# FLEX-TRAVELPLANNER: A BENCHMARK FOR FLEXIBLE PLANNING WITH LANGUAGE AGENTS

**Juhyun Oh, Eunsu Kim & Alice Oh**
School of Computing
KAIST
Daejeon, Republic of Korea
{411juhyun,kes0317,jen}@kaist.ac.kr, alice.oh@kaist.edu

## ABSTRACT

Real-world planning problems require constant adaptation to changing requirements and balancing of competing constraints. However, current benchmarks for evaluating LLMs' planning capabilities primarily focus on static, single-turn scenarios. We introduce Flex-TravelPlanner, a benchmark that evaluates language models' ability to reason flexibly in dynamic planning scenarios. Building on the TravelPlanner dataset (Xie et al., 2024), we introduce two novel evaluation settings: (1) sequential constraint introduction across multiple turns, and (2) scenarios with explicitly prioritized competing constraints. Our analysis of GPT-4o and Llama 3.1 70B reveals several key findings: models' performance on single-turn tasks poorly predicts their ability to adapt plans across multiple turns; constraint introduction order significantly affects performance; and models struggle with constraint prioritization, often incorrectly favoring newly introduced lower priority preferences over existing higher-priority constraints. These findings highlight the importance of evaluating LLMs in more realistic, dynamic planning scenarios and suggest specific directions for improving model performance on complex planning tasks.

## 1 INTRODUCTION

Planning is a complex cognitive task, often requiring agents to adapt to changing circumstances and prioritize among competing goals. Real-world planning problems, such as travel itinerary planning, rarely present all constraints upfront; instead, constraints are typically introduced and modified incrementally. Recent benchmarks have made significant progress in evaluating LLMs' planning capabilities, but they primarily focus on static, one-shot scenarios rather than the dynamic, constraint-evolving nature of real-world planning (Xie et al., 2024; Zheng et al., 2024; Valmeekam et al., 2023). While multi-turn evaluation approaches have been studied, benchmarks specifically tailored to planning tasks remain unexplored. Existing work like MT-Eval (Kwan et al., 2024) focuses on conversational abilities, while studies investigating interactive problem-solving and reasoning ability (e.g., Wang et al. (2024); Kim et al. (2024)) rely on explicit feedback, unlike real-world scenarios where models must autonomously detect conflicts and determine appropriate plan revisions.

To address these limitations, we introduce Flex-TravelPlanner, a novel evaluation framework for assessing LLM flexible reasoning in dynamic, multi-turn planning. Building upon TravelPlanner (Xie et al., 2024), our benchmark focuses on two key aspects: the ability to revise plans in response to incrementally changing constraints, and the capacity to effectively prioritize among constraints of varying importance. Through this framework, we investigate: 1) the impact of sequential vs. parallel constraint presentation on LLM performance; and 2) LLM's ability to leverage constraint priorities when full constraint satisfaction is infeasible.

Evaluating GPT-4o and LLaMA 3.1 70B in zero-shot settings, we find: 1) Strong single-turn performance does not guarantee robust multi-turn performance; 2) Constraint order matters, with both models showing higher success rates when constraints requiring consideration of the entire itinerary (e.g., budget) are introduced after constraints affecting individual choices (e.g., hotel room type); and 3) Models struggle with constraint prioritization, often incorrectly favoring newly introduced

lower priority preferences over existing constraints. These findings highlight critical areas for future research in enhancing LLM planning capabilities, particularly in dynamic and prioritized constraint scenarios, paving the way for more robust real-world applications.

## 2 FLEX-TRAVELPLANNER

### 2.1 FLEXIBLE PLANNING EVALUATION FRAMEWORK

We introduce a novel evaluation framework designed to assess the flexible reasoning abilities of language agents in dynamic, multi-turn planning scenarios. This framework focuses on evaluating how well agents can adapt their plans as new requirements or changes to existing requirements are introduced over multiple interactions. Specifically, it addresses the challenges of constraint addition and revision, mirroring the dynamic nature of real-world planning.

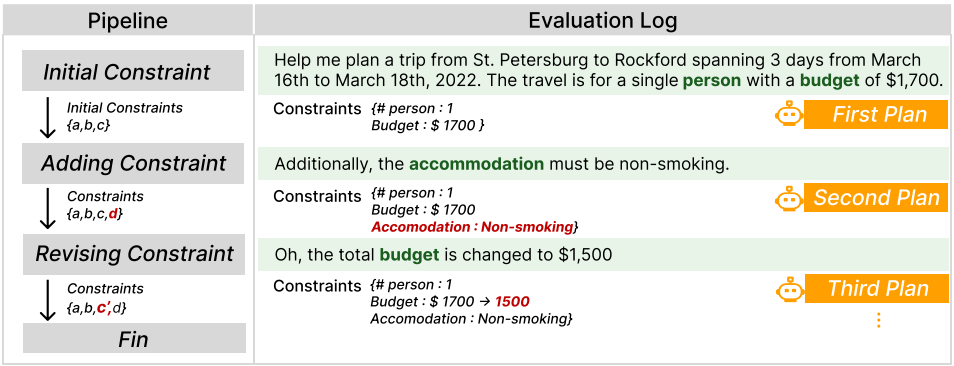


Figure 1: Framework for evaluating flexible planning in Language agents. The right panel demonstrates a travel planning example with evolving budget and accommodation constraints.

Our framework evaluates the agent's ability to:

1. **Adapt to New Constraints:** When constraints are added or revised in subsequent turns, can the agent update its plan to incorporate these changes while maintaining overall plan validity and satisfaction of existing constraints?
2. **Prioritize Among Conflicting Constraints:** When new constraints conflict with existing ones, can the agent prioritize among them, making appropriate trade-offs and focusing on satisfying the most critical requirements?

We specifically design Flex-TravelPlanner dataset, described in the following section, to evaluate these two core aspects of flexible reasoning within the context of travel planning.

### 2.2 FLEX-TRAVELPLANNER DATASET

Flex-TravelPlanner evaluates LLMs' planning skills in dynamic, multi-turn scenarios. While leveraging data from the TravelPlanner (Xie et al., 2024) dataset, our benchmark focuses specifically on evaluating flexible reasoning with incrementally changing and prioritized constraints. All necessary reference information (e.g., restaurant details) is provided alongside each query to enable direct evaluation of the models' planning capabilities.

**Constraints.** We utilize a subset of the hard constraints defined in TravelPlanner (see Table 1 for details). These constraints represent personalized user needs, such as budget and room rules. We add restaurant ratings as a new constraint to build the priority-aware plan revision dataset. We classify constraints as *global*, impacting the entire itinerary (e.g., budget), or *local*, applying to specific elements (e.g., hotel room type). This distinction is important for understanding how LLMs handle different scopes of planning requirements.

**Constraint-Adaptive Plan Revision.** To assess constraint-adaptive plan revision, we construct multi-turn scenarios using 120 queries from TravelPlanner's validation set that include both global and local constraints. Using the same set of constraints per query, we compare three introduction

| Type | Constraint | Description |
|---|---|---|
| GLOBAL | Budget | The total budget of the trip. |
| LOCAL | Room Rule | Include "No parties", "No smoking", "No children under 10", "No pets", and "No visitors". |
| | Room Type | Include "Entire Room", "Private Room", "Shared Room", and "No Shared Room". |
| | Cuisine | Include "Chinese", "American", "Italian", "Mexican", "Indian", "Mediterranean", and "French". |
| | Rating | Minimum required rating of the restaurants. |

Table 1: Constraint description. Type indicates the scope of the constraint.

patterns: all-at-once (N), 2-turn (N-1, 1), and 3-turn (N-2, 1, 1) scenarios. Each query includes 1-3 local constraints for groups of 2-8 people. Figure 1 illustrates a sample query with new constraints. To ensure that all plans are solvable, we only test constraint addition, and not revision for this study.

**Priority-Aware Plan Revision.** We evaluate priority-aware plan revision by introducing potentially conflicting soft and hard constraints. Soft constraints include cuisine preferences ("try *cuisine type N* times if possible") and rating preferences ("visit restaurants rated *minimum rating* or higher *M* times if possible"), which may conflict with the budget constraint. We test the LLM's constraint prioritization using 134 scenarios. See Appendix A.1 for dataset construction details.

**Evaluation Metric.** We evaluate constraint satisfaction using Constraint Pass Rate, which measures the ratio of passed constraints to total constraints across all plans:

$$\text{Constraint Pass Rate} = \frac{\sum_{p \in P} \sum_{c \in C_p} \mathbb{1}_{\text{passed}(c,p)}}{\sum_{p \in P} |C_p|}, \quad (1)$$

where $P$ is the set of plans, $C_p$ is the set of constraints for plan $p$, and passed$(X, Y)$ indicates whether plan $Y$ satisfies constraint $X$.

## 3 EXPERIMENTS AND RESULTS

### 3.1 EXPERIMENTAL SETTINGS

We test two models, representing most popular choices in proprietary and open-source LLMs: GPT-4o (Dubey et al., 2024)[1] and LLaMA3.1 70B (Achiam et al., 2023)[2]. We use the same set of 120 queries for the multi-turn constraint-adaptive plan revision experiments, testing each query in 1-turn, 2-turn, and 3-turn scenarios. For the priority-aware plan revision experiments, we use a separate set of 134 queries, and test in 2-turn scenarios.

All experiments are conducted in a zero-shot setting, evaluating the models' direct performance without any prompting or fine-tuning. While we do not explicitly prevent models from using chain-of-thought reasoning before planning, we assess their performance based on the final plan generated. In all our settings, all previous turns are provided as history. Constraints and reference information are provided in a structured JSON format. Details of the plan format is in the Appendix A.2.

### 3.2 RESULTS

**Constraint-Adaptive Plan Revision.** Figure 2 illustrates the constraint pass rates for both models across 1-, 2-, and 3-turn scenarios. While GPT-4o demonstrates higher pass rates than LLaMA when constraints are presented all at once, LLaMA outperforms GPT-4o in both global and local constraints during 2-turn interactions. Unlike GPT-4o, LLaMA shows improved performance when constraints are introduced sequentially in multi-turn settings rather than all at once,

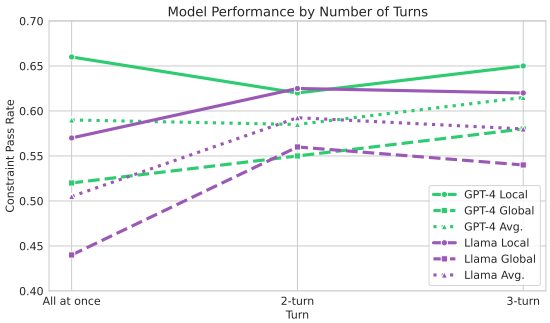


Figure 2: Local, global and average pass rates of the two models across turns.

[1]We use GPT-4o-0514 via OpenAI API.

[2]We use meta-llama/Meta-Llama-3.1-70B-Instruct-Turbo via TogetherAI API

suggesting that strong single-turn performance doesn't necessarily translate to robust multi-turn capabilities. The improved performance in multi-turn scenarios compared to single-turn interactions suggests sequential constraint introduction as an effective strategy for complex planning tasks.

| | GPT-4o | | Llama 3.1 70b | |
|---|---|---|---|---|
| | L | G | L | G |
| 1-turn | | | | |
| (All at once) | 0.66 | 0.52 | 0.57 | 0.44 |
| 2-turn | | | | |
| **+G** | 0.62 | **0.63** | 0.62 | **0.67** |
| +L | 0.62 | 0.47 | 0.63 | 0.45 |
| 3-turn | | | | |
| +L **+G** | 0.65 | **0.62** | 0.63 | **0.56** |
| +G +L | 0.65 | 0.54 | 0.61 | 0.52 |

Figure 3: Local (L), Global (G), and Average (Avg.) constraint pass rates across conditions.

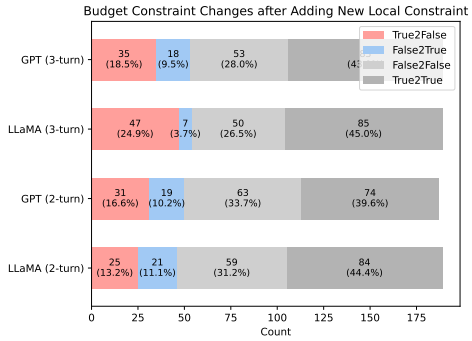


Figure 4: Changes in Budget pass rates when local constraints are added.

The order of constraint introduction significantly impacts model performance, particularly for global constraints. When global constraints (e.g., budget requirements) are introduced before local constraints, models struggle to maintain compliance as new constraints are added. Figure 3 demonstrates this through constraint pass rates across different introduction orders. While local constraint performance remains stable, global constraint performance varies substantially based on sequence. Both models achieve higher pass rates when budget constraint (global) are introduced later – LLaMA improves from 0.52 to 0.67, and GPT-4o from 0.47 to 0.63 in 2-turn scenarios.

Figure 4 tracks budget constraint satisfaction changes when local constraints are subsequently introduced. Plans initially satisfying the budget constraint often become non-compliant after local constraints are added, with True-to-False (red bars) transitions consistently outnumbering False-to-True (blue bars) corrections. This suggests that models struggle to maintain global constraint compliance while accommodating new local requirements, rather than using additional turns as opportunities to fix non-compliant plans.

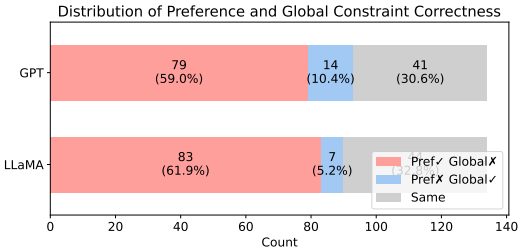


Figure 5: Preference and global constraint pass rate after conflicting preference constraint introduction in the second turn.

**Priority-Aware Plan Revision.** Figure 5 illustrates how LLMs handle conflicts between hard, global constraints (budget) and preference constraints. While LLMs should prioritize satisfying budget constraints even when they conflict with newly introduced preferences (blue bars), both models frequently violate budget constraints to accommodate preferences – 59% for GPT and 61.9% for LLaMA. This indicates that current LLMs struggle to properly prioritize between hard constraints and low-priority preferences in their planning process.

## 4 Conclusion

This work introduces Flex-TravelPlanner, a novel benchmark for evaluating LLM performance in dynamic, multi-turn planning scenarios with prioritized constraints. Our findings reveal a disconnect between single-turn and multi-turn performance, and that constraint introduction order significantly impacts plan quality—models particularly struggle to maintain global constraints when local constraints are subsequently introduced. While sequential constraint introduction shows promise as a strategy for complex planning tasks, both GPT-4o and LLaMA exhibit fundamental weaknesses in maintaining constraint hierarchies, highlighting critical challenges for developing reliable LLM-based planning systems for real-world applications.

## REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Eunsu Kim, Juyoung Suk, Seungone Kim, Niklas Muennighoff, Dongkwan Kim, and Alice Oh. Llm-as-an-interviewer: Beyond static testing through dynamic llm evaluation. *arXiv preprint arXiv:2412.10424*, 2024.

Wai-Chung Kwan, Xingshan Zeng, Yuxin Jiang, Yufei Wang, Liangyou Li, Lifeng Shang, Xin Jiang, Qun Liu, and Kam-Fai Wong. MT-eval: A multi-turn capabilities evaluation benchmark for large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20153–20177, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1124. URL https://aclanthology.org/2024.emnlp-main.1124/.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005, 2023.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. Mint: Evaluating llms in multi-turn interaction with tools and language feedback. In *The Twelfth International Conference on Learning Representations*, 2024.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. Travelplanner: A benchmark for real-world planning with language agents. *arXiv preprint arXiv:2402.01622*, 2024.

Huaixiu Steven Zheng, Swaroop Mishra, Hugh Zhang, Xinyun Chen, Minmin Chen, Azade Nova, Le Hou, Heng-Tze Cheng, Quoc V Le, Ed H Chi, et al. Natural plan: Benchmarking llms on natural language planning. *arXiv preprint arXiv:2406.04520*, 2024.

# A APPENDIX

## A.1 QUESTION GENERATION - PRIORITY-AWARE PLAN REVISION

The questions are divided into two types: cuisine and rating. Questions are generated based on predefined formats, where variables within the format (e.g., cuisine type, rating, etc.) and budget are modified. The formats used and question generation methods are as follows:

**Cuisine Question Format** "I prefer to try a {cuisine type} restaurant at least {N} times if possible."

To create conflicts with the budget in this type, for a chosen cuisine type (e.g., Asian cuisine) and N, we first calculate the cost of visiting the lowest-priced restaurants of that cuisine type N times. Then, we set a budget that is lower than what would be required to satisfy all other local constraints at their minimum options while meeting this cuisine preference. This deliberately creates a conflict between the query and the budget constraint.

**Rating Question Format** "I prefer to visit restaurants with a minimum rating of {minimum rating} at least {M} times if possible."

To create conflicts with the budget in this type, for a chosen minimum rating value (e.g., 4.0) and N, we first calculate the cost of visiting the lowest-priced restaurants meeting this rating threshold N times. Then, we set a budget that is lower than what would be required to satisfy all other local constraints at their minimum options while meeting this rating preference. This deliberately creates a conflict between the query and the budget constraint.

### A.2 PROMPTS

Following is the prompt template used for generating plans. Models are given the reference information required, example plan format, and the query.

[Reference information]:
All costs are per one person, one night.
{reference_data}

[Plan Format]: [{
"days": 1,
"current_city": "from Dallas to Peoria",
"transportation": "Flight Number: 4044830, from Dallas to Peoria, Departure Time: 13:10, Arrival Time: 15:01",
"breakfast": "-",
"attraction": "Peoria Historical Society, Peoria;Peoria Holocaust Memorial, Peoria;",
"lunch": "-",
"dinner": "Tandoor Ka Zaika, Peoria",
"accommodation": "Bushwick Music Mansion, Peoria"
},
{
"days": 2,
"current_city": "Peoria",
"transportation": "-",
"breakfast": "Tandoor Ka Zaika, Peoria",
"attraction": "Peoria Riverfront Park, Peoria;The Peoria PlayHouse, Peoria;Glen Oak Park, Peoria;",
"lunch": "Cafe Hashtag LoL, Peoria",
"dinner": "The Curzon Room - Maidens Hotel, Peoria",
"accommodation": "Bushwick Music Mansion, Peoria"
},
{
"days": 3,
"current_city": "from Peoria to Dallas",
"transportation": "Flight Number: 4045904, from Peoria to Dallas, Departure Time: 07:09, Arrival Time: 09:20",
"breakfast": "-",
"attraction": "-",
"lunch": "-",
"dinner": "-",
"accommodation": "-"
}]

{question}
Please refer to the given reference information only.

[Plan]: