

TP-RAG: Benchmarking Retrieval-Augmented Large Language Model Agents for Spatiotemporal-Aware Travel Planning

Hang Ni¹, Fan Liu¹, Xinyu Ma², Lixin Su², Shuaiqiang Wang²

Dawei Yin²✉, Hui Xiong¹ Hao Liu¹✉,

¹The Hong Kong University of Science and Technology (Guangzhou), ²Baidu Inc.

{hni017, fliu236}@connect.hkust-gz.edu.cn

{xinyuma2016, sulixinict, shqiang.wang}@gmail.com

yindawei@acm.org, {xionghui, liuh}@ust.hk

Abstract

Large language models (LLMs) have shown promise in automating travel planning, yet they often fall short in addressing nuanced spatiotemporal rationality. While existing benchmarks focus on basic plan validity, they neglect critical aspects such as route efficiency, POI appeal, and real-time adaptability. This paper introduces *TP-RAG*¹, the first benchmark tailored for retrieval-augmented, spatiotemporal-aware travel planning. Our dataset includes 2,348 real-world travel queries, 85,575 fine-grained annotated POIs, and 18,784 high-quality travel trajectory references sourced from online tourist documents, enabling dynamic and context-aware planning. Through extensive experiments, we reveal that integrating reference trajectories significantly improves spatial efficiency and POI rationality of the travel plan, while challenges persist in universality and robustness due to conflicting references and noisy data. To address these issues, we propose *EvoRAG*, an evolutionary framework that potentially synergizes diverse retrieved trajectories with LLMs' intrinsic reasoning. *EvoRAG* achieves state-of-the-art performance, improving spatiotemporal compliance and reducing commonsense violation compared to ground-up and retrieval-augmented baselines. Our work underscores the potential of hybridizing Web knowledge with LLM-driven optimization, paving the way for more reliable and adaptive travel planning agents.

1 Introduction

Emerging studies have explored the potential of LLMs to serve as travel agents capable of interpreting natural language inquiries, and autonomously generating travel plans that comprise daily tourist activities detailed with various Points of Interest (POIs) (Wong et al., 2023). Despite their promise, existing works (Xie et al., 2024; Singh et al., 2024;

¹Our dataset and code are available at <https://github.com/usail-hkust/TP-RAG>.

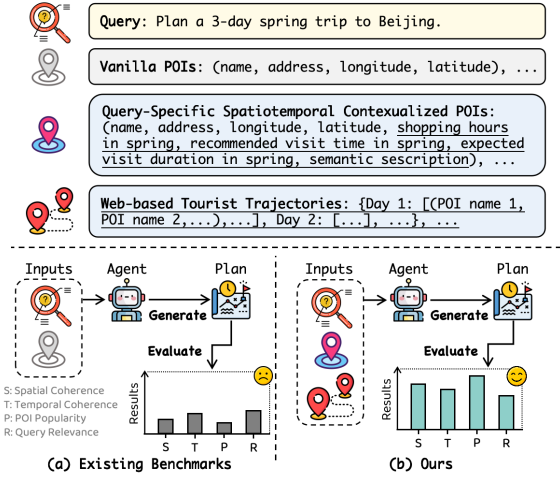


Figure 1: TP-RAG distinguishes itself from existing works by: (1) query-specific spatiotemporal contextualization and (2) trajectory-level knowledge utilization.

Hao et al., 2024) focus primarily on whether the plans meet basic commonsense requirements (*e.g.*, complete information, actual POIs), while overlooking the nuanced spatiotemporal rationality that is critical for practicality, such as spatiotemporal coherence (*e.g.*, transit efficiency and schedule comfort), POI attractiveness (*i.e.*, scenic spots with local cultural characteristics and high popularity), and temporal adaptability (*i.e.*, capturing time-evolving POI information, such as seasonal closures and altered opening hours). These oversights may result in flawed plans with inefficient routes, exhausting journeys, unappealing POIs, or limited flexibility. Therefore, this study investigates the capabilities of LLM agents in travel planning while emphasizing spatiotemporal awareness.

Contemporary research primarily focuses on benchmarking the abilities of LLM agents in travel planning (Xie et al., 2024; Singh et al., 2024) or exploring sophisticated strategies to enhance planning effectiveness (Tang et al., 2024; Xie and Zou, 2024). However, these studies rely exclusively on the POI-level knowledge, *i.e.*, the metadata of the candidate POIs, and agents' internal reasoning capabilities to

construct travel plans from scratch. This ground-up approach faces inherent limitations stemming from insufficient spatiotemporal reasoning (Manvi et al., 2024; Li et al., 2024; Chu et al., 2024) and the generation of hallucinated or outdated content (Huang et al., 2025; Gao et al., 2023), which restrict the spatiotemporal coherence of the produced travel plans. Notably, beyond POI-level knowledge, the Web offers a wealth of up-to-date travel documents that encapsulate real-life experience and collective wisdom into practical tourist trajectories, *i.e.*, the sequences of POIs interconnected by spatiotemporal logic, which are neglected by current studies. To address this gap, advances in Retrieval-augmented Generation (RAG) approaches (Fan et al., 2024) can enable LLM agents to integrate such trajectory-level knowledge represented in Web documents, which provide substantial spatiotemporal-aware insights for travel planning.

In this paper, we introduce a new travel planning benchmark, *TP-RAG*, to investigate whether retrieval-augmented LLM agents can effectively leverage trajectory-level knowledge to produce spatiotemporally coherent travel plans. Our benchmark comprises a dataset grounded in the real-world search engine, featuring high-quality data sources. It is designed to develop and evaluate LLM agents in generating spatiotemporal coherent travel plans, adhering to user queries and utilizing relevant POI and trajectory information. Since privacy concerns, in our dataset, we enclose the verbalized trajectories extracted from the newest Web documents instead of using full document content. Totally, our dataset includes 2,348 travel queries, 85,575 geotagged POIs and 18,784 tourist trajectory references. Unlike prior datasets, our dataset incorporates query-customized latest spatiotemporal attributes into POI information, enabling time-adaptive and spatiotemporal-aware planning. In addition, the inclusion of tourist trajectories encourages retrieval-augmented LLM agents to employ the vast repository of Web knowledge for plan enhancements. The comparison between *TP-RAG* and existing benchmarks is illustrated in Figure 1.

Based on our benchmark, we evaluate various LLM-based travel planning methods. The results reveal notable limitations of advanced LLM agents which are constrained by internal knowledge, while highlighting promising prospects for the utilization of Web-based tourist trajectories in spatiotemporal travel planning. Our in-depth analysis further uncovers concerns regarding the universality

and robustness of retrieval-augmented travel planning approaches. To address these issues, we propose *EvoRAG*, a LLM-based evolutionary framework that iteratively optimizes travel plans through population-based selection, crossover and mutation of varied trajectory knowledge. It effectively blends the merits of divergent retrieved knowledge and agents’ intrinsic planning capacity, while alleviating the impact of noisy information, the superiority of which is demonstrated by our experiments.

Our main contributions are three-fold: (1) *TP-RAG*, the first travel planning benchmark for trajectory-level retrieval-augmented and spatiotemporal-aware travel planning, using around 1 billion GPT-4o tokens for dataset construction. (2) Extensive experiments (*i.e.*, over 5,000 A800 GPU hours) with various travel planning methods in different evaluation dimensions, showcasing both the opportunities and challenges of incorporating trajectory-level knowledge. (3) A simple yet effective method, *EvoRAG*, that further counters the limitations of retrieval-augmented travel planning.

2 Related Work

2.1 Benchmarks of LLM-based Travel Planning

To assess the capabilities of LLM agents in complex and realistic planning tasks, recent benchmarks have proliferated in travel planning, which stands out as a significant domain. One line of research, into which our study falls, delves into the LLM-centric travel planning (Xie et al., 2024; Singh et al., 2024; Zhang et al., 2024; Chaudhuri et al., 2025). For example, TravelPlanner (Xie et al., 2024) investigates long-horizon travel planning in multi-constraint scenarios. Beyond relying solely on LLMs, another line of benchmarks examine hybrid approaches that leverage LLMs for natural language interpretation paired with symbolic solvers to ensure solution validity through formal verification (Hao et al., 2024; de la Rosa et al., 2024; Shao et al., 2024). Despite progress, these benchmarks fail to incorporate sufficient fine-grained spatiotemporal contexts into planning, and are confined to agents’ internal reasoning processes, which hinder the real-world deployment.

2.2 LLM Agent for Travel Planning

Large Language Model (LLM) agents have demonstrated remarkable effectiveness across a wide

range of applications (Liu et al., 2025a,b). Among these, travel planning remains an intricate problem involving intent comprehension, information seeking, and long-horizon planning. To automate this task, current research on LLM travel agents bifurcates into two paradigms: LLM-driven and hybrid. LLM-driven approaches seek to enhance LLM agents’ intrinsic planning capacities via advanced techniques, such as multi-agent collaboration which achieves coordination among LLM specialists (Xie and Zou, 2024; Zhang et al., 2025) and LLM-based optimization that iteratively refines the quality of travel schedules (Yuan et al., 2024; Lee et al., 2025). In contrast, hybrid approaches tackle LLM agents’ limitations through the integration of computational planning modules, such as route optimizers (Tang et al., 2024), heuristic POI selection algorithms (Chen et al., 2024a), and symbolic solvers (Ju et al., 2024).

2.3 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) systems have emerged as pivotal solutions for enhancing LLMs with external knowledge (Fan et al., 2024; Li et al., 2025). Recent research efforts have been devoted to building reliable benchmarks to evaluate RAG performance, emphasizing two assessment aspects: retrieval efficacy (e.g., relevance, utility) (Lyu et al., 2025; Saad-Falcon et al., 2024) and generation quality (e.g., accuracy, coherence) (Chen et al., 2024b; Qi et al., 2024b), with our study concentrating on the latter. Beyond open-domain scenarios, while some studies have demonstrated success in specific domains such as medical, legal and financial fields (Xiong et al., 2024; Pipitone and Alami, 2024; Wang et al., 2024), travel applications remain nascent, restricted in POI-level tasks such as question-answering (QA) (Song et al., 2024; Yu et al., 2025) and city or POI recommendation (Banerjee et al., 2024; Qi et al., 2024a). And there is a lack of benchmarks for travel planning tasks that require real-time knowledge integration and multi-objective resolution.

3 TP-RAG

3.1 Background

We focus on generating single-city, multi-day travel plan consisting of attraction POIs, with some critical definitions delineated below:

Query. A travel query q is articulated by the user in natural language, and comprises significant ele-

ments such as the city name, travel duration, and personalized travel constraints.

Point of Interest (POI). A Point of Interest (POI) $p = (p_n, p_s, p_t, p_d)$ refers to a specific location that holds significance or interest for travelers. Its attributes include the POI name p_n , spatial details p_s (e.g., address, geocoordinates), temporal references p_t (e.g., opening hours, recommended visit time, expected visit duration), and POI’s semantic description p_d . The candidate POIs dependent on the query q are defined as $P^q = \{p_i^q\}_{i=1}^{|P^q|}$.

Trajectory. A tourist trajectory is extracted from the retrieved travel-related Web document, and is denoted as $t = (p_{n,1}, p_{n,2}, \dots, p_{n,|t|})$ consisting only POI names. The trajectories relevant to the query q are defined as $T^q = \{t_i^q\}_{i=1}^{|T^q|}$.

Problem 1. Travel Planning. Given a user query q , query-dependent POI candidates $P^q = \{p_i^q\}_{i=1}^{|P^q|}$, and query-relevant trajectories $T^q = \{t_i^q\}_{i=1}^{|T^q|}$, the travel plan $I = [(p_{n,1}^q, t_{s,1}, t_{e,1}), \dots, (p_{n,|I|}^q, t_{s,|I|}, t_{e,|I|})]$ is derived by LLM agents via $I = A(q, P^q, T^q)$, where t_s and t_e denote the scheduled start and end times of the POI visit, respectively.

3.2 Dataset Construction

As illustrated in Figure 2, the construction pipeline of our dataset comprises the following steps: (1) query generation; (2) POI collection; (3) trajectory collection; culminating in a (4) quality control procedure, which are detailed below.

Query Generation. Unlike previous works that rely on simulated queries, we sample the latest realistic queries from the data sessions in the Baidu search engine², one of China’s largest search engines. These queries are characterized by their conciseness, effectively expressing users’ tourist needs through brief statements, such as ‘plan a 3-day trip to Beijing’.

Based on the collected seed queries, we curate the LLM to establish a standard for query formulation, which is identified by several fundamental elements: (1) city name, (2) travel duration, and (3) personal constraints that reflect user preferences. For the destination city, we select the most popular cities in China, given their wealth of tourist resources and public travel narratives. Regarding travel duration, we divide our queries into short-, medium- and long-term categories based on the

²<https://www.baidu.com>

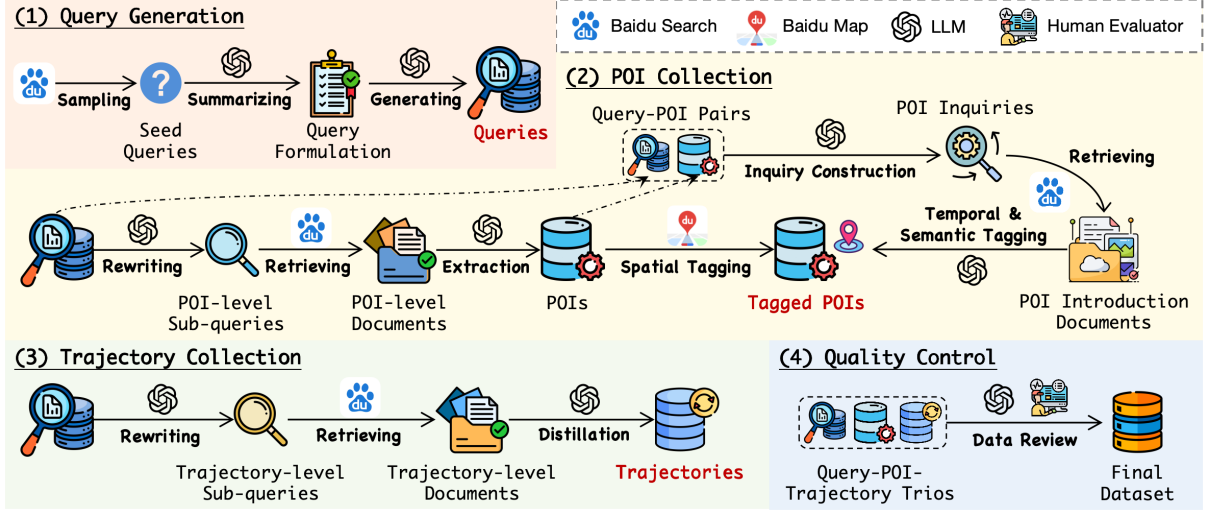


Figure 2: Dataset construction pipeline.

number of travel days. Depending on the presence of personalized constraints, our queries can also be categorized into generic and personal types. To ensure the diversity of personal queries, we further classify them by constraint types, encompassing seasonal preferences, holiday-specific requirements, POI category restrictions, traveler demographics, and trip compactness parameters. Refer to Appendix A.1 for more details of the query data. This formulation systematically tests LLM agent’s capabilities in handling different user focuses and planning horizons. In accordance with the standard, we engage the LLM to generate a practical and extensive query dataset.

POI Collection. To gather candidate attraction POIs and their associated spatiotemporal attributes for each query, we resort to LLMs augmented by search engines. Initially, we exploit a query rewriting strategy (Ma et al., 2023), whereby the LLM reformulates the original queries into POI-level sub-queries tailored for attraction recommendations. The Baidu search engine is then utilized to retrieve pertinent documents for each sub-query. Given that the documents generally contain a lot of irrelevant snippets, we apply the LLM to extract POI information exactly from each document.

Furthermore, to annotate fine-grained spatiotemporal information and contextual semantics for the extracted POIs, we combine two processes: (1) *Spatial Tagging*: We leverage the Baidu Map platform³ to enrich POIs with comprehensive spatial information including addresses and geocoordinates (i.e., latitude and longitude). This process also fa-

cilitates the standardization of POIs into unified names, thereby preventing noisy and hallucinated POIs. (2) *Temporal and Semantic Tagging*: For each query-POI pair (e.g., *Plan a spring trip in Beijing - the Great Wall*), we create query-tailored POI inquiries (e.g., *What’s the recommended travel time period of the Great Wall in spring?*) to search for related documents about this POI, representing the real-time needs (e.g., Spring trip). Subsequently, the retrieval-augmented LLM is employed to summarize nuanced temporal and semantic insights, entailing opening hours, recommended visit times, expected visit durations, and semantic POI descriptions of the POIs, which foster agents’ awareness of spatiotemporal coherence and POI distinctiveness in travel planning.

Trajectory Collection. To collect the trajectory-level knowledge, we first retrieve up-to-date documents that pertain to real-life travel experiences relevant to the given query. In order to address user privacy concerns, full documents are not disclosed in our dataset. Instead, we implement a LLM-based desensitization process that distills tourist trajectories from the original lengthy documents. Specifically, we prompt LLMs to directly extract the sequence of POIs from the document and filter private and irrelevant information, which avoids altering the original trajectory. To maintain the integrity of the data, we also instruct the LLM to refrain from answering if the document lacks plausible trajectory-level information. Additionally, to emulate the retrieval noise commonly encountered in real-world scenarios, we retain the disturbances present within these trajectories. As outlined in

³<https://lbsyun.baidu.com/>

the quality control stage below, we also prepare a denoised version of trajectory data for further in-depth analysis in Section 4.3.

Quality Control. To ensure the quality of the generated dataset, we employ a dual-evaluation approach involving both LLMs and human evaluators to systematically review and refine all *query-POI-trajectory* data instances. The process begins with LLM-based verification, where models are prompted to identify and correct factual inaccuracies and noisy information according to predefined criteria, such as (1) *whether the POIs and trajectories align with the query*; (2) *whether POI information is correct and non-repetitive*; (3) *whether the trajectory is logically complete and free of redundant POIs*. Subsequently, crowd-sourced annotators perform additional validation by cross-checking both the original instances and the LLM-refined records. This human evaluation is supplemented with edge-case analysis and iterative discussions between the authors and annotators to resolve ambiguities and ensure consistency. Furthermore, data instances lacking sufficient trajectory references are automatically filtered out. This comprehensive validation protocol ensures the reliability and feasibility of our retrieval-augmented spatiotemporal travel planning framework.

In conclusion, our dataset consists of 2,348 travel queries, including 115 generic queries and 2,233 personal queries. This dataset is associated with 85,575 attraction POIs (averaging 36.45 POIs per request) derived from 5,018 unique attractions, and 18,784 retrieved trajectories (averaging 8 trajectories per request). Throughout the data construction process, we utilize GPT-4o (OpenAI, 2024), and the prompts are presented in Appendix E.1.

3.3 Evaluation

Beyond commonsense constraints, our evaluation system illuminates the nuanced aspects concerning spatiotemporal rationality and the semantic prominence of POIs. Unlike TripCraft (Chaudhuri et al., 2025), we conceptualize spatiotemporal travel planning as a complex problem without unique optimal solutions, challenging the reliability of annotating ideal plans. Following existing works (Tang et al., 2024), our evaluation system integrates rule-based metrics alongside LLM-as-a-Judge techniques, emphasizing five critical evaluation dimensions: commonsense, spatial, temporal, POI semantic and query relevance. This approach effectively circumvents costly annotations and mitigates evaluation

biases, with metric descriptions provided below.

Commonsense. Commonsense metrics measure whether the generated plan adheres to basic validity standards, including: (1) *Failure Rate (FR)*: the percentage of legitimate POIs without hallucination; (2) *Repetition Rate (RR)*: the frequency of POI repetition within the plan.

Spatial. The spatial metric evaluates the route efficiency of the plan. Specifically, we use *Distance Margin Ratio (DMR)* to quantify the distance gap from the theoretically optimal route.

Temporal. Temporal metrics assess the rationality of the scheduled visit periods of POIs, which embrace (1) *Start Time Rationality (STR)*: whether the arranged arrival time for POI visit is appropriate; (2) *Duration Underflow Ratio (DUR)*: the extent to which the planned visit duration meets expectations; (3) *Time Buffer Ratio (TBR)*: the proportion of buffer time available throughout the plan, indicating the degree of tourist comfort.

POI Semantic. The semantic metric examines popularity and distinctiveness of the selected POIs. In particular, we design a *POI Popularity (PP)* metric to measure the recall rate within the retrieved attraction leaderboard.

Query Relevance. The relevance metrics focus on whether the user demands specified in the queries (e.g., time-sensitive desires) are fulfilled, concerning two aspects: (1) *POI Relevance (PR)*: the alignment between planned POIs and the user query; (2) *Time Schedule Relevance (TSR)*: the pertinence of the arranged POI visit period and personal needs.

We apply LLM-as-a-Judge for STR, PR and TSR, while the others are automatic metrics. In Appendix B, we further elaborate details about the metrics, and validate that LLM evaluators aligns with humans well. To provide a more transparent depiction of the overall effectiveness, we supplement five rank-based metrics: R_S , R_T , R_P , R_R and R_C , which respectively denote the performance rank of methods from spatial, temporal, POI semantic and query relevance dimensions, and a comprehensive view averaging all aspects.

4 Experiment

4.1 Baselines

Travel Planning Methods. We evaluate two categories of travel agents: (1) **Ground-up Travel Agents** which include *Direct*, *Chain of Thought (CoT)* (Wei et al., 2022), *Reflexion* (Shinn et al., 2023) and two multi-agent frameworks: *Multi-*

Table 1: Main results (%) of different methods on our dataset. The best strategies are marked in bold, while the second-best ones are underlined.

Method	FR↓	RR↓	DMR↓	DUR↓	TBR↑	STR↑	PP↑	PR↑	TSR↑	R_S ↓	R_T ↓	R_P ↓	R_R ↓	R_C ↓
GPT-4o														
Direct	0.32	0.00	67.92	3.03	22.01	77.22	50.82	80.51	<u>92.52</u>	6.00	7.67	7.00	3.00	5.92
CoT	<u>0.39</u>	<u>0.01</u>	69.4	2.78	22.13	76.96	50.09	79.92	91.99	8.00	8.00	10.00	6.50	8.12
Reflexion	0.67	0.34	73.35	3.84	21.46	77.3	50.52	80.34	92.71	10.00	8.67	8.00	<u>3.50</u>	7.54
MAC	1.40	0.72	66.11	3.67	24.07	75.41	46.38	81.76	89.04	3.00	7.00	11.00	6.00	6.75
MAD	0.68	0.07	74.95	3.68	20.67	77.14	50.52	79.53	92.4	11.00	9.67	8.00	6.50	8.79
RAG($M=8$)	2.05	<u>0.01</u>	68.37	2.57	<u>23.82</u>	77.19	<u>58.00</u>	<u>80.55</u>	91.58	7.00	4.67	<u>2.00</u>	5.00	4.67
RAG($M=4$)	2.08	0.02	67.75	2.55	23.71	77.35	56.11	80.53	91.67	5.00	3.67	4.00	4.50	<u>4.29</u>
RAG($M=1$)	2.43	0.04	<u>66.05</u>	2.47	22.59	77.44	53.38	80.1	91.5	<u>2.00</u>	4.00	6.00	8.50	5.12
RAG+Extr.($M=4$)	1.91	0.02	<u>66.99</u>	2.41	23.5	77.87	56.82	80.31	91.69	4.00	2.00	3.00	6.00	3.75
RAG+Extr.($M=1$)	2.72	0.06	65.76	<u>2.42</u>	22.95	<u>77.79</u>	53.85	80.39	91.64	1.00	<u>3.00</u>	5.00	6.00	3.75
RAG+Abst.	3.20	0.02	69.49	2.64	22.23	76.66	59.15	79.36	90.67	9.00	7.67	1.00	10.50	7.04
Qwen2.5-72B-Instruct														
Direct	0.38	0.03	71.67	6.49	24.82	78.33	48.53	79.68	92.86	8.00	<u>5.00</u>	8.00	9.00	7.50
CoT	<u>0.42</u>	<u>0.04</u>	70.51	6.64	24.66	78.77	47.09	80.12	93.16	7.00	5.33	10.00	8.00	7.58
Reflexion	2.39	1.62	85.38	8.08	25.37	77.54	49.15	79.65	92.14	10.00	7.67	7.00	10.00	8.67
MAC	1.10	2.21	70.08	5.74	23.65	76.48	43.3	<u>81.28</u>	90.00	6.00	7.33	11.00	6.50	7.71
MAD	3.30	1.49	87.47	9.46	26.53	77.75	47.49	80.23	91.21	11.00	7.00	9.00	9.00	9.00
RAG($M=8$)	3.41	0.11	69.39	<u>6.35</u>	23.48	78.4	<u>55.15</u>	81.67	93.29	5.00	6.00	<u>2.00</u>	<u>3.00</u>	4.00
RAG($M=4$)	3.15	0.11	68.77	6.54	24.06	79.08	53.62	81.26	93.86	3.00	<u>5.00</u>	4.00	2.00	3.50
RAG($M=1$)	2.58	0.09	68.07	6.89	25.27	78.48	51.62	81.07	93.4	1.00	5.33	6.00	4.50	4.21
RAG+Extr.($M=4$)	3.46	0.16	69.03	6.49	24.33	<u>79.03</u>	54.79	81.21	<u>93.81</u>	4.00	4.33	3.00	<u>3.00</u>	<u>3.58</u>
RAG+Extr.($M=1$)	3.16	0.20	<u>68.29</u>	6.73	<u>25.59</u>	78.33	52.36	80.92	93.42	<u>2.00</u>	<u>5.00</u>	5.00	4.50	4.12
RAG+Abst.	3.78	0.17	72.4	7.32	25.11	78.04	56.14	80.5	93.29	9.00	7.33	1.00	6.00	5.83
DeepSeek-R1														
Direct	0.55	0.01	68.78	3.07	22.94	76.68	50.27	80.68	91.7	7.00	4.67	7.00	6.50	6.29
RAG($M=8$)	2.31	0.03	65.31	3.94	<u>23.30</u>	<u>77.21</u>	<u>53.87</u>	82.47	92.66	1.00	<u>3.00</u>	<u>2.00</u>	2.00	2.00
RAG($M=4$)	1.62	0.03	66.76	3.78	23.37	76.95	53.02	<u>82.00</u>	92.45	5.00	2.67	4.00	4.00	3.92
RAG($M=1$)	<u>1.23</u>	0.04	66.42	<u>3.73</u>	22.97	77.37	51.82	81.54	<u>92.89</u>	3.00	<u>3.00</u>	6.00	3.50	3.88
RAG+Extr.($M=4$)	2.00	0.03	66.45	4.04	23.28	76.94	53.69	<u>82.00</u>	92.66	4.00	4.33	3.00	<u>2.50</u>	3.46
RAG+Extr.($M=1$)	1.25	0.03	<u>66.19</u>	3.76	23.23	76.94	51.95	81.66	93.02	<u>2.00</u>	3.67	5.00	<u>2.50</u>	<u>3.29</u>
RAG+Abst.	2.04	<u>0.02</u>	67.57	4.96	23.16	76.63	54.28	80.5	92.6	6.00	6.33	1.00	6.00	4.83

Agent Collaboration (MAC) which applies a divide-and-conquer solution (Zhang et al., 2025), and *Multi-Agent Debate (MAD)* that facilitates a discussion session for plan refinement (Ni et al., 2024); (2) **Retrieval-augmented Travel Agents** implemented by the RAG strategy utilizing trajectory-level knowledge. $RAG(M=m)$ denotes the retrieval-augmented method using m trajectories. To test agents’ capabilities for explicit context utilization, we consider two simple post-retrieval techniques (Xu et al., 2024): extractive compression method $RAG+Extr.(M=m)$ and abstractive compression method $RAG+Abst.$, which intentionally purify the retrieved content for enhancements.

Base Models. The core of agent-based planning methods lies in the LLM. Therefore, we evaluate various advanced LLMs including GPT-4o (OpenAI, 2024), Qwen2.5-72B-Instruct (Yang et al., 2024), LLaMA3.3-70B-Instruct (AI, 2024), as well as DeepSeek-R1 (Guo et al., 2025). More details of the baselines and evaluation setups are provided in Appendix C.1.

4.2 Main Results

In this section, we discuss the performances of various methods and models on our benchmark as presented in Table 1. Due to the space limit, we leave the results on LLaMA3.3 in Appendix C.2. Our critical observations summarized below:

Advanced LLM agents struggle with spatiotemporal travel planning. Cutting-edge strategies (i.e., CoT, Reflexion and multi-agent techniques) underperform direct prompting in holistic ranking R_C , revealing complex task decomposition or answer reflection can lead to error accumulation and degeneration of spatiotemporal reasoning.

Trajectory knowledge holds potential for enhanced travel planning. According to R_C , retrieval-augmented planning methods generally outperform ground-up ones, highlighting the value of trajectory knowledge. The performance gain primarily stems from spatial and POI semantic dimensions, while some models (e.g., GPT-4o) exhibit slight declines in temporal and relevance aspects, which can be attributed to agents’ limited capacity

to resolve confusing and verbose contexts.

Sophisticated methods compromise agents’ commonsense awareness. A notable increase of inaccessible or repetitive POIs is observed within both advanced LLM agents and retrieval-augmented ones, suggesting that overly complex methods tend to perplex LLM agents since long-context and complicated inputs.

Reasoning-optimized LLMs are not as desired in travel planning. Despite architectural advances, specialized reasoning models like Deepseek-R1 fail to show a remarkable advantage over other foundational models in spatiotemporal travel planning, even lagging in the temporal dimension.

Retrieval-augmented planning methods lack stability. Knowledge richness and post-processing techniques show inconsistent benefits across different base models, indicating context sensitivity. We leave the in-depth analysis in Section 4.3.

Performances in distinct metrics are inconsistent. No single solution dominates all evaluation dimensions. Trade-offs between different metrics are commonly observed in most cases, illustrating the complexity of our multi-objective spatiotemporal travel planning task.

Refer to Appendix C.3 for results on sub-datasets across various query categories, which generally align with our full-scale experiments in Table 1.

4.3 In-depth Analysis

In this section, we conduct a thorough examination of retrieval-augmented methods in terms of universality, planning mechanisms and robustness. We implement our analysis experiments using Qwen2.5-72B-Instruct as an example.

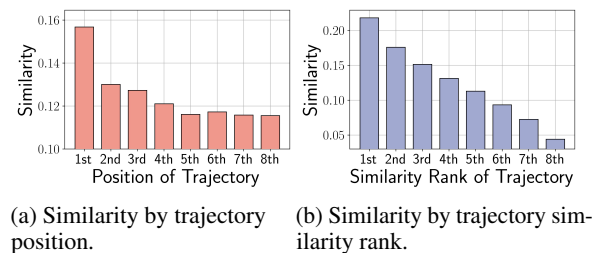


Figure 3: Similarities between plans and trajectories.

Universality Analysis. To explore whether the retrieved knowledge is always necessary, we calculate the win rates of retrieval-augmented methods compared to the Direct baseline, achieving an average of 87.41% across seven spatiotemporal metrics. This demonstrates that trajectory references are not universally efficacious, as evidenced by a 12.59%

failure gap. Refer to Appendix C.4 for the detailed statistics of win rates.

Utilization Analysis. To delve into how LLM agents utilize trajectory knowledge, we employ the similarity $\sigma(I, t)$ between the plan I generated by RAG($M=8$) and each associated trajectory t , as a proxy indicating the extent of LLMs’ utilization of the trajectory. Specifically, the similarity function $\sigma(I, t) = \beta \cdot \sigma_{\text{POI}}(I, t) + (1 - \beta) \cdot \sigma_{\text{order}}(I, t)$ implies the extent of utilization for each trajectory. σ_{POI} denotes the Jaccard similarity of the POIs contained in the plan and trajectory, while σ_{order} is the Kendall Tau coefficient between the ranks of common POIs in the plan and trajectory. β controls the balance between similarities of POI sets and POI order. Figure 3a unravels the similarities between the plan and trajectories at different positions within the prompt context, indicating a preference for information at the beginning, which is more relevant to the travel query. When we reorder the distribution by descending reference similarity, the results in Figure 3b elucidate that LLM agents tend to selectively use several references rather than assimilate all trajectories.

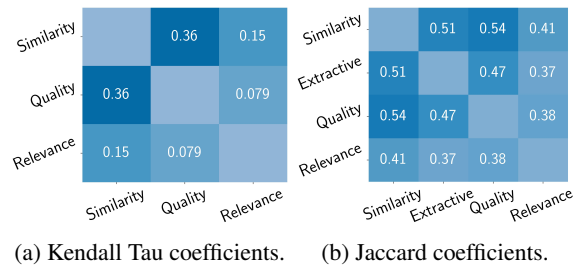


Figure 4: Correlation analysis.

Furthermore, to probe why retrieval-augmented methods have positive effects, we analyze the ordinal correlations among three variables: (1) extent of utilization for the trajectory (*i.e.*, similarity σ), (2) quality of the trajectory, (3) query relevance of the trajectory. To represent the "quality" variable of the methods, we use POI semantic metric as a proxy since it shows the greatest improvements according to Table 1, which facilitates exploration into the mechanisms behind the superiority of retrieval-augmented methods. Figure 4a reveals a significant consistency between similarities and quality, demonstrating LLM agents’ abilities to identify high-quality knowledge for travel planning beyond query relevance.

Besides, we contrast implicit and explicit extractive utilization methods (*i.e.*, RAG($M=8$) and

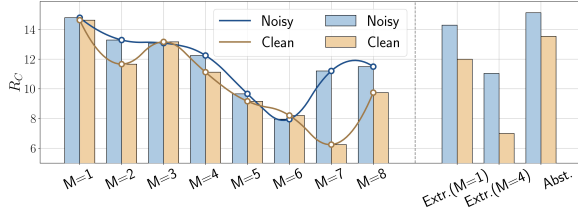


Figure 5: The sensitivity analysis of retrieval-augmented methods with different retrieval quantity, based on noisy and clean trajectory knowledge.

RAG+Extr.($M=4$) in Figure 4b. Since the explicit extractive method RAG + Extr. ($M=4$) directly determines the selected trajectory references, where the utilization indicator like similarity is inaccessible. Thus we select the top-4 trajectory set according to the ranks of three variables: similarity (*i.e.*, the extent of trajectory utilization for the implicit method RAG($M=8$)), trajectory quality, and trajectory relevance, so as to contrast them with the set extracted by RAG + Extr. ($M=4$). We discern that these two extractive ways are distinct in reference utilization with only 0.51 Jaccard consistency, while the implicit method gains a slight edge in quality alignment (*i.e.*, 0.54 versus 0.47).

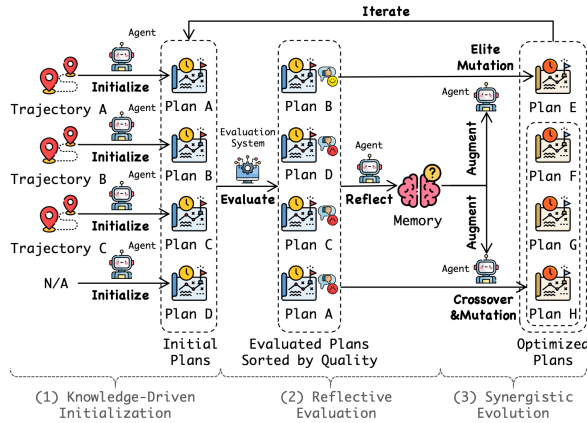


Figure 6: The workflow of EvoRAG.

Sensitivity Analysis. To investigate the sensitivity of retrieval-augmented planning methods with respect to the quantity and quality of retrieval data, we test retrieval-augmented strategies with varying number of trajectory references, and compare the results of using noisy and denoised trajectories (as mentioned in Section 3.2). The analysis results in Figure 5 depict that the R_C performance reaches its peak by integrating 6 or 7 trajectories. Reducing the volume of knowledge (*i.e.*, $M < 6$) leads to a significant decline in efficacy, uncovering the necessity of diverse reference knowledge, and confirming the non-uniqueness of the travel planning problem. Conversely, excessive knowledge (*i.e.*,

$M = 8$) also undermines the effectiveness, resulting in difficulties assimilating diverging references and filtering out low-quality insights, despite their relevance to user queries. Moreover, the performance comparison between noisy and clean (*i.e.*, denoised) trajectories highlights the negative impact of the POI noise on the spatiotemporal validity, particularly for post-processing methods.

Takeaways. (1) LLM agents effectively enhance the spatiotemporal rationality of travel planning aided by retrieved trajectories. (2) LLM agents utilize trajectories extractively, in a manner that aligns with reference quality. (3) Retrieval-augmented agents face challenges in uniformly promoting spatiotemporal travel planning across all queries and evaluation metrics, and robustly integrating conflicting and noisy references.

5 EvoRAG

Method. To address the aforementioned issues, we propose *EvoRAG*, a knowledge-evolution optimization framework, as illustrated in Figure 6. It includes three procedures: (1) *knowledge-driven initialization*; (2) *reflective evaluation*; and (3) *synergistic evolution*. Based on initialization, the plans are iteratively optimized by alternating cycles of evaluation and evolution. We elaborate the algorithmic workflow *EvoRAG* in Algorithm 1.

(1) *Knowledge-Driven Initialization.* To incorporate divergent retrievable knowledge, we curate LLM agents to generate initial plans, individually based on $|T|$ tourist trajectories. These plans are independent and poised for evolutionary optimization, bypassing the deficiencies of LLM agents in assimilating discordant references. Additionally, we include an initial plan derived solely from LLM agents' intrinsic knowledge (implemented by the Direct baseline) to counter the limited universality of retrieval-augmented methods.

(2) *Reflective Evaluation.* In each iteration, to ensure that LLM agents can comprehend the optimization objectives, we evaluate the quality of plans across various metrics (detailed in Section 3.3). To further facilitate LLM agents in discerning how to improve solutions, we encourage them to deliberately analyze the evaluation results of different plans and reflect on their strengths and weaknesses. This self-aware reflection process is managed by a memory module, which is iteratively updated to maintain the expertise of LLM agents learned from optimization and evaluation experiences.

Algorithm 1: The workflow of EvoRAG.

Input: query q , POI candidates P^q , trajectories T^q ; LLM agents $A_p, A_r, A_u^{mo}, A_u^{cm}$ respectively for plan initialization, evaluation reflection, mutation-only plan updating and crossover-mutation plan updating, evaluation function \mathcal{E} ; maximum number of optimization iterations G , mutation-only ratio α ;

Output: The optimal plan I^* .

Initialize population $\mathcal{I}_0 = \{I_{(0,i)}\}_{i=0}^{|T^q|}$ via $I_{(0,0)} = A_p(q, P^q)$, $I_{(0,i)} = \{A_p(q, P^q, t_i^q)\}$, $i = 1, 2, \dots, |T^q|$;

Initialize planning reflection memory R_0 ;

Initialize $g \leftarrow 0$;

while $g < G$ **do**

1. Evaluate and rank the plans $E_g = [(I_{(g,i)}^o, e_{(g,i)})]_{i=0}^{|T^q|} = \mathcal{E}(\{I_{(g,i)}\}_{i=0}^{|T^q|})$ in the descending order of efficacy, where e denotes the evaluation details;
2. Reflect about the evaluation results and update the memory $R_{g+1} = A_r(R_g, E_g)$;
3. Perform mutation for the top α proportion of population and generate
 $\mathcal{I}_{g+1,a} = \{I_{(g+1,i)}\}_{i=0}^{\alpha \cdot |T^q|} = A_u^{mo}(q, P^q, R_{g+1}, [(I_{(g,i)}^o, e_{(g,i)})]_{i=0}^{\alpha \cdot |T^q|})$;
4. Perform crossover and mutation for population and generate
 $\mathcal{I}_{g+1,b} = \{I_{(g+1,i)}\}_{i=0}^{(1-\alpha) \cdot |T^q|} = A_u^{cm}(q, P^q, R_{g+1}, E_g)$;
5. $\mathcal{I}_{g+1} = \text{Union}(\mathcal{I}_{g+1,a}, \mathcal{I}_{g+1,b})$;
6. $g = g + 1$;

end

Evaluate and rank the plans $E_G = [(I_{(G,i)}^o, e_{(G,i)})]_{i=0}^{|T^q|} = \mathcal{E}(\{I_{(G,i)}\}_{i=0}^{|T^q|})$ in the descending order of efficacy;

Evaluate and rank the best plans from all the iterations $E^* = [(s_i^*, e_i^*)]_{i=0}^G = \mathcal{E}(\{I_{(i,0)}^o\}_{i=0}^G)$;

return $I^* = I_0^*$.

(3) *Synergistic Evolution.* Based on evaluation feedback and reflective memory, the best α proportion of plans are retained for optimization by LLM agents, termed elite mutation. Furthermore, to unify the advantages of trajectory knowledge from distinct perspectives, we selectively synthesize the plans (*i.e.*, crossover), avoiding knowledge isolation stemming from the separate initialization. Specifically, we repeatedly select dissimilar plans to perform crossover and mutation to ensure diversity, until we obtain $(|T| + 1)$ solutions.

The framework enables the LLM agent to focus on plan optimization utilizing multifarious external and internal knowledge, while becoming more robust to noisy and nonsensical information. The prompts used are shown in Appendix E.4.

ground-up and retrieval-augmented planning methods across almost all dimensions (except for query relevance R_R), as illustrated in Figure 7. *EvoRAG* also achieves notable reductions in commonsense failures with a 0.4% POI failure rate (FR) and a 0.06% POI repetition rate (RR), which are comparable to the *Direct* baseline. The detailed setups and complete experimental results are provided in Appendix D. This further underscores the proficiency of integrating trajectory-level knowledge with LLM-based optimization, paving the way for future advancements in LLM-driven travel agents.

6 Conclusion

This work investigates the role of online knowledge in improving LLM agents for spatiotemporal travel planning. We introduce *TP-RAG*, a benchmark that integrates spatiotemporal POI characteristics and trajectory-level Web knowledge. Experiments across advanced LLMs demonstrate that retrieval-augmented methods generally improve the fine-grained quality of plans, but they are not universally effective or robust. Our proposed *EvoRAG* framework counters these issues through evolutionary optimization, achieving state-of-the-art results by balancing divergent knowledge. This sets the stage for developing more powerful travel agents with exceptional spatiotemporal awareness.

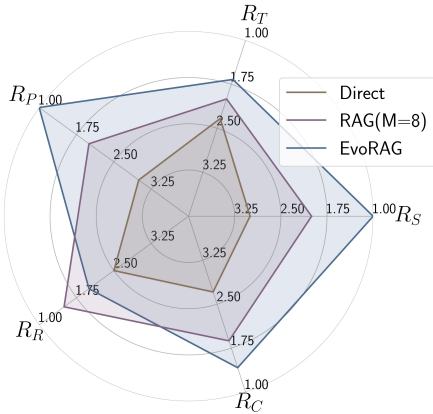


Figure 7: Comparison of EvoRAG and baselines.

Comparison Results. Based on Qwen2.5-72B-Instruct model, *EvoRAG* generally surpasses both

Limitations

Planning Setup. The proposed *TP-RAG* aims to examine the capabilities of LLM agents in spatiotemporal travel planning by utilizing online knowledge. Therefore, we specifically focus on attraction planning without the use of tools for agents. We believe that our benchmark can be expanded to include realistic planning scenarios encompassing meals, accommodations, and transportation, and enabling adaptive, tool-based information retrieval.

Query Scenario. Our dataset is limited to search scenarios which feature concise user queries. To address more complex queries, a viable approach involves decomposing the query and sourcing relevant online information for each segment, which we plan to explore in future work.

Data Source. Our dataset is built using the Baidu search engine, with a focus on sourcing documents in Chinese. Though there may be some regional biases due to limitations in Chinese cities, our construction pipeline, backed by advanced search engines, can be adapted for other regions globally. This adaptability contributes to opportunities of creating a more comprehensive dataset.

Evaluation. In our paper, we conceptualize spatiotemporal travel planning as a multi-objective optimization problem. However, the intricate nature of these objectives complicates the calculation of Pareto fronts, which serve as the golden reference plan for our task. Furthermore, the lack of reliable ground truths precludes the consideration of fine-tuning strategies related to RAG or post-retrieval compression. Future research may focus on developing methods to annotate trustworthy ground truths for this task, ensuring that evaluations are free from multifaceted biases.

Baseline. Due to the challenges in collecting ground-truth information, we have opted not to consider training-based baseline methods in our current work. Although exploring the potential of training a travel agent is indeed valuable, developing a reliable specialist model for the complex spatiotemporal travel planning methods is something we leave for future research.

Ethical Statement

Artifacts Licensing and Compliance.

(1) *Used Artifacts.* Our dataset is constructed from publicly available content obtained via Baidu Search and Baidu Map under their data usage policies. For proprietary data such as queries and web

documents, we have ensured compliance through academic use authorization. Base LLM models were accessed via HuggingFace under their respective licenses; GPT-4o was used under OpenAI’s terms. Code resources such as the TSP solver are used in accordance with their original licenses. All artifacts are used strictly for research within permitted scopes.

(2) *Released Artifacts.* Our data, aggregated from Baidu and anonymized to prevent re-identification, is released for non-commercial research under CC BY-NC-SA 4.0. Any use beyond academic research is prohibited.

Anonymization and Privacy Protection. Trajectories and POI information are automatically extracted using LLMs to remove personal identifiers and sensitive content. We further conduct manual verification to ensure complete anonymization and eliminate any offensive material. All released data are securely processed and stripped of identifiable metadata, preventing any possibility of re-identification.

Biases and Ethical Considerations. Despite these efforts, we acknowledge that data collection and processing methods may introduce biases, particularly concerning the representativeness of marginalized or vulnerable populations. Users are cautioned that the dataset should only be used in research and experimental settings, with careful consideration of these limitations and potential ethical risks.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62572417, No.92370204), National Key R&D Program of China (Grant No.2023YFF0725004), the Guangzhou Basic and Applied Basic Research Program under Grant No. 2024A04J3279, Education Bureau of Guangzhou Municipality.

References

- Meta AI. 2024. [Llama-3.3-70b model](#).
- Ashmi Banerjee, Adithi Satish, and Wolfgang Wörndl. 2024. Enhancing tourism recommender systems for sustainable city trips using retrieval-augmented generation. *arXiv preprint arXiv:2409.18003*.
- Soumyabrata Chaudhuri, Pranav Purkar, Ritwik Raghav, Shubhojit Mallick, Manish Gupta, Abhik Jana, and Shreya Ghosh. 2025. Tripcraft: A benchmark for spatio-temporally fine grained travel planning. *arXiv preprint arXiv:2502.20508*.

- Aili Chen, Xuyang Ge, Ziquan Fu, Yanghua Xiao, and Jiangjie Chen. 2024a. Travelagent: An ai assistant for personalized travel planning. *arXiv preprint arXiv:2409.08069*.
- Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024b. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.
- Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2024. Timebench: A comprehensive evaluation of temporal reasoning abilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1204–1228.
- Tomas de la Rosa, Sriram Gopalakrishnan, Alberto Pozanco, Zhen Zeng, and Daniel Borrajo. 2024. Tripal: Travel planning with guarantees by combining large language models and automated planners. *arXiv preprint arXiv:2406.10196*.
- Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Yilun Hao, Yongchao Chen, Yang Zhang, and Chuchu Fan. 2024. Large language models can plan your travels rigorously with formal verification tools. *arXiv e-prints*, pages arXiv–2404.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.
- Da Ju, Song Jiang, Andrew Cohen, Aaron Foss, Sasha Mitts, Arman Zharmagambetov, Brandon Amos, Xian Li, Justine Kao, Maryam Fazel-Zarandi, and 1 others. 2024. 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*, pages 240–249.
- Kuang-Huei Lee, Ian Fischer, Yueh-Hua Wu, Dave Marwood, Shumeet Baluja, Dale Schuurmans, and Xinyun Chen. 2025. Evolving deeper llm thinking. *arXiv preprint arXiv:2501.09891*.
- Fangjun Li, David C Hogg, and Anthony G Cohn. 2024. Reframing spatial reasoning evaluation in language models: a real-world simulation benchmark for qualitative reasoning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 6342–6349.
- Yuchen Li, Hengyi Cai, Rui Kong, Xinran Chen, Jiamin Chen, Jun Yang, Haojie Zhang, Jiayi Li, Jiayi Wu, Yiqun Chen, and 1 others. 2025. Towards ai search paradigm. *arXiv preprint arXiv:2506.17188*.
- Fan Liu, Jindong Han, Tengfei Lyu, Weijia Zhang, Zhe-Rui Yang, Lu Dai, Cancheng Liu, and Hao Liu. 2025a. Foundation models for scientific discovery: From paradigm enhancement to paradigm transition. *Authorea Preprints*.
- Fan Liu, Zherui Yang, Cancheng Liu, Tianrui Song, Xiaofeng Gao, and Hao Liu. 2025b. Mm-agent: Llm as agents for real-world mathematical modeling problem. *arXiv preprint arXiv:2505.14148*.
- Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. 2025. Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *ACM Transactions on Information Systems*, 43(2):1–32.
- Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.
- Rohin Manvi, Samar Khanna, Gengchen Mai, Marshall Burke, David B Lobell, and Stefano Ermon. 2024. Geollm: Extracting geospatial knowledge from large language models. In *ICLR*.
- Hang Ni, Yuzhi Wang, and Hao Liu. 2024. Planning, living and judging: A multi-agent llm-based framework for cyclical urban planning. *arXiv preprint arXiv:2412.20505*.
- OpenAI. 2024. [Gpt-4o: Openai’s multimodal language model](#). Accessed: 2025-03-24.
- Nicholas Pipitone and Ghita Houir Alami. 2024. Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain. *arXiv preprint arXiv:2408.10343*.
- Jinhu Qi, Shuai Yan, Yibo Zhang, Wentao Zhang, Rong Jin, Yuwei Hu, and Ke Wang. 2024a. Rag-optimized tibetan tourism llms: Enhancing accuracy and personalization. *arXiv preprint arXiv:2408.12003*.

- Zehan Qi, Rongwu Xu, Zhijiang Guo, Cunxiang Wang, Hao Zhang, and Wei Xu. 2024b. Long2rag: Evaluating long-context & long-form retrieval-augmented generation with key point recall. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4852–4872.
- Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. Ares: An automated evaluation framework for retrieval-augmented generation systems. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 338–354.
- Jie-Jing Shao, Xiao-Wen Yang, Bo-Wen Zhang, Baizhi Chen, Wen-Da Wei, Lan-Zhe Guo, and Yu-feng Li. 2024. Chinatravel: A real-world benchmark for language agents in chinese travel planning. *arXiv preprint arXiv:2412.13682*.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.
- Harmanpreet Singh, Nikhil Verma, Yixiao Wang, Manasa Bharadwaj, Homa Fashandi, Kevin Ferreira, and Chul Lee. 2024. 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*, pages 486–514.
- Sihan Song, Chuncheng Yang, Li Xu, Haibin Shang, Zhuo Li, and Yinghui Chang. 2024. Travelrag: A tourist attraction retrieval framework based on multi-layer knowledge graph. *ISPRS International Journal of Geo-Information*, 13(11):414.
- Yihong Tang, Zhaokai Wang, Ao Qu, Yihao Yan, Zhaofeng Wu, Dingyi Zhuang, Jushi Kai, Kebing Hou, Xiaotong Guo, Jinhua Zhao, and 1 others. 2024. 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*, pages 1413–1432.
- Shuting Wang, Jiejun Tan, Zhicheng Dou, and Ji-Rong Wen. 2024. Omnieval: An omnidirectional and automatic rag evaluation benchmark in financial domain. *arXiv preprint arXiv:2412.13018*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- IpKin Anthony Wong, Qi Lilith Lian, and Danni Sun. 2023. Autonomous travel decision-making: An early glimpse into chatgpt and generative ai. *Journal of Hospitality and Tourism Management*, 56:253–263.
- Chengxing Xie and Difan Zou. 2024. A human-like reasoning framework for multi-phases planning task with large language models. *arXiv preprint arXiv:2405.18208*.
- Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. Travelplanner: A benchmark for real-world planning with language agents. In *International Conference on Machine Learning*, pages 54590–54613. PMLR.
- Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6233–6251.
- Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. Re-comp: Improving retrieval-augmented lms with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- Dazhou Yu, Riyang Bao, Gengchen Mai, and Liang Zhao. 2025. Spatial-rag: Spatial retrieval augmented generation for real-world spatial reasoning questions. *arXiv preprint arXiv:2502.18470*.
- Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Dongsheng Li, and Deqing Yang. 2024. Evoagent: Towards automatic multi-agent generation via evolutionary algorithms. In *NeurIPS 2024 Workshop on Open-World Agents*.
- Cong Zhang, Xin Deik Goh, Dexun Li, Hao Zhang, and Yong Liu. 2025. Planning with multi-constraints via collaborative language agents. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10054–10082.
- Xuan Zhang, Yang Deng, Zifeng Ren, See Kiong Ng, and Tat-Seng Chua. 2024. Ask-before-plan: Proactive language agents for real-world planning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10836–10863.

A Dataset Details

In this section, we present further details of our dataset. It is noteworthy that our original dataset was developed in Chinese based on the Baidu search engine, and we will also provide an English version of all queries, POI data, and trajectories to facilitate global research.

Query Category	Constraint Category	#Query
Generic	-	115
Personal	Season	425
	Holiday	608
	POI Category	634
	Traveler Category	451
	Trip Compactness	115
	Total	2233
Total	-	2348

Table 2: Query distribution according to constraints.

A.1 Query Data

We select the most popular Chinese travel cities to construct our query dataset. In specific, we refer to [Sohu Travel 2022’s rankings](#), and select the top 30 cities in mainland China. Below is the city list: *Chongqing, Wuhan, Sanya, Luoyang, Beijing, Nanjing, Shanghai, Xi’an, Qingdao, Guiyang, Fuzhou, Xiamen, Hangzhou, Shaoxing, Guilin, Jinan, Zhaoqing, Foshan, Chengdu, Changchun, Suzhou, Rizhao, Yantai, Huangshan, Yangzhou, Zhangjiajie, Guangzhou, Nanning, Jilin, Binzhou.*

In Table 3, we detail the personalized constraints according to their categories. And we report the query data distribution in terms of the constraint type In Table 2. To diversify the travel duration, we generate queries specified with 3, 4, and 5 days.

A.2 POI and Trajectory Data

For each query, we retrieve 10, 5, 20 documents for POI collection, real-time POI refinement, and trajectory collection, respectively. We associate 8 valid trajectories for each query and abandon instances that are insufficient in number. To emulate the retrieval-augmented planning scenario for each query independently, it is essential that the POIs featured in the retrieved trajectory should be included in the candidate set. Thus, we locate the POIs that are newly present in the trajectories and add them to the candidate POI set.

The construction of the overall dataset costs around 1 billion GPT-4o tokens.

B Metric Details

In this section, we detail each evaluation metric as follows:

- **Failure Rate (FR):** This metric quantifies the number of attractions absent from the candidate set, which may indicate hallucinations by the LLM agents.
- **Repetition Rate (RR):** We measure the frequency of POI repetition in plans to assess basic commonsense awareness of the agents.
- **Distance Margin Ratio (DMR):** This metric evaluates the margin ratio between the total distance required to transfer between attractions in the generated plan and the optimal distance determined by the Traveling Salesman Problem (TSP) solver⁴.
- **Start Time Rationality (STR):** It is essential to determine whether the scheduled times for visiting attractions are appropriate. Due to the lack of uniform standard, we prompt LLMs to consult POI-level temporal information and verify the plausibility of scheduled arrival times by binary judgment (*i.e.*, yes or no), then calculating the total acceptable rate.
- **Duration Underflow Ratio (DUR):** This metric assesses how well the planned visit durations align with the expected time spans for each attraction. We directly used the visiting duration tags of POIs in our dataset, and then compute the average duration underflow ratio.
- **Time Buffer Ratio (TBR):** Given that overly tight schedules are generally inadvisable, this metric evaluates the flexibility of plans by estimating the proportion of buffer time available between attractions throughout the plan.
- **POI Popularity (PP):** Popular attractions are typically favored, thus, we curate LLMs to offer a golden popularity ranking based on the retrieved attraction leaderboard data. We then calculate the top M recall, where M represents the number of selected POIs in the plan.
- **POI Relevance (PR):** To check the alignment of the delivered plans with the soft constraints in personal queries, we nudge LLMs to judge the matchness of each POI in a binary manner.
- **Time Schedule Relevance (TSR):** Similarly, LLMs are elicited to gauge whether the time intervals arranged for attractions are consistent with the personalized requirements.

⁴<https://github.com/fillipe-gsm/python-tsp>

Constraint Category	Constraints
Season	Spring, Summer, Autumn, Winter
Holiday	Spring Festival, Qingming Festival, Labor Day, Gragon Boat Festival, Mid-Autumn Festival, National Day
POI Category	Natural Landscapes, Historical & Cultural Heritage, Leisure & Recreation Areas, Art & Technology Hubs, City Sightseeing, Religious & Spiritual Sites
Traveler Category	Senior, Single, Couple, Parent-child
Trip Compactness	Special Forces-style

Table 3: Constraint taxonomy.

Dimension	Metric	Description
Commonsense	FR	Legitimacy of POIs
	RR	Non-redundancy of POIs.
Spatial	DMR	Route efficiency.
Temporal	STR	Rationality of arrival time.
	DUR	Rationality of visit duration.
	TBR	Comfort level of schedule.
POI Semantic	PP	POI popularity.
Query Relevance	PR	POI relevance.
	TSR	Time schedule relevance.

Table 4: Metric taxonomy.

A summary of the proposed evaluation metrics is presented in Table 4. The prompts for LLM-based evaluation are showed in Appendix E.2. Moreover, we conduct human evaluation to validate the proficiency of LLM-based evaluation, provided with the same instructions as those for LLMs. We randomly sample 100 plans generated by GPT-4o, and test the performance of Qwen2.5-72B-Instruct in evaluating results across three LLM-based metrics: STR, PR, and TSR. For each metric, we report three measurements: (1) agreement rate between the judgments of LLMs and humans; (2) Kendall Tau and (3) Spearman coefficient of the method rankings across all sampled queries. Table 5 demonstrates the alignment between LLM and human evaluators on our metrics.

Metric	Agree. Rate	Kendall Tau	Spearman Coeff.
STR	93.70	60.24	66.61
PR	95.54	68.07	71.72
TSR	97.55	74.34	77.66

Table 5: The alignment performance between LLM and human evaluators across three metrics (%).

C Experiments

C.1 Experimental Setup

In this section, we provide some details of our experiments. For stability, we set the temperature as 0 for all base models. For the cases that agents

fail to generate a grammatically correct answer, we retry several times until a success, because the delivery failure is not considered in our evaluation system. Since the strong reasoning ability of DeepSeek-R1, we omit the implementation of complex ground-up planning strategies (*i.e.*, CoT, Reflexion, MAC, MAD) on the model. The prompts designed for these baseline methods are presented in Appendix E.3. To assess our baselines, we adopt Qwen2.5-72B-Instruct as the LLM evaluator. Our experiments are conducted on A800-80G GPUs, totaling over 5,000 GPU hours (except for the GPT-4o experiments, which were performed using OpenAI’s APIs). Due to the high computational cost and large number of experiments, we only conducted a single run.

C.2 Results on LLaMA3.3-70B-Instruct

We present the experimental results implemented by LLaMA3.3-70B-Instruct in Table 6, which are generally consistent with the outcomes in Table 1.

C.3 Results on Different Query Categories

We report our benchmark results on separate query data via radar charts, as illustrated in Figure 8, Figure 9, Figure 10 and Figure 11. The analysis reveals that retrieval-augmented strategies significantly outperform ground-up methods, such as direct prompting, in most evaluation metrics. This observation aligns with the findings of our full-scale experiments detailed in Section 4.2.

C.4 Universality Analysis

We detail the statistics of win rates of retrieved-augmented planning methods over the Direct baseline in Table 8, which is implemented with Qwen2.5-72B-Instruct model.

D Methodology

In this section, we elaborate the experimental details. For efficiency, we set G as 1. The evaluate

Method	FR↓	RR↓	DMR↓	DUR↓	TBR↑	STR↑	PP↑	PR↑	TSR↑	R_S ↓	R_T ↓	R_P ↓	R_R ↓	R_C ↓
LLaMA3.3-70B-Instruct														
Direct	0.67	0.01	90.55	7.41	24.14	<u>77.07</u>	46.27	79.75	89.32	8.00	<u>4.67</u>	7.00	7.50	6.79
CoT	<u>0.97</u>	0.01	86.45	7.77	24.76	78.10	45.52	80.31	87.97	7.00	4.33	10.00	6.00	6.83
Reflection	1.37	0.22	94.71	8.19	26.40	76.73	45.73	<u>80.28</u>	<u>89.96</u>	10.00	7.00	9.00	2.00	7.00
MAC	3.55	3.73	97.68	9.61	31.46	75.47	42.11	79.98	88.38	11.00	7.67	11.00	7.00	9.17
MAD	1.37	0.16	91.77	7.81	<u>26.42</u>	76.26	46.27	79.05	90.14	9.00	7.00	7.00	6.00	7.25
RAG($M=8$)	3.95	0.08	84.92	5.99	20.80	76.94	56.00	79.91	89.09	5.00	5.33	1.00	7.00	4.58
RAG($M=4$)	2.81	0.07	83.67	6.10	21.02	76.77	53.60	79.99	89.68	4.00	6.33	4.00	<u>3.00</u>	<u>4.33</u>
RAG($M=1$)	2.53	<u>0.06</u>	85.36	6.56	22.68	77.00	50.43	79.79	89.55	6.00	<u>4.67</u>	6.00	5.50	5.54
RAG+Extr.($M=4$)	3.33	0.09	83.60	<u>6.05</u>	21.06	76.70	54.66	79.73	89.30	3.00	6.67	<u>2.00</u>	8.50	5.04
RAG+Extr.($M=1$)	3.18	0.09	<u>82.42</u>	6.62	22.57	76.77	52.23	79.83	89.38	<u>2.00</u>	6.67	5.00	6.00	4.92
RAG+Abst.	3.97	0.22	81.96	6.60	22.91	76.86	<u>54.44</u>	79.36	89.42	1.00	5.33	3.00	7.50	4.21

Table 6: Main results (%) of LLaMA3.3-70B-Instruct on our dataset.

Method	FR↓	RR↓	DMR↓	DUR↓	TBR↑	STR↑	PP↑	PR↑	TSR↑	R_S ↓	R_T ↓	R_P ↓	R_R ↓	R_C ↓
Direct	0.38	0.03	71.67	6.49	24.82	78.33	48.53	79.68	92.86	9.00	5.67	9.00	9.50	8.29
CoT	0.42	<u>0.04</u>	70.51	6.64	24.66	78.77	47.09	80.12	93.16	8.00	6.33	11.00	8.50	8.46
Reflection	2.39	1.62	85.38	8.08	25.37	77.54	49.15	79.65	92.14	11.00	8.67	8.00	11.00	9.67
MAC	1.10	2.21	70.08	5.74	23.65	76.48	43.3	81.28	90.00	7.00	8.00	12.00	7.50	8.62
MAD	3.30	1.49	87.47	9.46	<u>26.53</u>	77.75	47.49	80.23	91.21	12.00	8.00	10.00	10.00	10.00
RAG($M=8$)	3.41	0.11	69.39	<u>6.35</u>	23.48	78.40	55.15	<u>81.67</u>	93.29	6.00	6.67	3.00	<u>3.50</u>	4.79
RAG($M=4$)	3.15	0.11	68.77	6.54	24.06	<u>79.08</u>	53.62	81.26	93.86	4.00	5.67	5.00	2.50	<u>4.29</u>
RAG($M=1$)	2.58	0.09	<u>68.07</u>	6.89	25.27	78.48	51.62	81.07	93.40	<u>2.00</u>	6.33	7.00	5.00	5.08
RAG+Extr.($M=4$)	3.46	0.16	69.03	6.49	24.33	79.03	54.79	81.21	<u>93.81</u>	5.00	<u>5.00</u>	4.00	<u>3.50</u>	4.38
RAG+Extr.($M=1$)	3.16	0.20	68.29	6.73	25.59	78.33	52.36	80.92	93.42	3.00	6.00	6.00	5.00	5.00
RAG+Abst.	3.78	0.17	72.40	7.32	25.11	78.04	<u>56.14</u>	80.50	93.29	10.00	8.33	<u>2.00</u>	6.50	6.71
EvoRAG	<u>0.40</u>	0.06	44.45	6.54	28.22	81.63	58.05	85.82	92.19	1.00	2.33	1.00	5.00	2.33

Table 7: Comparison between EvoRAG and baseline methods on our dataset implemented by Qwen2.5-72B-Instruct.

function E is implemented by our evaluation system introduced in Section 3.3. And the complete experiment results are exhibited in Table 7. Due to the high computational cost and large number of experiments, we only conducted a single run.

Metric	DMR	DUR	TBR	STR	PP	PR	TSR
Win Rate	80.96	83.30	81.37	87.69	91.35	89.57	97.67

Table 8: Win rate statistics of retrieval-augmented agents exceeding the Direct one, across various metrics.

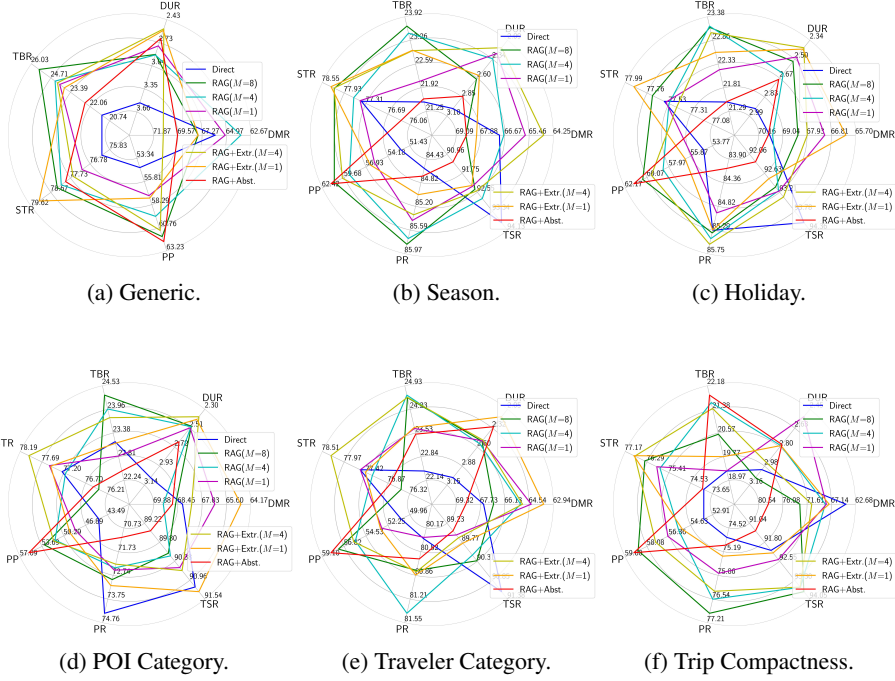


Figure 8: Comparison between Direct and RAG strategies on query data with various categories, implemented by GPT-4o.

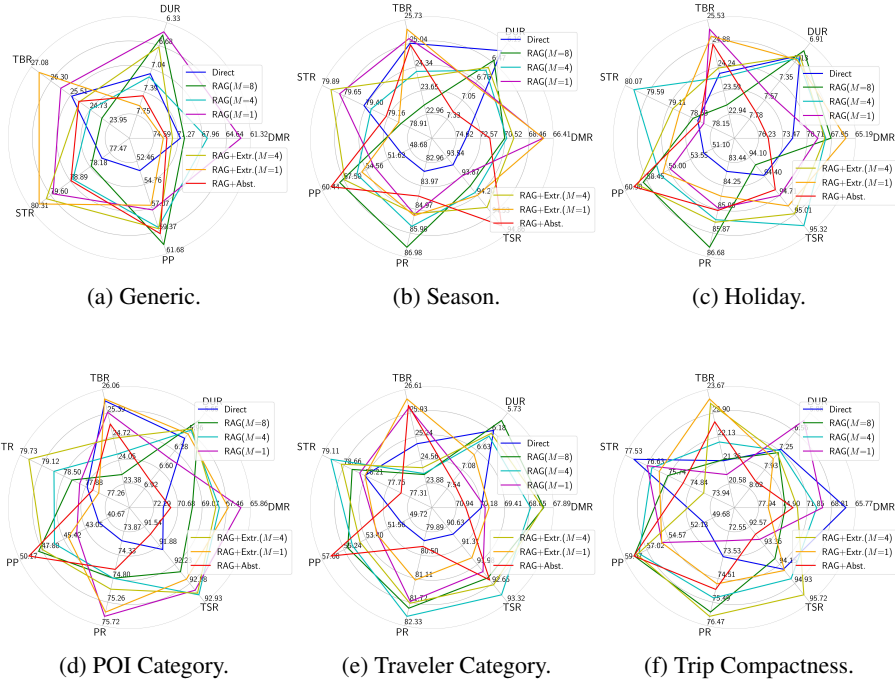


Figure 9: Comparison between Direct and RAG strategies on query data with various categories, implemented by Qwen2.5-72B-Instruct.

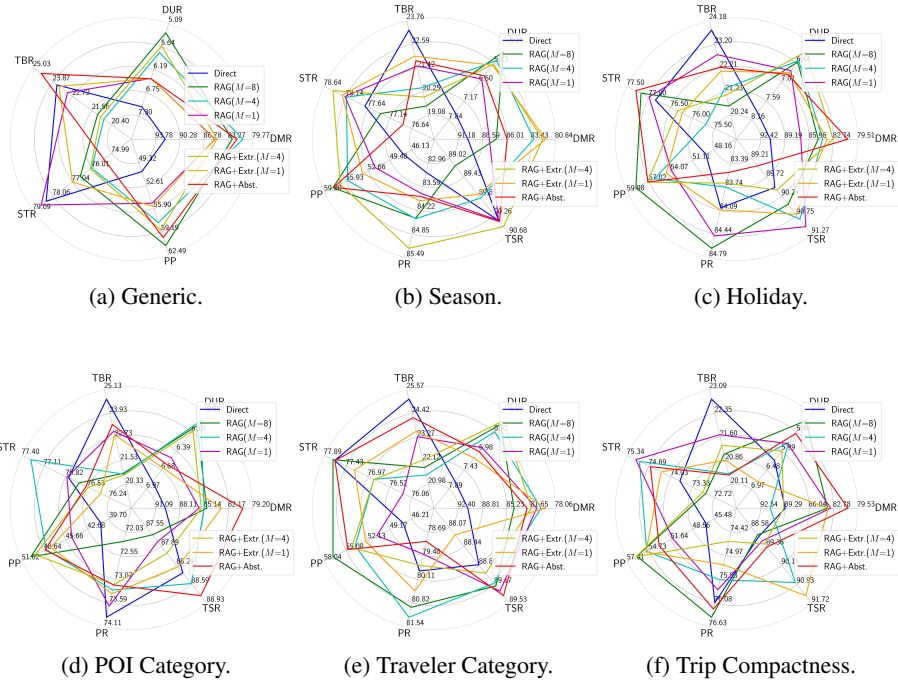


Figure 10: Comparison between Direct and RAG strategies on query data with various categories, implemented by LLaMA3.3-70B-Instruct.

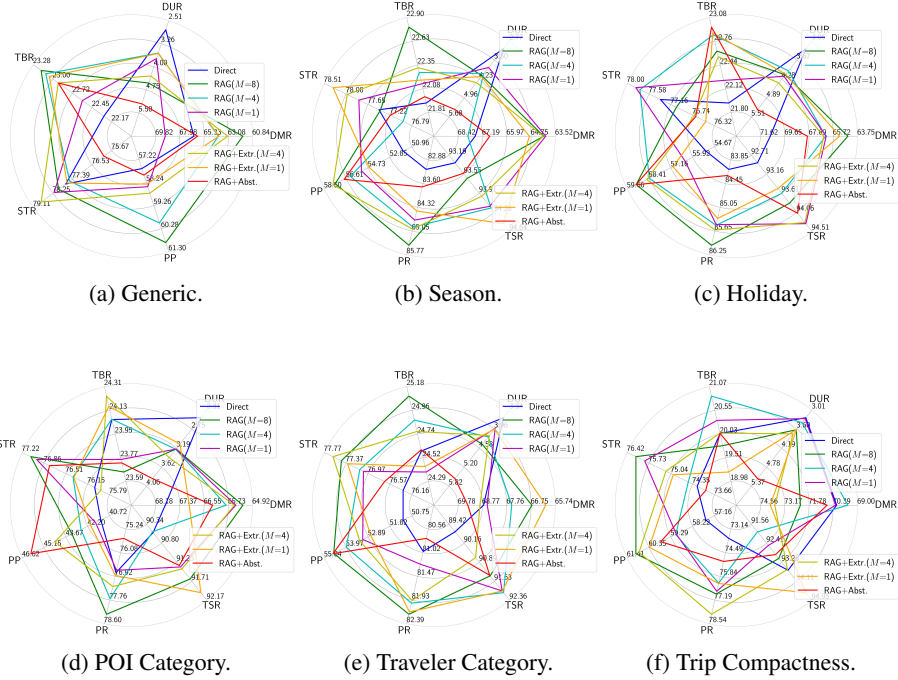


Figure 11: Comparison between Direct and RAG strategies on query data with various categories, implemented by DeepSeek-R1-671B.

E Prompt Templates

E.1 Data Generation

Query Formulation

Based on the seed query examples, please create a standard for query formulation, i.e., which fundamental elements the query may include, as well as which potential words can represent these elements.

<SEED QUERIES>

Answer Format in JSON:

```
{"Element 1": ["Potential Word 1", "Potential Word 2", ...], "Element 2": ["Potential Word 1", "Potential Word 2", ...], ...}
```

Query Generation

Based on the selected popular cities, please generate <NUMBER OF QUERIES> new natural language queries, that adhere to the standard of query formulation.

<POPULAR CITIES>

<QUERY FORMULATION>

Answer Format in JSON:

```
["Query 1", "Query 2", ...]
```

POI-level Query Rewriting

Based on the user query <QUERY>, please generate <NUMBER OF SUB-QUERIES> sub-queries that retain the meaning of the original query while facilitate retrieving Web documents about POI recommendations via the search engine. In addition, the generated sub-queries must be diverse.

Answer Format in JSON:

```
["Sub-query 1", "Sub-query 2", ...]
```

POI Extraction

Based on the following <NUMBER OF DOCUMENTS> retrieved documents, please identify and extract all related tourist attraction Points of Interest (POIs).

Tourist Attraction POI Definition:

Tourist attraction POIs include natural scenic areas, historical sites, cultural landmarks, parks, museums, commercial streets, resorts, theme parks, amusement parks, zoos and botanical gardens, specialty malls, cinemas, temples, palaces, etc..

POI Extraction Requirements:

1. Extraction: Use your knowledge and reasoning

abilities to identify and extract all tourist attraction-related POIs from the documents.

2. Explanation: For each extracted POI, provide a brief explanation of why it is considered a tourist attraction POI.

Special Notes:

1. Ensure the extraction is thorough, and avoid missing any potential POIs. Include every tourist attraction POI mentioned in the documents. Extract at least 10 and up to 30 POIs.

2. All POIs must come from the documents. Do not fabricate any POIs. Specify the exact source by referencing the relevant document ID.

3. Do not extract streets, roads, public facilities, broad geographic regions or city names.

4. Each POI name should represent a single tourist attraction. Avoid connecting multiple POIs with a hyphen ("-") or extracting duplicates or overlapping POIs.

<DOCUMENTS>

Answer Format in JSON:

```
[{"Extraction reason": "xxx", "Source": ["Document ID", ...], "POI name": "xxx"}, ...]
```

POI Inquiry Construction

Based on the user query <QUERY>, please generate <NUMBER OF QUERIES> queries for a specific attraction POI <POI>, that are relevant to the original query while facilitate retrieving real-time Web documents related to this POI via the search engine. In addition, the generated sub-queries must be diverse.

Answer Format in JSON:

```
["Query 1", "Query 2", ...]
```

Temporal and Semantic Tagging

Based on the retrieved documents, please analyze several things for the POI <POI>: (1) Opening Hours; (2) Recommended Visit Time; (3) Expected Visit Duration; (4) POI Description with Special Notes.

Requirements: 1. Provide intermediate analysis and reasoning steps.

2. Opening hours should specify the specific opening hours of the attraction in 24-hour format, such as "9:00-14:00" or "0:00-24:00" for all-day opening. Recommended visit time should offer specific recommended arrival times, also in 24-hour format. If there are no time restrictions, indicate "Open All Day." Expected visit duration should be in hours, such as "3" and "4.5". Description must be brief, not exceeding 50 words.

<DOCUMENTS>

Answer Format in JSON:

```
{
  "Reasoning steps": "xxx",
  "Opening hours": "xxx",
  "Recommended visit time": "xxx",
  "Expected visit duration": "xxx",
  "POI description": "xxx"
}
```

Trajectory-level Query Rewriting

Based on the user query <QUERY>, please generate <NUMBER OF SUB-QUERIES> sub-queries that retain the meaning of the original query while facilitate retrieving Web documents about travel guides via the search engine. In addition, the generated sub-queries must be diverse.

Answer Format in JSON:

```
[
  "Sub-query 1",
  "Sub-query 2",
  ...
]
```

Trajectory Extraction

Based on the given document below, please identify and extract the exact tourist trajectory, which consists of attraction POIs and is relevant to the user query <QUERY>.

Requirements:

1. Determine whether the given document contains a clear travel route. If not, respond with "None".
2. Organize the trajectory according to the number of travel days, with subtitles like "Day 1", "Day 2", and "Day 3". For each day, the itinerary should not be empty, especially the first and last day. If there are multiple trajectories in the given plan, choose one to extract and do not mix multiple solutions together.
3. In each extracted trajectory, the tourist attraction POIs and the order of visit must strictly follow the document. Do not hallucinate!
4. The attractions in the trajectory should be answered using the standardized POI names from the given POI reference list. If a POI is not listed in the reference list, it should still be included but with an special note.
5. Verify newly added attractions to ensure they are genuine tourist attractions, excluding non-POI items such as "return journey", "flag-raising ceremony", "free time", etc..
6. Summarize additional information about the attraction mentioned in the trajectory in a remark section. If there is no additional information, leave it blank. Do not extract information from the POI reference list for the remarks; it must be from the document.

<DOCUMENT>

<POI REFERENCE LIST>

Answer Format in JSON:

```
{
  "Day 1": [
    {
      "POI name": "xxx",
      "Whether in the reference list": "xxx",
      "Remark": "xxx"
    },
    ...
  ],
  ...
} or {
  "None": "None"
}
```

POI Quality Control

Given the initial list of tourist attraction POIs, please purify these POIs.

POI Purification Requirements:

1. Check whether the attraction POI is included in the user-specified city <CITY>. If not, please remove them.
2. Based on your understanding of these tourist attraction POIs, recheck if they are all indeed tourist attractions. If not, please remove non-tourist attraction POIs such as restaurants, hotels, specific leisure and entertainment venues (arcades, spas, cinemas, etc.).
3. Remove duplicate tourist attraction POIs, keeping only one instance. For example, if "Tiananmen" and "Tiananmen Square" both appear, keep "Tiananmen". If "Wangfujing" and "Wangfujing Street" both appear, keep "Wangfujing". If "National Aquatics Center" and "National Aquatics Center - North Entrance" both appear, keep "National Aquatics Center".
4. Some POIs may have a hierarchical relationship, such as "Summer Palace" and "Kunming Lake". In such cases, only keep "Summer Palace" and remove "Kunming Lake".
5. Filter the list from the initial tourist attraction POI list, and refrain from fabricating attractions.

<INITIAL POI LIST>

Please provide a textual explanation for the discarded tourist attraction POIs and respond in JSON format with the final processed list of purified POIs.

Answer Format in JSON:

```
[
  "POI name 1",
  "POI name 2",
  ...
]
```

Trajectory Quality Control

There is a tourist trajectory with a lot of noise (false scenic POIs, incomplete or incorrect names of scenic POIs). Please denoise the existing trajectory based on the standard name list of POIs.

Requirements:

1. If the existing trajectory contains non-attraction POIs that do not exist in the standard name list, consider making modifications or deletions:
(1) POI Modification: The names of POIs in the existing planning may be incomplete, non-standard, or inaccurate. Please find a standard and accurate name from the standard

name list as a replacement. For example, change "Aosen Park" to "Olympic Forest Park".

(2) POI Deletion: If there is no suitable replacement in the standard name list, please directly delete the POI from the current trajectory.

2. Ensure that the names of POIs in the modified trajectory are all in the standard name list. Prohibit the addition of new unrelated scenic POIs during the modification process.

<STANDARD POI INFORMATION>

<CURRENT TRAJECTORY>

Answer Format in JSON:

```
{ "Day 1": [{ "POI name": "xxx", "Remark": "xxx" }, ... ], ... }
```

E.2 Evaluation

Start Time Rationality (STR) Evaluation

Given the user query <QUERY>, associated reference information, and the planned start visit times for attraction POIs, evaluate whether the arrival time is reasonable.

Requirements: 1. Judge whether the start visit time falls into the opening hours.

2. Judge whether the start visit time conforms to the recommended arrival time.

3. Answer "Appropriate" or "Inappropriate" for each POI.

4. Do not omit any POI.

<REFERENCE TEMPORAL INFORMATION OF POIS>

<PLANNED START VISIT TIMES OF POIS>

Please give brief textual explanations of why the time slots for the attractions that are deemed inappropriate are not suitable, and provide the final evaluated results.

Answer Format in JSON:

```
{ "POI name 1": "xxx", "POI name 2": "xxx", ... }
```

POI Relevance (PR) Evaluation

Given the user query <QUERY> and the planned attraction POIs, evaluate whether the POI satisfies the personalized demands in the query. Answer "Satisfied" or "Unsatisfied" for each POI. Do not omit any POI.

<PLANNED POIS>

Please give brief textual explanations for POIs that are deemed unsatisfied, and provide the final evaluated results.

Answer Format in JSON:

```
{ "POI name 1": "xxx", "POI name 2": "xxx", ... }
```

Time Scheduling (TSR) Evaluation

Given the user query <QUERY> and the planned time slots of attraction POIs, evaluate whether the scheduled time slot satisfies the personalized demands in the query. Answer "Satisfied" or "Unsatisfied" for each POI. Do not omit any POI.

<PLANNED TIME SLOTS of POIS>

Please give brief textual explanations for POIs' time slots that are deemed unsatisfied, and provide the final evaluated results.

Answer Format in JSON:

```
{ "POI name 1": "xxx", "POI name 2": "xxx", ... }
```

E.3 Baseline

Direct

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

Basic Requirements:

1. Structure the article according to the number of days, such as "Day 1", "Day 2", and "Day 3". If the query does not specify the number of days, use your knowledge and the attractions list to deduce the duration of the travel plan.

2. Plan the visit to attractions POI in the order of scheduled visit times. Select attractions only from the provided reference list, and do not include attractions outside the list.

3. Plan specific start and end times of visit in 24-hour format for each POI. Ensure no overlap in visit times for different POIs, leave gaps between activities.

More Planning Requirements:

a. General Requirements:

1. Spatial: Consider the clustering of attraction POIs based on their geographical locations. Assign attraction POIs with close geographical locations to the same day and those with distant locations to different days. Ensure attractions on the same day are not too far apart.

2. Time:

(1) Each POI must be visited during its opening hours. Prioritize the recommended start times for attractions and ensure sufficient time for each visit (based on the expected duration of the attraction).

(2) In general, the total travel schedule for each day should not be too tight, ensuring the overall travel time is not too long, the number of attractions visited is not too large, and there is enough free time for meals, accommodation, and transportation.

3. Attractions Semantics: In general, prioritize popular and unique attractions that reflect the city's characteristics.

b. Personalized Requirements:

1. If there are additional personalized constraints in the query, understand and summarize these requirements. When selecting attractions and planning the plan, consider these personalized constraints.

2. Some personalized requirements may conflict with general requirements. In such cases, prioritize the personalized requirements. For example, if the query is "Special Forces-style Tourist", the overall itinerary time, number of attractions visited per day, free time, duration of attraction visits, and start times of visits may not have specific constraints, allowing for a more compact itinerary. If the query is related to specific demands (e.g., "Natural Landscape Tourism", "Family Travel", "Autumn Travel", "Chinese New Year Travel", etc.), just choose the most popular attractions that meet the query constraints.

<POI REFERENCE LIST>

Answer Format in JSON:

```
{"Day 1": [{"POI name": "xxx", "Start visit time": "xxx", "End visit time": "xxx"}, ...], ...}
```

Chain of Thought (CoT)

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

Basic Requirements:

1. Structure the article according to the number of days, such as "Day 1", "Day 2", and "Day 3". If the query does not specify the number of days, use your knowledge and the attractions list to deduce the duration of the travel plan.

2. Plan the visit to attractions POI in the order of scheduled visit times. Select attractions only from the provided reference list, and do not include attractions outside the list.

3. Plan specific start and end times of visit in 24-hour format for each POI. Ensure no overlap in visit times for different POIs, leave gaps between activities.

More Planning Requirements:

a. General Requirements:

1. Spatial: Consider the clustering of attraction POIs based on their geographical locations. Assign attraction POIs with close geographical locations to the same day and those with distant locations to different days. Ensure attractions on the same day are not too far apart.

2. Time:

(1) Each POI must be visited during its opening hours. Prioritize the recommended start times for attractions and ensure sufficient time for each visit (based on the expected duration of the attraction).

(2) In general, the total travel schedule for each day should not be too tight, ensuring

the overall travel time is not too long, the number of attractions visited is not too large, and there is enough free time for meals, accommodation, and transportation.

3. Attractions Semantics: In general, prioritize popular and unique attractions that reflect the city's characteristics.

b. Personalized Requirements:

1. If there are additional personalized constraints in the query, understand and summarize these requirements. When selecting attractions and planning the plan, consider these personalized constraints.

2. Some personalized requirements may conflict with general requirements. In such cases, prioritize the personalized requirements. For example, if the query is "Special Forces-style Tourist", the overall itinerary time, number of attractions visited per day, free time, duration of attraction visits, and start times of visits may not have specific constraints, allowing for a more compact itinerary. If the query is related to specific demands (e.g., "Natural Landscape Tourism", "Family Travel", "Autumn Travel", "Chinese New Year Travel", etc.), just choose the most popular attractions that meet the query constraints.

<POI REFERENCE LIST>

Please provide a step-by-step plan to solve the problem, and then present the final plan.

Answer Format in JSON:

```
{"Day 1": [{"POI name": "xxx", "Start visit time": "xxx", "End visit time": "xxx"}, ...], ...}
```

For plan initialization of Reflexion method, we directly utilize the template of Direct.

Reflexion (Feedback)

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

Planning Requirements:

a. General Requirements:

1. Spatial: Consider the clustering of attraction POIs based on their geographical locations. Assign attraction POIs with close geographical locations to the same day and those with distant locations to different days. Ensure attractions on the same day are not too far apart.

2. Time:

(1) Each POI must be visited during its opening hours. Prioritize the recommended start times for attractions and ensure sufficient time for each visit (based on the expected duration of the attraction).

(2) In general, the total travel schedule for each day should not be too tight, ensuring the overall travel time is not too long, the number of attractions visited is not too large, and there is enough free time for meals,

accommodation, and transportation.

3. Attractions Semantics: In general, prioritize popular and unique attractions that reflect the city's characteristics.

b. Personalized Requirements:

1. If there are additional personalized constraints in the query, understand and summarize these requirements. When selecting attractions and planning the plan, consider these personalized constraints.

2. Some personalized requirements may conflict with general requirements. In such cases, prioritize the personalized requirements. For example, if the query is "Special Forces-style Tourist", the overall itinerary time, number of attractions visited per day, free time, duration of attraction visits, and start times of visits may not have specific constraints, allowing for a more compact itinerary. If the query is related to specific demands (e.g., "Natural Landscape Tourism", "Family Travel", "Autumn Travel", "Chinese New Year Travel", etc.), just choose the most popular attractions that meet the query constraints.

There is an initial plan in place, please review whether this plan meets the requirements for attraction planning and provide specific feedback for modifications.

<POI REFERENCE LIST>

<INITIAL PLAN>

Reflexion (Refinement)

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

Basic Requirements:

1. Structure the article according to the number of days, such as "Day 1", "Day 2", and "Day 3". If the query does not specify the number of days, use your knowledge and the attractions list to deduce the duration of the travel plan.

2. Plan the visit to attractions POI in the order of scheduled visit times. Select attractions only from the provided reference list, and do not include attractions outside the list.

3. Plan specific start and end times of visit in 24-hour format for each POI. Ensure no overlap in visit times for different POIs, leave gaps between activities.

More Planning Requirements:

a. General Requirements:

1. Spatial: Consider the clustering of attraction POIs based on their geographical locations. Assign attraction POIs with close geographical locations to the same day and those with distant locations to different days. Ensure attractions on the same day are not too far apart.

2. Time:

(1) Each POI must be visited during its opening hours. Prioritize the recommended start times for attractions and ensure sufficient time for each visit (based on the expected duration of the attraction).

(2) In general, the total travel schedule for each day should not be too tight, ensuring the overall travel time is not too long, the number of attractions visited is not too large, and there is enough free time for meals, accommodation, and transportation.

3. Attractions Semantics: In general, prioritize popular and unique attractions that reflect the city's characteristics.

b. Personalized Requirements:

1. If there are additional personalized constraints in the query, understand and summarize these requirements. When selecting attractions and planning the plan, consider these personalized constraints.

2. Some personalized requirements may conflict with general requirements. In such cases, prioritize the personalized requirements. For example, if the query is "Special Forces-style Tourist", the overall itinerary time, number of attractions visited per day, free time, duration of attraction visits, and start times of visits may not have specific constraints, allowing for a more compact itinerary. If the query is related to specific demands (e.g., "Natural Landscape Tourism", "Family Travel", "Autumn Travel", "Chinese New Year Travel", etc.), just choose the most popular attractions that meet the query constraints.

There is an initial travel plan in place, as well as feedback for modifications to the plan. Please generate a revised plan in the same format as the original plan.

<POI REFERENCE LIST>

<INITIAL PLAN>

<FEEDBACK>

Answer Format in JSON:

```
{"Day 1": [{"POI name": "xxx", "Start visit time": "xxx", "End visit time": "xxx"}, ...], ...}
```

Multi-Agent Collaboration (MAC) applies a divide-and-conquer paradigm. First, a manager agent decomposes the planning problem into several sub-problems. The executor agents strive to address these sub-problems independently and the sub-solutions are finally summarized by the manager to directly answer the original question.

Multi-Agent Collaboration (MAC) - Manager Agent (Decomposition)

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

Basic Requirements:

1. Structure the article according to the number of days, such as "Day 1", "Day 2", and "Day 3". If the query does not specify the number of days, use your knowledge and the attractions list to deduce the duration of the travel plan.
2. Plan the visit to attractions POI in the order of scheduled visit times. Select attractions only from the provided reference list, and do not include attractions outside the list.
3. Plan specific start and end times of visit in 24-hour format for each POI. Ensure no overlap in visit times for different POIs, leave gaps between activities.

More Planning Requirements:

a. General Requirements:

1. Spatial: Consider the clustering of attraction POIs based on their geographical locations. Assign attraction POIs with close geographical locations to the same day and those with distant locations to different days. Ensure attractions on the same day are not too far apart.

2. Time:

- (1) Each POI must be visited during its opening hours. Prioritize the recommended start times for attractions and ensure sufficient time for each visit (based on the expected duration of the attraction).

- (2) In general, the total travel schedule for each day should not be too tight, ensuring the overall travel time is not too long, the number of attractions visited is not too large, and there is enough free time for meals, accommodation, and transportation.

3. Attractions Semantics: In general, prioritize popular and unique attractions that reflect the city's characteristics.

b. Personalized Requirements:

1. If there are additional personalized constraints in the query, understand and summarize these requirements. When selecting attractions and planning the plan, consider these personalized constraints.

2. Some personalized requirements may conflict with general requirements. In such cases, prioritize the personalized requirements. For example, if the query is "Special Forces-style Tourist", the overall itinerary time, number of attractions visited per day, free time, duration of attraction visits, and start times of visits may not have specific constraints, allowing for a more compact itinerary. If the query is related to specific demands (e.g., "Natural Landscape Tourism", "Family Travel", "Autumn Travel", "Chinese New Year Travel", etc.), just choose the most popular attractions that meet the query constraints.

Please do not directly answer this question, but carefully consider how to break down the problem and plan the execution sequence.

1. Break down the original planning problem into several subproblems (up to four), and detail the planning requirements for each subproblem.

2. The names of the subproblems correspond to the order of execution, with later subproblems taking the outputs of the previous subproblems as inputs.

Answer Format in JSON:

```
{"Sub-problem 1": {"Sub-problem description": "xxx", "Planning requirements": ["xxx", "xxx", ...]}, ... }
```

Multi-Agent Collaboration (MAC) - Executor Agent

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

<POI REFERENCE LIST>

Please do not directly answer this question, but carefully solve one sub-problem about it as follows:

<SUB-PROBLEM INFORMATION>

The outputs of previous subproblem:

<PREVIOUS OUTPUTS>

Please answer according to the requirements of the subproblem (answer in JSON format, keep it brief, not exceeding 500 characters).

Multi-Agent Collaboration (MAC) - Manager Agent (Summarization)

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

Basic Requirements:

1. Structure the article according to the number of days, such as "Day 1", "Day 2", and "Day 3". If the query does not specify the number of days, use your knowledge and the attractions list to deduce the duration of the travel plan.

2. Plan the visit to attractions POI in the order of scheduled visit times. Select attractions only from the provided reference list, and do not include attractions outside the list.

3. Plan specific start and end times of visit in 24-hour format for each POI. Ensure no overlap in visit times for different POIs, leave gaps between activities.

More Planning Requirements:

a. General Requirements:

1. Spatial: Consider the clustering of attraction POIs based on their geographical

locations. Assign attraction POIs with close geographical locations to the same day and those with distant locations to different days. Ensure attractions on the same day are not too far apart.

2. Time:

(1) Each POI must be visited during its opening hours. Prioritize the recommended start times for attractions and ensure sufficient time for each visit (based on the expected duration of the attraction).

(2) In general, the total travel schedule for each day should not be too tight, ensuring the overall travel time is not too long, the number of attractions visited is not too large, and there is enough free time for meals, accommodation, and transportation.

3. Attractions Semantics: In general, prioritize popular and unique attractions that reflect the city's characteristics.

b. Personalized Requirements:

1. If there are additional personalized constraints in the query, understand and summarize these requirements. When selecting attractions and planning the plan, consider these personalized constraints.

2. Some personalized requirements may conflict with general requirements. In such cases, prioritize the personalized requirements. For example, if the query is "Special Forces-style Tourist", the overall itinerary time, number of attractions visited per day, free time, duration of attraction visits, and start times of visits may not have specific constraints, allowing for a more compact itinerary. If the query is related to specific demands (e.g., "Natural Landscape Tourism", "Family Travel", "Autumn Travel", "Chinese New Year Travel", etc.), just choose the most popular attractions that meet the query constraints.

<POI REFERENCE LIST>

You have already thought about how to break down this problem and got answers to each sub-problem as follows:

<SUB-PROBLEM OUTPUTS>

Now please solve the original planning problem.

Answer Format in JSON:

```
{ "Day 1": [{"POI name": "xxx", "Start visit time": "xxx", "End visit time": "xxx"}, ...], ... }
```

Multi-Agent Debate (MAD) launches a discussion session allowing criticism agents with different perspectives to give feedback for the initial plan generated by a planner agent. The planner agent collects the feedback and try to make an enhanced travel plan. For the planner agent, we use the templates of Direct and Reflexion (Refinement) for plan initialization and refinement respectively.

Multi-Agent Debate (MAD) - Spatial-Perspective Criticism Agent

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

Spatial Requirements: Consider the clustering of attraction POIs based on their geographical locations. Assign attraction POIs with close geographical locations to the same day and those with distant locations to different days. Ensure attractions on the same day are not too far apart.

There is an initial plan in place, please review whether this plan meets the spatial requirements for attraction planning and provide specific feedback for modifications.

<POI REFERENCE LIST>

<INITIAL PLAN>

Multi-Agent Debate (MAD) - Temporal-Perspective Criticism Agent

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

Temporal Requirements:

1. Each POI must be visited during its opening hours. Prioritize the recommended start times for attractions and ensure sufficient time for each visit (based on the expected duration of the attraction).

2. In general, the total travel schedule for each day should not be too tight, ensuring the overall travel time is not too long, the number of attractions visited is not too large, and there is enough free time for meals, accommodation, and transportation.

There is an initial plan in place, please review whether this plan meets the temporal requirements for attraction planning and provide specific feedback for modifications.

<POI REFERENCE LIST>

<INITIAL PLAN>

Multi-Agent Debate (MAD) - Semantic-Perspective Criticism Agent

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

Semantic Requirements: In general, prioritize popular and unique attractions that reflect the city's characteristics.

There is an initial plan in place, please review whether this plan meets the semantic requirements for attraction planning and

provide specific feedback for modifications.

<POI REFERENCE LIST>

<INITIAL PLAN>

Multi-Agent Debate (MAD) - Relevance-Perspective Criticism Agent

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

Personalized Requirements:

1. If there are additional personalized constraints in the query, understand and summarize these requirements. When selecting attractions and planning the plan, consider these personalized constraints.
2. Some personalized requirements may conflict with general requirements. In such cases, prioritize the personalized requirements. For example, if the query is "Special Forces-style Tourist", the overall itinerary time, number of attractions visited per day, free time, duration of attraction visits, and start times of visits may not have specific constraints, allowing for a more compact itinerary. If the query is related to specific demands (e.g., "Natural Landscape Tourism", "Family Travel", "Autumn Travel", "Chinese New Year Travel", etc.), just choose the most popular attractions that meet the query constraints.

There is an initial plan in place, please review whether this plan meets the personal requirements for attraction planning and provide specific feedback for modifications.

<POI REFERENCE LIST>

<INITIAL PLAN>

Retrieval-Augmented Planning

Our task is to generate a travel plan based on the query <QUERY> and associated POI references and retrieved trajectories.

Basic Requirements:

1. Structure the article according to the number of days, such as "Day 1", "Day 2", and "Day 3". If the query does not specify the number of days, use your knowledge and the attractions list to deduce the duration of the travel plan.
2. Plan the visit to attractions POI in the order of scheduled visit times. Select attractions only from the provided reference list, and do not include attractions outside the list.
3. Plan specific start and end times of visit in 24-hour format for each POI. Ensure no overlap in visit times for different POIs, leave gaps between activities.

More Planning Requirements:

a. General Requirements:

1. Spatial: Consider the clustering of attraction POIs based on their geographical locations. Assign attraction POIs with close geographical locations to the same day and those with distant locations to different days. Ensure attractions on the same day are not too far apart.

2. Time:

- (1) Each POI must be visited during its opening hours. Prioritize the recommended start times for attractions and ensure sufficient time for each visit (based on the expected duration of the attraction).

- (2) In general, the total travel schedule for each day should not be too tight, ensuring the overall travel time is not too long, the number of attractions visited is not too large, and there is enough free time for meals, accommodation, and transportation.

3. Attractions Semantics: In general, prioritize popular and unique attractions that reflect the city's characteristics.

b. Personalized Requirements:

1. If there are additional personalized constraints in the query, understand and summarize these requirements. When selecting attractions and planning the plan, consider these personalized constraints.

2. Some personalized requirements may conflict with general requirements. In such cases, prioritize the personalized requirements. For example, if the query is "Special Forces-style Tourist", the overall itinerary time, number of attractions visited per day, free time, duration of attraction visits, and start times of visits may not have specific constraints, allowing for a more compact itinerary. If the query is related to specific demands (e.g., "Natural Landscape Tourism", "Family Travel", "Autumn Travel", "Chinese New Year Travel", etc.), just choose the most popular attractions that meet the query constraints.

<POI REFERENCE LIST>

<TRAJECTORIES>

Answer Format in JSON:

```
{ "Day 1": [{"POI name": "xxx", "Start visit time": "xxx", "End visit time": "xxx"}, ... ], ... }
```

For post-retrieval methods, we first apply compression on the raw trajectories and then use the Retrieval-Augmented Planning template for travel planning.

Extractive Trajectories Compression

Given several reference tourist trajectories about the query <QUERY>. Please select the <EXTRACTIVE NUMBER> schemes that best meet the requirements below, and answer with the

trajectory ID with explanations.

Extractive Requirements:

a. General Requirements:

1. Spatial: Consider the clustering of attraction POIs based on their geographical locations. Assign attraction POIs with close geographical locations to the same day and those with distant locations to different days. Ensure attractions on the same day are not too far apart.

2. Time:

(1) Each POI must be visited during its opening hours. Prioritize the recommended start times for attractions and ensure sufficient time for each visit (based on the expected duration of the attraction).

(2) In general, the total travel schedule for each day should not be too tight, ensuring the overall travel time is not too long, the number of attractions visited is not too large, and there is enough free time for meals, accommodation, and transportation.

3. Attractions Semantics: In general, prioritize popular and unique attractions that reflect the city's characteristics.

b. Personalized Requirements:

1. If there are additional personalized constraints in the query, understand and summarize these requirements. When selecting attractions and planning the plan, consider these personalized constraints.

2. Some personalized requirements may conflict with general requirements. In such cases, prioritize the personalized requirements. For example, if the query is "Special Forces-style Tourist", the overall itinerary time, number of attractions visited per day, free time, duration of attraction visits, and start times of visits may not have specific constraints, allowing for a more compact itinerary. If the query is related to specific demands (e.g., "Natural Landscape Tourism", "Family Travel", "Autumn Travel", "Chinese New Year Travel", etc.), just choose the most popular attractions that meet the query constraints.

<POI REFERENCE LIST>

<TRAJECTORIES>

Answer Format in JSON:

```
{"Explanation": "xxx", "Extractive IDs": ["x", "x", ...]}
```

Abstractive Trajectories Compression

Given several reference tourist trajectories about the query <QUERY>, please summarize, generalize, merge, and compress these information into a single trajectory according to the given requirements, with specific explanation and remarks.

Summarization Requirements:

a. General Requirements:

1. Spatial: Consider the clustering of attraction POIs based on their geographical locations. Assign attraction POIs with close geographical locations to the same day and those with distant locations to different days. Ensure attractions on the same day are not too far apart.

2. Time:

(1) Each POI must be visited during its opening hours. Prioritize the recommended start times for attractions and ensure sufficient time for each visit (based on the expected duration of the attraction).

(2) In general, the total travel schedule for each day should not be too tight, ensuring the overall travel time is not too long, the number of attractions visited is not too large, and there is enough free time for meals, accommodation, and transportation.

3. Attractions Semantics: In general, prioritize popular and unique attractions that reflect the city's characteristics.

b. Personalized Requirements:

1. If there are additional personalized constraints in the query, understand and summarize these requirements. When selecting attractions and planning the plan, consider these personalized constraints.

2. Some personalized requirements may conflict with general requirements. In such cases, prioritize the personalized requirements. For example, if the query is "Special Forces-style Tourist", the overall itinerary time, number of attractions visited per day, free time, duration of attraction visits, and start times of visits may not have specific constraints, allowing for a more compact itinerary. If the query is related to specific demands (e.g., "Natural Landscape Tourism", "Family Travel", "Autumn Travel", "Chinese New Year Travel", etc.), just choose the most popular attractions that meet the query constraints.

<POI REFERENCE LIST>

<TRAJECTORIES>

Answer Format in JSON:

```
{"Explanation": "xxx", "Results": {"Day 1": [{"POI name": "xxx", "Remark": "xxx"}, ...], ...}}
```

E.4 Methodology

For plan initialization, we adopt the prompt templates of Direct and Retrieval-Augmented Planning.

Evaluation Reflection

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

<POI REFERENCE LIST>

You have previously generated a batch of planning results and evaluated them:

1. Planning results are as follows (sorted in descending order of optimality):

<PREVIOUS PLANNING RESULTS>

2. Evaluation criteria (i.e., optimization objectives) are as follows:

<CRITERIA & OBJECTIVES>

Please analyze the differences in these plans based on the evaluation results. Considering the optimization objectives, reflect on what makes a good plan and how to achieve an even better plan.

You have already considered the following:

<PREVIOUS REFLECTION>

Now, please refine your previous reflection, providing a concise analysis for each optimization objective. Just provide your final reflection, no need to output the analysis process.

Plan Updating - Mutation Only

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

<POI REFERENCE LIST>

You have previously generated a batch of planning results and evaluated them:

1. Planning results are as follows (sorted in descending order of optimality):

<PREVIOUS PLANNING RESULTS>

2. Evaluation criteria (i.e., optimization objectives) are as follows:

<CRITERIA & OBJECTIVES>

Based on the evaluation results, you have the following considerations:

<REFLECTION>

Now, improve and optimize these plans. The improved set of <NUMBER OF PLANS> new plans should be distinct from each other and from the previous plans, but they should be more optimal than the previous results across all evaluation criteria.

Answer Format in JSON:

```
[{"Day 1": [{"POI name": "xxx", "Start visit time": "xxx", "End visit time": "xxx"}, ...], ...}, ...]
```

Plan Updating - Crossover & Mutation

Our task is to generate a travel plan based on the query <QUERY> and associated POI references.

<POI REFERENCE LIST>

You have previously generated a batch of planning results and evaluated them:

1. Planning results are as follows (sorted in descending order of optimality):

<PREVIOUS PLANNING RESULTS>

2. Evaluation criteria (i.e., optimization objectives) are as follows:

<CRITERIA & OBJECTIVES>

Based on the evaluation results, you have the following considerations:

<REFLECTION>

Please follow the steps below to generate new plans:

1. Selection: Choose two plans from the previous results that are less similar.

2. Crossover: Merge and process these two plans to create a new plan that combines the strengths of the original two plans.

3. Mutation: Based on your thoughts, further improve and optimize the new plan after the crossover.

Iterate and repeat the above steps until you generate <NUMBER OF PLANS> new plans. These new plans should be distinct from each other and from the previous plans, but they should be more optimal than the previous results across all evaluation criteria.

Answer Format in JSON:

```
[{"Day 1": [{"POI name": "xxx", "Start visit time": "xxx", "End visit time": "xxx"}, ...], ...}, ...]
```

F Statement on the Use of AI Assistance

In the preparation of this manuscript, AI-based tools (*i.e.*, ChatGPT and DeepSeek) were employed solely for the purpose of language polishing and text refinement. All conceptualization, experimental design, scientific reasoning, and the drawing of conclusions were conducted entirely by the authors. The AI assistant did not contribute to the generation of ideas, interpretation of results, or substantive content creation. Final responsibility for the manuscript rests with the authors.