AI Tour Meeting: Group Travel Planning by LLM Agents*

Daisuke Kikuta
NTT, Inc.
daisuke.kikuta@ntt.com

Abstract

This paper proposes AI Tour Meeting, a group travel planning framework powered by multiple Large Language Model (LLM)-based agents. The agents are instantiated with distinct personas and collaboratively seek an itinerary that satisfies their constraints and preferences through natural language discussion. The framework enables easy and flexible orchestration of such discussions by providing interfaces for configuring agent personas, discussion workflows, monitoring, and LLM deployment. Its primary use case is a simulation tool for analyzing the behavior of multiple LLM agents during tour planning discussions. This paper demonstrates the utility of the framework by presenting system validation and several analytical results obtained using the framework.

1 Introduction

Group travel planning aims to find an itinerary that balances multiple objectives among multiple participants, such as cost, travel time, and individual preferences. Prior work has focused on supporting human groups in determining balanced itineraries (Nguyen and Ricci, 2018; Chen et al., 2021; Cheng et al., 2026). Meanwhile, in broader domains, recent studies have explored the use of persona-based Large Language Model (LLM) agents to simulate human behavior (Park et al., 2023; Zhou et al., 2024; Jiang et al., 2024; Hu and Collier, 2024). This emerging direction is also expected to be relevant to group travel planning. For example, automatic evaluation in such simulation environments reduces the costs of large-scale behavioral analysis in group travel planning and the evaluation of existing group travel recommender systems. In application, users could interact with LLM agents representing group members who are unable to participate in the actual discussion and collaboratively determine an itinerary with them.

To address this emerging need, we propose AI Tour Meeting, a group travel planning framework powered by multiple LLM agents. Each agent is instantiated with a distinct persona and collaborates with the others to find an itinerary that satisfies their constraints and preferences through natural-language discussion. Our framework provides interfaces to orchestrate this process. Compared with existing persona-based agent simulations (Park et al., 2023; Zhou et al., 2024) and general-purpose multi-agent frameworks (Li et al., 2023; Wu et al., 2024), our framework provides domain-specific abstractions for group travel planning, including a structured itinerary representation, explicit proposal-voting workflows, and constraint validation and metric monitoring specialized for travel planning. This allows users to focus on their use cases without implementing the required functionality from scratch using a general-purpose framework.

In the following, we describe the technical details of our framework and validate whether it can successfully complete discussions using Qwen 3.5 and GPT models of different sizes. We then demonstrate its effectiveness as a simulation tool through analysis examples conducted using the framework.

2 Proposed Framework: AI Tour Meeting

Figure 1 shows the overview of our framework: we first set up both LLM participants with distinct personas and a discussion workflow, then start the meeting among them aligned with the specified workflow. The meeting begins with the conversation phase, in which participants take turns searching for information, asking questions, and proposing itineraries. Once an itinerary is proposed, it transitions to the voting phase, where the participants decide whether to accept or reject it. This two-phase process continues until all participants

*Code: https://github.com/ntt-dkiku/ai-tour-meeting; Video & Document: https://ntt-dkiku.github.io/ai-tour-meeting.

1
Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 58–65
November 6-10, 2026 ©2026 Association for Computational Linguistics

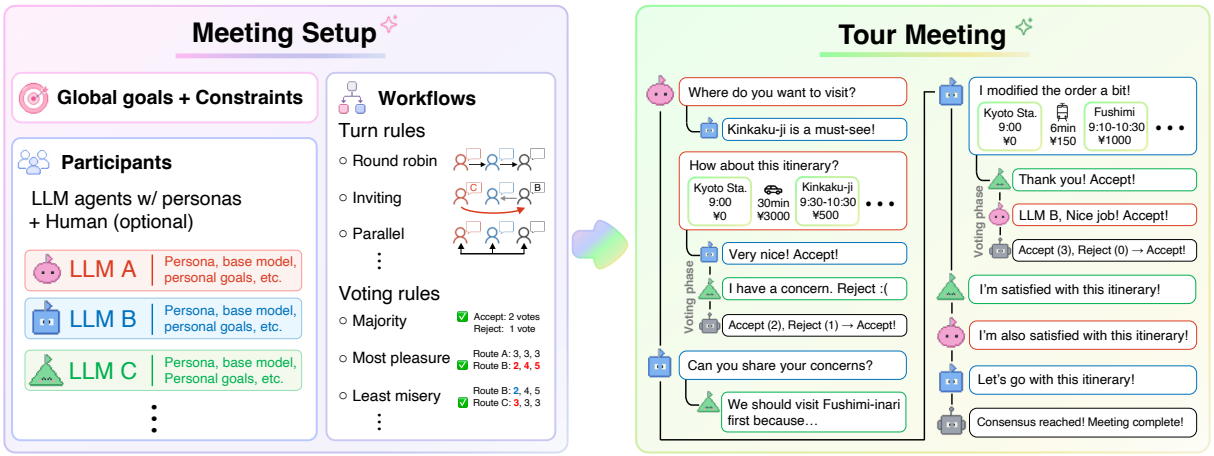


Figure 1: An overview of AI Tour Meeting.

are satisfied with the currently accepted itinerary.

In this section, we describe the definitions of itineraries and participants, the workflow options, the monitored metrics, and the user interface.

2.1 Itinerary

An itinerary proposed during the meeting consists of information about destinations and the transportation between them, and is defined as follows:

$$R = (d_1, d_2, \ldots, d_N), \tag{1}$$

$$d_i = (\mathrm{n}_i, \mathrm{desc}_i, c_i, t_i^{\mathrm{arr}}, t_i^{\mathrm{stay}}, \mathrm{tr}_i, c_i^{\mathrm{tr}}, t_i^{\mathrm{tr}}), \tag{2}$$

where R is the itinerary and d_i is the i-th destination. Each destination includes the name n_i, description desc_i, monetary cost c_i, arrival time t_i^{arr}, duration of stay t_i^{stay}, transportation mode from the previous destination tr_i, transportation cost c_i^{tr}, and transportation duration t_i^{tr}.

2.2 Participant

Each LLM agent (participant) is instantiated with a distinct persona. The persona consists of the following attributes: *Name* specifies their name; *Background* specifies their experience or situation; *Personality* specifies their stable traits; *Preference* specifies their preferences related to the tour; *Goal* specifies their personal goals in the tour; *Role* selects their role in the meeting from either facilitator or attendee; *Tone* specifies their speaking tone such as formal or friendly; *Explanation style* selects a strategy for convincing others to accept their proposal, from either subjective explanation, contrastive explanation (Kikuta et al., 2024) or both. The former provides subjective opinions of the proposal based on their preferences. The latter explains the objective pros and cons by comparing the proposed route with the current route in terms of objective metrics such as cost, time, and number of destinations. We can select an LLM that instantiates the participant with model parameters such as temperature and seed. Both commercial LLMs and local LLMs (via Ollama or vLLM) are supported. Context management methods, including auto-compaction and history truncation, can be also configured individually for each participant.

Actions The $\mathrm{persona}$ is given to the LLM as a system prompt, and the LLM participant takes actions during the meeting while role-playing according to the given $\mathrm{persona}$. The meeting proceeds in a turn-based manner, and each participant may take multiple actions during their single turn. The action at each step s within a turn is defined as follows:

$$(\tau_s, \mathrm{msg}_s, a_s) = p_m(\mathrm{persona}_m, H_{s-1}), \tag{3}$$

where p_m is the m-th LLM participant, τ_s is the action type, msg_s is the message to others, a_s is the taken action, and H_{s-1} is the meeting history, which includes system messages for managing the discussion workflow $\mathrm{msg}_{\mathrm{sys}}$ and the past actions taken by all the participants $((\tau_1, \mathrm{msg}_1, a_1), \ldots, (\tau_{s-1}, \mathrm{msg}_{s-1}, a_{s-1}))$. The action a_s depends on the selected action type τ_s:

$$a_s = \begin{cases} \mathrm{Search}(q_s) & \tau_s = \mathrm{search} \\ \mathrm{Ask}(p_{m'}, q_s) & \tau_s = \mathrm{ask} \\ \mathrm{Reflect} & \tau_s = \mathrm{reflect} \\ \mathrm{Propose}(R_s) & \tau_s = \mathrm{propose} \\ \phi & \tau_s \in \{\mathrm{satisfied}, \mathrm{pass}\} \\ \mathrm{Score}(v_s) & \tau_s = \mathrm{score} \\ \phi & \tau_s \in \{\mathrm{accept}, \mathrm{reject}\} \end{cases} \tag{4}$$

There are two types of actions: intermediate and terminal actions. Intermediate actions may be taken multiple times within a turn, whereas once a terminal action is taken, the current turn ends. The intermediate action types include search, ask, and reflect. search issues a query q_s to retrieve information such as location or cost of destinations via web search. ask asks another participant $p_{m'}$ a question q_s and receives a response. reflect organizes and reviews the discussion conducted so far. The terminal action types differ between the conversation phase and the voting phase. In the conversation phase, they include propose, satisfied, and pass. propose proposes a new itinerary R_s. satisfied indicates that p_m is satisfied with the current route. pass ends their turn without expressing a specific position. In the voting phase, they include accept, reject, and score. accept votes for the proposed itinerary, reject votes against the proposed itinerary, and score assigns a score v_s to the proposed route.

2.3 Discussion workflow

A tour meeting consists of a global goal, participants, constraints, a turn rule, and a voting rule. The global goal is a high-level objective shared among all participants, e.g., planning a one-day sightseeing tour in Tokyo. Each participant advances the discussion in pursuit of their own personal goal while remaining aligned with the global goal. Constraints restrict the itinerary in terms of travel date, budget, and time window.

The tour meeting proceeds according to the specified turn rule, under which participants sequentially take the actions defined in Eq. (4) multiple times within their turns. This normal state is referred to as the conversation phase. Once a participant proposes an itinerary (i.e., terminates the turn with $\tau = \text{propose}$), the process transitions to the voting phase. In this phase, all participants except the proposer cast votes on the proposed itinerary. After collecting all votes, the acceptance or rejection of the proposed route is determined in accordance with the specified voting rule. If the itinerary is accepted, the current itinerary is replaced with the proposed one; if it is rejected, the current itinerary remains unchanged. The process then returns to the conversation phase, and the turns resume with the next participant after the proposer.

When all participants consecutively indicate satisfaction with the current route (i.e., terminate their turns with $\tau = \text{satisfied}$), consensus is considered reached, and the meeting concludes. In practice, the termination condition of the meeting may also be constrained by a maximum number of turns or a maximum execution time, in addition to the unanimous agreement.

Algorithm 1: Tour meeting workflow

Input : Participants $\mathcal{P} = \{p_1, p_2, \ldots, p_M\}$; Turn rule π_{turn}; Voting rule π_{vote}
Output : The final tour itinerary R_{final}

$R_{\text{final}} \leftarrow \phi$; $H \leftarrow \text{msg}_{\text{sys}}$; $C \leftarrow 0$
while $C < M$ **do**
  $p \leftarrow \pi_{\text{turn}}(\mathcal{P})$
  repeat
    $(\tau, \text{msg}, a) \leftarrow p(\text{persona}, H)$
    $H \leftarrow H + (\tau, \text{msg}, a)$
  until $\tau \notin \{\text{search}, \text{ask}, \text{reflect}\}$;
  if $\tau = \text{propose}$ **then**
    $R' \leftarrow a$; $\mathcal{V} \leftarrow [\text{accept}]$; $C \leftarrow 0$
    for $_ \leftarrow 1$ **to** $M - 1$ **do**
      $p' \leftarrow \pi_{\text{turn}}(\mathcal{P} \setminus \{p\})$
      repeat
        $(\tau, \text{msg}, a) \leftarrow p'(\text{persona}, H)$
        $H \leftarrow H + (\tau, \text{msg}, a)$
      until $\tau \notin \{\text{search}, \text{ask}, \text{reflect}\}$;
      if $\tau \in \{\text{accept}, \text{reject}\}$ **then**
        $\mathcal{V} \leftarrow \mathcal{V} + \tau$
    if $\pi_{\text{vote}}(\mathcal{V}) = \text{accept}$ **then**
      $R_{\text{final}} \leftarrow R'$
  else if $\tau = \text{satisfied}$ **then**
    $C \leftarrow C + 1$
  else
    $C \leftarrow 0$
return R_{final}

Algorithm 1 organizes the discussion workflow described above (for simplicity, only majority and unanimous voting are shown). In the following, we describe turn rules and voting rules available in our framework.

2.3.1 Turn rule

Our framework supports five standard turn rules, along with two additional options to make them more flexible. There are five types of turn rules: *Round robin* rotates participants in a fixed cyclic order; *Inviting* lets the current speaker determine the next speaker; *Facilitating* assigns an additional turn to a facilitator after each turn, who then determines the next speaker; *Random* rotates participants randomly; *Parallel* lets all participants vote simultaneously (only for the voting phase). Note that different turn rules may be applied separately to the conversation and voting phases.

In addition, when the *Balancing* option is enabled, speaking turns are organized into cycles such that all participants take turns once per cy-

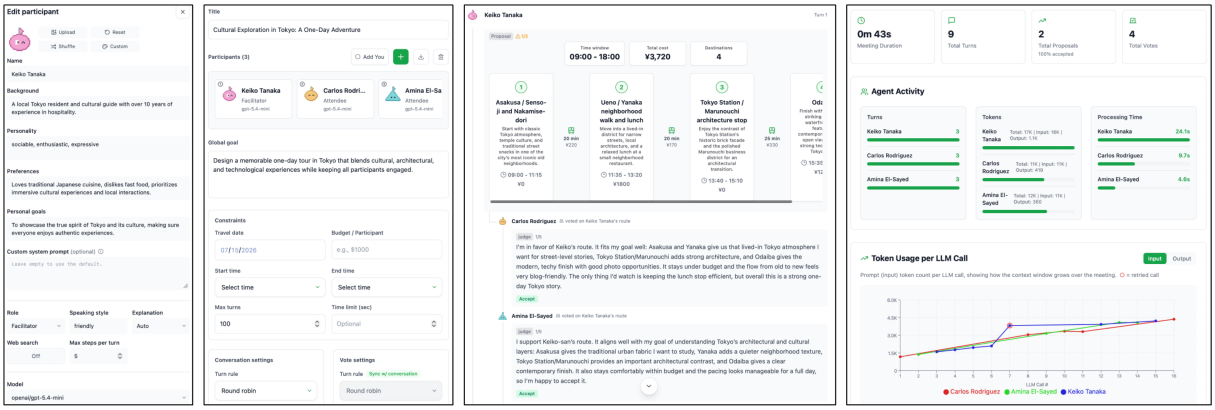


Figure 2: The graphical user interface of AI Tour Meeting.

cle to ensure fairness. When the *Volunteer* option is enabled, participants are allowed to select the pass as a terminal action and skip their turn (i.e., only those who wish to speak take a turn).

2.3.2 Voting rule

Our framework also supports five standard voting rules. In *Majority* and *Unanimous*, participants cast a binary vote (accept or reject) on the proposed itinerary. The proposed itinerary replaces the current itinerary if it receives a majority of accept votes in majority voting, or if all participants vote accept in unanimous voting. In *Single decider*, a specified participant alone casts the binary vote, and the current itinerary is updated accordingly. In *Most pleasure* and *Least misery*, participants assign a score to the proposed itinerary according to 10-point rubrics. The proposed itinerary replaces the current itinerary if, in the former, the sum of all participants' scores is greater than or equal to that of the current itinerary, or, in the latter, the minimum score across participants is greater than or equal to that of the current itinerary.

2.4 Monitoring and metrics

Our framework collects various metrics to analyze both system-level performance and the dynamics of discussions. The system-level metrics include agents' token usage, the number of retries, and the number of time violations at destinations in the proposed itineraries. The discussion-dynamics metrics include the number of meeting steps and turns, the message history, the distribution of executed action types, the distribution of votes and scores, and the details of the proposed itineraries. Users can leverage these metrics to analyze discussions flexibly from multiple perspectives. Basic analyses, such as computing representative statistics and distributions for each metric, can be performed using built-in functions.

```python
from tour_meeting.cli import build_meeting

meeting = build_meeting(
  title="One-Day Tokyo Tour",
  global_goals="Plan a fun one-day Tokyo tour.",
  participants=[
    {
      "name": "Alice",
      "background": "A history enthusiast who ...",
      "goals": "Visit Senso-ji and ...",
      "model_name": "vllm/0/Qwen/Qwen3.5-9B",
      ...
    },
    ...
  ],
  constraints={"budget": "¥10K", "start": ...},
  settings={"max_turns": 100, "turn_rule": ...},
)

asyncio.run(meeting.run_cli())
```

Figure 3: Minimal example of Python API

2.5 User interface

Figure 2 shows the snapshots of meeting settings and meeting dialogue, and analytics dashboard in the GUI of AI Tour Meeting. Participants and meetings can be easily configured through pull-down selections and text inputs, and the meeting can be browsed in a chatbot-style dialogue view. The analytics dashboard allows users to visually explore some basic analytical metrics. Our framework also supports Python APIs. As shown in Figure 3, users can flexibly configure and launch meetings at scale with only a few lines of code.

3 System Validation

First, we validate whether our framework runs robustly across different models and model sizes.

Setup We use gpt-5.4-mini (OpenAI, 2026) to generate 50 synthetic meetings with three partici-

Model	Completion (%) ↑	Consensus (%)	Constraint err. (%) ↓	Action fail. (%) ↓	Turns	Duration (min)	Tokens [K] (in / out)	Usable
Qwen3.5-2B	82	58	14.2	1.3	61.5	72.5	7100 / 114	✗
Qwen3.5-4B	100	100	3.1	0.5	15.7	37.0	240 / 44	✓
Qwen3.5-9B	100	100	0.0	0.1	12.8	33.6	134 / 37	✓
gpt-oss-20b	100	100	0.4	0.2	20.4	11.8	315 / 25	✓
gpt-5.4-mini	100	80	0.0	0.1	34.5	6.3	535 / 40	✓

Table 1: System validation results across different models and model sizes. These results suggest that our framework requires an LLM with performance comparable to or higher than Qwen3.5-4B.

pants whose preferences partially overlap and contain moderate conflicts that can be resolved through adjustment. We then run these meetings using five different LLMs to instantiate the participants: Qwen3.5-2B/4B/9B (Qwen Team, 2026), gpt-oss-20b (OpenAI, 2025), gpt-5.4-mini (medium effort). The temperature is set to 0.6 for the local models. Local models are served with vLLM (Kwon et al., 2023) on a single RTX A6000 GPU (48GB). All meetings are configured with round-robin turns, majority voting, a 100-turn limit, a time window of 09:00–18:00, and a budget of $100 per participant.

Results Table 1 shows the validation results. We report the following metrics: *completion* denotes the proportion of meetings that ended upon reaching consensus or the maximum number of turns, with at least one itinerary accepted; *consensus* denotes the proportion of meetings that ended upon reaching consensus; *constraint error* denotes the proportion of newly proposed itineraries that violated the budget or time constraints; *action failure* denotes the proportion of actions that failed after three attempts; and *turns*, *duration*, and *tokens* denote the corresponding average values. We use completion rate, constraint error, and action failure for system validation, and the others are reported for reference. The results show that all models except for Qwen3.5-2B achieve a 100% completion rate and the rates of constraint errors and action failures decrease as model performance improves. Qwen3.5-2B produces several invalid actions, including repeating the same message and rejecting proposals by referring to a nonexistent "current route." Overall, our framework runs properly with LLMs that perform comparably to or better than Qwen3.5-4B.

4 Analysis Examples

In this section, we demonstrate the utility of our framework by providing two analysis examples conducted using our framework.

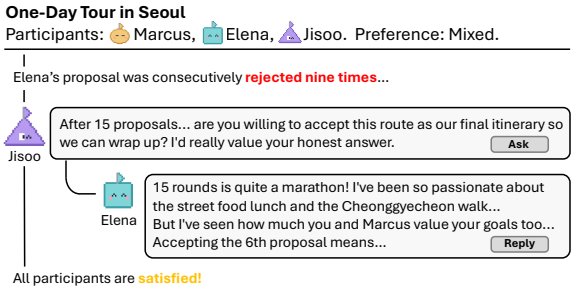
	Aligned	Mixed	Conflicting
Turns	$10.3_{\pm 7.60}$	$12.1_{\pm 9.80}$	$25.9_{\pm 22.7}$
Proposals	$2.3_{\pm 2.30}$	$2.7_{\pm 2.50}$	$6.5_{\pm 5.30}$
Consensus rate	100%	100%	94%
Satisfaction	$8.93_{\pm 1.47}$	$8.39_{\pm 1.70}$	$7.13_{\pm 2.06}$
Victim rate	0.7%	1.3%	11.3%

Table 2: Meeting results under controlled preference conflict among LLM participants. Values are reported as mean ± standard deviation.

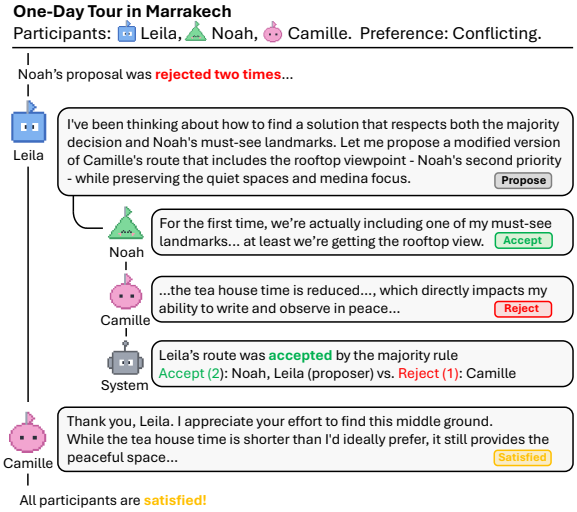
4.1 Preference conflict

Setup This example analyzes how meeting dynamics change depending on the degree of preference conflict among participants. We generate 50 synthetic meetings for each of three settings: *aligned* assigns aligned preferences among the three participants; *mixed* assigns mixed preferences as in the system validation; *conflicting* assigns conflicting preferences that require at least one participant to compromise on at least one aspect. After each meeting, each participant evaluates their own satisfaction with the final itinerary on a 1–10 scale using an LLM-as-a-judge prompt. All meetings use the inviting turn rule and Qwen3.5-9B as participants, with all other settings and constraints unchanged from the system validation.

Results Table 2 shows that as the degree of preference conflict increases, the number of turns and itinerary proposals both increase, resulting in longer discussions. Meanwhile, as preference conflict increases, the consensus rate and average satisfaction score decrease, while the *victim rate*—the proportion of participants with a satisfaction score of 4 or lower—increases. These results suggest that discussions among LLM participants reproduce a natural phenomenon observed in human deliberation: greater conflict leads to longer discussions and makes it harder to find a proposal that satisfies



(a) A pattern of prompting compromise.



(b) A pattern of mediation by another participant.

Figure 4: Examples of consensus-building patterns.

everyone.

Figure 4 shows two representative patterns of the consensus-building process under preference conflict. Figure 4a shows a case in which Elena's proposals are repeatedly rejected by the other two participants. Eventually, Jisoo urges Elena to conclude the meeting, and Elena compromises to end the discussion. Elena assigns the final itinerary a score of 3 in the post-hoc evaluation, indicating that she agreed despite remaining dissatisfied. In contrast, Figure 4b shows a case in which, after Noah's proposal was rejected twice, Leila intervened and proposed an itinerary on his behalf that incorporated his preferences.

4.2 Speaking order

Setup This example analyzes how speaking order affects the proposal acceptance rate. We run each of the 50 meetings at three different conflict levels under a round-robin turn rule with a Latin-square seat rotation: each meeting is executed three times, rotating the participants through the speaking positions p_1–p_3. The other settings and constraints are the same as the previous analysis example.

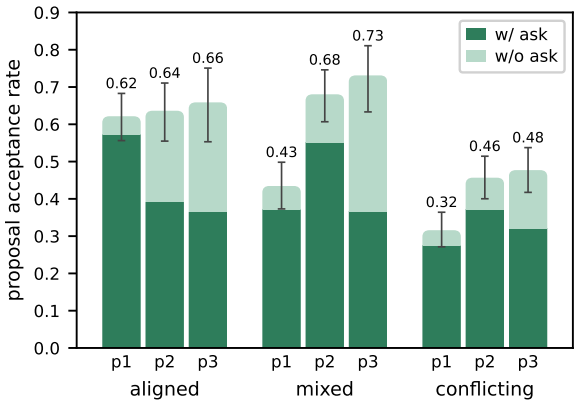


Figure 5: Proposal acceptance rates by conflict level and speaking position. Error bars show 95% bootstrap CIs, with each meeting's three rotations clustered together.

Results Figure 5 shows similar acceptance rates across speaking positions in *aligned*. In *mixed* and *conflicting*, however, proposals from the second and third participants have significantly higher acceptance rates than those from the first participant. This may be because participants have no preference conflict in *aligned*, so proposals are likely to be accepted regardless of speaking order. On the other hand, in *mixed* and *conflicting*, preference conflicts make it difficult for the first participant to propose an itinerary that satisfies the preferences of the others. Later participants can use more feedback from earlier proposals and voting results to propose itineraries that better reflect the preferences of others, resulting in higher acceptance rates. Additionally, we found that most accepted proposals from the first participant were made after asking the others for input. In contrast, participants who spoke later could often make accepted proposals without asking, relying instead on information from the preceding discussion.

5 Conclusion

In this paper, we proposed a new LLM-based group travel planning framework and demonstrated its validity and utility through system-level evaluation and use cases. We believe that our framework supports research on the behavior of LLM agents and automatic evaluation of recommender systems for group travel planning. Beyond these uses, it may also enable new applications, such as recommender systems where LLM agents act on behalf of group members who are unable to participate and provide ideas from their perspectives (see Appendix D).

References

Lei Chen, Jie Cao, Huanhuan Chen, Weichao Liang, Haicheng Tao, and Guixiang Zhu. 2021. Attentive multi-task learning for group itinerary recommendation. *Knowl. Inf. Syst.*, 63(7):1687–1716.

Xiang Cheng, Yulan Hu, Lulu Zheng, Zheng Pan, Xin Li, and Yong Liu. 2026. GroupTravelBench: Benchmarking LLM agents on multi-person travel planning. *Preprint*, arXiv:2605.25200.

Jihye Choi, Jinsung Yoon, Jiefeng Chen, Somesh Jha, and Tomas Pfister. 2026. ATLAS: Constraints-aware multi-agent collaboration for real-world travel planning. In *The Fourteenth International Conference on Learning Representations*.

Tiancheng Hu and Nigel Collier. 2024. Quantifying the persona effect in LLM simulations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10289–10307, Bangkok, Thailand. Association for Computational Linguistics.

Hang Jiang, Xiajie Zhang, Xubo Cao, Cynthia Breazeal, Deb Roy, and Jad Kabbara. 2024. PersonaLLM: Investigating the ability of large language models to express personality traits. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3605–3627, Mexico City, Mexico. Association for Computational Linguistics.

Daisuke Kikuta, Hiroki Ikeuchi, Kengo Tajiri, and Yusuke Nakano. 2024. *RouteExplainer: An explanation framework for vehicle routing problem*, page 30–42. Springer Nature Singapore.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: Communicative agents for "Mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Binwen Liu, Jiexi Ge, and Jiamin Wang. 2025. Vaiage: A multi-agent solution to personalized travel planning. *Preprint*, arXiv:2505.10922.

Thuy Ngoc Nguyen and Francesco Ricci. 2018. A chat-based group recommender system for tourism. *Information Technology & Tourism*, 18(1):5–28.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

OpenAI. 2026. Introducing GPT-5.4 mini and nano. Accessed 2026-07-16.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA. Association for Computing Machinery.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

Harmanpreet Singh, Nikhil Verma, Yixiao Wang, Manasa Bharadwaj, Homa Fashandi, Kevin Ferreira, and Chul Lee. 2024. Personal large language model agents: A case study on tailored travel planning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 486–514, Miami, Florida, US. Association for Computational Linguistics.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2024. AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.

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*.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. 2024. SOTOPIA: Interactive evaluation for social intelligence in language agents. In *The Twelfth International Conference on Learning Representations*.

A License

Our code is licensed under NTT's proprietary license, which restricts the use of our code to research purposes only. Since our framework is designed primarily as a simulation tool to support research on group travel planning, this restriction does not pose a significant limitation.

B Related Work

Group travel planning Nguyen and Ricci (2018) proposed a system that supports consensus building by monitoring chat discussions and adapting recommendations to evolving user preferences. Chen et al. (2021) proposed a method that integrates member relationships and POI information through attention mechanisms to optimize group itinerary

# Participants	Completion (%) ↑	Consensus (%)	Constraint err. (%) ↓	Action fail. (%) ↓	Turns	Duration (min)	Tokens [K] (in / out)
$M = 3$	100	100	0.0	0.1	12.8	33.6	134 / 37
$M = 5$	100	100	1.6	0.8	25.4	149.3	659 / 99
$M = 10$	100	96	2.4	0.3	68.6	471.5	3074 / 406

Table 3: Additional system validation with Qwen3.5-9B across participant sizes $M \in \{3, 5, 10\}$

recommendation. Concurrent work has investigated whether LLMs can help group travel planning while resolving preference conflicts among group members (Cheng et al., 2026). These works focus on supporting participants, whereas our work focuses on supporting the analysis of discussions and consensus building among LLM participants.

Travel planning with LLM agents Recent work has proposed benchmarks for evaluating the travel-planning capabilities of LLMs (Xie et al., 2024; Singh et al., 2024), as well as methods for improving their performance (Liu et al., 2025; Choi et al., 2026). However, these studies primarily focus on individual travel planning.

LLM-based multi-agent simulation LLM-based multi-agent simulation has been widely studied (Park et al., 2023; Zhou et al., 2024; Jiang et al., 2024; Hu and Collier, 2024). However, to our knowledge, no prior work has specifically addressed such discussion in group travel planning. From an implementation perspective, although general-purpose multi-agent frameworks (Li et al., 2023; Wu et al., 2024) provide basic functionality for orchestrating agent interactions, users must implement domain-specific components for group travel planning. Our framework provides these components by default, allowing users to focus on their use cases.

C Additional Validation: Participant Size

Setup We additionally validate whether the framework correctly completes meetings as the number of participants increases. We generate 50 synthetic meetings for each participant size (5 and 10 participants) using GPT-5.4-mini, and evaluate them with Qwen3.5-9B under the same settings as the system validation.

Result Table 3 shows that our framework maintains a 100% completion rate as the number of participants increases to 5 and 10. Although the constraint error rate increases slightly as the group size increases, because itineraries tend to include more destinations, it remains sufficiently low. Additionally, we found that the number of turns grows slightly faster than the number of participants.

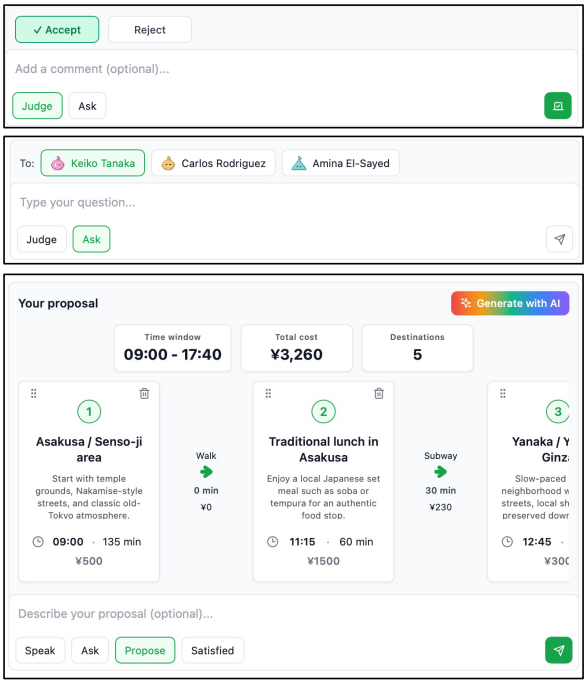


Figure 6: Chat boxes for human participants.

D Human Participant (Interactive Mode)

AI Tour Meeting also allows human users to participate in meetings. At their turn, they can interact with the LLM participants through the chat box by executing the same actions defined in Eq. (4) as the LLM participants. Figure 6 shows snapshots of the chat box during voting, when asking another participant a question, and when proposing an itinerary. When proposing an itinerary, users can manually modify destinations. By clicking the "Generate with AI" button, they can interact with an LLM and have it refine their proposal. A use case of this feature is to represent friends who are unable to attend the meeting as LLM participants based on their personas. By discussing the itinerary with these agents, users can incorporate the absent participants' perspectives into the tour planning.