LifePlanner: Evaluating LLM Agents for Geo-spatial Planning with Social Media Data

Anonymous ACL submission

Abstract

Geo-spatial planning, like trip design, is a realistic testbed for LLM agents because it requires grounded tool use, noisy evidence retrieval, and multi-constraint reasoning. Most benchmarks, however, only provide clean geospatial data and tools, missing the open-ended social signals that people use in daily planning. We introduce LifePlanner, a benchmark that enriches map data with large-scale local social media posts and provides access through an MCP toolset. LifePlanner evaluation covers 4 daily planning tasks, each across 3 difficulty levels. Experiments show frontier LLMs perform well on simple retrieval but degrade sharply on complex planning, with the Pass Rate dropping to 40.2%. Results show that failures mainly stem from incomplete evidence acquisition from such a large multi-modal database, imprecise tool use, and weak constraint integration rather than model size or reasoning length, suggesting that future progress requires effective grounded planning instead of scaling alone.

1 Introduction

Large Language Models (LLMs) are evolving from passive text generators into agentic systems capable of tool use and long-horizon reasoning (Parisi et al., 2022; Shen et al., 2023; Patil et al., 2024). Geo-spatial planning has therefore become an increasingly compelling task for evaluating such systems(Dihan et al., 2025; Cheng et al., 2026; Song et al., 2026). Its core appeal lies in a comprehensive testing: long-context processing over multi-modal and heterogeneous spatial evidence, tool use for grounded spatial analysis, and constrained reasoning for cost-preference trade-offs and spatially faithful planning.

However, as summarized in Table1, existing geo-spatial planning benchmarks remain limited along two key dimensions. The first is the *environment*, i.e., the database and tools available to LLMs. Most benchmarks are built on clean geospatial databases or manually curated resources (Shao et al., 2026; Ni et al., 2025; Qu et al., 2025; Dihan et al., 2025; Cheng et al., 2026; Song et al., 2026; Chen et al., 2025). Some expose LLMs to map-based tools for information retrieval and analysis, yet they rarely incorporate *open-ended social signals such as social-media posts, which are often central to how real users acquire timely, noisy, and locally grounded* spatial knowledge. The second is the *evaluation protocol*. Existing tasks are often narrow in intent and weakly stratified in difficulty, making it hard to derive systematic trends and insights across planning tasks and complexity levels.

Work	Env.		Eval.	
	Tool Box	*Social Media*	*Multi Task*	*Multi Diff.*
TravelPlanner (Xie et al., 2024)	✓	✗	✗	✓
CityEval (Feng et al., 2025a)	✗	✗	✓	✗
USTBench (Lai et al., 2025)	✗	✗	✓	✗
CityBench (Feng et al., 2025c)	✗	✗	✓	✗
MapEval (Dihan et al., 2025)	✓	✗	✓	✗
TripScore (Qu et al., 2025)	✗	✗	✗	✗
TP-RAG (Ni et al., 2025)	✗	✗	✗	✗
TripTailor (Wang et al., 2025a)	✓	✗	✗	✓
ChinaTravel (Shao et al., 2026)	✓	✗	✗	✓
COMPASS (Qin et al., 2026)	✓	✗	✗	✓
MobilityBench (Song et al., 2026)	✓	✗	✓	✗
DeepPlanning (Zhang et al., 2026)	✓	✗	✗	✗
TravelBench (Cheng et al., 2026)	✓	✗	✓	✗
LifePlanner (Ours)	✓	✓	✓	✓

Table 1: Comparison with prior geo-spatial benchmarks. *Multi Task* indicates the inclusion of varying geo-spatial task types, and *Multi Diff.* denotes the presence of hierarchical task difficulty levels within each task.

To address these limitations, we introduce LifePlanner, a scalable framework and benchmark for geo-spatial planning in realistic daily scenarios. At the *environment* level, LifePlanner augments standard geospatial information with large-scale, region-specific social-media data and provides agents with tool access for grounded, up-to-

date spatial analysis. At the *evaluation* level, LifePlanner defines a multi-task benchmark spanning four daily planning categories, progressively moving from point-level to area-level planning: Place Perception, Nearby Discovery, Routing, and Trip Design. Each category is further organized into three structured difficulty levels: L0 requires a single search over the given database, L1 requires multiple retrievals for calculation, and L2 requires decision-making over retrieved evidence under additional user constraints. This design enables systematic assessment across both task diversity and planning complexity.

LifePlanner reveals three key limitations of current LLM agents in realistic urban planning. (1) Task difficulty exposes a sharp gap between simple retrieval and complex constrained planning. (2) Failures arise less from hallucination than from *incomplete evidence acquisition and weak constraint integration*. (3) Model scaling and longer reasoning traces are insufficient; future agents need stronger training for effective tool use, targeted exploration on a large (multi-modality) database, and faithful evidence-grounded planning. To ensure reproducibility and robust measurement, all data collection tools and data will be open sourced.

2 Related Works

2.1 LLM Agents in City Scenario Planning

Urban scenario planning severely challenges Large Language Models (LLMs) due to diverse tasks and heterogeneous information. Previous studies have demonstrated that relying solely on parametric knowledge is fundamentally unreliable for complex geospatial reasoning (Wang et al., 2024; Yuan et al., 2025). To address this, existing research predominantly pursues domain knowledge injection or external information augmentation.

Under knowledge injection, prior work utilizes instruction-tuning with city-level structure data (Feng et al., 2025a), or incorporates multi-modal inputs (e.g., street and satellite views) to broaden spatial capabilities (Feng et al., 2025b; Wang et al., 2025b; Chu et al., 2026). However, despite achieving a certain degree of generalization on specific spatial tasks, inherent domain discrepancies severely limit cross-city applicability, necessitating resource-intensive, city-specific fine-tuning.

Alternatively, external augmentation bypasses parametric limitations via Retrieval-Augmented Generation (RAG) and dynamic tool invocation. RAG frameworks integrate external data into the context, tailored for geo-spatial tasks via SQL-based filtering (Yu et al., 2025) and document-based itinerary planning (Ni et al., 2025). Extending beyond static retrieval, tool augmentation grants agents executable actions to dynamically acquire information (Xie et al., 2024). To simplify complex tool utilization, recent works decompose queries into sequential steps for specialized sub-agents (Zhe et al., 2025; Hasan et al., 2026). Moving beyond pure geographic planning, latest developments leverage user profiles for preference-aligned planning (Liu et al., 2025; Lan et al., 2025).

In this work, alongside user-side preference integration, we significantly expand the information capacity and density on the environment side. By designing a customized, comprehensive toolset, we empower the agent to not only query standard cartographic platforms but also dynamically acquire real-time insights from social media. This paradigm shift authentically simulates the inherent noise, richness, and complexity of planning tasks within real-world urban spaces.

2.2 Evaluation on Geo-Spatial Reasoning

To evaluate internalized capabilities of language models, CityEval Feng et al. (2025a) assesses text-centric spatial tasks, while USTBench (Lai et al., 2025) and CityBench (Feng et al., 2025c) extend this to complex decision-making and multimodal data. Conversely, external augmentation benchmarks evaluate agents using retrieved data or dynamic tools. TripScore and TP-RAG (Qu et al., 2025; Ni et al., 2025) adopt a static context approach, providing necessary information upfront. Moving toward dynamic interaction, MapEval (Dihan et al., 2025) and TravelPlanner (Xie et al., 2024) require active data fetching via cartographic or domain-specific APIs. Recent works further assess interactive multi-turn preference inference (Qin et al., 2026; Cheng et al., 2026), multi-constraint optimization (Shao et al., 2026; Wang et al., 2025a; Zhang et al., 2026), and fine-grained, real-world mobility planning (Song et al., 2026).

Despite these advancements, existing benchmarks predominantly restrict agents to highly refined, structured data sources (e.g., cartographic databases or dedicated search APIs), which diverges sharply from actual human behavior. In reality, individuals invest significant effort in exploring noisy, unstructured data source, such as social media, and deeply integrating these infor-

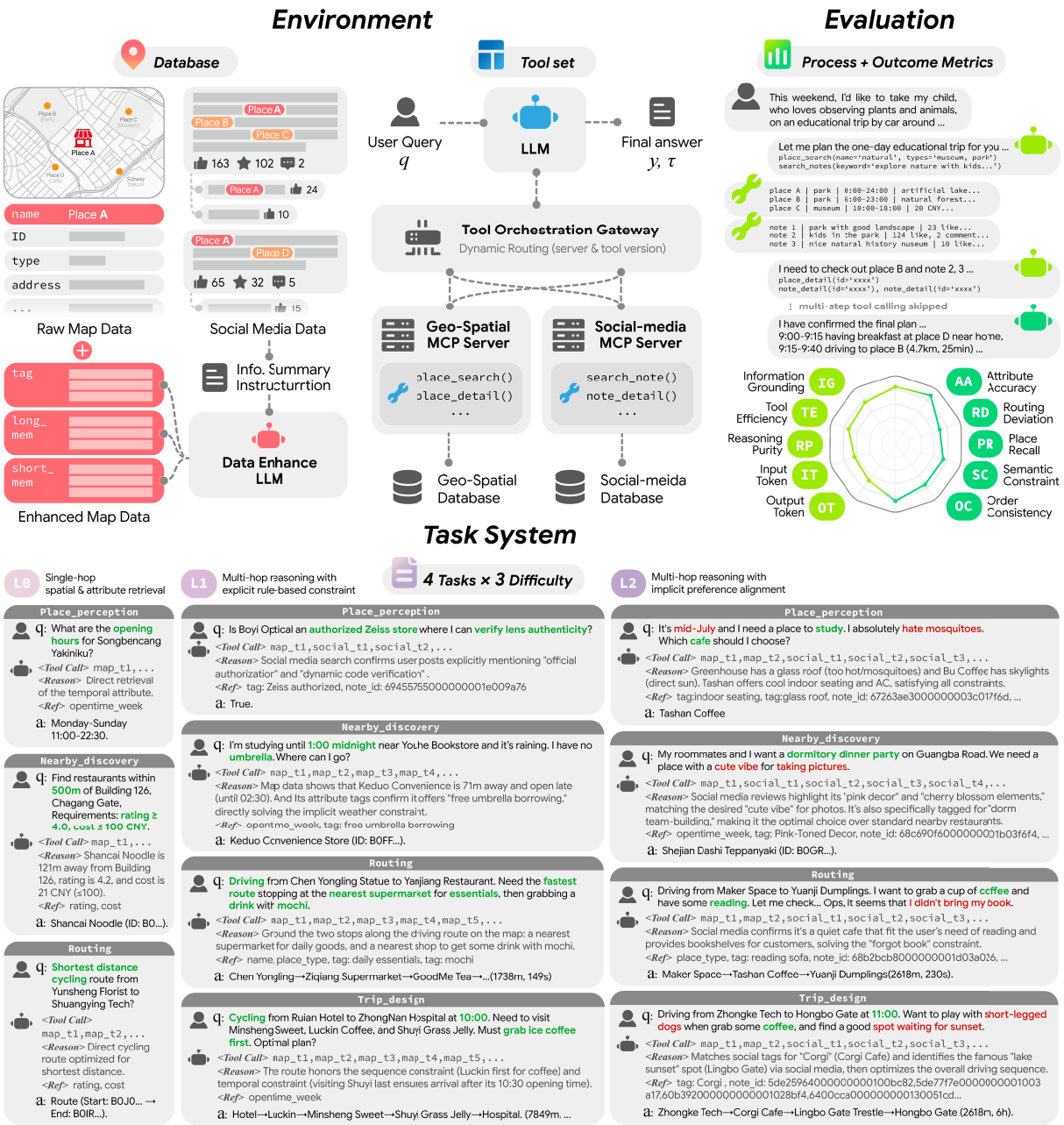

Figure 1: **Lifeplanner Overview.** LifePlanner consists of two main components: an environment and an evaluation protocol. The environment provides geospatial and social media data with a corresponding toolbox, while the evaluation assesses both the reasoning process and final outputs of LLM agents. The benchmark covers four task categories, each stratified by complexity. Complexity is defined by retrieval depth and constraint reasoning: L0 requires a single database retrieval step (green), L1 requires multiple retrieval steps and unconstrained calculation, and L2 further requires implicit constraints (red) to be integrated with retrieved evidence for a globally valid plan.

mation into their daily planning. To bridge this gap, LifePlanner introduces socially-driven preferences (e.g., locating "hidden gem" spot or limited-time events) that traditional structured platforms alone cannot satisfy. By compelling agents to actively navigate simulated social media and distill actionable information from high-noise environments, our benchmark pushes evaluation significantly closer to authentic real-world life planning.

3 LifePlanner

LifePlanner consists of two core components: a realistic *environment* for LLM interaction and a systematic *evaluation* protocol for measuring planning capability. The environment covers a database of approximately 10 km^2 of urban space, containing over 3,600 locations and about 200,000 social-media notes and comments, with tools for data retrieval and basic analysis. The evaluation proto-

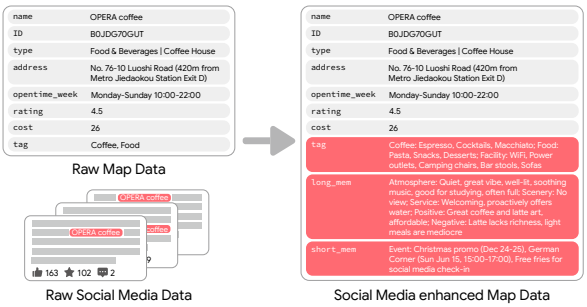

Figure 2: Lifeplanner uses social media data to enrich map data and construct a rich, realistic environment. Note that evaluated LLMs cannot access the time-sensitive information (mem field) extracted from social media.

col includes 667 predominantly open-ended queries across four daily planning categories and three complexity levels; for each, LLMs return a structured JSON output for multi-perspective scoring.

3.1 Environment

Database for LLM querying. We build the LifePlanner database through a four-stage pipeline:

- *Geospatial backbone construction.* We query the AMap API (Amap, 2002) and OpenStreetMap (OSM) (Coast, 2004) for places, road networks, and map attributes in the target region. After cleaning and deduplication, each place is assigned a unique spatial anchor ID as shown in Fig. 2 that carries map records, routing information, and subsequent social-media evidence.
- *Social-media evidence collection.* We augment each spatial anchor with locally grounded, unstructured evidence from social-media, specifically RedNote (RedNote, 2014). For each place, we retrieve 20 relevant notes by name search and collect 10 comments for each note. Each note is stored with its title, full content, publish time, engagement metrics (likes, collects, shares, and comments), and reference tags; each comment is stored with its publish time, full content, and like count. Before further processing, we remove personal information to protect privacy.
- *Attribute distillation.* We convert the raw social-media corpus into auxiliary structured attributes for dataset construction. Gemini-2.5-Flash is used to extract static venue properties as tags, such as offerings and facilities, and dynamic properties, including long-term impressions (e.g., ambiance and service quality) and short-term states (e.g., new arrivals or temporary closures). As shown in Fig. 2, they are merged to the corresponding spatial anchor record.
- *Access separation for evaluation.* The enriched information is not directly exposed to the tested LLMs in full. Instead, an LLM must first identify the target spatial anchor an query its corresponding map or social-media evidence through the tools described below. This tool-mediated access improves retrieval efficiency and reduces hallucination caused by overwhelming the model with large volumes of irrelevant raw data.

Toolbox for LLM interaction. We implement a toolbox compatible to Model Context Protocol (MCP)(MCP, 2024) that emulates how users consult map services and social-media platforms in real life. Its core functions are retrieval tools, which allow the LLM to search for target places or and obtain corresponding information by ID, name, tag, or free-form description(Gao et al., 2023). We also provide basic geo-spatial analysis tools, such as neighborhood identification. During evaluation, the LLM specifies a tool-use intent and a structured query, which are routed by our tool gateway to the appropriate backend tool for feedback. Finally, the LLM should returns both the final answer a and its tool-use chain $\tau = \{(c_t, o_t)\}_{t=1}^{T}$, where c_t and o_t denote the t-th tool call and its feedback. Evaluation scores both a and τ. Details are in Appendix.

3.2 Evaluation Protocol

Task Taxonomy. LifePlanner defines tasks along the scope of spatial decision-making, progressively moving from point-level understanding to area-level itinerary planning:

- *Place Perception*: identify, verify or compare properties of specific places.
- *Nearby Discovery*: find suitable candidates around a given spatial context.
- *Routing*: identify and integrate optimal waypoints along an active route under specific en-route constraints.
- *Trip Design*: construct comprehensive, multi-stop itineraries that jointly satisfy overarching user goals, personal preferences, and spatial-temporal constraints.

Difficulty Levels. Each task category is stratified by the operations required to solve it:

- *L0*: single-step lookup over geospatial data.

- *L1*: multi-step retrieval and computation, such as candidate comparison, route checking, or attribute aggregation.
- *L2*: hybrid reasoning over retrieved evidence, *implicit constraints, costs, and preferences* to select the best final plan.

Note that *Trip Design* starts from L1 because even its simplest form requires multi-step planning. As shown in Fig. 1, this yields 11 scenario-complexity combinations, enabling systematic analysis across both task diversity and planning complexity.

Evaluation Case Generation. We generate evaluation cases with an LLM-based data engine. Given a task template and candidate answer $a^\star$, the eigen has full access to the environment, including map records, raw social-media evidence, condensed dynamic attributes (Sec. 3.1), and tools. It first derives a valid solution path by selecting a set of tools, identifying supporting information, and verifying answer uniqueness; it then formulates a query q by injecting semantic and spatial constraints that make this path $\tau^\star$ necessary. We manually cross-validate each case for solvability, reference correctness, and answer uniqueness. Note that the tested LLM receives only the final query q and must reconstruct the hidden solution path through step-by-step database and tool exploration.

Metric Design. We evaluate both final outcomes and intermediate tool-use processes. The outcome metrics assess whether the final answer satisfies task requirements, while the process metrics assess whether the tool-use trajectory supports the answer efficiently and faithfully.

Outcome metrics to score a:

- *Attribute Accuracy (AA)*. AA evaluates whether the attributes x of these places p in a (Fig. 2) satisfy the explicit attribute requirements x_q specified in the query q, such as opening hours if it's a store. The score is 1 if all requirements are satisfied, and 0 otherwise:

$$AA = \mathbb{1}[x \models x_q]. \tag{1}$$

- *Place Recall (PR)*. PR measures the consistency between the predicted place set p in a and the ground-truth target set $p^\star$ in $a^\star$ using intersection over union:

$$PR = \frac{|p \cap p^\star|}{|p \cup p^\star|}. \tag{2}$$

- *Routing Deviation (RD)*. For routing-related tasks, RD evaluates whether the distance/-time v of the predicted route in a matches the ground-truth $v^\star$ of $a^\star$. Incorrect waypoints will bring a route-cost deviation:

$$RD_v = PR \cdot \max\left(0, 1 - \frac{|v - v^\star|}{v^\star}\right). \tag{3}$$

- *Order Consistency (OC)*. For routing and planning tasks, OC checks whether the places shared by a and $a^\star$ appear in the correct order. Let $r(\cdot)$ be the ground-truth rank function, and let $s = p \cap p^\star$ denote the ordered intersection of p and $p^\star$, with $n = |s|$:

$$OC = PR \cdot \left(1 - \frac{\sum_{i<j} \mathbb{1}[r(s_i) > r(s_j)]}{\binom{n}{2}}\right). \tag{4}$$

- *Semantic Constraint Satisfaction (SC)* evaluates constraints that cannot be fully captured by rules. Three LLM judges, GPT-5.4, Claude Haiku 4.5, and GLM-5.1, receive a, $a^\star$, the grading rubric, and verification tools. Each judge assigns a score in $[0, 1]$, and the final score is their average.
- *Pass Rate (PR)*. PR is the most comprehensive indicator that directly evaluates whether a task is successful. It reports the percentage of cases whose SC exceeds a threshold α, and all other calculable indicators above are 1.

Process metrics to score τ:

- *Information Grounding (IG)* measures whether the intermediate information $o \in \tau$ covers the required information $\tau^\star$, where o denotes identities such as a place attribute or social-media note according to the case:

$$IG = \frac{1}{|\{o\}|} \sum_{\{o\}} \mathbb{1}[o \in \tau^\star]. \tag{5}$$

- *Tool Efficiency (TE)* penalizes redundant tool use by comparing the number of calls in the reference tool-use set $|\tau^\star|$ with the actual number of calls $|\tau|$:

$$TE = \min\left(1, \frac{|\tau^\star|}{|\tau|}\right). \tag{6}$$

- *Reasoning Purity (RP)* assesses whether the information and reasoning in τ are faithful to the database, without hallucination or distortion. We provide τ and the corresponding database evidence to LLM judges for scoring.
- *Token Usage* records computational overhead, including input tokens from prompts, queries, tool feedback, and context, as well as all output tokens generated by the LLM agent.

Model	Place Perception			Nearby Discovery			Routing			Trip		Overall	
	L0	*L1*	*L2*	*L0*	*L1*	*L2*	*L0*	*L1*	*L2*	*L1*	*L2*	*Avg.*	*#Tokens*
Closed-source Models													
Claude-Sonnet-4.6	**99.0**	87.5	91.4	74.0	**67.3**	**54.3**	**100.0**	46.0	**38.7**	69.8	25.7	62.7	121.8k
Qwen3.6-Plus	**99.0**	84.7	88.6	**87.5**	61.5	48.6	98.0	44.0	35.5	69.8	25.7	**63.3**	142.3k
GPT-5.4	97.0	83.3	80.0	75.0	44.2	40.0	95.0	44.0	29.0	66.0	**28.6**	57.9	98.9k
Gemini-3.1-Pro-Preview	98.0	86.1	71.4	78.9	53.9	48.6	**100.0**	20.0	16.1	41.5	14.3	59.4	56.5k
Open-source Models													
GLM-5.1	**99.0**	**88.9**	**94.3**	80.8	61.5	40.0	99.0	**56.0**	32.3	**73.6**	25.7	61.5	75.2k
Kimi-K2.6	96.0	**88.9**	82.9	76.0	**67.3**	45.7	95.0	36.0	35.5	64.2	25.7	59.8	123.2k
MiniMax-M2.7	96.0	70.8	71.4	80.8	**67.3**	42.9	99.0	12.0	12.9	39.6	11.4	56.7	65.8k
Qwen3.5-27B	92.0	81.9	82.9	79.8	59.6	42.9	95.0	30.0	22.6	62.3	11.4	59.4	191.2k
Qwen3.5-35B-A3B	97.0	70.8	74.3	78.9	57.7	40.0	92.0	8.0	12.9	45.3	0.0	56.7	126.4k
Qwen3.5-9B	83.0	47.2	17.1	64.4	21.2	25.7	83.0	4.0	0.0	32.1	2.9	41.4	90.7k

Table 2: *PR@0.9 in %* (Pass rate with $\alpha = 0.9$) of LLMs. We **bold** the best and underlined the second-best.

Metric	L0	L1	L2
Process			
Information Grounding	96.5	80.4	64.6
Tool Efficiency	76.2	53.7	44.6
Reasoning Purity	96.7	89.4	86.4
Outcome			
Attribute Accuracy	96.1	86.9	80.8
Place Recall	90.7	73.8	55.0
Routing Deviation	97.2	69.7	47.6
Order Consistency	-	84.7	44.7
Semantic Constraint	-	74.0	65.2
Pass Rate@0.9	89.4	57.7	40.2
Token Usage	24.6k	137.5k	251.1k

Table 3: Average performance in % of all LLMs across different task difficulty levels. "-" indicate that Order Consistency and Semantic Constraint are not applicable to the single-hop retrieval nature of L0 tasks.

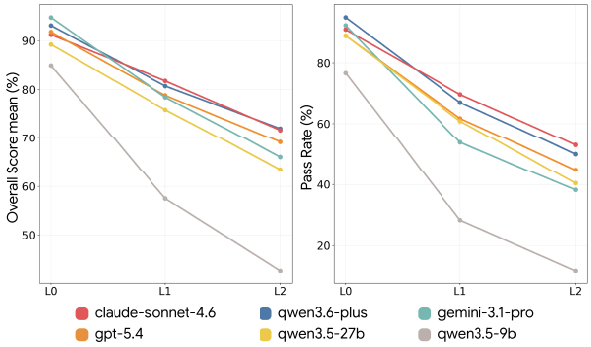

Figure 3: Average scores (left) and strict Pass Rates (right) of representative models across the three difficulty levels (L0, L1, L2).

4 Experiments

4.1 Models

We evaluate LifePlanner on a broad set of representative LLMs, including frontier closed-source models (Claude-Sonnet-4.6 (Anthropic, 2026), Gemini-3.1-Pro-Preview (Google, 2026), GPT-5.4 (OpenAI, 2026), and Qwen3.6-Plus (Qwen, 2026)) and competitive open-source models (GLM-5.1 (GLM-5-Team et al., 2026), Kimi-K2.6 (Kimi, 2026), MiniMax-M2.7 (MiniMax, 2026), Qwen3.5-27B, Qwen3.5-35B-A3B, and Qwen3.5-9B). All models are evaluated under the same task and tool settings. See Appendix for more Details.

4.2 Outcome analysis

Table 2 presents the comprehensive evaluation results of all models across the 11 tasks. Table 3 shows model-averaage results across difference metrics. We detail our key findings below.

Difficulties. Task difficulty is the most consistent source of performance degradation. From L0 to L2, Table 2, Table 3, and Figure 3 show that the average Pass Rate drops from 89.4% to 40.2%, while Token Usage increases from 24.6k to 251.1k. This shows that L2 tasks require longer exploration, meaning multi-hop evidence collection, and faithful integration of spatial, semantic, ordering, and implicit constraints. However, the outcome metrics of Table 3 show that current models still perform poorly in evidence retrieval and ordering, and also struggle with constraint understanding and integration, with L2 Semantic Constraint reaching only 65.2%. Together, these limitations lead to only a 40.2% success rate on L2 tasks.

Tasks. The other trend exists in tasks and performance. In relatively simple place-recognition tasks, models only need limited intent understanding and a small number of tool queries; strong models can still achieve over 90% Pass Rate even at L2. In contrast, trip planning requires models to satisfy

multiple retrieval requirements and user constraints simultaneously, with most intermediate steps depending on correct tool use. As a result, even strong models reach only about 25% Pass Rate on L2 trip design tasks. This demonstrates that our task design effectively distinguishes different dimensions of model capability, while also showing that current models remain far from reliable on complex constrained planning.

Models. In table 2, stronger models generally perform better, but scaling alone does not close the planning gap. Qwen3.6-Plus and Claude-Sonnet-4.6 achieve the highest overall Pass Rates, while GLM-5.1 reaches competitive performance among open-source models. Within the same model family, Qwen3.5-27B substantially outperforms Qwen3.5-9B, indicating that scaling is indeed beneficial. However, the ranking is not monotonic with size: GPT-5.4 does not outperform the smaller Qwen3.5-27B and GLM-5.1. This suggests that after a basic capability threshold, planning behavior and evidence integration matter more than parameter scale alone.

Efficiency. More exploration is necessary for difficult tasks, but it is not sufficient. Gemini-3.1-Pro-Preview uses the fewest tokens and is highly efficient, yet its early-stopping behavior, i.e., reaching a conclusion before enough evidence has been gathered, hurts performance on difficult routing and trip design cases. Conversely, Qwen3.5-27B consumes 191.2k tokens per task but underperforms GLM-5.1, which uses only 75.2k. Thus, effective planning requires not only collecting more information, but also precisely analyzing relevant evidence, avoiding redundant calls while needing enough calls, and integrating constraints into the final answer. This remains a demanding test of an agent's overall capability.

These findings validate our benchmark as a comprehensive platform, highlighting the substantial gap that future LLM agents must bridge: *evidence retrieval in a large database, efficient tool usage, and planning with constraints*.

4.3 Process analysis

We further analyze the process metrics in Table 3/ Fig. 4 and cases in Fig. 6 to identify why LLM agents fail during exploration.

Relative faithful reasoning. Reasoning Purity remains high even on L2 tasks, reaching 86.4% on average. This indicates that most models can preserve the factual accuracy of retrieved database information without severe hallucination or distortion. However, this ability is not uniform: weaker models, such as Qwen3.5-9B in Fig. 4, show consistently lower Reasoning Purity on L2 tasks, suggesting that faithful use of retrieved evidence still depends on model capability.

Incomplete information acquisition. Information Grounding drops to 64.6% on L2 tasks, showing that agents frequently make decisions before collecting all required evidence, or exhaust the tool budget without obtaining the necessary information. The gap across models in Fig. 4 is substantial. Gemini-3.1-Pro-Preview is trained to terminate early, using the fewest tokens but also retrieving a smaller fraction of required evidence. Conversely, Qwen3.5-27B consumes many more tokens, yet its grounding ratio remains below stronger closed-source models such as Claude-Sonnet-4.6. This confirms that increasing exploration length alone does not solve the problem; agents need more effective evidence analysis and search strategies.

Inefficient tool usage. Tool Efficiency is the weakest process metric, dropping to 44.6% on L2 tasks. This suggests that models often rely on redundant or poorly targeted tool calls instead of using tools efficiently. As shown in Figure 5, tool-use failures are dominated not by inability to call tools, but by imprecise tool use. Only a small fraction of calls fail because of malformed formats or invalid argument types. Most failures instead return limited or empty results, indicating that the model issues queries that are too broad, underspecified, or misaligned with the target evidence. Figure 6 (left) shows a typical failure in which an incorrect tag traps the model in repeated retrieval of irrelevant results until the tool budget is exhausted. In other words, current LLMs generally know how to invoke tools, but still struggle to formulate precise queries and use returned information efficiently.

Weak constraint integration. Even when relevant evidence is retrieved, LLM agents often fail to convert it into a plan satisfying all constraints. Consistent with the Semantic Constraint results of Table 3, models may ignore implicit or soft requirements, over-optimize a salient objective such as shortest distance, or misread retrieved evidence when composing the final answer. Figure 6 (right) shows a common case where the model fails to produce a globally reasonable plan because it does not jointly consider the closing-time order of different stores, only fulfilling shortest distance needs. Thus, the main bottleneck is not only access to in-

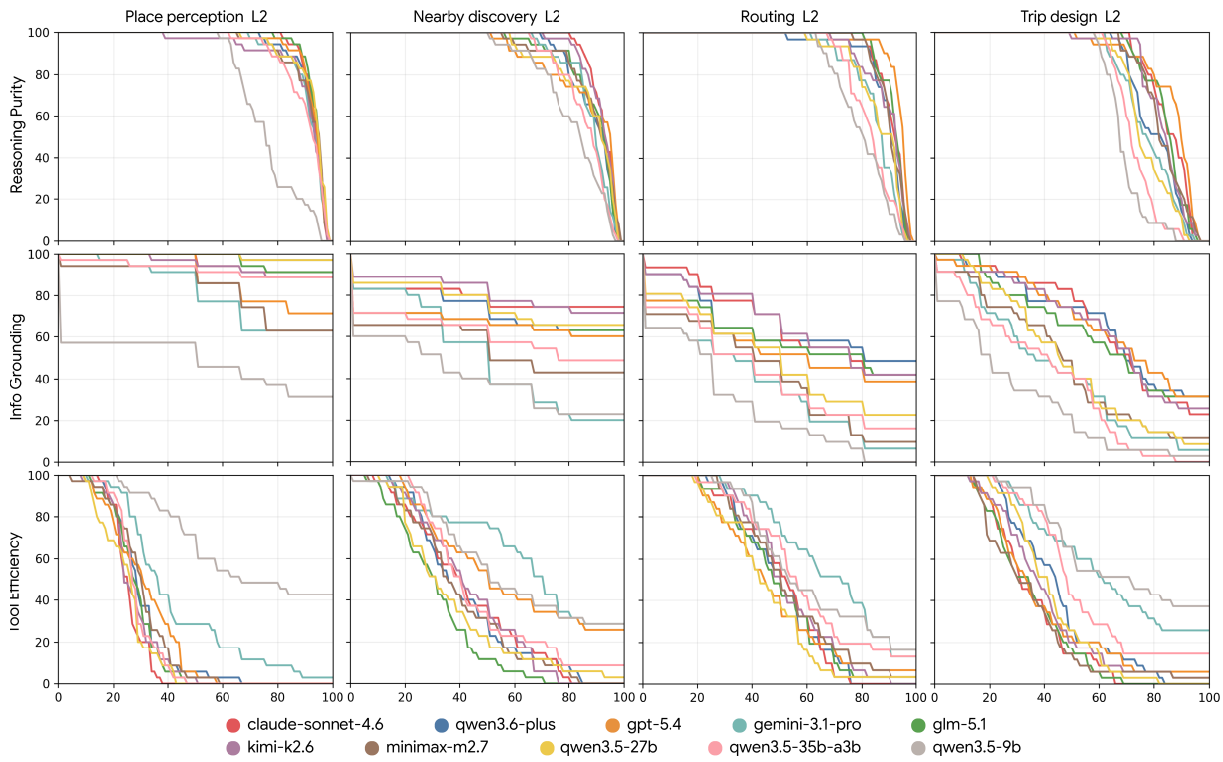

Figure 4: Cross-model exceedance curves for the **Reasoning Purity, Information Grounding and Tool Efficiency** score across different type L2 tasks. A point (x, y) indicates that $y\%$ of cases achieved a score $\geq x$, x is in %.

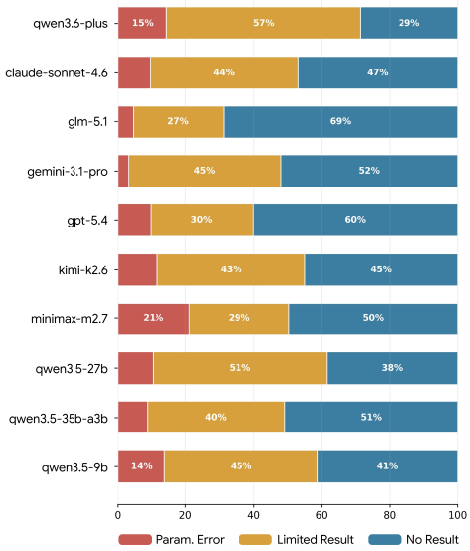

Figure 5: Distribution of tool failures. It illustrates the percentage of tool-call failing for parameter errors, truncations (too many results) for loose conditions, and empty/error returns for imprecise conditions.

formation, but the ability to maintain and integrate multiple constraints in long-horizon reasoning.

Similar conclusions are achieved to outcome analysis that failures are not mainly caused by severe hallucination, but by weak exploration: LLM agents often miss key evidence, issue imprecise tool queries, and fail to turn retrieved information and constraints into reliable plans.

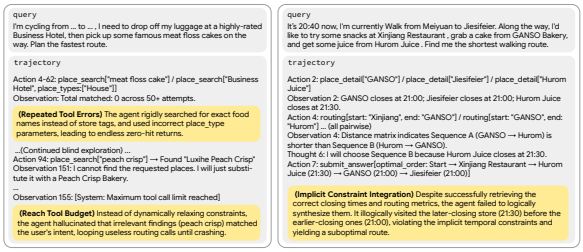

Figure 6: Representative failure cases. Yellow highlights indicate the main errors: (Left) the model repeatedly retrieves the wrong target with an incorrect tag until the tool budget is exhausted; (Right) the model fails to jointly consider constraints such as store closing times when composing the final plan.

5 Conclusion

We present LifePlanner, a realistic geo-spatial planning benchmark that combines structured map data, large-scale social-media evidence, MCP-based tools, and multi-level planning tasks. Our evaluation shows a sharp gap between simple retrieval and complex multi-constraint planning. Failures mainly arise from incomplete evidence acquisition in a large database, inefficient tool usage, and weak constraint integration for joint planning rather than hallucination. Moreover, model scaling and longer reasoning are insufficient without precise tool queries and constrained plan aggregation.

Limitations

Design limitations. Although LifePlanner is designed to scale to massive and dynamic real-world information, the current benchmark has several limitations. First, the social media corpus comes from a single large-scale collection, resulting in a temporally concentrated snapshot. This limits our ability to evaluate how agents handle information conflicts caused by temporal changes. Second, the current benchmark focuses on single-turn, multi-hop planning with all constraints given upfront. Extending LifePlanner to multi-turn interactive planning, where user preferences evolve during the conversation, is an important direction for future work.

Potential Risks. At now, all external artifacts, including map services, social media data, and model APIs, are used in accordance with their respective access terms and are intended only for research and evaluation; derived resources should not be used for user profiling, commercial redistribution, or purposes beyond the original access conditions. Because LifePlanner incorporates social media content, privacy protection is a central concern. Although the benchmark is designed for research on planning agents rather than user profiling, raw posts may contain sensitive or personally identifiable information. We therefore remove personal identifiers during preprocessing and retain only task-relevant evidence needed for evaluation. Any future release or extension of the dataset should continue to follow platform policies, minimize exposed user content, and apply strict anonymization to prevent re-identification or unintended use of personal information.

References

Amap. 2002. Amap. https://www.amap.com/.

Anthropic. 2026. Introducing claude sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6.

Jiabin Chen, Haiping Wang, Jinpeng Li, Yuan Liu, Zhen Dong, and Bisheng Yang. 2025. Spatialllm: From multi-modality data to urban spatial intelligence. *Preprint*, arXiv:2505.12703.

Xiang Cheng, Yulan Hu, Xiangwen Zhang, Lu Xu, Lide Tan, Zheng Pan, Xin Li, and Yong Liu. 2026. Beyond itinerary planning-a real-world benchmark for multi-turn and tool-using travel tasks. *Preprint*, arXiv:2512.22673.

Meng Chu, Yukang Chen, Haokun Gui, Shaozuo Yu, Yi Wang, and Jiaya Jia. 2026. TraveLLaMA: A multimodal travel assistant with large-scale dataset and structured reasoning. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 3390–3398. AAAI Press.

Steve Coast. 2004. Open street map. https://www.openstreetmap.org/.

Mahir Labib Dihan, Md Tanvir Hassan, Md Tanvir Parvez, Md Hasebul Hasan, Md Almash Alam, Muhammad Aamir Cheema, Mohammed Eunus Ali, and Md. Rizwan Parvez. 2025. MapEval: A map-based evaluation of geo-spatial reasoning in foundation models. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

Jie Feng, Tianhui Liu, Yuwei Du, Siqi Guo, Yuming Lin, and Yong Li. 2025a. CityGPT: Empowering urban spatial cognition of large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.2, KDD 2025, Toronto ON, Canada, August 3-7, 2025*, pages 591–602. ACM.

Jie Feng, Shengyuan Wang, Tianhui Liu, Yanxin Xi, and Yong Li. 2025b. Urbanllava: A multi-modal large language model for urban intelligence. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pages 6209–6219. IEEE.

Jie Feng, Jun Zhang, Tianhui Liu, Xin Zhang, Tianjian Ouyang, Junbo Yan, Yuwei Du, Siqi Guo, and Yong Li. 2025c. CityBench: Evaluating the capabilities of large language models for urban tasks. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.2, KDD 2025, Toronto ON, Canada, August 3-7, 2025*, pages 5413–5424. ACM.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, Haofen Wang, and 1 others. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1):32.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, and 168 others. 2026. Glm-5: from vibe coding to agentic engineering. *Preprint*, arXiv:2602.15763.

Google. 2026. Gemini 3.1 pro: A smarter model for your most complex tasks. https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/.

Md Hasebul Hasan, Mahir Labib Dihan, Tanzima Hashem, Mohammed Eunus Ali, and Md Rizwan Parvez. 2026. MapAgent: A hierarchical agent for geospatial reasoning with dynamic map tool integration. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 1296–1322, Rabat, Morocco. Association for Computational Linguistics.

Kimi. 2026. Kimi k2.6: Advancing open-source coding. https://www.kimi.com/blog/kimi-k2-6.

Siqi Lai, Yansong Ning, Zirui Yuan, Zhixi Chen, and Hao Liu. 2025. USTBench: Benchmarking and dissecting spatiotemporal reasoning of llms as urban agents. *Preprint*, arXiv:2505.17572.

Xiaochong Lan, Jie Feng, Jiahuan Lei, Xinlei Shi, and Yong Li. 2025. Localgpt: Benchmarking and advancing large language models for local life services in meituan. *Preprint*, arXiv:2506.02720.

Yiping Liu, Hyungchul Chung, Zixuan Xu, Kitae Jang, Sikai Chen, and Tiantian Chen. 2025. LLM-TripPlanner: A large-language-model-based agent for personalized trip planning. In *28th IEEE International Conference on Intelligent Transportation Systems, ITSC 2025, Gold Coast, Australia, November 18-21, 2025*, pages 927–934. IEEE.

MCP. 2024. Model context protocol. https://github.com/modelcontextprotocol.

MiniMax. 2026. Minimax m2.7 - model self-improvement, driving productivity innovation through technological breakthroughs. https://www.minimax.io/models/text/m27.

Hang Ni, Fan Liu, Xinyu Ma, Lixin Su, Shuaiqiang Wang, Dawei Yin, Hui Xiong, and Hao Liu. 2025. TP-RAG: Benchmarking retrieval-augmented large language model agents for spatiotemporal-aware travel planning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12392–12418, Suzhou, China. Association for Computational Linguistics.

OpenAI. 2026. Introducing gpt-5.4. https://openai.com/zh-Hans-CN/index/introducing-gpt-5-4/.

Aaron Parisi, Yao Zhao, and Noah Fiedel. 2022. TALM: Tool augmented language models. *Preprint*, arXiv:2205.12255.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2024. Gorilla: Large language model connected with massive apis. In *Advances in Neural Information Processing Systems*, volume 37, pages 126544–126565.

Tian Qin, Felix Bai, Ting-Yao Hu, Raviteja Vemulapalli, Hema Swetha Koppula, Zhiyang Xu, Bowen Jin, Mert Cemri, Jiarui Lu, Zirui Wang, and Meng Cao. 2026. COMPASS: Benchmarking constrained optimization in llm agents. *Preprint*, arXiv:2510.07043.

Yincen Qu, Huan Xiao, Feng Li, Gregory Li, Hui Zhou, Xiangying Dai, and Xiaoru Dai. 2025. TripScore: Benchmarking and rewarding real-world travel planning with fine-grained evaluation. *Preprint*, arXiv:2510.09011.

Qwen. 2026. Qwen3.6-plus: Towards real world agents. https://qwen.ai/blog?id=qwen3.6.

RedNote. 2014. Rednote. https://www.xiaohongshu.com/.

Jie-Jing Shao, Bo-Wen Zhang, Xiao-Wen Yang, Baizhi Chen, Si-Yu Han, Jinghao Pang, Wen-Da Wei, Guohao Cai, Zhenhua Dong, Lan-Zhe Guo, and Yu-Feng Li. 2026. Chinatravel: An open-ended travel planning benchmark with compositional constraint validation for language agents. *Preprint*, arXiv:2412.13682.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. In *Advances in Neural Information Processing Systems*, volume 36, pages 38154–38180.

Zhiheng Song, Jingshuai Zhang, Chuan Qin, Chao Wang, Chao Chen, Longfei Xu, Kaikui Liu, Xiangxiang Chu, and Hengshu Zhu. 2026. Mobilitybench: A benchmark for evaluating route-planning agents in real-world mobility scenarios. *Preprint*, arXiv:2602.22638.

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Sharon Li, and Neel Joshi. 2024. Is A picture worth A thousand words? delving into spatial reasoning for vision language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Kaimin Wang, Yuanzhe Shen, Changze Lv, Xiaoqing Zheng, and Xuanjing Huang. 2025a. TripTailor: A real-world benchmark for personalized travel planning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9705–9723, Vienna, Austria. Association for Computational Linguistics.

Qiongyan Wang, Xingchen Zou, Yutian Jiang, Haomin Wen, Jiaheng Wei, Qingsong Wen, and Yuxuan Liang. 2025b. Urban-r1: Reinforced mllms mitigate geospatial biases for urban general intelligence. *Preprint*, arXiv:2510.16555.

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 *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 54590–54613. PMLR / OpenReview.net.

Dazhou Yu, Riyang Bao, Ruiyu Ning, Jinghong Peng, Gengchen Mai, and Liang Zhao. 2025. Spatial-rag: Spatial retrieval augmented generation for real-world geospatial reasoning questions. *Preprint*, arXiv:2502.18470.

Liangqi Yuan, Dong-Jun Han, Christopher Brinton, and Sabine Brunswicker. 2025. LLMAP: LLM-assisted multi-objective route planning with user preferences. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 7866–7894, Suzhou, China. Association for Computational Linguistics.

Yinger Zhang, Shutong Jiang, Renhao Li, Jianhong Tu, Yang Su, Lianghao Deng, Xudong Guo, Chenxu Lv, and Junyang Lin. 2026. DeepPlanning: Benchmarking long-horizon agentic planning with verifiable constraints. *Preprint*, arXiv:2601.18137.

Tao Zhe, Rui Liu, Fateme Memar, Xiao Luo, Wei Fan, Xinyue Ye, Zhongren Peng, and Dongjie Wang. 2025. Constraint-aware route recommendation from natural language via hierarchical llm agents. *Preprint*, arXiv:2510.06078.

Table 4: Distribution of tasks across different categories and difficulty levels. Note that L0 is not applicable to *Trip* tasks, as they inherently require multi-location retrieval and navigation.

Task Category	L0	L1	L2	Total
Place Perception	100	72	35	207
Nearby Discovery	104	52	35	191
Routing	100	50	31	181
Trip Planning	–	53	35	88
Total	**304**	**227**	**136**	**667**

A Benchmark Details

A.1 Dataset Composition

In this section, we provide a detailed breakdown of our dataset composition and the underlying data structure. Table 4 outlines the quantitative distribution of tasks across different categories and difficulty levels. Table 5 defines the fields utilized across all task instances with the corresponding descriptions, while Table 6 systematically maps which specific fields are triggered under each task family and difficulty tier. Figure 7 presents a concrete task instance, demonstrating how unstructured user queries, implicit time/social media constraints, and multi-source evidence are structured within our dataset.

A.2 Tool Schema

In this section, we detail the comprehensive toolset provided to the agents, implemented via the Model Context Protocol (MCP). To accurately simulate human-like progressive exploration, the **Standard Agent Toolset** (Table 7) is designed to return lightweight, granular initial results. This forces the evaluated agent to actively invoke drill-down tools (e.g., `place_detail`, `note_detail`) to acquire further insights, mirroring authentic human search behavior.

Conversely, to maximize dataset generation efficiency and prevent any omission of complex constraints, the **Data Engine Agent Toolset** (Table 8) features enhanced tools. These tools return comprehensive records upfront—including social memory, temporal attributes, and tags—and provide specialized tools for random sampling and multi-stop permutations. Note that the descriptions and parameter explanations listed in both tables represent the exact, raw schema descriptions injected into the agents' prompts.

A.3 Database Schema

In this section, we delineate the schema of the underlying environment databases. It is important to note that the fields detailed in Table 9 (Map Database) and Table 10 (Social Media Database) only represent the attributes explicitly retrievable by the agents via the MCP tools. Internal environmental data essential for system operation—such as the topological geometries of road networks (ways/nodes) used for routing, or the tokenized texts and vector embeddings utilized for social media full-text search (`notes_fts`)—are abstracted away and intentionally not exposed to the agents.

B Evaluation Details

B.1 Metric Applicability

Table 11 details the applicability of each evaluation metric across different task categories and difficulty levels within the benchmark.

B.2 Hyperparameters

To minimize variance and ensure strict evaluation reproducibility, the temperature parameter for all evaluator and judge models is consistently set to 0. The temperature for the data engine agent (during task generation) is set to 0.7 to encourage generative diversity. Furthermore, to bound the computational overhead of exploration, a maximum limit of 30 tool invocations is enforced per single task instance.

C Prompt list

In this section, we provide the complete system prompts used to construct the autonomous agents in our framework. Figure 8 presents the foundational system prompt for the evaluated benchmark agent, which outlines its persona, task objectives, and reasoning constraints when navigating the environment. Figure 9 details the system prompt for the Data Engine Agent, emphasizing its specialized directives for comprehensive data retrieval and implicit constraint synthesis.

D AI usage in research.

Annotation. We used LLM-based data engines to synthesize evaluation cases, generate candidate answers, and construct reference tool-use paths from predefined templates and retrieved evidence. All generated cases were manually checked by the

Parent Field	Child Field	Description
type	–	Flag indicating task subtype and difficulty level. (e.g., routing_L1, trip_L2)
question	–	Natural-language user query presented to the benchmark agent.
answer	(*Self*)	Root payload object or array containing the final task results or scalar answer.
	name	Name of a returned place, start/end point, waypoint, or itinerary stop.
	place_id	Map identifier corresponding to the returned location.
	reason	Natural-language explanation for a verification, comparison, route selection, or stop choice.
	start / end	Origin and destination place for route or trip.
	vehicle	Travel mode used for routing (e.g., walking, cycling, driving).
	mode	Route optimization target (e.g., time, distance).
	total_distance_m	Total route distance in meters.
	total_time_s	Total route time in seconds.
	waypoints	List of intermediate stops used for routing or candidate stops used for trip planning.
	optimal_order	Visit order of places in the waypoints list.
ref	(*Self*)	Evidence container attached to an answer or stop to support the decision.
	attribute_key	List of supporting database fields or map tag values used as evidence.
	note_id	List of note IDs cited when decisions rely on social-media evidence.
	implicit_logic	Flag indicating whether the selection depended on implicit or subjective reasoning.

Table 5: Field definitions for query instances across different tasks. Nested fields are logically grouped under their parent objects.

authors for solvability, reference correctness, and answer uniqueness.

Evaluation. We used LLM-as-a-judge models for metrics that require semantic assessment, including Semantic Constraint Satisfaction and Reasoning Purity. These judgments were combined with rule-based metrics and verification tools, and the authors reviewed the evaluation design and outputs.

Writing. During the preparation of this work, the authors used ChatGPT to improve language and readability. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Task Family	Level	Included Fields
Nearby	L0	type, question, answer, name, place_id, reason, ref, attribute_key
	L1	type, question, answer, name, place_id, reason, ref, attribute_key, note_id
	L2	type, question, answer, name, place_id, reason, ref, attribute_key, note_id, implicit_logic
Place	L0	type, question, answer, ref, attribute_key, note_id
	L1	type, question, answer, reason, ref, attribute_key, note_id
	L2	type, question, answer, reason, ref, attribute_key, note_id, implicit_logic
Routing	L0	type, question, answer, start / end, name, place_id, vehicle, mode, total_distance_m, total_time_s, reason
	L1	type, question, answer, start / end, waypoints, optimal_order, name, place_id, vehicle, mode, total_distance_m, total_time_s, reason, ref, attribute_key
	L2	type, question, answer, start / end, waypoints, optimal_order, name, place_id, vehicle, mode, total_distance_m, total_time_s, reason, ref, attribute_key, note_id, implicit_logic
Trip	L1	type, question, answer, start / end, waypoints, optimal_order, name, place_id, vehicle, mode, total_distance_m, total_time_s, reason, ref, attribute_key
	L2	type, question, answer, start / end, waypoints, optimal_order, name, place_id, vehicle, mode, total_distance_m, total_time_s, reason, ref, attribute_key, note_id, implicit_logic

Table 6: Field coverage by task family and difficulty level. Raw instance fields have been mapped to the simplified ontology defined in Table 5. Note that *Trip* lacks an L0 configuration due to the inherit complexity.

Tool Name	Description	Parameters
Map Domain		
place_search	Search places by name with lightweight POI output (place_id, name, type, rating, cost). Supports type, rating, cost, and geo-tag filtering.	**query_name**: REQUIRED: Place name in local language **place_types**: POI types list **min_rating / max_rating**: Min/Max rating (1-5) **min_cost / max_cost**: Min/Max cost per capita (CNY) **geo_tag**: Filter by tag/name keywords list in local language, matches if ANY keyword hits
place_detail	Get full POI information for one place, including name, type, address, hours, rating, cost, tags, and alias.	**place_id**: POI place_id (from place_search or nearby_search)
nearby_search	Search nearby places around a POI or along a route buffer. Standard agent version returns a lightweight POI list with distance.	**center**: Center: POI place_id (str) OR Route object {'start': 'ID', 'end': 'ID', 'vehicle': 'driving'} **radius_meters**: Search radius/buffer in meters (default 100) **place_types**: POI types list **min_rating / max_rating**: Min/Max rating (1-5) **min_cost / max_cost**: Min/Max cost per capita (CNY) **geo_tag**: Filter by tag/name keywords list in local language, matches if ANY keyword hits
routing	Calculate an optimal route between two POIs and return total distance/time, geometry, and turn instructions.	**start_place_id**: Start POI place_id **end_place_id**: End POI place_id **avoid_way_ids**: Way IDs to avoid (ways.gid) **mode**: Route mode: 'time' (shortest time) or 'distance' (shortest distance) **vehicle**: Travel mode: 'walking', 'cycling', or 'driving'
submit_answer	Submit the final structured JSON answer. The evaluated agent must call this last instead of printing plain JSON.	**result**: Your final answer as a JSON object or array following the required format
Social Media Domain		
search_notes	Search social media posts by keyword logic and return a lightweight post list (note_id, title, time, counts).	**keyword**: Search keyword, multi keywords must be separated by 'AND'/'OR'/'NOT' logic
note_detail	Get full post details for one social media note, including description and comments.	**note_id**: Post ID (from search_notes result)

Table 7: The Standard Agent Toolset. Tools are designed to be lightweight, requiring the evaluated agent to execute multi-step reasoning and drill-down tool invocations to fulfill complex queries.

Tool Name	Description	Parameters
Map Domain		
place_search	Search places by name with enhanced filters and return full POI records. Data Engine version can filter by both static geo tags and social-memory keywords.	**query_name**: REQUIRED: Place name in local language **place_types**: POI types list, Separated by commas (,) **min_rating / max_rating**: Min/Max rating (1-5) **min_cost / max_cost**: Min/Max cost per capita **geo_tag**: Filter by tag/name keywords list **social_mem**: Filter by social insights in local language
nearby_search	Search nearby places around a POI or along a route buffer and return full POI records, including address, opentime_week, tag, alias, long_mem, and short_mem.	**center**: Center: POI place_id (str) OR Route object {'start': 'ID', 'end': 'ID', 'vehicle': 'driving'} **radius_meters**: Search radius/buffer in meters **place_types**: POI types list, Separated by commas **min_rating / max_rating**: Min/Max rating (1-5) **min_cost / max_cost**: Min/Max cost per capita **geo_tag**: Filter by tag/name keywords list **social_mem**: Filter by long_mem/short_mem keywords (social media insights) in local language
routing	Calculate an optimal route between two POIs and return total distance/time, geometry, and turn instructions.	**start_place_id**: Start POI place_id **end_place_id**: End POI place_id **avoid_way_ids**: Way IDs to avoid (ways.gid) **mode**: Route mode: 'time' or 'distance' **vehicle**: Travel mode: 'walking', 'cycling', or 'driving'
sample_place	Randomly sample POIs from the map database for prompt initialization. Returns full POI information including memory fields.	**num_samples**: Number of random POIs to sample (typically 1-50)
multi_stop_routing	Enumerate and score route permutations for multi-stop trip planning. Supports open trips and fixed start/end settings.	**stop_ids**: List of POI place_ids to visit (typically 2-6 stops) **start_id**: Fixed start place_id (must be in stop_ids) **end_id**: Fixed end place_id (must be in stop_ids) **vehicle**: Travel mode: 'walking', 'cycling', 'driving' **mode**: Optimize: 'time' or 'distance'
submit_answer	Submit the final structured JSON answer generated by the Data Engine agent.	**result**: Your final answer as a JSON object or array following the required format
Social Media Domain		
search_notes	Search social media posts and return full post information with comments. Required when generated data cites social-memory evidence.	**keyword**: Search keyword, multi keywords must be separated by 'AND'/'OR'/'NOT' logic

Table 8: The Data Engine Agent Toolset. Enhanced MCP tools directly return highly dense and comprehensive records to ensure complete information retrieval during dataset generation without information loss.

Returned Key	Exposed by Tool(s)	Explanation
osm_id	place_search, nearby_search, sample_place, place_detail	Primary POI identifier.
name	place_search, nearby_search, sample_place, place_detail	POI name.
place_type	place_search, nearby_search, sample_place, place_detail	Categorization of the POI (e.g., Auto Service|Dedicated Charging Station).
address	place_search (DE), nearby_search (DE), sample_place, place_detail	Physical address of the POI.
opentime_week	place_search (DE), nearby_search (DE), sample_place, place_detail	Weekly business opening hours, if available.
rating	place_search, nearby_search, sample_place, place_detail	User rating score.
cost	place_search, nearby_search, sample_place, place_detail	Average consumption cost per person.
tag	place_search (DE), nearby_search (DE), sample_place, place_detail	Static attribute tags associated with the POI.
alias	place_search (DE), nearby_search (DE), sample_place, place_detail	Known aliases or alternate names.
long_mem	place_search (DE), nearby_search (DE), sample_place	Long social-memory summary synthesized and attached to the POI.
short_mem	place_search (DE), nearby_search (DE), sample_place	Short social-memory summary synthesized and attached to the POI.
steps[].way_ids	routing	Underlying routing edge identifiers exposed in step-level route output.
road_name	routing	Road names used in step-level route navigation instructions.

Table 9: Retrievable fields from the Map Database. Fields marked with (DE) are exclusively exposed upfront to the Data Engine Agent during the initial search.

Returned Key	Exposed by Tool(s)	Explanation
note_id	search_notes, note_detail	Primary identifier for the social media post.
title	search_notes, note_detail	Title of the post.
desc	search_notes (DE), note_detail	Full description and textual content of the post.
time	search_notes, note_detail	Publication timestamp of the post.
liked_count	search_notes, note_detail	Number of likes received by the post.
collected_count	search_notes, note_detail	Number of times the post has been saved or favorited.
comment_count	search_notes, note_detail	Total number of comments under the post.
share_count	search_notes (DE), note_detail	Number of times the post has been shared.
tag_list	search_notes (DE), note_detail	Comma-separated list of tags or hashtags attached to the post.
comment_id	search_notes (DE), note_detail	Unique identifier for an individual comment.
comments[].time	search_notes (DE), note_detail	Timestamp of when the specific comment was posted.
comments[].content	search_notes (DE), note_detail	The text content of the specific comment.
comments[].like_count	search_notes (DE), note_detail	Number of likes received by the specific comment.

Table 10: Retrievable fields from the Social Media Database. The Data Engine Agent (DE) retrieves detailed comment data directly from the search tool to ensure data integrity during generation.

Metric	Place Perception			Nearby Discovery			Routing			Trip Design	
	L0	*L1*	*L2*	*L0*	*L1*	*L2*	*L0*	*L1*	*L2*	*L1*	*L2*
Attr Accuracy	✓	✓	✓				✓	✓	✓	✓	✓
POI Hit Rate				✓	✓	✓	✓	✓	✓	✓	✓
Routing Deviation							✓	✓	✓	✓	✓
Order Consistency										✓	✓
Semantic Constraint		✓	✓		✓	✓		✓	✓	✓	✓
Info Grounding	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tool Efficiency	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Reasoning Purity	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Token Usage	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 11: Applicability matrix of evaluation metrics across different task types and difficulty levels.

Example for Task Instance


```
{
  "type": "trip_L2",
  "question": "I'm leaving work at Wuhan Zhongke Kaiwu Technology Co., Ltd. and
      driving to meet a friend at Hongbo Gate. It's 11:02 a.m. now. On the way, I
       want to stop somewhere to play with some short-legged dogs, and also visit
       the popular photo spot where I can walk above the lake and waiting for the
       sunset. Plan the quickest driving route."
  ,
  "answer": {
    "start": {
      "name": "Wuhan Zhongke Kaiwu Technology Co., Ltd.",
      "place_id": "B0FFG2DQ5U",
      "selectedreason": "User-specified origin."
    },
    "end": {
      "name": "Hongbo Gate",
      "place_id": "B0LR9NNY1H",
      "reason": "User-specified destination."
    },
    "waypoints": [
      {
        "name": "Corgi & Shiba Inu Pet Cafe (Jiedaokou Branch)",
        "place_id": "B0FFMBUXIP",
        "reason": "Dedicated Corgi and Shiba Inu cafe that verified through
            social media, meeting the requirements for playing with short-legged
            dogs. Business hours are 11:00-21:00, arrival time is approximately
            11:06, within opening hours.",
        "ref": {
          "note_id": "5de2596400000000100bc82,68fee6b500000000301305a",
          "attribute_key": "tag:corgi,opentime_week",
          "implicit_logic": "True"
        }
      },
      {
        "name": "Lingbo Gate Trestle",
        "place_id": "B0KANCP2BM",
        "reason": "A popular viewing spot, you can walk on the trestle on the
            lake and is a popular spot to enjoy the sunset. Social media notes
            indicate that this is a well-known spot for sunrise/sunset viewing.",
        "ref": {
          "note_id": "6769ff6700000000900fe52,6759819b000000000203527b",
          "implicit_logic": "True"
        }
      }
    ],
    "optimal_order": [
      {
        "name": "Wuhan Zhongke Kaiwu Technology Co., Ltd.",
        "place_id": "B0FFG2DQ5U"
      },
      {
        "name": "Corgi & Shiba Inu Pet Cafe (Jiedaokou Branch)",
        "place_id": "B0FFMBUXIP"
      },
      {
        "name": "Lingbo Gate Trestle",
        "place_id": "B0KANCP2BM"
      },
      {
        "name": "Hongbo Gate",
        "place_id": "B0LR9NNY1H"
      }
    ],
    "vehicle": "driving",
    "mode": "time",
```

```
    "total_distance_m": 8618,
    "total_time_s": 726
  }
}
```

Figure 7: An example of task instance in LifePlanner.

System Prompt for the Evaluated Agent

```
You are a geographic information assistant. Use the available tools to answer the
     user's question.

CRITICAL RULES:
1. NEVER hallucinate or fabricate data. ALL reasoning MUST be grounded in tool
    return results.
2. You MUST call tools to get real data before answering.
3. If your answer references social media insights, you MUST include ALL relevant
     note_ids in the ref field. 4. Use the EXACT values (names, IDs, distances,
    times) returned by tools.
5. After completing your analysis, you MUST call submit_answer() to deliver your
    final answer as a structured JSON object. Do NOT output JSON as plain text -
    always use submit_answer().


---

You MUST call submit_answer(result=<your_answer>) where <your_answer> is a single
     JSON object containing ALL of the following fields:
{
    "start": {
      "name": "{NAME}",
      "place_id": "{PLACE_ID}",
      "reason": "Why this place was selected based on criteria."
    },
    "end": {
      "name": "{NAME}",
      "place_id": "{PLACE_ID}",
      "reason": "Why this place was selected based on criteria."
    },
    "waypoints": [
      {
        "name": "{NAME}",
        "place_id": "{PLACE_ID}",
        "reason": "Why this place was selected based on criteria.",
        "ref": {
          "note_id": "{NOTE_ID1},{NOTE_ID2},...",
          "attribute_key": "comma-separated keys. tag:keyword,rating,cost",
          "implicit_logic": "True"
        }
      }
    ],
    "optimal_order": [
      {"name": "{NAME}", "place_id": "{PLACE_ID}", "visit_order": 1},
      ...
    ],
    "vehicle": "{VEHICLE}",
    "mode": "{MODE}",
    "total_distance_m": <number>,
    "total_time_s": <number>,
    "constraints_satisfied": ["description of each constraint met"]
  }

IMPORTANT: Pass the ENTIRE object above as the `result` parameter. Do NOT split
```

```
    fields into separate parameters. Use the EXACT field names shown above.


---

Now answer the following question:
[benchmark question text inserted here at runtime]
```

Figure 8: The complete system prompt provided to the evaluated agent, defining its reasoning guidelines and tool-use constraints.

System Prompt for the Data Engine Agent

```
You are a Synthetic Data Generation Agent for a Map Assistant Benchmark.
Your goal is to reverse-engineer realistic multi-stop trip planning queries based
     on the provided map environment data.
Follow the "Core Principles": Groundedness, Diversity, strict JSON format,
    Uniqueness, output Language, and Mandatory Tool Calls.

**Core Principles:**
1. **Groundedness**: Your generated questions MUST be answerable using routing
    results from the tools. Do not hallucinate places or routes not present in
    the tool output.
2. **Diversity**:
   - Vary the phrasing (e.g., "Plan a trip visiting...", "I need to go to these
       places...", "What's the best order to visit...").
   - Vary vehicle types: walking, cycling, driving. Vary constraints: time
       windows, ordering, efficiency.
   - **CRITICAL**: For L2, diversify the tag/mem conditions. Explore various
       categories and attributes.
3. **Format**: Output strictly in the specified JSON format. No markdown fencing
    (```json) outside the JSON block.
4. **Necessity of Uniqueness**:
   - The selected places must be the ONLY ones matching the criteria in the
       entire map region.
   - Verify uniqueness using `place_search` with strict filters.
5. **Language**: ALL generated content MUST be in ENGLISH, except for place names
     and social media content (which should remain in their original language).
6. **Mandatory Tool Calls**:
   - You MUST call `multi_stop_routing()` to get ACTUAL total_distance_m and
       total_time_s values. DO NOT fabricate these numbers.
   - The `total_distance_m` and `total_time_s` in your final output MUST exactly
       match the tool's return values.
   - Skipping tool calls and guessing route metrics is STRICTLY FORBIDDEN.


**Map Region**: {Current City}. Use local language of current city for place
    names and queries in this region.


---

# CURRENT TASK: L2: Mixed Reasoning + Dynamic Waypoint Selection

**Description**: Select waypoints based on tag/mem conditions, then plan optimal
    route.

# OUTPUT FORMAT
{
  "type": "trip_reasoning",
  "question": "A complex query requiring selection of waypoints based on implicit
      criteria, then route planning.",
  "answer": {
```

```
    "start": {
        "name": "{NAME}",
        "osm_id": "{OSM_ID}",
        "reason": "Why this place was selected based on criteria.",
        "ref": {
          "note_id": "{NOTE_ID1},{NOTE_ID2},...",
          "attribute_key": "comma-separated keys. tag:keyword,rating,cost",
          "implicit_logic": "True"
        }
      }
    "end": {
        "name": "{NAME}",
        "osm_id": "{OSM_ID}",
        "reason": "Why this place was selected based on criteria."
        "ref": {
          "note_id": "{NOTE_ID1},{NOTE_ID2},...",
          "attribute_key": "comma-separated keys. tag:keyword,rating,cost",
          "implicit_logic": "True"
        }
      }
    "waypoints": [
      {
        "name": "{NAME}",
        "osm_id": "{OSM_ID}",
        "reason": "Why this place was selected based on criteria.",
        "ref": {
          "note_id": "{NOTE_ID1},{NOTE_ID2},...",
          "attribute_key": "comma-separated keys. tag:keyword,rating,cost",
          "implicit_logic": "True"
        }
      }
    ],
    "optimal_order": [
      {"name": "{NAME}", "osm_id": "{OSM_ID}", "visit_order": 1},
      ...
    ],
    "vehicle": "{VEHICLE}",
    "mode": "{MODE}",
    "total_distance_m": <number>,
    "total_time_s": <number>,
    "constraints_satisfied": ["description of each constraint met"]
  }
}

## REF FIELD RULES (MUST follow):
- **tag content**: Write `"tag:{specific_text}"`. Never just `"tag"`.
- **Other DB fields**: Write field name directly, e.g. `"rating"`, `"cost"`, `"
    opentime_week"`.
- **Multiple sources**: Comma-separate, e.g. `opentime_week"`, `"rating,cost"`.
- **note_id**: REQUIRED when reasoning uses any social media insight (long_mem/
    short_mem content). Must be real ID(s) from `search_notes()`. Comma-separate
    multiple IDs.
- The long_mem/short_mem fields are summaries of social posts - always trace back
     to the original note_id(s).
- **NEVER** write `"attribute_key": "long_mem"` or `"short_mem"`. Find the source
     note and cite the actual DB fields that informed the decision.
- If no matching note is found via `search_notes()`, discard that place and
    select a different one.


---

# TASK INSTRUCTIONS

**Note**: The system has provided 10 sampled places above for REFERENCE only.
You MUST call tools (`place_search`, `multi_stop_routing`, `search_notes`) to
    validate and get actual data.
```

```
Choose ONE scenario:

--- SCENARIO A: Full Discovery Trip (No Fixed Points) ---
*Goal*: Find N places matching criteria, then plan optimal route visiting all.

[A1. Define Selection Criteria]:
  Design a multi-constraint question combining 2+ info sources (tag, rating, cost
      , opentime, social media insights). The locations sampled are used as
      sources of inspiration.
  Good questions layer MULTIPLE constraints, e.g.:
  - "pet cafe with Corgis" -> geo_tag = "Corgis" + place_type = "Food & Beverages
      "
  - "affordable KTV with good reviews" -> rating + cost + social notes

[A2. Search and Validate Uniqueness (MANDATORY)]:
  **YOU MUST CALL THIS TOOL - DO NOT SKIP**
  Call `place_search` with criteria in A1.
  **CRITICAL**: Result count must be 3-5 (>=3 required - with only 2 waypoints
      there is no ordering to optimize, it degenerates to a simple A -> B routing
       question).
  - If too many (>5): add more constraints.
  - If too few (<3): relax constraints or change criteria.
  These become the waypoints.

[A3. Compute Optimal Order (MANDATORY)]:
  **YOU MUST CALL THIS TOOL - DO NOT SKIP**
  Call `multi_stop_routing(stop_ids=[selected IDs], vehicle={VEHICLE}, mode={MODE
      })`.
  This returns ALL permutations sorted by total time/distance.
  Extract the ACTUAL `total_distance_m` and `total_time_s` from the tool response
      .
  Select the first (optimal) route from the result.

[A4. Question Formulation]:
  Construct a persona-based question that implies the criteria.
  Example: "I'm a dog owner. Plan a cafe-hopping route to all pet-friendly coffee
       shops in the area."
  Do NOT directly mention the tag-use natural language.

[A5. Reference Verification (MANDATORY for social insights)]:
  **IF using social insights (long_mem/short_mem), YOU MUST CALL `search_notes`**
  For each selected waypoint using social insights:
  - Call `search_notes` with relevant keywords.
  - Extract {NOTE_ID}s.

[A6. Final Output]:
  Include `waypoints` with `reason` and `ref`.
  **CRITICAL**: Use the EXACT values from tool response for total_distance_m and
      total_time_s.

--- SCENARIO B: Constrained Trip (Fixed Start/End, Constraints FORCE Detour) ---
*Goal*: Given fixed start/end, find N intermediate places matching criteria, then
     plan route where constraints FORCE a different order than the unconstrained
    optimal.

[B1. Select Fixed Endpoints]:
  From the pre-sampled places below, select {START} and {END} place.

[B2. Define Selection Criteria for Waypoints]:
  Design multi-constraint criteria combining 2+ info sources (tag, rating, cost,
      opentime, social notes). Each waypoint should require distinct reasoning.
  **CRITICAL**: >=2 intermediate waypoints required - with only 1, the order is
      trivially start -> waypoint -> end and there is nothing to optimize.

[B3. Search and Validate (MANDATORY)]:
  **YOU MUST CALL THIS TOOL - DO NOT SKIP**
  Call `place_search` with criteria in B2.
  Verify uniqueness: result count should be 2-4 (>=2 intermediates required -
```

863

```
      with only 1, the visit order is trivial).
  These become intermediate waypoints.

[B4. Compute Unconstrained-Optimal First (MANDATORY)]:
  **YOU MUST CALL THIS TOOL - DO NOT SKIP**
  Call `multi_stop_routing(stop_ids=[START_ID, ...intermediate IDs..., END_ID],
      start_id={START_ID}, end_id={END_ID}, ...)`.
  This returns ALL orderings with fixed start/end, sorted by total time/distance.
  **Record the #1 (unconstrained-optimal) ordering.**

[B5. Design CONFLICTING Constraints]:
  **CRITICAL**: Design constraint(s) that ELIMINATE the unconstrained-optimal
      ordering from B4.
  Use place data (opentime_week, logical dependencies, social insights) to
      justify constraints that force a different order.

  **VALIDATION**: Verify the unconstrained-optimal ordering VIOLATES your
      constraint. If it doesn't, redesign.

[B6. Select Constrained-Optimal]:
  From the sorted permutations in B4, find the FIRST ordering that satisfies ALL
      constraints.
  **FINAL CHECK**: It MUST differ from the unconstrained-optimal. If same, go
      back to B5.
  Extract the ACTUAL `total_distance_m` and `total_time_s`.

[B7. Question Formulation]:
  Construct a persona/scenario question with implicit criteria and constraints.
  Do NOT reveal that the constraint forces a detour.

[B8. Reference Verification (MANDATORY for social insights)]:
  **IF using social insights (long_mem/short_mem), YOU MUST CALL `search_notes`**
  Verify social insights with `search_notes`.

[B9. Final Output]:
  Include `waypoints` with `reason` and `ref`.
  **CRITICAL**: Use the EXACT values from tool response for total_distance_m and
      total_time_s.


---

# Examples

## Example 1

skipped, see Figure: Example for Task Instance

## Example 2

skipped, see Figure: Example for Task Instance

## Example 3

skipped, see Figure: Example for Task Instance


---

**System has automatically sampled 10 places to help you get started:**

`sample_place(num=10)` returned:
```json
{
  "pois": [
    {
      "osm_id": "B001B1H2DS",
      "name": "Luoying Lake",
```
```

```
      "place_type": "Scenery Spot|Tourist Attraction",
      "address": "No. 16 Luojia Mountain",
      "opentime_week": "Monday-Sunday 00:00-24:00",
      "rating": 4.0,
      "cost": null,
      "tag": "Facility: Fountain; Rule: Shuttle Bus Stop",
      "alias": "Jian Lake",
      "long_mem": "Atmosphere: Quiet, mirror-like surface; Scenery: Water lilies,
           lotus, winter snow; Feedback: Beautiful view",
      "short_mem": "Crowd: Extremely crowded during Cherry Blossom season"
    },
    {
      "osm_id": "B0FFI211Q3",
      "name": "COACH (Chicony Plaza)",
      "place_type": "Shopping|Brand Bags and Suitcases Store",
      "address": "1F Chicony Plaza, No. 6 Luoyu Road (Near Metro Jiedaokou
          Station Exit B)",
      "opentime_week": "Monday-Sunday 10:00-22:00",
      "rating": 4.2,
      "cost": null,
      "tag": "Brand: COACH; Category: Luxury Bags; Facility: Mall Parking",
      "alias": "",
      "long_mem": "Atmosphere: Premium, clean; Service: Professional staff;
          Target: High-end shopping",
      "short_mem": "Event: Presale (2nd-23rd) up to 45% off, Official sale (24th
          -25th) spend 500 get 500; Special Hours: Open until 23:00 on 24th-25th"
    },
    {
      "osm_id": "B0LAJRQMJH",
      "name": "Sanqi Homebar (Jiedaokou)",
      "place_type": "Sports & Recreation|Pub",
      "address": "Room 2103, Tower C, Fuhua Building (90m from Metro Jiedaokou
          Station Exit B)",
      "opentime_week": "Tuesday-Sunday 19:00-02:00",
      "rating": 3.9,
      "cost": 138,
      "tag": "Drinks: Cocktails, Beer; Service: Karaoke, Board games; Facility:
          Private rooms, Terrace, Smoking area; Rule: Women-friendly, Reservation
           required, Closed on Mondays",
      "alias": "",
      "long_mem": "Atmosphere: Cozy, blues lighting, relaxing; Audience: Gen Z,
          university students, introvert-friendly; Service: Bartender
          recommendations; Feedback: Great cocktails, affordable",
      "short_mem": "Event: Valentine's Day limited edition cocktails available"
    },
    ... 7 samples skipped
  ]
}
```

Now, please generate a new data sample based on the sampled places above.

Figure 9: The specialized system prompt utilized by the Data Engine Agent to orchestrate data retrieval and query generation.