Real-Time Adaptive Tour Recommendation with Agentic AI

Kwan Hui Lim
Singapore University of Social Sciences
School of Business
Singapore, Singapore
khlim@suss.edu.sg

Aldy Gunawan
Singapore Management University
School of Computing and Information Systems
Singapore, Singapore
aldygunawan@smu.edu.sg

Carmen Kar Hang Lee
Singapore University of Social Sciences
School of Business
Singapore, Singapore
carmenleekh@suss.edu.sg

Abstract

Tour-recommendation systems, including the recent wave of LLM-based travel planners, typically stop at plan generation: given a user request, they produce a single itinerary and stop. Once the trip starts, however, tourists are left to handle disruptions such as rain, growing queues, transit breakdowns, or user fatigue. We present AdaptTour, an agentic system that continues replanning after departure; its contribution is the overall control architecture. Mid-trip signals (weather, queue length, transit disruption, user fatigue, and free-text feedback) are fused into a single context snapshot, typed triggers pass through a debounced registry, and a replanner locks the executed prefix and re-optimises the tail, re-enforcing every constraint at each pass: money budget, accessibility, per-member vetoes and boosts, and carbon preference. The LLM is used only to score candidates, with a deterministic fallback for omitted candidates, and to generate the rationale presented to the traveller; selection, sequencing, routing, and constraint checking remain deterministic. Weather is live, while crowd, transit, and fatigue are simulated for the demonstration and explicitly labelled as such in the interface. The system provides a curated five-city catalogue of POIs (Singapore, Melbourne, London, New York, Paris), per-city fare tables, five travel modes, and OSRM-routed map polylines cached on disk for offline-capable visualisation. Source code, tests, and JSONL scenario traces are publicly available to support reproduction.

CCS Concepts

• **Computing methodologies** → **Intelligent agents**; *Planning and scheduling*; Natural language processing; • **Information systems** → **Recommender systems**.

Keywords

tour recommendation, agentic AI, large language models, real-time replanning, multi-signal fusion, system demonstration

ACM Reference Format:
Kwan Hui Lim, Aldy Gunawan, and Carmen Kar Hang Lee. 2026. Real-Time Adaptive Tour Recommendation with Agentic AI. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 07–11, 2026, Rome, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3799682.3840259



This work is licensed under a Creative Commons Attribution 4.0 International License.
CIKM '26, Rome, Italy
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2539-5/2026/11
https://doi.org/10.1145/3799682.3840259

1 Introduction

Tour and trip recommendation has been studied for over two decades as a constrained sequence optimization problem, with applications spanning urban tourism, lifelong learning excursions, and crisis response [6, 9, 14, 22]. Recent years have seen a rapid shift towards travel planners built on large language models (LLMs), with systems such as ItiNera [21], TravelAgent [3], and DeepTravel [18] generating full itineraries directly from natural-language user requests, while the TravelPlanner benchmark [24] reports that even GPT-4 succeeds on under 1% of queries. While these systems have improved the quality of initial plans, they share a common limitation: once the itinerary is generated, the agent stops. Mid-trip disruptions such as rain, long queues, transit breakdowns, or user fatigue are not handled by the system, while TravelPlanner [24], as a dominant benchmark, evaluates systems based on initial-plan quality alone.

Beyond this missing trip-time loop, our review of recent travel demos and benchmarks [1–4, 11, 17, 18, 20] identified six capabilities that no single recent system provides jointly: money budgets at the optimiser, accessibility as a pre-retrieval filter, per-visit structured reasoning, mid-trip per-member vetoes and boosts, sandboxed booking with an audit log, and per-leg CO_2e accounting.

Contributions. We present AdaptTour, an agentic system for real-time adaptive tour recommendation whose contribution is an agentic control architecture rather than a set of new features: (i) a closed mid-trip loop in which typed triggers (weather onset, crowd spike, transit disruption, fatigue, free-text feedback) pass through a debounced registry to a replanner that locks the executed prefix and re-optimises the tail, re-enforcing every constraint on each pass rather than applying it only at plan generation; (ii) a division of labour that keeps the loop auditable, with the LLM confined to scoring candidates (a deterministic fallback covers any candidates it omits) and writing rationale text, while retrieval, filtering, selection, sequencing, and routing remain deterministic; (iii) fusion of the signals into one `ContextSnapshot` with per-signal ablation flags, allowing each signal's contribution to be isolated, while the six capabilities above are realised as constraints that the loop re-checks rather than prompt hints; and (iv) a curated five-city catalogue with five travel modes, per-city fare tables, wheelchair, sensory-friendly and pace toggles, and OSRM-routed polylines cached on disk for offline reproducibility. The system is open-source and available at https://github.com/kwanhui/public-adaptiveTourAgent-CIKM2026, and a short demonstration video is available at https://youtu.be/qWlkl9_vnp8.

2 Related Work

2.1 Tour Recommendation

Tour recommendation has been studied for over a decade, with the dominant formulation casting an itinerary as an instance of the Orienteering Problem [22], where a planner selects a subset of POIs to maximise interest under time and travel-distance constraints. Surveys of mobile and personalised tourism systems [6, 9] show that most deployed solutions assume a static input context: the tourist's profile, the POI graph, and the visit budget are fixed before the trip begins. Our prior work follows this lineage, with PersTour [14] optimising personalised itineraries using visit-duration estimates and later extensions incorporating queuing time [8], group-level crowd dynamics [16], and learned dynamic POI generation [10]. A second line of work uses LLMs to generate itineraries directly from natural-language requests. LLM-Mob [23], Li et al. [13], LLMMove [5], and the retrieval-augmented RALLM-POI [12] apply LLMs to next-POI prediction, while ItiNera [21] couples an LLM with a spatial optimiser to assemble multi-stop walking itineraries. TravelPlanner [24] provides the first large benchmark for tool-using language agents for travel planning and reports that even GPT-4 succeeds on under 1% of queries. Across these works, the LLM is primarily used as a one-shot generator: an itinerary is generated for a request, but no mechanism feeds mid-trip events back into the planning process.

2.2 Adaptive Tour Recommendation with Agentic LLMs

The intersection of adaptive tour recommendation and agentic LLMs remains relatively unexplored. General agent frameworks such as ReAct [25] and Reflexion [19] establish the building blocks for interleaved reasoning and tool use, together with verbal self-reflection on outcomes, allowing an LLM to update its plan as new observations arrive. Gundawar et al. [7] apply the LLM-Modulo pattern to travel planning by pairing an LLM proposer with external verifiers, reporting a 4.6× improvement for GPT-4-Turbo on TravelPlanner; however, their loop operates only during plan construction and does not consume real-world signals from a trip in progress. Choi et al. [4] push this further with a constraints-aware multi-agent framework whose planner-checker loop, interleaved search, and multi-turn feedback increase the TravelPlanner final pass rate from 23.3% to 44.4%. Their adaptivity remains within plan construction and stops once a plan is accepted; in contrast, ours operates during execution, locking completed stops and re-optimising only the remaining tail for as long as the trip continues. Recent travel-planning systems [1–4, 11, 17, 18, 20] have advanced initial-plan quality but stop short of maintaining a persistent mid-trip controller over weather, crowd, transit, and user-state signals. Our system addresses this gap with a closed feedback loop operating over a context snapshot fused from multiple data sources.

3 System Description

We developed AdaptTour as a Python service with a single-page web UI (Figure 1), further organised into nine core modules around two cross-cutting flows: a closed mid-trip replanning loop and a multi-signal context fusion pipeline.

3.1 Signal Sources and Trigger Registry

Mid-trip signals come from a pluggable set of sources: a recorded JSONL replayer for the demonstration scenarios, an Open-Meteo weather adapter that requires no API key, and synthesised crowd and transit sources. Only weather is real; crowd, transit, and fatigue are simulated for demo purposes, but can be modified to connect to real external sources. The trigger registry fires seven typed triggers (rain onset above 1mm/h, storm, crowd spike, venue closure, transit disruption, fatigue, free-text request; a storm suppresses rain onset) under per-trigger debounce (60s for rain) and cooldown (10min), both clocked on scenario time so the demo can advance the day while bounding how often a flickering signal re-fires.

3.2 Multi-Signal Context Fusion

The `fusion` module merges source batches into one snapshot carrying the latest weather, crowd, transit, and user-state readings, together with itinerary position, executed-prefix metadata, and running cost and CO_2e totals. The three simulated signals are deterministic generators, not learned models, and we next elaborate their construction. Crowd intensity is POI popularity times the larger of two hour-of-day Gaussian peaks centred at 12:00 and 16:00, thresholded at 0.25 and 0.55 into low, medium, and high, with queue estimates of 5, 15, 35, and 60 minutes as intensity rises through 0.85. Transit returns the routing-table baseline duration for an edge and never synthesises a disruption; disruptions come from a recorded trace, one of which inflates a Singapore leg to 65 minutes at 13:00 and restores it to 30 at 15:30, or from a transit keyword in a user note. Fatigue accumulates at 0.06 per hour plus 0.04 per visit, scaled by $1 + 0.05(n - 1)$ for a party of n, reduced by 0.15 after a break within 30 minutes, and increased by 0.35 by an "I'm tired" note, with `fatigue_high` firing at 0.85. Each signal sits behind a typed reading, so a live feed can replace it without modifying the fusion module or replanner. A `disable=` flag zeroes specific signals before the snapshot reaches the replanner, providing a single ablation surface for per-signal isolation studies. Each `TriggerEvent` carries its kind, severity, affects, and snapshot id, so the JSONL log alone can reconstruct any decision.

3.3 Agentic Replanning Loop

The loop is a closed-loop controller: typed triggers pass through a debounced registry into prefix-locked re-optimisation of the tail, with every constraint re-enforced on each pass rather than only at plan time. The driver in `replanner.loop` ticks once per signal observation, builds a snapshot, and runs the registry; when a trigger fires, the replanner executes two stages.

Stage 1: constrained tail re-scoring. The replanner locks the executed prefix (completed visits cannot be revoked) and re-decides only the remaining. Every trigger fired on the tick is handled in a single replan rather than separately, and triggers compose by type rather than through an explicit priority order. Hard constraints intersect: closed venues, hand-removed stops, and member-vetoed categories are unioned into a single exclusion set, while rain or storm adds `require_indoor=True`. Soft signals combine into one score vector, applying a 0.5 penalty to crowded candidates and a 1.25 lift to a boosted member's category, while a free-text note is parsed for weather, transit, and fatigue keywords and also carried verbatim

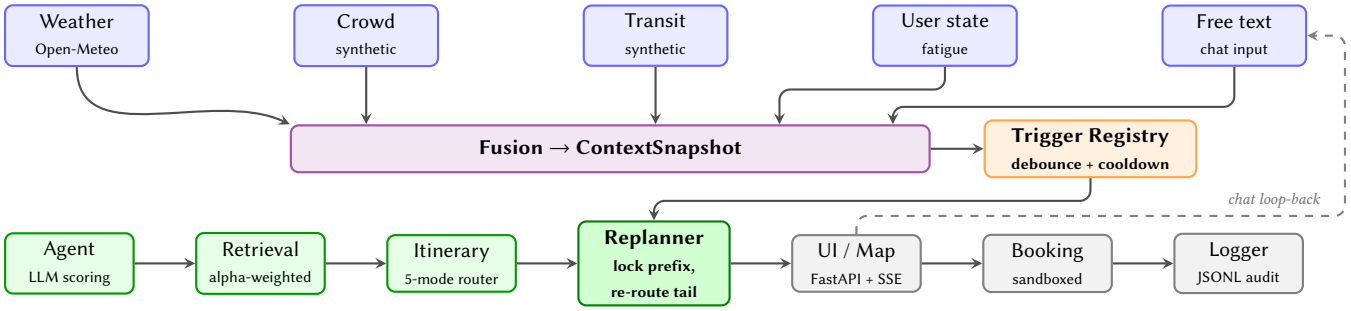


Figure 1: AdaptTour architecture. Five signal sources feed the fusion layer, which produces a `ContextSnapshot`. The trigger registry applies per-trigger debounce and cooldown, then the replanner pins the executed prefix and re-routes the tail by calling the agent / retrieval / itinerary pipeline. Outputs land in the UI, the sandboxed booking actuator, and the JSONL logger. The dashed loop-back arrow shows that the UI's free-text chat input is itself a signal source, closing the adaptive loop.

into the scoring prompt so the model can act on intent missed by the parser. One joint re-optimisation then resolves the combined signals, so a hard constraint always takes precedence over a soft preference, while a burst of simultaneous signals costs one pair of LLM calls rather than one pair per trigger; the registry's cooldowns further bound how often the loop can re-fire. The candidate set is passed to a single LLM scoring call at $T = 0.0$ for repeatable rankings, returning per-POI scores as JSON read through a fence-tolerant parser. Any POI omitted by the model falls back to a deterministic blend of popularity and category alignment, so the optimiser never receives an empty slot.

Stage 2: rationale generation. A second call at $T = 0.3$ generates a rationale under sixty words explaining the visible change, which is logged both to the chat panel and to the JSONL audit trail.

Across the loop, the model does two things: per-POI scoring and rationale generation; retrieval, filtering, selection and sequencing under the time and money caps, mode choice, debounce, prefix locking, and diffing remain deterministic, similar to our guardrailed LLM tutor [15]. The agent module wraps a provider-agnostic client (default `gpt-4o-mini`) with usage tracking, per-session cost caps, and structured-output parsing, so a provider can be substituted without modifying the replanner.

3.4 Retrieval, Routing, and Geometry

The `retrieval` module first applies structured filters (kid-friendly, indoor, open-at-hour, wheelchair, dietary, low-stimulation), then ranks the surviving POIs by α-weighted category alignment blended with normalised popularity (α=0.7). Only the top k reach the scoring prompt, with k=12 at every call site, so the candidate list ranked by the model is bounded independently of catalogue size; the deterministic pass ahead of it performs a single filtering sweep over the catalogue followed by a sort of the survivors. Prompt width is not yet bounded end to end: the context note sent with the candidates still lists every crowded or closed POI in the snapshot, and scoping it to the upcoming and candidate POIs is a known local change required before city-scale deployment.

The catalogue currently includes five cities (Singapore, Melbourne, London, New York, Paris), each with 15 POIs and 5 to 10 accommodations under a uniform schema (open hours, category, popularity, party-multiplied fee, access flags). Travel is routed across five modes (walk, bike-share, transit, ride-hailing, drive) with per-city fare tables and brand labels (Singapore MRT, London Tube, Paris métro); `pick_mode` selects based on distance, party size, remaining money, and the low-carbon and wheelchair flags, applying per-mode circuity factors and wait times. Map polylines are road-snapped OSRM routes for each leg, cached on disk and committed so that the visualisation reproduces offline.

3.5 Beyond the Replanning Loop

Several constraints, outputs, and actuators sit alongside the loop; the individual features, which have appeared in isolation in prior systems, are not the key contribution, but their re-enforcement on every mid-trip revision rather only at initial plan generation. **Money budget** caps cumulative entry-fee plus transit-fare spend (party-multiplied) within the greedy optimiser, alongside the time cap. **Accessibility** is enforced where wheelchair, dietary, and low-stimulation flags pre-filter POIs during retrieval, while the wheelchair flag steers routing, addressing the critique of Mayr et al. [17]. **Per-visit reasoning** composes a `reasoning_text` string and a numeric `reasoning_scores` trace from values already computed by the optimiser (score gap to the runner-up, fatigue position, mode driver), using a template rather than an additional LLM call. **Mid-trip group dynamics** gives each `GroupMember` vetoes that block a category outright and boosts that lift it; unlike plan-time aggregation [16], a veto cast mid-trip from the UI re-routes the tail and persists across subsequent replans. **Sandboxed booking** writes POI-ticket and accommodation bookings to an audit trail with synthetic confirmation codes and no external calls for now. **Per-leg CO_2e** is recorded per visit using factors from Banerjee et al. [1] and IPCC AR6, with a low-carbon flag downweighting high-emission options in the optimiser score. A three-way pace selector tunes the fatigue coefficient (0.50/0.40/0.25) and per-stop buffer (+15/0/−5 min).

3.6 User Interface

The UI comprises a FastAPI server (`ui.server`) and a single static page. The form carries city, start/end datetimes (single- or multi-day automatically), a profile preset (Solo, Couple, Friends, Family),

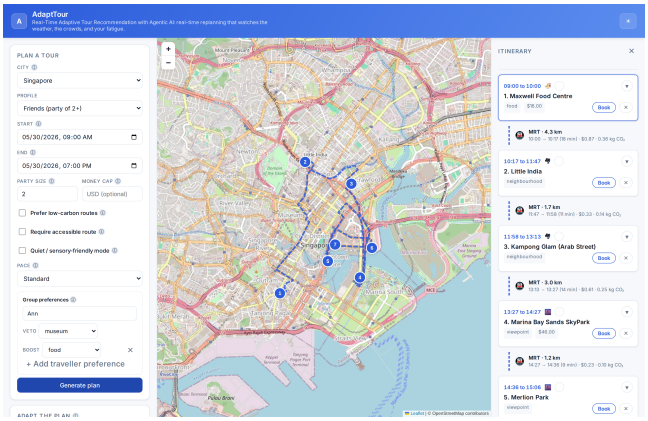


Figure 2: Live demo for a Friends-of-two group in Singapore with a per-member veto (no museums) and boost (food). The right rail lists each stop with its window and fee, per-leg transport cards (mode, routed distance, fare, CO_2e), and *Book* and remove controls.

the low-carbon, wheelchair, and sensory-friendly toggles, the pace selector, and, for Friends and Family, a per-member veto and boost panel. Trigger buttons and a free-text chat box feed the same fusion pipeline as weather and crowd. A sticky metrics card shows session LLM cost, tour spend, CO_2e, and replan count. Visit cards carry the time window, name, category, fee, `reasoning_text` under "Why this stop", the score trace and alternatives behind icons, *Book* and *remove* controls, and an *Undo*. Transport-leg cards render the chosen mode (city-aware label), routed distance, time, fare, and CO_2e, matching the Leaflet polyline colours. Server-Sent Events push every replan, and the map redraws in place; when hard filters prune the catalogue, the UI reports how many stops remain and may suggest relaxing one if required.

4 Demonstration Scenarios

We describe four scenarios; three are provided as JSONL replays under `demo/sample-inputs/scenarios/`, and every run logs `snapshot`, `trigger`, and `replan` events, allowing any decision to be replayed. Deterministic components (catalogue, fares, routing) reproduce exactly; POI choices depend on the live model, so the runs below are representative. In our demo, attendees can select a city and profile, generate a tour, and then fire triggers or type a request to watch the plan re-form.

Use Case 1: Mid-trip replanning under rain. A family-of-four Singapore tour starts at 09:00; the recorded weather trace turns to rain (4.5mm/h, peaking at 6mm/h) at 11:30, when two outdoor stops are scheduled. After a 60-second debounce, the `weather_rain_onset` trigger fires; the replanner locks the morning prefix, the fused rain condition sets `require_indoor`, and rescoring swaps the Gardens by the Bay segment for the ArtScience and National museums. The chat panel logs the diff ("2 dropped, 2 inserted") and a rationale, while total LLM cost stays under $0.05 with `gpt-4o-mini`.

Use Case 2: Multi-day tour with accommodation matching. A family of two requests a Singapore trip from 2026-06-01 to 2026-06-03. The `find-accommodation` endpoint filters the catalogue to two candidates that pass the kid-friendly and near-MRT filters under a $300/night cap and scores them with the LLM; the selected hotel anchors three daily itineraries, with POIs pinned across days so no entry repeats. One model-dependent run returns three days and fourteen visits at $497.82 in fees and fares, with 3.21kg of CO_2e.

Use Case 3: Sustainability comparison via low-carbon flag. For a family-of-two day plan with a money budget loose enough to admit ride-hailing, toggling the `prefer_low_carbon` flag switches medium-distance hops from ride-hailing to walking, cycling, and transit. With the visited POIs held fixed, toggling the flag in one such run reduces trip CO_2e from 6.07kg to 1.38kg, a 4.4× reduction, and the per-visit cards show the chosen mode for each leg so that each routing decision is visible rather than buried in an aggregate.

Use Case 4: Free-text mid-trip refinement. A typed chat message, "the kids want dinner now, skip the next museum", pushes a `user_request` trigger through the same fusion pipeline as weather and crowd, and the note is fed verbatim into the scoring prompt. The UI first echoes a short interpretation ("Understood: working a food stop into the plan nearby") so the traveller sees how the request was interpreted rather than receiving no feedback, covering intents that the keyword parser does not map to a typed signal (heat, hunger, cost, drop-this). In a sample run, the replanner drops the upcoming museum, inserts a nearby food stop, and updates the rationale. This addresses the mid-trip refinement gap left open by Singh et al. [20], whose conversational handling operates only at plan time.

Reproducibility. Switching the city dropdown re-runs the same plan against any of the five catalogues (Singapore → Paris switches MRT to métro at $0.10 → $0.65 per km). Sweeping pace over an 09:00 to 17:00 window returns 3, 4, and 4 stops for a family-of-two profile (405/390/370 minutes of in-POI time), while `require_wheelchair` drops inaccessible POIs and excludes cycle from `pick_mode`. The four core contracts (retrieval, optimisation, routing, and the replanning loop) are covered by the repository's test suite, parametrised across the five cities; OSRM geometry is cached and committed, and `ATAU_DISABLE_OSRM=1` allows viewers to run offline. A per-IP rate limiter and a per-session cost cap bound LLM spend, while a GitHub Action deploys the same container to a hosted Space.

5 Conclusion and Future Work

We presented AdaptTour, an agentic system that continues replanning after departure: typed triggers pass through a debounced registry, and a replanner locks the executed prefix and re-optimises the tail, with money budgets, accessibility, group vetoes, and carbon preference re-enforced on every revision. The contribution is this control loop and integrated features rather than a single capability; the LLM scores candidates and writes rationales, while selection and sequencing remain deterministic. Our evaluation is scenario-based for now, the crowd, transit, and fatigue signals are simulated, although it can be easily linked to the appropriate data providers. The source codes, JSONL traces, and demonstration video are publicly available. The next step of this work is to extend to a quantitative study using the disruption scenarios of TripTide [11], including measurement of latency. Future directions include extensions with cross-day replanning, a mobile deployment, real-provider booking and connections to other real-life data sources.

GenAI Usage Disclosure

Generative AI assistance was used to help refine the writing and organisation of text that was originally drafted by the authors, as well as assist with the refactoring and documentation of source codes. The authors reviewed and edited all such text and take full responsibility for the content of this paper.

References

[1] Ashmi Banerjee, Tunar Mahmudov, Emil Adler, Fitri Nur Aisyah, and Wolfgang Wörndl. 2025. Modeling sustainable city trips: integrating CO_2e emissions, popularity, and seasonality into tourism recommender systems. *Information Technology & Tourism* 27, 1 (2025), 189–226.

[2] Soumyabrata Chaudhuri, Pranav Purkar, Ritwik Raghav, Shubhojit Mallick, Manish Gupta, Abhik Jana, and Shreya Ghosh. 2025. TripCraft: A Benchmark for Spatio-Temporally Fine Grained Travel Planning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*. 17035–17064.

[3] Aili Chen, Xuyang Ge, Ziquan Fu, Yanghua Xiao, and Jiangjie Chen. 2024. TravelAgent: An AI Assistant for Personalized Travel Planning. *arXiv preprint arXiv:2409.08069* (2024).

[4] Jihye Choi, Jinsung Yoon, Jiefeng Chen, Somesh Jha, and Tomas Pfister. 2025. ATLAS: Constraints-Aware Multi-Agent Collaboration for Real-World Travel Planning. arXiv:2509.25586

[5] Shanshan Feng, Haoming Lyu, Fan Li, Zhu Sun, and Caishun Chen. 2024. Where to Move Next: Zero-shot Generalization of LLMs for Next POI Recommendation. In *Proceedings of the IEEE Conference on Artificial Intelligence (CAI)*.

[6] Damianos Gavalas, Charalampos Konstantopoulos, Konstantinos Mastakas, and Grammati Pantziou. 2014. Mobile recommender systems in tourism. *Journal of Network and Computer Applications* 39 (2014), 319–333.

[7] Atharva Gundawar, Mudit Verma, Lin Guan, Karthik Valmeekam, Siddhant Bhambri, and Subbarao Kambhampati. 2024. Robust Planning with LLM-Modulo Framework: Case Study in Travel Planning. *arXiv preprint arXiv:2405.20625* (2024).

[8] Sajal Halder, Kwan Hui Lim, Jeffrey Chan, and Xiuzhen Zhang. 2022. POI recommendation with queuing time and user interest awareness. *Data Mining and Knowledge Discovery* 36, 6 (2022), 2379–2409.

[9] Sajal Halder, Kwan Hui Lim, Jeffrey Chan, and Xiuzhen Zhang. 2024. A survey on personalized itinerary recommendation: From optimisation to deep learning. *Applied Soft Computing* 152 (2024), 111200.

[10] Sajal Halder, Kwan Hui Lim, Jeffrey Chan, and Xiuzhen Zhang. 2025. Deep Learning of Dynamic POI Generation and Optimisation for Itinerary Recommendation. *ACM Transactions on Recommender Systems* 3, 4 (2025), 51:1–51:29.

[11] Priyanshu Karmakar, Soumyabrata Chaudhuri, Shubhojit Mallick, Manish Gupta, Abhik Jana, and Shreya Ghosh. 2025. TripTide: A Benchmark for Adaptive Travel Planning under Disruptions. *arXiv preprint arXiv:2510.21329* (2025).

[12] Kunrong Li and Kwan Hui Lim. 2025. RALLM-POI: Retrieval-Augmented LLM for Zero-Shot Next POI Recommendation with Geographical Reranking. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence (PRICAI)*. 210–218.

[13] Peibo Li, Maarten de Rijke, Hao Xue, Shuang Ao, Yang Song, and Flora D. Salim. 2024. Large Language Models for Next Point-of-Interest Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 1463–1472.

[14] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2018. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. *Knowledge and Information Systems* 54, 2 (2018), 375–406.

[15] Kwan Hui Lim, Jing Ren, Yimiao Zhang, Stefanie Chye, and Amy Wong. 2026. SAGE: A Guardrailed LLM Tutoring System with Scaffolded Self-Regulated Learning for Adult Education. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26)*. ACM, New York, NY, USA. doi:10.1145/3799682.3840260

[16] Junhua Liu, Aldy Gunawan, Kristin L. Wood, and Kwan Hui Lim. 2025. Optimizing group utility in itinerary planning: a strategic and crowd-aware approach. *Journal of Big Data* 12, 1 (2025), 201.

[17] Gregor Mayr, Markus Reiter-Haas, and Elisabeth Lex. 2025. Mind the Gap: Urban-Rural Disparities in Wheelchair Accessibility for POI Recommendations. In *Recommender Systems for Sustainability and Social Good (RecSoGood) Workshop, in conjunction with ACM RecSys*. 108–116.

[18] Yansong Ning, Rui Liu, Jun Wang, Kai Chen, Wei Li, Jun Fang, Kan Zheng, Naiqiang Tan, and Hao Liu. 2025. DeepTravel: An End-to-End Agentic Reinforcement Learning Framework for Autonomous Travel Planning Agents. *arXiv preprint arXiv:2509.21842* (2025).

[19] Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[20] Harmanpreet Singh, Nikhil Verma, Yixiao Wang, Manasa Bharadwaj, Homa Fashandi, Kevin Ferreira, and Chul Lee. 2024. Personal Large Language Model Agents: A Case Study on Tailored Travel Planning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track (EMNLP Industry)*. 486–514.

[21] Yihong Tang, Zhaokai Wang, Ao Qu, Yihao Yan, Zhaofeng Wu, Dingyi Zhuang, Jushi Kai, Kebing Hou, Xiaotong Guo, Jinhua Zhao, Zhan Zhao, and Wei Ma. 2024. ItiNera: Integrating Spatial Optimization with Large Language Models for Open-domain Urban Itinerary Planning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track (EMNLP Industry)*. 1413–1432.

[22] Pieter Vansteenwegen, Wouter Souffriau, and Dirk Van Oudheusden. 2011. The Orienteering Problem: A Survey. *European Journal of Operational Research* 209, 1 (2011), 1–10.

[23] Xinglei Wang, Meng Fang, Zichao Zeng, and Tao Cheng. 2023. Where Would I Go Next? Large Language Models as Human Mobility Predictors. *arXiv preprint arXiv:2308.15197* (2023).

[24] 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 *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

[25] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.