, but most still face limitations in robust spatial reasoning and in integrating diverse real-time data streams into the planning loop.

LLM as Translator. Translator-based approaches shift the focus from direct itinerary generation to bridging natural language and structured reasoning systems. In these methods, LLMs convert user queries into machine-interpretable formats—such as symbolic constraint sets, semantic graphs, or formal planning languages—that are then processed by external solvers. For instance, Hao et al. [17] formulated travel planning as a satisfiability modulo theories (SMT) problem, enabling precise constraint handling. ItiNera [18], TRIP-PAL [19], and TTG [20] followed similar pipelines, combining LLM-based parsing with solver-based optimization. ChinaTravel [21] contributed an open benchmark for scalable evaluation of travel planning, focusing on aligning generated plans with real-world travel demands. This paradigm offers strong guarantees on constraint satisfaction and optimality, but often relies on static or incomplete knowledge bases, making it less adaptive to dynamic, multi-source inputs and less capable of leveraging LLMs' generative flexibility for nuanced user preferences.

3. Preliminaries

Definition 1 (Urban Travel Plan). *An urban travel plan* p *is a structured itinerary spanning* M *consecutive days for* N *travelers within an urban environment. It can be represented in a JSON-like format containing fields such as date, attractions, restaurants, accommodations, and transportation, along with optional metadata.*

Definition 2 (Online Trip Data). *Online trip data* $\mathcal{D}_{\text{on}}$ *denotes real-time travel information retrieved from external APIs during planning. It includes attributes of attractions (name, price), restaurants (name, price, cuisine), and accommodations (name, price, hotel type), providing up-to-date references for generating feasible and cost-aware itineraries.*

Definition 3 (Offline City Data). *Offline city data* $\mathcal{D}_{\text{off}}$ *refers to static, city-specific information collected before planning. It comprises road networks, POI datasets, and tourism-related data (e.g., attractions, restaurants, hotels) obtained from public sources. This data serves as a persistent knowledge base that enhances the spatial reasoning and domain knowledge of the underlying LLM.*

Problem Statement. Given a user query q in natural language, the goal of urban travel planning is to generate an itinerary p under accessible online data $\mathcal{D}_{\text{on}}$:

$$p = \mathcal{F}\left(q, \mathcal{D}_{\text{on}}\right)$$

where $\mathcal{F}$ denotes an *agentic planner* built upon LLMs and augmented with offline city data $\mathcal{D}_{\text{off}}$.

4. AgentTravel

Figure 1 demonstrates the architecture of *AgentTravel*, which integrates knowledge-augmented modeling, agentic real-time planning, and multi-perspective evaluation. The process starts with a natural-language query, which activates TravelAgent to coordinate interactions among TravelLLM, real-time trip data, and a structured memory that tracks the itinerary in progress. The generated plan is then assessed by TravelBench, which combines KnowEval and TripEval for a comprehensive evaluation of knowledge grounding and planning quality.

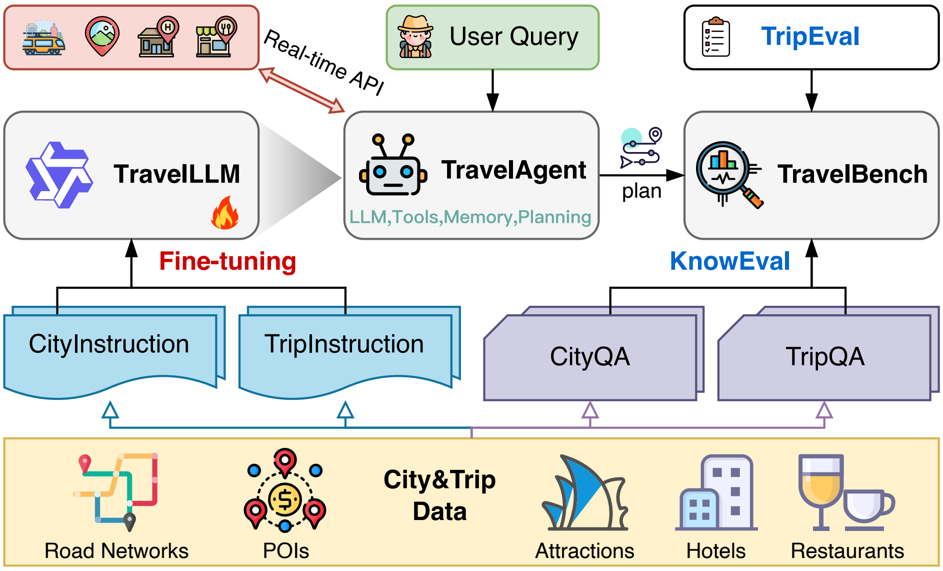

Figure 1: Overview of AgentTravel.

4.1. TravelLLM

TravelLLM is a knowledge-augmented large language model tailored for urban travel planning. It equips the base LLM with two complementary capabilities often missing in general-purpose models: (1) spatial reasoning over urban environments, including road networks, POI relations, and travel distances; and (2) domain-specific travel knowledge, such as details about attractions, accommodations, and restaurants.

We use Qwen 2.5-7B as the backbone model and apply Low-Rank Adaptation (LoRA) for efficient domain and spatial knowledge injection. The model is fine-tuned on a hybrid corpus that combines two domain-specific instruction sets: *CityInstruction* and *TripInstruction*.

4.1.1. CityInstruction: Urban Spatial Knowledge

CityInstruction focuses on enhancing an LLM's spatial understanding and reasoning capabilities in urban contexts. It is built from instruction–response pairs derived from our curated *offline city data* $\mathcal{D}_{\text{off}}$, covering two primary categories:

- **Intersection**: mapping intersection names to geographic coordinates (`name2coords`), performing reverse lookups from coordinates to names (`coords2name`), and computing distances between two intersections (`between_distance`).
- **Points of Interest**: linking POI names to their corresponding addresses (`name2address`) and categories, enabling the model to recognize and reason about relevant locations.

These instructions equip the model with fine-grained spatial grounding, facilitating more accurate reasoning over locations, navigation, and proximity when generating travel itineraries.

4.1.2. TripInstruction: Travel-Specific Knowledge

TripInstruction focuses on travel-specific entities, enriching the model's understanding of attractions, accommodations, and restaurants to produce realistic and personalized itineraries. It is also derived from $\mathcal{D}_{\text{off}}$ and includes three main categories:

- **Attractions**: mapping attraction names to their addresses (`name2address`), ticket information (`name2ticket`), and operating hours (`name2opentime`), allowing the model to recommend feasible and timely visits.
- **Hotels**: providing hotel addresses (`name2address`) and average prices (`name2price`), enabling accommodation suggestions that fit budget and location constraints.
- **Restaurants**: associating restaurant names with their addresses (`name2address`), price (`name2price`), and cuisine types (`name2cuisine`), supporting meal planning for users.

By incorporating these fine-grained attributes, the model gains domain-specific grounding to generate itineraries that are both factually accurate and preference-aware. To retain broad conversational and task-following abilities while injecting urban knowledge, we augment the domain-specific instructions with three open instruction datasets: ShareGPT [1], UltraChat [22], and Open-Platypus [23]. This hybrid mix stabilizes the model's general reasoning and dialogue quality during LoRA fine-tuning, mitigating over-specialization to the travel domain.

4.2. TravelAgent

TravelAgent is the agentic controller in the framework, responsible for translating user requirements into concrete, constraint-aware itineraries through real-time interaction with online trip data $\mathcal{D}_{\text{on}}$ and the knowledge-enhanced model *TravelLLM*. It operates through three tightly coupled modules: a structured memory for state tracking, a domain-specific toolbox for real-time data retrieval, and a ReAct-style planning loop for interleaved reasoning and action.

---

[1] https://huggingface.co/datasets/shareAI/ShareGPT-Chinese-English-90k

4.2.1. Structured Memory for State Tracking

Urban travel planning involves numerous interdependent elements and evolving contextual factors. *TravelAgent* maintains a day-by-day *structured memory* that records itinerary details (e.g., attractions, meals, accommodations, transportation, and estimated per-capita costs), thus providing a persistent state for iterative updates as planning progresses. The schema for each day is defined as:

```
{
    "date": str,
    "num_people": int,
    "visit_attractions": list,
    "breakfast": {"name": str, "cuisines": str},
    "lunch": {"name": str, "cuisines": str},
    "dinner": {"name": str, "cuisines": str},
    "accommodation": {"name": str, "type": str},
    "transportation": {"org-dst": str},
    "cost_per_capita": dict
}
```

4.2.2. Domain-Specific Toolbox

The *domain-specific toolbox* is a suite of parameterized functions implemented via JSON-schema-based calls, enabling *TravelAgent* to retrieve, filter, and integrate external travel information during itinerary construction. Each tool serves a specific role in the planning workflow:

- `MemoryInit` – initializes global trip parameters, such as travel dates and number of travelers, providing a consistent context for subsequent planning steps.
- `AttractionSearch` – queries online trip data sources to obtain detailed information about candidate attractions, including names, locations, and basic attributes.
- `NearbyRestaurantSearch` – identifies restaurants within a specified radius of a given point of interest, allowing the integration of geographically coherent dining options.
- `NearbyHotelSearch` – retrieves available accommodations in the vicinity of a target location, facilitating proximity-based lodging selection.
- `TransportationSearch` – returns feasible transportation routes between two locations, supporting realistic scheduling and connectivity.
- `MemoryWrite` – updates the structured memory with newly retrieved or revised itinerary elements, ensuring that intermediate planning states remain accessible for reasoning.
- `PlanOutput` – compiles the current itinerary state into a coherent, user-facing travel plan representation.

By encapsulating external interactions in modular, parameterized tools, the framework can adapt to diverse data providers, geographic contexts, and planning requirements without altering its core reasoning and control logic.

4.2.3. ReAct-Style Planning Loop

TravelAgent follows a ReAct-style planning paradigm [24], interleaving reasoning and tool invocation in an iterative feedback loop. At each iteration, the agent performs three coordinated steps: (1) **State Interpretation**: analyzes the structured memory to evaluate progress and identify missing or inconsistent elements. (2) **Action Selection**: decides between internal reasoning (e.g., sequencing attractions, allocating time slots) and external tool invocation (e.g., querying restaurants, retrieving routes). (3) **State Update**: integrates the results of reasoning or retrieved data into the structured memory, incrementally refining the itinerary state.

4.3. TravelBench

TravelBench is a two-part benchmark designed to evaluate both *knowledge grounding* and *itinerary quality* for LLM-based urban travel planning. Unlike prior evaluations such as TravelPlanner [1], it is built on (1) curated real-world POI and route datasets from major tourist cities, and (2) a unified framework that jointly assesses factual knowledge of urban entities and the feasibility of multi-day itineraries under commonsense and user-preference constraints.

4.3.1. KnowEval

KnowEval assesses an LLM's capability to retrieve and reason over factual urban knowledge before the planning stage. It consists of two complementary subsets: **CityQA**, which focuses on spatial knowledge such as road networks and general POIs, and **TripQA**, which targets domain-specific travel entities including attractions, hotels, and restaurants.

Each subset is further structured around fine-grained attribute categories derived from the curated offline dataset $\mathcal{D}_{\text{off}}$. Specifically, CityQA covers: (1) *Road attributes* - OD pairs, connectivity, and distances; (2) *POI attributes* - name-to-address mappings. TripQA includes: (1) *Attractions* - address, ticket price, and opening hours; (2) *Hotels* - address and average price; (3) *Restaurants* - address, average price, and cuisine tags.

We converte the knowledge item into a multiple-choice question (MCQ) automatically generated by `GPT-4o-mini` from $\mathcal{D}_{\text{off}}$ and validated by human annotators for factual accuracy and clarity. All question text is presented in Chinese to maintain fidelity with real-world POI names and descriptions, but the underlying methodology is language-agnostic and can be readily applied to other languages or regions by replacing the source datasets. This ensures that the evaluation is grounded in authentic curated resources while remaining broadly extensible.

4.3.2. TripEval

TripEval evaluates the *feasibility* and *personalization quality* of travel plans generated by LLM-based agents. It operates on the structured memory produced by the agent and applies a suite of rule-based validators that cross-reference curated POI databases and real-time transportation APIs. The evaluation metrics are grouped into two major categories, as summarized in Table 1.

Commonsense Constraints	
Valid Fields	All required fields in the travel plan are populated.
Valid Days	The number of planned days matches the requested trip length.
Valid Attractions	Every listed attraction is real and publicly accessible.
Valid Restaurants	Every listed restaurant is real and currently operating.
Valid Accommodations	All accommodations are valid and bookable.
Available Transportation	Transportation between locations is feasible.
No Repeated Attractions	No attraction is visited more than once.
No Repeated Restaurants	No restaurant is visited more than once.
Preference Constraints	
Reasonable Budget	The total cost remains within the user-specified budget.
Favorite Cuisine	The itinerary includes the user's preferred cuisines.
Preferred Hotel Type	Accommodation matches the specified hotel category.

Table 1
Constraint categories in TripEval.

5. Experiments

5.1. Settings

5.1.1. City & Trip Datasets

We construct the datasets from five representative tourist cities in China: Beijing, Shanghai, Guangzhou, Chengdu, and Xi'an. These cities were selected for the rich cultural heritage, diverse urban layouts, and high tourist activity, making them ideal testbeds for evaluating urban travel planning systems.

The *city-level data* is sourced from OpenStreetMap[2] and Amap[3] , covering road networks and POIs. The *trip-level data* comes from Ctrip[4] , including attractions, accommodations, and restaurants with rich attributes such as prices, operating hours, and category labels. Table 2 summarizes the dataset statistics. All data is in Chinese to match real-world place names and descriptions, but this does not impact the generality of our approach. The framework and evaluation pipeline are *language-agnostic* and can be applied to other languages or cities.

	City Data			**Trip Data**		
	Num. Roads	Num. Intersections	Num. POIs	Num. Attractions	Num. Hotels	Num. Restaurants
Beijing	33,794	20,327	288,852	3,471	1,473	132,379
Shanghai	38,281	18,871	424,198	3,967	1,417	117,880
Guangzhou	25,142	17,556	483,344	3,552	1,406	82,603
Chengdu	28,564	16,389	422,244	3,312	1,411	100,405
Xi'an	23,176	14,215	279,080	3,107	1,439	53,263

Table 2
Statistics of City and Trip Datasets.

5.1.2. Query Generation

To simulate realistic and diverse user requests for itinerary planning, we develop an automated pipeline that generates natural-language queries paired with structured JSON representations. Given a target city and difficulty level, the generator samples key trip parameters - duration, number of travelers, start date, and budget - through controlled randomization. Budgets are derived from a per-capita-per-day baseline cost and adjusted by multiplicative factors for different hotel categories, ensuring internal consistency across trip attributes.

Preference constraints are injected in three tiers: (1) **No preference** - budget constraint only; (2) **Single preference** - one hotel category or one to three preferred cuisines; (3) **Combined preferences** - both hotel category and multiple cuisines. We generate 100 queries per city with difficulty levels, and prompt `GPT-4o-mini` to produce a fluent, user-like query.

5.1.3. Metrics

We evaluate model performance using five complementary metrics: **Delivery Rate (DR)**: the percentage of itineraries successfully completed within the allowed number of reasoning and tool-invocation steps; **Commonsense Pass Rate (CPR)**: the proportion of itineraries satisfying all commonsense constraints defined in *TripEval* (e.g., valid POIs, non-repetition, feasible transportation); **Preference Pass Rate (PPR)** – the proportion satisfying all user-specified preference constraints (e.g., budget, cuisine, accommodation type); **Final Pass Rate (FPR)** – the percentage of itineraries simultaneously meeting both commonsense and preference constraints; **Accuracy (ACC)** – the fraction of correctly answered multiple-choice questions in *KnowEval*, reflecting factual and spatial knowledge grounding.

---

[2] https://www.openstreetmap.org/

[3] https://lbs.amap.com/

[4] https://ctrip.com/

	Beijing (#200)		Shanghai (#200)		Guangzhou (#200)		Chengdu (#200)		Xi’an (#200)	
Model	**CityQA**	**TripQA**	**CityQA**	**TripQA**	**CityQA**	**TripQA**	**CityQA**	**TripQA**	**CityQA**	**TripQA**
Qwen2.5-7B	0.420	0.580	0.445	**0.645**	0.475	**0.655**	0.475	0.515	0.450	0.585
GLM4-9B	0.430	0.465	0.420	0.555	0.425	0.535	0.470	0.410	0.450	0.530
Gemma3-12B	0.325	0.490	0.420	0.475	0.330	0.455	0.435	0.455	0.390	0.550
GPT4o-mini	**0.500**	0.530	**0.500**	0.610	0.500	0.585	**0.550**	0.430	0.490	0.620
TravelLLM	0.445	**0.630**	0.410	0.625	**0.525**	0.620	0.505	**0.535**	**0.550**	**0.635**

Table 3

Comparison of different LLMs on KnowEval. Bold denotes the best result, underline denotes the second-best.

	Beijing (#100)				Shanghai (#100)				Guangzhou (#100)				Chengdu (#100)				Xi’an (#100)			
Model	**DR**	**CPR**	**PPR**	**FPR**	**DR**	**CPR**	**PPR**	**FPR**	**DR**	**CPR**	**PPR**	**FPR**	**DR**	**CPR**	**PPR**	**FPR**	**DR**	**CPR**	**PPR**	**FPR**
Qwen2.5-7B	0.97	0.18	0.43	0.15	0.89	0.12	0.25	0.04	0.91	0.11	0.18	0.00	0.94	0.18	0.47	0.11	0.90	0.19	0.53	0.19
GLM4-9B	0.94	0.20	0.51	0.19	0.98	0.06	0.31	0.02	0.91	0.12	0.34	**0.08**	0.97	0.04	0.50	0.03	0.96	0.17	0.55	0.16
Gemma3-12B	0.29	0.00	0.17	0.00	0.34	0.00	0.15	0.00	0.31	0.00	0.14	0.00	0.31	0.02	0.09	0.01	0.13	0.00	0.07	0.00
GPT4o-mini	1.00	0.41	0.40	0.19	1.00	0.41	0.02	0.01	1.00	0.39	0.14	0.07	1.00	0.08	0.13	0.03	1.00	0.11	0.52	0.05
AgentTravel	0.98	0.42	0.34	**0.24**	0.99	0.20	0.12	**0.10**	1.00	0.31	0.05	0.01	0.99	0.14	0.42	**0.15**	1.00	0.31	0.40	**0.24**

Table 4

Results of different LLMs on TripEval. Bold denotes the best result, underline denotes the second-best.

5.2. Results

We evaluate *AgentTravel* against several competitive LLM baselines on both *KnowEval* and *TripEval*. To ensure a fair and controlled comparison, all models operate within the same *TravelAgent* planning framework, sharing an identical prompting template, structured memory schema, ReAct-style reasoning loop, and domain-specific toolbox.

Table 3 reports results on **CityQA** and **TripQA** across five cities. *TravelLLM* ranks first or second in nearly all cases, showing the best overall balance. On **TripQA**, TravelLLM achieves the highest scores in Beijing, Chengdu, and Xi’an, and competitive results in Shanghai and Guangzhou. These gains confirm that domain-specific fine-tuning improves factual recall and reasoning on travel entities. On **CityQA**, GPT-4o-mini leads in Beijing, Shanghai, and Chengdu, while TravelLLM performs better in Guangzhou and Xi’an. This shows that city-level adaptation can match or surpass larger models in localized spatial reasoning.

Table 4 reports delivery (DR), commonsense (CPR), preference (PPR), and final pass rate (FPR) across five cities. *AgentTravel* achieves near-perfect delivery ($\geq$0.98) across all settings, indicating strong execution stability. GPT-4o-mini performs best on commonsense reasoning, while *AgentTravel* remains competitive in Beijing and Xi’an, outperforming other open models. On personalization, performance is moderate but consistent, slightly below Qwen and GLM in some cities. Notably, *AgentTravel* attains the highest FPR in four cities, reflecting improved overall feasibility.

Despite these advances, LLM-based travel planning remains challenging. Our results suggest that integrating knowledge-grounded reasoning with structured memory offers a promising path toward more reliable and adaptive LLM planners.

6. Conclusion

This paper introduced AgentTravel, a unified framework for LLM-based urban travel planning, combining knowledge-grounded modeling, agentic reasoning, and multi-perspective evaluation. Experiments across five Chinese cities show that domain- and city-specific fine-tuning strengthens factual reasoning, while structured agentic planning improves itinerary feasibility. Despite these gains, LLM-based travel planning remains a challenging task, requiring better commonsense reasoning, preference alignment, and adaptability to real-world data.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4 in order to: Grammar and spelling check, Paraphrase and reword. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] J. Xie, K. Zhang, J. Chen, T. Zhu, R. Lou, Y. Tian, Y. Xiao, Y. Su, Travelplanner: A benchmark for real-world planning with language agents, in: Forty-first International Conference on Machine Learning, 2024.

[2] UTA AI, Wanderboat: Ai travel planning assistant, 2024. URL: https://wanderboat.ai/, accessed: 2025-06-18.

[3] H. S. Zheng, S. Mishra, H. Zhang, X. Chen, M. Chen, A. Nova, L. Hou, H.-T. Cheng, Q. V. Le, E. H. Chi, et al., Natural plan: Benchmarking llms on natural language planning, arXiv preprint arXiv:2406.04520 (2024).

[4] A. Chen, X. Ge, Z. Fu, Y. Xiao, J. Chen, Travelagent: An ai assistant for personalized travel planning, arXiv preprint arXiv:2409.08069 (2024).

[5] K.-H. Lee, I. Fischer, Y.-H. Wu, D. Marwood, S. Baluja, D. Schuurmans, X. Chen, Evolving deeper llm thinking, arXiv preprint arXiv:2501.09891 (2025).

[6] J. Feng, T. Liu, Y. Du, S. Guo, Y. Lin, Y. Li, Citygpt: Empowering urban spatial cognition of large language models, in: Proceedings of the 31th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2025.

[7] J. Feng, J. Zhang, T. Liu, X. Zhang, T. Ouyang, J. Yan, Y. Du, S. Guo, Y. Li, Citybench: Evaluating the capabilities of large language models for urban tasks, in: Proceedings of the 31th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2025.

[8] S. Chaudhuri, P. Purkar, R. Raghav, S. Mallick, M. Gupta, A. Jana, S. Ghosh, Tripcraft: A benchmark for spatio-temporally fine grained travel planning, arXiv preprint arXiv:2502.20508 (2025).

[9] H. Singh, N. Verma, Y. Wang, M. Bharadwaj, H. Fashandi, K. Ferreira, C. Lee, Personal large language model agents: A case study on tailored travel planning, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, 2024, pp. 486–514.

[10] J. Oh, E. Kim, A. Oh, Flex-travelplanner: A benchmark for flexible planning with language agents, arXiv preprint arXiv:2506.04649 (2025).

[11] C. Xie, D. Zou, A human-like reasoning framework for multi-phases planning task with large language models, in: ICML Workshop on Large Language Models and Cognition, 2024.

[12] Y. Wang, Z. Wu, J. Yao, J. Su, Tdag: A multi-agent framework based on dynamic task decomposition and agent generation, Neural Networks (2025) 107200.

[13] R. Gui, Z. Wang, J. Wang, C. Ma, H. Zhen, M. Yuan, J. Hao, D. Lian, E. Chen, F. Wu, Hypertree planning: Enhancing llm reasoning via hierarchical thinking, arXiv preprint arXiv:2505.02322 (2025).

[14] A. Miin, T. Wei, Smart language agents in real-world planning, arXiv preprint arXiv:2407.19667 (2024).

[15] W. Chen, S. Koenig, B. Dilkina, Reprompt: Planning by automatic prompt engineering for large language models agents, arXiv preprint arXiv:2406.11132 (2024).

[16] X. Zhang, Y. Deng, Z. Ren, S.-K. Ng, T.-S. Chua, Ask-before-plan: Proactive language agents for real-world planning, in: Findings of the Association for Computational Linguistics: EMNLP 2024, 2024, pp. 10836–10863.

[17] Y. Hao, Y. Chen, Y. Zhang, C. Fan, Large language models can solve real-world planning rigorously with formal verification tools, in: Proceedings of the 2025 Conference of the Nations of the Americas

Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2025, pp. 3434–3483.

[18] Y. Tang, Z. Wang, A. Qu, Y. Yan, Z. Wu, D. Zhuang, J. Kai, K. Hou, X. Guo, J. Zhao, et al., Itinera: Integrating spatial optimization with large language models for open-domain urban itinerary planning, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, 2024, pp. 1413–1432.

[19] T. de la Rosa, S. Gopalakrishnan, A. Pozanco, Z. Zeng, D. Borrajo, Trip-pal: Travel planning with guarantees by combining large language models and automated planners, arXiv preprint arXiv:2406.10196 (2024).

[20] D. Ju, S. Jiang, A. Cohen, A. Foss, S. Mitts, A. Zharmagambetov, B. Amos, X. Li, J. T. Kao, M. Fazel-Zarandi, et al., To the globe (TTG): Towards language-driven guaranteed travel planning, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2024, pp. 240–249.

[21] J.-J. Shao, X.-W. Yang, B.-W. Zhang, B. Chen, W.-D. Wei, L.-Z. Guo, Y.-f. Li, Chinatravel: A real-world benchmark for language agents in chinese travel planning, arXiv preprint arXiv:2412.13682 (2024).

[22] N. Ding, Y. Chen, B. Xu, Y. Qin, Z. Zheng, S. Hu, Z. Liu, M. Sun, B. Zhou, Enhancing chat language models by scaling high-quality instructional conversations, arXiv preprint arXiv:2305.14233 (2023).

[23] A. N. Lee, C. J. Hunter, N. Ruiz, Platypus: Quick, cheap, and powerful refinement of llms, arXiv preprint arXiv:2308.07317 (2023).

[24] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, React: Synergizing reasoning and acting in language models, in: International Conference on Learning Representations (ICLR), 2023.

A. CityInstruction & TripInstruction Examples

Example (Intersection-`name2coords`):

```
{
  "instruction": "Please provide the geographical coordinates of a given
      intersection",
  "input": "Zhouzhang Road and Fangyi Road Intersection",
  "output": "115.6906259, 39.5750395"
}
```

Example (POI-`name2address`):

```
{
   "instruction": "Please provide the address of a given Point of Interest.",
   "input": "Sanyuan Ecological Park in Beijing",
   "output": "No. 8, Xiaoyunli, Sanyuan Park, Taiyanggong Township, Chaoyang
       District"
}
```

Example (Attractions-`name2ticket`):

```
{
   "instruction": "Please tell me the ticket price of a given attraction.",
   "input": "Old Summer Palace in Beijing",
   "output": "The ticket price for the Old Summer Palace in Beijing is 10 CNY."
}
```

Example (Restaurants-`name2cuisine`)

```
{
  "instruction": "What is the main cuisine offered by this restaurant?",
  "input": "Zi Shan Restaurant, Mandarin Oriental Wangfujing, Beijing",
  "output": "Zi Shan Restaurant offers ['Cantonese Cuisine', 'Cantonese Dim Sum
      ']."
}
```

B. CityQA & TripQA Examples

CityQA Example (Road Connectivity):
Q: Which road is directly connected to Yunquan Road?
A. Haidian South Road B. Zhiquan Road C. Yuanboyuan South Road D. Zhaitang Street
Answer: B

TripQA Example (Hotel Price):
Q: What is the average per-capita price of Lavande Hotel (Beijing Headquarters Base)?
A. 285 CNY B. 699 CNY C. 535 CNY D. 313 CNY
Answer: D

C. Prompts

C.1. ReAct Planning Prompt

```
You are a travel planning assistant. Your task is to help users create detailed
↪  daily travel itineraries (in Chinese) by strictly following the instructions
↪  below.

### Responsibilities
1. Understand user requirements: Accurately extract travel start/end dates, number
↪  of people, budget, preferences, etc.
2. Retrieve information using tools: Use designated tools to gather data on
↪  attractions, restaurants, accommodations, and transportation.
3. Preliminary setup: Before starting the planning task, use `MemoryInit` to
↪  initialize the memory and set up essential information such as travel dates and
↪  group size.
4. Timely record-keeping: Each time a restaurant, accommodation, or transportation
↪  item is obtained, immediately write it to the memory using `MemoryWrite`.
5. Step-by-step itinerary construction: First determine the full list of
↪  attractions to be visited across the trip. Then, collect and record restaurants,
↪  accommodations, and transportation information on a day-by-day basis.

### Task Execution Flow
#### Phase 1: Plan Attractions Across the Entire Trip
1. Use `MemoryInit` to initialize the memory with travel dates and number of people.
2. Call `AttractionSearch` to retrieve information about attractions in the target
↪  city.
3. Select appropriate attractions and assign them to each day in a balanced manner
↪  (avoid overcrowded schedules).
4. Use `MemoryWrite` to record the attractions for Day 1. Repeat this for each day
↪  until all attractions have been assigned and recorded.
```

```
#### Phase 2: Daily Information Collection and Logging
For each day, perform the following steps in sequence:
1. Call `NearbyRestaurantSearch` to obtain breakfast options.
2. Write the breakfast information to the memory using `MemoryWrite`.
3. Repeat the above two steps for lunch.
4. Repeat the above two steps for dinner.
5. Call `NearbyHotelSearch` to find accommodation near the day's attractions.
6. Record accommodation details with `MemoryWrite`.
7. Call `TransportationSearch` to get transportation plans between all visited
↪  attractions for the day.
8. Log the transportation details using `MemoryWrite`.


### Using Thought-Action-Observation Loop
- Thought: Express your current reasoning using natural language. Do not include any
↪  tool calls in this phase.
- Action: Based on your thought, invoke the appropriate tool using valid parameters.
↪  Use the system’s function-calling mechanism where possible.
- Observation: Examine the tool's output and use it to guide the next thought.

### Important Guidelines
- Do not use attraction/restaurant/hotel names unless they come from the results
↪  returned by `AttractionSearch` or `NearbySearch`.
- Each piece of information must be collected and recorded **independently**;
↪  merging multiple tasks is not allowed.
- To avoid forgetting data, each collected item must be immediately written to the
↪  memory using `MemoryWrite`.
- For days with multiple attractions, transportation between each pair must be
↪  queried and written separately.
- Each Action phase should involve only one tool invocation for a single specific
↪  task. Multiple tool uses in one action are not allowed.
- After all daily information has been recorded in the memory, call `PlanOutput` to
↪  generate the final complete travel plan.
```

C.2. Knowledge Evaluation Prompt

```
Here is a multiple-choice question related to urban travel knowledge. You need to
↪  choose the most appropriate answer from A, B, C, and D. Please output only the
↪  letter corresponding to the correct answer, with no additional content.
```

D. Example Query and Generated Plan

Below is an example of a user query in English and the corresponding structured travel plan produced by our system.

User Query

```
I would like a 1-day travel plan in Beijing for 2 people, starting on July 20, 2025,
↪  with a budget of around 2,200 CNY.
```

Generated Plan

```
{
    "date": "2025-07-20",
    "num_people": 2,
    "visit_attractions": [
        "Summer Palace",
        "Palace Museum (Forbidden City)",
        "Temple of Heaven"
    ],
    "breakfast": {
        "name": "Palace Museum Restaurant",
        "cuisines": "Chinese"
    },
    "lunch": {
        "name": "Tingliguan Restaurant (Summer Palace Branch)",
        "cuisines": "Chinese"
    },
    "dinner": {
        "name": "Donglaishun Restaurant (Temple of Heaven Branch)",
        "cuisines": "Beijing Cuisine"
    },
    "accommodation": {
        "name": "Atour Light Hotel Beijing Qianmen Temple of Heaven",
        "type": "Comfort"
    },
    "transportation": {
        "Summer Palace → Palace Museum": "From the Summer Palace, walk 791 meters
        ↪  to ...",
        "Palace Museum → Temple of Heaven": "From the Palace Museum, walk 870
        ↪  meters to ..."
    },
    "cost_per_capita": {
        "Palace Museum": 60,
        "Summer Palace": 30,
        "Temple of Heaven": 10,
        "breakfast": 86,
        "lunch": 153,
        "dinner": 147,
        "accommodation": 300,
        "transit": 8.0
    }
}
```