AI-Driven Intelligent Itinerary Generation: A Conceptual Framework Integrating Semantic Reasoning and Geospatial Intelligence

Cheng-Yun Ho[1] and Cheng-Yuan Ho[2]

[1] Department of Mechanical Engineering, Lunghwa University of Science and Technology, Taoyuan City, 333326, Taiwan
[2] Information Management, National Taiwan University, Taipei City, 106319, Taiwan
[1] cyho@mail.lhu.edu.tw

Abstract. This paper introduces a conceptual framework for intelligent itinerary generation that integrates two complementary dimensions: *semantic reasoning* enabled by large language models, and *geospatial intelligence* enabled by Google Maps-based factual validation. The framework is grounded in two methods: (A) a recursive filling large language model (LLM) + Google Maps system representing an incremental synthesis strategy, and (B) a full-stack Gemini + Google Maps implementation that performs point-to-point route verification. At the conceptual level, this work compares their design philosophies, complementary strengths, and suitable deployment scenarios, and further generalizes a high-level design principle: *semantic intent, spatial validation, layered synthesis*. The paper also discusses conceptual evaluation dimensions, potential hybrid strategies, and future research opportunities, including contextual sensing integration, learning-to-rank extensions, and human-AI co-creation interaction design.

Keywords: semantic reasoning, geospatial intelligence, itinerary generation, recursive synthesis, point-to-point routing.

1 Introduction

Intelligent itinerary generation is a representative case of human-artificial intelligence (AI) collaborative decision-making: the system must integrate user intention, preference, and real-world geographic and temporal constraints, in order to produce itineraries that are both executable and compelling. With the rapid proliferation of large language models (LLMs), researchers and engineering teams have increasingly attempted to employ LLMs to address the semantic layer requirements (e.g., preference interpretation, candidate attraction generation, natural-language itinerary narration). However, the content generated by LLMs often lacks spatial precision, leading to location ambiguity or misplacements. Consequently, geospatial services such as Google Maps Geocoding, Places, and Routes Application Programming Interfaces (APIs) must serve as a validation layer to ensure feasibility and navigational credibility.

This paper uses two methods as the empirical starting point: one adopts a recursive filling paradigm to incrementally construct itineraries, while the other emphasizes point-to-point route retrieval as the grounding mechanism for route feasibility. Rather than focusing on implementation-level engineering specifics, this study aims to propose a conceptual framework that can be reused in subsequent research or system engineering work, and examines the two strategies in terms of their cognitive assumptions, engineering implications, and usage contexts.

2 Related Work

Conceptually, prior work relevant to this study can be grouped into three streams: (1) the role of generative AI in application-level systems, (2) geospatial computation and route planning methods, and (3) human-AI co-creation and context-aware recommendation. Generative AI and recommendation systems: Recent research has demonstrated strong capability of LLMs in natural-language understanding, preference extraction, and candidate generation [1]-[3]. At the same time, prior studies consistently highlight the model's vulnerability to hallucination, particularly in factual and spatial precision [4]-[5]. As a result, although LLMs can provide semantically rich candidates, external factual sources remain necessary for validation [6]-[7]. Both implementation reports considered in this work adopt LLMs as the core mechanism for candidate generation, consistent with this emerging trend [8]-[9]. Geospatial intelligence and route optimization: Traditional itinerary and routing optimization research includes traveling salesman problem (TSP), vehicle routing problem (VRP), and time-window constrained heuristics [10]-[12]. In production systems, commercial geospatial APIs such as Google Maps provide reliable place resolution, distance estimation, and business-hour metadata, serving as the primary source of real-world constraints [13]. The second implementation report adopts pairwise routing and heuristic ranking as its architectural basis, representing an engineering instantiation where route services are treated as feasibility criteria [8], [14]-[15]. Human-AI co-creation and context awareness: Research in recommender systems suggests that user acceptance correlates strongly with controllability [10], [14]. Incremental or recursive generation, combined with mid-process user intervention, improves transparency and perceived trustworthiness [11]. The recursive filling strategy in the first implementation report embodies this human-AI interaction philosophy [6].

3 Conceptual Core Model

We propose a three-layer conceptual model composed of the Semantic Layer, the Geospatial Validation Layer, and the Generation and Interaction Layer. Semantic Layer parses the user's natural-language input-including travel intent, preferences, and non-negotiable constraints-and generates candidate locations, temporal allocation suggestions, or alternative options based on semantic patterns and commonsense knowledge. The strength of LLMs lies in preference abstraction and coherent textual description generation. Geospatial Validation Layer grounds semantic outputs into the physical world through geocoding, business-hour verification, and feasibility checks over inter-location routing. This layer treats geospatial APIs as authoritative sources of real-world truth. Generation and Interaction Layer integrates semantic outputs and geospatial constraints to produce executable itineraries using different reasoning strategies (e.g., recursive fill or point-to-point optimization). It provides editable user interface (UI)-level affordances to support micro-adjustment and human-in-the-loop refinement. These three layers correspond to the system engineering triad: front-end UI, back-end orchestration logic, and external intelligence services.

(A) Architectural Overview: The overall architecture is illustrated in Fig. 1. The user interacts with a React-based Web interface to specify destination city, date ranges, and preference constraints. These inputs are transmitted to a Node.js back-end using Axios or Fetch, where command parsing, schema validation, and model orchestration take place. The back-end communicates with external intelligence modules-including Google Maps APIs and LLM APIs (e.g., Gemini or Llama)-with all data exchanged in JSON to ensure structural consistency across modules.

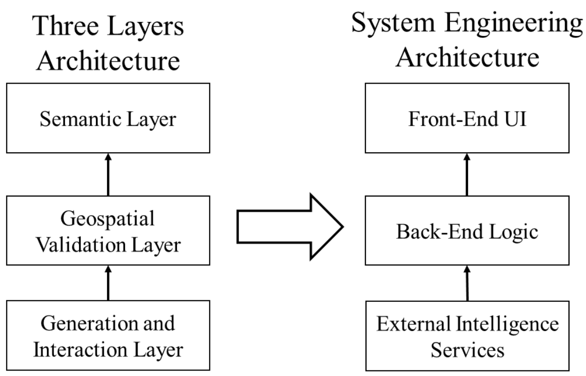

Fig. 1. Architectural overview.

(B) Semantic Layer: The Semantic Layer functions as the cognitive engine of the system. It parses the user's natural-language input-including places, preferences, and activity categories-and transforms these elements into structured prompt templates that are submitted to the LLM to generate candidate locations and itinerary-level descriptive text. In implementation, this layer operates through a Node.js back-end middleware module that communicates with LLM APIs, and applies Zod schemas to verify the structural integrity of returned outputs, preventing undefined fields or semantic inconsistencies. The strengths of the LLM at this layer lie in (i) preference inference at the semantic level and (ii) automatic production of coherent natural-language explanations (e.g., "Day 1: Visit Museum"). Consequently, the system produces dual outputs: semantic reasoning and structured representations.

(C) Geospatial Validation Layer: The Geospatial Validation Layer grounds the semantic outputs into physical feasibility. Each candidate location is converted into latitude-longitude coordinates via the Geocoding API, then cross-validated through the

Places API to obtain business hours, ratings, and category information. The system subsequently invokes the Directions or Routes API to compute inter-location routing, distances, and travel-time estimates, forming a Feasibility Matrix for itinerary construction. This layer implements asynchronous request patterns and caching to reduce repeated API calls. If the LLM returns ambiguous place names (e.g., “Central Park”), string-similarity matching is used to select candidates most consistent with the specified city context, enabling geospatial correction.

(D) Generation and Interaction Layer: The Generation and Interaction Layer integrates semantic suggestions and geospatial constraints into visualizable itineraries, also supporting user interaction. Rendering is handled through a React-based front-end, which exchanges state continuously with the back-end via RESTful APIs or Web Sockets. Two generation strategies are supported: (i) recursive filling, where the LLM incrementally completes one day at a time, and (ii) route-based sequencing, which constructs the day-level schedule based on shortest-path optimization. This layer also provides visualization components-including dynamic map rendering (Google Maps JS SDK) and day-card views-and supports direct reordering of locations on the front-end, where “regenerate” triggers new back-end inference, enabling a human-in-the-loop refinement loop.

(E) Data and Control Flow: End-to-end processing proceeds as follows: User input → front-end JSON request → back-end receives request → LLM semantic generation → geospatial verification → merged output returned to UI → user edits → new iteration. The design principle behind this workflow is layered cooperation: each layer specializes in its own function, exchanging results through well-formed data structures, minimizing coupling, and enabling drop-in replacement of models or APIs.

4 Comparative Strategies

In framing the conceptual architecture of intelligent itinerary generation, this study identifies two representative paradigms emerging from prior literature: semantic-first iterative synthesis and geography-first route grounding. These paradigms correspond to two distinct epistemic logics-one driven by semantic reasoning and iterative refinement, and the other grounded in spatial feasibility and optimization. The recursive filling strategy (Method A) draws its theoretical foundation from recent advances in stepwise reasoning and tool-augmented language modeling, such as Chain-of-Thought prompting [3], the ReAct framework [4], and SayCan [5]. These works collectively demonstrate that when large language models externalize intermediate reasoning steps and interface with factual environments, both interpretability and factual accuracy improve. Applied to itinerary generation, such “semantic–action–verification” cycles enable the model to progressively fill missing contextual elements while maintaining user alignment. Furthermore, TRIP-PAL [6] exemplifies the integration of LLMs and automated planners in multi-step travel itinerary planning, providing empirical grounding for iterative synthesis. Method A thus represents a semantic reasoning–driven paradigm, emphasizing explain ability, interactivity, and human-in-the-loop adaptability.

Conversely, the point-to-point route grounding strategy (Method B) originates from classical geospatial intelligence and routing optimization literature. From foundational studies of the Vehicle Routing Problem [12] to recent surveys on route recommendation [8], the field emphasizes computational frameworks based on distance matrices, adjacency graphs, and heuristic multi-objective optimization to ensure spatial feasibility. In practical engineering, commercial APIs such as Google Maps [13] provide the factual grounding-coordinates, distances, transit times, and business-hour metadata-necessary for real-world viability. Recent works on fairness-aware routing [15] further highlight that geography-first methods can ensure both feasibility and diversity. Method B therefore embodies a geographical constraint-driven paradigm, prioritizing physical executability and navigational precision. By juxtaposing these two paradigms, this study seeks to explore the complementarity between semantic reasoning and geospatial grounding within a unified framework. The comparative design enables us to examine how semantic generation can be anchored in spatial constraints, and how geographical optimization can, in turn, benefit from semantic preference modeling. This conceptual duality forms the foundation of the proposed Semantic-Geospatial-Interaction integration model.

At the conceptual level, this study abstracts the two methods into two classes of high-level generation strategies, and uses these strategies as the behavioral grammar of the Generation and Interaction Layer. The first class is the "Recursive Fill/Incremental Synthesis" strategy, whose core operational logic can be characterized as a cyclic pattern of "Semantic Layer → Geospatial Validation Layer → back-to-Semantic." The system begins with a coarse planning frame-such as origin, destination, total number of days, or thematic intent-and at each iteration the LLM generates only partial hypotheses, which are then screened by the geospatial validation layer for feasibility. If feasible, the candidate is appended to the existing sequence; if not, control returns to the semantic layer and the LLM is prompted to revise that segment. In other words, recursive fill is a semantic-first generation logic. In terms of the three-layer model: the semantic layer serves as the hypothesis generator; the validation layer serves as the gate keeper that adjudicates feasibility; and the behavioral layer manifests as a multi-round feedback loop instead of a single-shot generation. Empirically, recursive strategies exhibit high transparency, enabling users to witness the system's incremental "reasoning trajectory," thereby resulting in stronger human-in-the-loop leverage. However, its limitations are also explicit: semantic noise tends to accumulate (error propagation), and each iteration requires API invocation, implying higher computational cost and latency.

The second class of strategy is the "Point-to-Point Route Grounding" approach, whose procedural ordering is the reverse of the previous one. Its logic is not semantic-first, but geography-first. The system first constructs a reachability adjacency graph at the geospatial validation layer (e.g., distance matrix, polyline, or pairwise routing feasibility), then the semantic layer assigns semantic or preference scores to each candidate node, and finally the itinerary is derived via ranking or merging. In this design, the validation layer assumes computational and decision primacy, while the semantic layer merely contributes ranking scores; the behavioral layer tends to be single-shot rather than iterative. This strategy aligns naturally with navigation-oriented logic-map UI, polyline visualization, turn-by-turn semantics. Its limitation is that the quality of the candidate pool sets the ceiling of overall performance: if the candidate pool is poor, system quality collapses accordingly. Moreover, because its default form is single-shot, its human-editable affordance is weaker than that of the recursive strategy unless an additional incremental UI is explicitly introduced.

5 Conceptual Validation

In this study, Methods A and B are not introduced as standalone engineering products, nor as specific implementations to be replicated. Instead, they serve as representative instantiations of the two dominant paradigms observed across existing itinerary-generation and LLM-based planning approaches: the semantic-first iterative synthesis paradigm and the geography-first route-grounding paradigm. Accordingly, the purpose of referring to these systems is not to detail their implementation mechanics, but rather to use them as behavioral evidence for validating the proposed conceptual framework. Specifically, Method A's recursive filling workflow (see Fig. 2) exhibits a clear cyclic interaction between the semantic layer and the geospatial validation layer, yielding an observable pattern of "semantic expansion → geospatial correction → semantic refinement." This behavior reflects characteristics widely discussed in the literature on stepwise reasoning and intermediate-state externalization, indicating that Method A exemplifies the semantic-first strategy in a real deployment scenario rather than representing an isolated algorithmic choice.

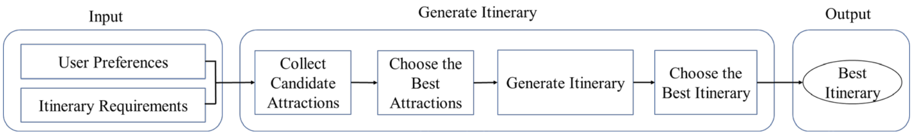

Fig. 2. Method A's conceptual validation.

In contrast, Method B (see Fig. 3) centers its operation around geospatial constraints, constructing feasibility matrices through Google Maps APIs and applying semantic scoring only after spatial viability has been ensured. This architecture echoes principles found in classical routing optimization research, including reachability analysis, polyline-based route representation, and structured indexing via place_id. Consequently, Method B is treated as a prototypical embodiment of the geography-first paradigm rather than as a single handcrafted solution. For these reasons, the two systems function as instances that reveal how different lead-layer configurations influence behavioral convergence, feasibility filtering, semantic alignment, and human-AI interaction dynamics within intelligent itinerary-generation workflows.

Integrating observations across all three layers, three conceptual validations are supported: (1) modular boundaries across the three-layer model allow cross-layer cooperation without dependency on specific models or APIs, demonstrating architectural extensibility; (2) semantic-geospatial-behavioral coupling forms a closed-loop dynamic exhibiting behavioral convergence, enabling progressive correction toward outputs that are both semantically coherent and physically feasible; (3) human-in-the-loop design not only increases error correction efficiency, but also provides implicit feedback benefiting future model re-training, validating the methodological value of the behavioral layer in intelligent itinerary generation systems.

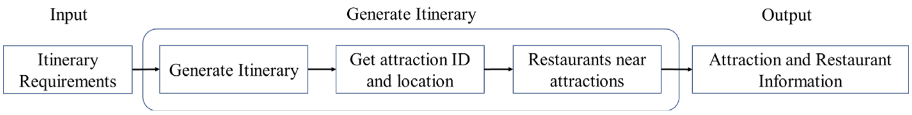

Fig. 3. Method B's conceptual validation.

6 Conceptual Evaluation Metrics

For the proposed three-layer integrated model, traditional system performance metrics-such as accuracy and execution time-are insufficient to capture its full value. Therefore, this section presents four conceptual evaluation dimensions, addressing semantic correctness, geospatial feasibility, human-AI interaction efficiency, and computational scalability (see Table. 1). These dimensions not only provide the basis for future quantitative experiments, but also serve as design-oriented criteria for validating conceptual models. Semantic coherence is defined as the degree to which the final system-generated itinerary aligns with the user's initial semantic input-including travel intent, preference conditions, and constraint descriptions. This metric corresponds primarily to the Semantic Layer of the conceptual model. It may be measured both subjectively and objectively: subjective assessment involves human rating to judge whether the itinerary meets the user requirements; objective assessment can apply semantic vector models to compute semantic distance between the original intent and the LLM output. This metric reflects whether preference extraction, semantic parsing, and content generation logic within the semantic layer remain coherent; higher semantic coherence indicates stronger preference understanding and inference quality by the LLM.

Table 1. Four conceptual evaluation dimensions.

Layer	Metric	Nature	Measurement	Tools
Semantic	Coherence	Cognitive Accuracy	Similarity, Human Rating	BERT/GPT Eval
Geospatial	Fidelity	Physical Feasibility	Hit Rate, Distance Error	Maps API
Behavioral	Efficiency	User Experience	Task Time, Interaction Count	Log Analyzer
Cross-layer	Operational Cost	Scalability	API Count, Latency	Vercel Monitor

Spatial fidelity is defined as the degree to which the generated itinerary aligns correctly with real-world geospatial information. This metric corresponds to the Geospatial Validation Layer, and is quantified using three primary measures: geographical hit rate, route feasibility index, and spatial distance error. Geographical hit rate measures whether LLM-generated place names can be successfully grounded to identifiable physical locations; route feasibility index is computed via distance matrices returned by the Google Directions API; spatial distance error compares model-assumed coordinates against true coordinates. This metric does not evaluate linguistic capability itself, but rather whether the generated plan can be executed in the physical world. High spatial fidelity implies robust integration of geocoding, routing, and business-hour APIs, and thus reflects stronger real-world awareness of system output. Human-AI Interaction Efficiency measures the interaction cost required for users to complete an acceptable itinerary. This metric corresponds to the Generation and Interaction Layer and is observed using task completion time, interaction count, and interaction path analysis. Task completion time reflects whether human-in-the-loop dynamics accelerate decision-making; interaction count quantifies depth of iterative cycles; interaction path analysis uses front-end event logs to trace edits, deletions, regeneration, and final acceptance. Higher interaction efficiency implies faster convergence with lower cognitive burden, and indicates that the behavioral layer affords sufficient operability and interpretability.

Finally, operational cost is regarded as a deployment-oriented metric that spans all three layers-semantic, geospatial validation, and behavioral. It is measured via API call count, average latency, and inference cost. API call count reflects dependence on external resources; average latency captures full-chain delay from request to response; inference cost depends on LLM token consumption and server-side computational load, and has direct business implication. This metric affects both user experience and deploy ability at commercial or large-scale levels. Therefore, operational cost provides a basis for cross-model and cross-strategy comparison, evaluating scalability and substitution flexibility among modules at different layers.

7 Conclusion and Future Work

This work proposed, from a conceptual perspective, a three-layer integrated model-Semantic Layer, Geospatial Validation Layer, and Generation and Interaction Layer-to demonstrate how the semantic abstraction capability of LLMs can be combined with the factual grounding capability of geospatial APIs (e.g., Google Maps) to produce executable, interactive, and iteratively refinable intelligent itinerary generation systems. Two methods were analyzed as strategy exemplars: (i) a semantic-first recursive filling strategy, and (ii) a geography-first route-grounding strategy. These two strategies correspond to different forms of lead layer configuration within the model, revealing actionable design implications: whether semantic hypotheses should be established first, or whether physical-world constraints should be evaluated first, directly affects behavioral convergence rate, interaction transparency, and deployment cost. Overall, the key conceptual conclusion is that the effectiveness of an LLM is not determined by the model alone, but by “which layer it occupies” and “how it circulates and interacts with the geospatial validation layer.” The three-layer boundary therefore operationalizes human-AI co-construction: the semantic layer proposes generative hypotheses, the validation layer imposes real-world constraints, and the behavioral layer provides loci for human intervention. Accordingly, itinerary generation should not be framed solely as an natural language processing (NLP) problem, but as a closed-loop human-AI system design problem defined by "semantic → geographic → interactive" iteration.

Future work proceeds in three directions: (1) Joint semantic-geospatial embedding: enabling transformer models to internalize geospatial priors instead of treating validation as an external module; (2) Contextual adaptation: integrating real-time weather, events, and traffic signals to shift from static generation toward context-sensitive generation; (3) Experimental validation of human-AI co-creation: evaluating lead layer configurations via user studies, using interaction efficiency and semantic coherence as primary metrics. In summary, this study abstracted two methods into conceptual strategy grammars, and articulated an evolvable integration model. This framework applies not only to itinerary generation but also to broader domains requiring "semantic generation and physical-world consistency," such as intelligent spatial layout, business district recommendation, or future automated urban planning. We expect this work to serve as a reference foundation for semantic-geospatial-interaction integrated framework design.

References

1. T. B. Brown *et al.*, "Language Models are Few-Shot Learners," in *Proc. NeurIPS*, 2020.
2. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.
3. J. Wei *et al.*, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in *Proc. NeurIPS*, 2022.
4. S. Yao, J. Zhao, D. B. Kong, et al., "ReAct: Synergizing Reasoning and Acting in Language Models," arXiv:2210.03629, 2022.
5. M. Ahn, S. Brohan, D. J. Ho, et al., "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," arXiv:2204.01691, 2022.
6. T. de la Rosa, "TRIP-PAL: Travel Itinerary Planning with Guarantees by Combining Large Language Models and Automated Planners," arXiv:2406.10196, 2024.
7. Y. Hao, "Large Language Models Can Solve Real-World Planning Problems," arXiv:2404.11891, 2024.
8. S. Zhang, Z. Luo, L. Yang, F. Teng, and T. Li, "A Survey of Route Recommendations: Methods, Applications, and Opportunities," arXiv:2403.00284 (also published/available via InfoFusion publisher), 2024.
9. "A Survey of Large Language Models in Tourism," Qeios preprint, Feb. 2024.
10. Z. Ding and A. Chan, "Evaluating Human–AI Text Co-Creation through the Lens of Headline Generation," *Findings of EMNLP*, 2023.
11. J. Heer and B. Shneiderman, "Interactive Dynamics for Visual Analysis," *Commun. ACM*, vol. 55, no. 4, pp. 45–54, Apr. 2012.
12. P. Toth and D. Vigo, *The Vehicle Routing Problem*, SIAM, 2002. (book)
13. Google Developers, "Google Maps Platform - Web Services Documentation," Google, online. [Online]. Available: https://developers.google.com/maps/documentation. (Accessed Jan. 2025).
14. S. Mesbah, I. Arous, J. Yang, and A. Bozzon, "HybridEval: A Human–AI Collaborative Approach for Evaluating Design Ideas at Scale," in *Proc. WWW '23*, 2023, pp. 3837–3848.

15. A. Ferrara *et al.*, “Beyond Shortest Paths: Node Fairness in Route Recommendations,” *PVLDB*, 2024.