Can LLMs Truly Plan? A Comprehensive Evaluation of Planning Capabilities

Gayeon Jung[†], **HyeonSeok Lim**[§], **Minjun Kim**[§], **Joon-Ho Lim**[¶], **KyungTae Lim**[‡*], **Hansaem Kim**[†*]
[†]Yonsei University, [§]Seoul National University of Science and Technology
[¶]Tutorus Labs, [‡]Korea Advanced Institute of Science and Technology
wjdrkdus98@yonsei.ac.kr, {gustjrantk,mjkmain}@seoultech.ac.kr
jhlim@tutoruslabs.com, ktlim@kaist.ac.kr, khss@yonsei.ac.kr

Abstract

The existing assessments of planning capabilities of large language models (LLMs) remain largely limited to single-language or specific representation formats. To address this gap, we introduce the Multi-Plan benchmark comprising 204 multilingual and multi-format travel planning scenarios. In experimental results obtained with state-of-the-art LLMs, the Multi-Plan benchmark effectively highlights the performance disparities among models, notably showing superior results for reasoning-specialized models. Interestingly, language differences exhibited minimal impact, whereas mathematically structured representations significantly improved planning accuracy for most models, underscoring the crucial role of the input format. These findings enhance our understanding of planning abilities of LLMs, offer valuable insights for future research, and emphasize the need for more sophisticated AI evaluation methods. This dataset is publicly available at huggingface.co/datasets/Bllossom/Multi-Plan.

1 Introduction

Large language models (LLMs) have demonstrated near-human performance in various tasks, including translation and summarization (Zhao et al., 2023; Chang et al., 2024). However, their potential and limitations in human-specific tasks like planning remain underexplored (Team et al., 2024; Wang et al., 2024). Planning ability is crucial for human-level intelligence yet remains challenging for LLMs, as it requires understanding multilayered constraints and sequential reasoning (Wei et al., 2025; Gui et al., 2025).

Current evaluations of LLM planning capabilities follow two paradigms: classical and practical. Classical planning focuses on generating abstract and formalized action sequences for the transition from initial to goal states, as exemplified by tasks such as block manipulation (Geffner and Bonet, 2013; Frances et al., 2017). Practical planning, conversely, addresses real-world complexities, such as itinerary development involving time and budget allocation, and prioritization (Russell and Norvig, 2016). Notably, existing benchmarks predominantly rely on specific formal languages like the planning domain definition language (PDDL) or single-language natural environments, primarily English (Valmeekam et al., 2023a; Zhang et al., 2024). This contrasts with recent multilingual and multi-format benchmarks such as HumanEval (Chen et al., 2021) and the multilingual grade school math (MGSM; Shi et al., 2022) benchmark, highlighting the need for comprehensive and balanced assessments to reflect LLM adaptability across diverse languages and formats.

To this end, we propose Multi-Plan, a benchmark designed to comprehensively assess LLM planning capabilities in multilingual and multi-format contexts. Multi-Plan consists of 204 travel planning scenarios, each including planning requests and corresponding correct plans expressed in Korean, English, and mathematical formats. All scenarios were manually constructed by human annotators to ensure consistency and reliability. Additionally, we conducted in-depth evaluations of planning accuracy for the latest LLMs developed by leading AI enterprises, shedding light on current challenges and future development directions. The key contributions of this study include:

- A comprehensive planning benchmark featuring Korean-English-mathematical multilingual and multi-format scenarios.
- An extensive diagnosis of planning capabilities across modern LLMs, including specialized reasoning models.
- Analysis of multidimensional factors influencing planning performance.

*Corresponding authors.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 13069–13084

2 Multi-Plan

We propose Multi-Plan, a benchmark designed to comprehensively evaluate the planning capabilities of LLMs across multilingual and multi-format contexts. This benchmark refers to Natural Plan (Zheng et al., 2024), but unlike Natural Plan, which exclusively relies on English, our benchmark encompasses Korean, English, and mathematical expressions.

2.1 Data Construction

Table 1 illustrates an example scenario from Multi-Plan. The data construction involved three main stages. We began by creating Korean planning request-answer pairs. Structured guidelines and templates covering city selection, duration settings, and constraint specifications were formulated to ensure that each scenario yields a unique optimal solution. Three human annotators manually generated 204 Korean scenarios, maintaining balanced distributions concerning the number of cities visited and minimizing the influence of variability on the model outputs. The detailed guidelines, templates, dataset compositions, and examples are provided in Appendix B, C.

In the second stage, the Korean dataset was translated into English using DeepL, with human annotators reviewing and refining translations to ensure accuracy and naturalness.

The final stage consisted of converting Korean planning requests into mathematically structured representations. Essential scenario components, including total travel duration, city-specific durations of stay, and available flight connections, were formalized mathematically or logically. Given the dataset's practical nature, we adopted a hybrid approach, mathematically representing critical constraints while preserving contextual information in natural language. The transformation employed Anthropic's Claude 3.5 Sonnet model using a few-shot prompting strategy for consistency and accuracy. Following this, the outputs generated by the model were meticulously reviewed by human workers to ensure their final quality. Consequently, each scenario in Multi-Plan includes requests in three distinct formats while also sharing the same natural language plan.

2.2 Dataset Features

Multi-Plan, underpinned by its multilingual (Korean, English) and multi-format (natural language, mathematical) design, strategically assesses genuine planning capabilities beyond mere language processing. Korean is an agglutinative language exhibiting a subject–object–verb (SOV) structure, linguistically contrasting with English, an isolating language characterized by a subject–verb–object (SVO) order (Kim, 2024; Park et al., 2016). Investigating planning consistency across these structurally distinct languages provides insights into LLM multilingual capabilities and language dependency.

Moreover, comparing natural language to mathematical representations yields insights into format-specific performance differences. For instance, Pallagani et al. (2023) highlighted superior performance in planning tasks by code-specialized models versus general text models, suggesting that structured formats positively influence model accuracy. Considering the structural similarities between travel planning and linear programming (Karloff, 2008), including constraints, optimization, and resource allocation, we reformulated natural language requests into mathematical representations. Such precise mathematical structures mitigate ambiguity, potentially enhancing planning accuracy.

```
Refer to the example provided below, and present
the solution to the given task using precisely the
same format as demonstrated in the example. Your
solution should be concise, and you should omit any
additional explanations.
[Example]
Task:{example_task},Solution:{example_solution}
Task:{task}, Solution:
```

Figure 1: An example of the evaluation prompt template.

3 Evaluation of Planning Performance

3.1 Experimental Setup

Model We evaluated various LLMs developed by OpenAI, Anthropic, and Google DeepMind, including reasoning-specialized and open-source models. Detailed model specifications and characteristics are provided in Appendix E.1.

Prompt Responses were elicited using prompts, as shown in Figure 1. Each response incorporated randomly selected examples from the dataset to ensure accurate comprehension and appropriate response formats.

English	Korean	Math
I am planning to visit three European cities over a total of 14 days. When traveling between cities, I will use only non-stop flights. Each transfer can be completed within a single day, and the flights do not reduce the time spent in any city. I would like to stay in Florence for 6 days and meet a friend there sometime between day 9 and day 14. I also want to spend 5 days in Barcelona and 5 days in Helsinki. Direct flights are available on the following routes: Barcelona ↔ Florence and Helsinki ↔ Barcelona. Using only these direct flights, please create a 14-day itinerary that covers all three cities.	총 14일 동안 유럽 3개 도시를 방문할 계획입니다. 도시 간을 이동할 때는 직항 항공편만 이용합니다. 도시 간 이동은 하루 안에 가능하며, 항공편은 각 도시에 머무르는 일정에 영향을 끼치지 않습니다. 6일 동안 피렌체를 방문하고 싶습니다. 9일에서 14일 사이에 피렌체에서 친구를 만나고 싶습니다. 5일 동안 바르셀로나를 방문하고 싶습니다. 5일 동안 헬싱키를 방문하고 싶습니다. 직항편이 있는 도시는 다음과 같습니다: 바르셀로나와 피렌체, 헬싱키와 바르셀로나. 직항 항공편을 이용하여 14일 동안 세 도시를 방문하는 여행 계획을 세워보세요.	**Variable Definitions**: H: Helsinki, B: Barcelona, F: Florence a_H: Arrival date in Helsinki, d_H: Departure date from Helsinki a_B: Arrival date in Barcelona, d_B: Departure date from Barcelona a_F: Arrival date in Florence, d_F: Departure date from Florence **Constraints**: $d_H - a_H + 1 = 5$, $d_B - a_B + 1 = 5$, $d_F - a_F + 1 = 6$, $\max(d_H, d_B, d_F) - \min(a_H, a_B, a_F) + 1 = 14$, $\lvert a_F, d_F \rvert \cap [9, 14] \neq \emptyset$ **Direct Flights**: $(B \leftrightarrow F)$, $(H \leftrightarrow B)$, Travel between cities takes at most one day and does not affect the duration of stays. Plan an itinerary visiting all three cities using direct flights.
Solution Plan (A): Days 1–5: Visit Helsinki for 5 days; Day 5: Fly from Helsinki to Barcelona; Days 5–9: Visit Barcelona for 5 days; Day 9: Fly from Barcelona to Florence; Days 9–14: Visit Florence for 6 days.		

Table 1: Examples from the Multi-Plan dataset constructed in English, Korean, and Mathematical Structuring. **Solution Plan (A)** illustrates a correct itinerary solution, and the **Variable Definitions**, **Constraints**, and **Direct Flights** sections specify the structured mathematical representation of the planning problem.

3.2 Result and Analysis

Using Multi-Plan, we assessed the planning accuracy of 11 models. The accuracy measurements relied on exact matches of the generated plans to the dataset answers, particularly date ranges and city names. Regular expressions extracted itinerary details (visited cities and stay durations) and travel logistics (departure and arrival points), verifying matches to establish correct answers. For instance, if a model produced "`Days 1-3: Visit Paris for 3 days`," the script would verify exact correspondence with the correct date range, duration, and city name. Detailed evaluation methods and regular expression scripts are provided in Appendix F.1. Accuracy comparisons among models are summarized in Table 2.

Model	Reasoning	Korean	English	Math
o3-mini	✓	82.84%	82.84%	80.39%
GPT-4o	✗	3.43%	3.43%	14.22%
GPT-4-turbo	✗	10.78%	10.78%	17.65%
GPT-4	✗	15.20%	15.20%	21.08%
Claude-3.7-Sonnet thinking	✓	46.57%	48.53%	55.39%
Claude-3.7-Sonnet standard	✗	29.41%	30.39%	37.27%
Claude-3.5-Haiku	✗	8.33%	8.33%	11.27%
Claude-3-Opus	✗	27.45%	27.45%	25.00%
Gemini-2.5-Pro	✓	82.35%	83.82%	82.84%
Gemini-2.0-Flash	✗	11.76%	9.80%	21.08%
Gemini-1.5-Pro	✗	6.37%	11.76%	7.84%

Table 2: Evaluation results of closed-source models on the Multi-Plan dataset. The 'Reasoning' column denotes whether the model was prompted with a reasoning-oriented instruction (✓) or a standard instruction without explicit reasoning (✗).

Distintive Performance Patterns in Multi-Plan Evaluating LLM planning capabilities using Multi-Plan revealed clear performance variations among models. Reasoning-specialized models such as o3-mini and Gemini-2.5-Pro consistently demonstrated very high accuracy across all formats, significantly outperforming Claude-3.7-Sonnet thinking, which recorded considerably lower accuracy. General models also exhibited notable intra-series variations, with Claude-3.7-Sonnet standard and Claude-3-Opus demonstrating relatively higher accuracy compared to the GPT-4 and Gemini series. These observations contrast with established benchmarks such as MMLU (Hendrycks et al., 2020) or GSM8K (Cobbe et al., 2021), where top-performing models typically exhibit closely similar performances. The results suggest that Multi-Plan is sensitive to subtle differences in model architectures and training methodologies, providing distinct insights that conventional benchmarks may overlook.

Superior Performance of Reasoning Models Reasoning-specialized models exhibited notably superior performance in the Multi-Plan evaluations. Specifically, o3-mini and Gemini-2.5-Pro achieved exceptionally high accuracy rates, averaging 82.03% and 83.01%, respectively. These models demonstrated robust capabilities in effectively addressing planning requests that included complex constraints. Claude-3.7-Sonnet thinking also

performed reasonably well, averaging 50.16% accuracy, significantly surpassing the maximum accuracy of 37% achieved by general LLMs. These findings empirically support the theoretical expectation that enhanced reasoning capabilities significantly improve performance in complex planning tasks.

Impact of Language on Planning Performance Because Multi-Plan contains identical planning scenarios in both Korean and English, it enables a detailed examination of how language affects LLM planning performance. The models showed only a minor average accuracy increase of approximately 0.71% for the English scenarios, with most models displaying an accuracy difference of less than 2% between Korean and English. This small performance variation, despite substantial structural and morphological differences between the two languages, underscores the advanced multilingual processing abilities of contemporary LLMs. This indicates that modern models have progressed significantly beyond basic linguistic comprehension and can effectively handle complex tasks across diverse languages.

Impact of Representation Format on Planning Performance This study examined the performance differences between natural language (Korean) and mathematically structured representations. Results indicated that 9 out of 11 models achieved improved accuracy with mathematically structured inputs. Notably, GPT-4o, Claude-3.7-Sonnet thinking, and Gemini-2.0-Flash exhibited improvements exceeding 8%. These findings highlight the practical importance of structured input formats in enhancing LLM performance for complex tasks, such as planning. Thus, the future research focusing on the development of LLM-based planning systems should consider adopting structured representation formats. However, the relative performance rankings among the models remained largely consistent regardless of the input format, underscoring that enhancing fundamental understanding and reasoning capabilities remains a critical task alongside input format optimization.

Performance Analysis of Open-Source Models In addition to closed-source models, we evaluated how open-source models perform on the Multi-Plan benchmark. A total of 11 open-source models were assessed using the same evaluation methodology applied to the closed-source models. Detailed results are provided in Table 3, and comprehensive model information is presented in Appendix E.2.

Model	Reasoning	**Korean**	**English**	**Math**
OLMo-2-7B-Instruct	✗	0.00%	1.96%	0.00%
SmolLM2-1.7B-Instruction	✗	0.00%	0.98%	0.00%
Llama3.1-8B-Instruction	✗	1.96%	6.86%	0.49%
Llama3.3-70B-Instruction	✗	7.84%	4.41%	12.75%
Qwen2.5-72B-Instruction	✗	4.41%	6.86%	15.20%
Llama-3.1-Nemotron-Nano-8B	✓	0.00%	2.45%	0.00%
Phi-4-reasoning-plus	✓	0.00%	0.49%	0.00%
DeepSeek-R1-Distill-Llama-8B	✓	0.49%	3.43%	0.00%
DeepSeek-R1-Distill-Qwen-7B	✓	0.00%	0.98%	0.00%
Qwen3-8B	✓	7.84%	10.29%	7.35%
Qwen3-32B	✓	25.98%	46.57%	26.47%

Table 3: Performance results of 11 open-source models on the Multi-Plan benchmark

Overall, our analysis reveals that open-source models still exhibit clear limitations and generally underperform compared to their closed-source counterparts. Specifically, non-reasoning models showed almost negligible performance at smaller scales, while larger models (70B or more parameters) demonstrated relatively better capabilities. Notably, these larger-scale models achieved stronger performance in the 'Math' domain, scoring 12.75% and 15.20%, surpassing their performance in Korean and English tasks. Meanwhile, reasoning-equipped models mostly exhibited limited capabilities as well, except for models from the Qwen3 series, which stood out significantly. In particular, the Qwen3-32B model achieved robust performance across all evaluated domains, with notable scores in Korean (25.98%), English (46.57%), and Math (26.47%), closely matching the performance of the commercial model 'Claude-3.7-Sonnet standard'.

Performance by Number of Cities Visited Examining the 'Cities Visited' column in Table 4, all models showed the highest accuracy for scenarios involving the fewest number of cities (three cities), with overall performance declining as the number of cities increased. In particular, GPT-4, Gemini-1.5, and 2.0 achieved accuracy close to or above half of the requests when visiting three cities, but their accuracy gradually decreased for scenarios involving four to six cities. Generally, increasing the number of cities adds complexity due to additional constraints, making planning more challenging; however, a consistently declining pattern was not clearly observed in this experiment. This may be due to the relatively small number of samples per scenario, suggesting the need for further analysis with more data. Nevertheless, the notably higher accuracy observed in requests with fewer cities clearly indicates that increased information processing requirements are a major cause

Model	Reasoning	Cities Visited				Continent		
		3	4	5	6	Europe	Asia	America
o3-mini	✓	26.63%	24.85%	24.26%	24.26%	35.50%	31.36%	33.14%
GPT-4-turbo	✗	50.00%	4.55%	27.27%	18.18%	54.55%	18.18%	27.27%
GPT-4	✗	67.74%	12.90%	9.68%	9.68%	41.94%	32.26%	25.81%
GPT-4o	✗	100.00%	0.00%	0.00%	0.00%	14.29%	28.57%	57.14%
Claude-3.7-Sonnet thinking	✓	28.42%	26.32%	28.42%	16.84%	36.84%	31.58%	31.58%
Claude-3.7-Sonnet standard	✗	35.00%	28.33%	21.67%	15.00%	45.00%	28.33%	26.67%
Claude-3.5-Haiku	✗	64.71%	17.65%	11.76%	5.88%	17.65%	47.06%	35.29%
Claude-3-Opus	✗	37.50%	14.29%	28.57%	19.64%	48.21%	23.21%	28.57%
Gemini-2.5-Pro	✓	26.19%	26.19%	22.62%	25.00%	33.93%	32.74%	33.33%
Gemini-2.0-Flash	✗	45.83%	8.33%	25.00%	20.83%	50.00%	12.50%	37.50%
Gemini-1.5-Pro	✗	46.15%	15.38%	23.08%	15.38%	53.85%	15.38%	30.77%

Table 4: Accuracy of travel-plan generation by number of cities visited (3–6) and by continent (Europe, Asia, America) for each model. The 'Cities Visited' columns indicate the percentage of correctly generated plans categorized by the number of cities in the itinerary. The 'Continent' columns show the distribution of correct plans based on the continent the travel plan is set in.

of performance degradation.

Performance by Continent The evaluation dataset was constructed with balanced representation across continents, allowing for precise analysis of regional accuracy variations. Analyzing the 'Continent' column in Table 4, we found that seven models recorded their lowest accuracy in Asia, whereas most models showed relatively higher success rates in Europe and America. This suggests that regional biases inherent in training data may have influenced model performance.

4 Conclusion

This study introduced **Multi-Plan**, a comprehensive benchmark designed to evaluate LLM planning capabilities across multilingual and multi-format contexts. By providing 204 travel planning scenarios in Korean, English, and mathematical formats, Multi-Plan facilitates a broad and balanced assessment of LLM planning abilities.

Evaluations using Multi-Plan yielded several key insights. First, Multi-Plan effectively distinguished model performance nuances, highlighting not only the overall superiority of reasoning-specialized models but also sensitively reflecting model architecture-specific characteristics. Second, the negligible accuracy differences observed between two structurally distinct languages, Korean and English, emphasized the maturity of multilingual processing capabilities in modern LLMs. Ultimately, the study contributed to enhancing our understanding of the planning capabilities and limitations of contemporary LLMs and highlighted critical considerations for future research and development in sophisticated AI systems.

5 Limitation

Despite its contributions, this study has several limitations. First, the Multi-Plan dataset comprises only 204 planning scenarios, which limits the generalizability of the findings. Expanding the scale and diversity of scenarios is a crucial direction for future work. Second, the evaluation focused primarily on closed-source models from major commercial providers. To address these limitations, incorporating a broader spectrum of open-source models is suggested to provide more comprehensive insights and encourage reproducibility. Third, we relied solely on accuracy as the evaluation metric, resulting in the failure to capture partial correctness or the quality of alternative plausible plans. Future studies should adopt additional evaluation metrics such as precision, recall, and F1 score to offer a more nuanced assessment of LLM planning performance. Addressing these limitations will contribute to a more thorough understanding of LLM capabilities and support the development of advanced AI systems capable of assisting or autonomously executing complex human planning tasks.

References

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. 2024. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Guillem Frances, Miquel Ramırez, Nir Lipovetzky, and Hector Geffner. 2017. Purely declarative action representations are overrated: Classical planning with simulators. In *Proc. IJCAI*, pages 4294–4301.

Hector Geffner and Blai Bonet. 2013. Classical planning: Full information and deterministic actions. In *A Concise Introduction to Models and Methods for Automated Planning*, pages 15–36. Springer.

Runquan Gui, Zhihai Wang, Jie Wang, Chi Ma, Huiling Zhen, Mingxuan Yuan, Jianye Hao, Defu Lian, Enhong Chen, and Feng Wu. 2025. Hypertree planning: Enhancing llm reasoning via hierarchical thinking. *arXiv preprint arXiv:2505.02322*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Howard Karloff. 2008. *Linear programming*. Springer Science & Business Media.

Jong-Bok Kim. 2024. *English and Korean in Contrast: A Linguistic Introduction*. John Wiley & Sons.

Vishal Pallagani, Bharath Muppasani, Keerthiram Murugesan, Francesca Rossi, Biplav Srivastava, Lior Horesh, Francesco Fabiano, and Andrea Loreggia. 2023. Understanding the capabilities of large language models for automated planning. *arXiv preprint arXiv:2305.16151*.

Hancheol Park, Gahgene Gweon, and Jeong Heo. 2016. Affix modification-based bilingual pivoting method for paraphrase extraction in agglutinative languages. In *2016 International Conference on Big Data and Smart Computing (BigComp)*, pages 199–206. IEEE.

Stuart J Russell and Peter Norvig. 2016. *Artificial intelligence: a modern approach*. pearson.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023a. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, 36:38975–38987.

Karthik Valmeekam, Sarath Sreedharan, Matthew Marquez, Alberto Olmo, and Subbarao Kambhampati. 2023b. On the planning abilities of large language models (a critical investigation with a proposed benchmark). *arXiv preprint arXiv:2302.06706*.

Kevin Wang, Junbo Li, Neel P Bhatt, Yihan Xi, Qiang Liu, Ufuk Topcu, and Zhangyang Wang. 2024. On the planning abilities of openai's o1 models: Feasibility, optimality, and generalizability. *arXiv preprint arXiv:2409.19924*.

Hui Wei, Zihao Zhang, Shenghua He, Tian Xia, Shijia Pan, and Fei Liu. 2025. Plangenllms: A modern survey of llm planning capabilities. *arXiv preprint arXiv:2502.11221*.

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. *arXiv preprint arXiv:2402.01622*.

Xuan Zhang, Yang Deng, Zifeng Ren, See-Kiong Ng, and Tat-Seng Chua. 2024. Ask-before-plan: Proactive language agents for real-world planning. *arXiv preprint arXiv:2406.12639*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

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. 2024. Natural plan: Benchmarking llms on natural language planning. *arXiv preprint arXiv:2406.04520*.

A Related Works

A.1 Classical Planning

Classical planning is a task of executing a sequence of actions to transition from an initial state to a goal state (Geffner and Bonet, 2013). This type of planning is addressed exclusively when the problem can be described in a declarative language such as PDDL (Planning Domain Definition Language), which decomposes states into variables (Frances et al., 2017; Russell and Norvig, 2016). Studies evaluating the capabilities of large language models (LLMs) based on classical planning have been conducted as follows.

Valmeekam et al. (2023b) sought to evaluate the abilities of LLMs using the BlocksWorld task, commonly considered a general planning problem. BlocksWorld involves stacking blocks to achieve specific goals, and this study established a benchmark based on this task to assess GPT-3 and BLOOM. The experimental results showed that only about 3% of the plans independently generated by LLMs were executable. Beyond this limitation, Valmeekam et al. (2023b) tested a Human-In-The-Loop approach, hypothesizing that while LLMs cannot independently produce correct plans, they could help humans by sharing insights gained during planning attempts. This approach led to a modest improvement in accuracy when humans utilized the insights provided by LLMs. The study significantly contributes to future research on evaluating the planning capabilities of LLMs by offering a novel benchmark. Additionally, it underscores the current limited independent planning capability of LLMs but highlights their potential value when collaborating with humans.

In addition, Valmeekam et al. (2023a) developed a benchmark named PlanBench to assess LLMs' reasoning capabilities in planning and handling changes. PlanBench comprises 26,250 tasks from both BlocksWorld, involving stacking blocks to reach specific goals, and Logistics, which focuses on transporting items to designated destinations. The benchmark enables comprehensive evaluation beyond simple plan generation, assessing whether LLMs can create optimal cost plans, accurately predict post-execution states, and adapt plans effectively to unexpected changes. Results from PlanBench revealed that even advanced LLMs such as GPT-4 and Instruct-GPT generally struggle with effectively generating simple plans and reasoning logically about changes. This research clearly outlines the current capabilities and limitations of LLMs in planning tasks, indicating the usefulness of PlanBench as a tool for future LLM evaluation and development.

Pallagani et al. (2023) aimed to examine the extent to which LLMs could be used for plan generation. They established a benchmark using six classical planning domains—Ferry, BlocksWorld, Miconic, Tower of Hanoi, Grippers, and Driverlog—which are representable in PDDL. Evaluations using T5, CodeT5, text-davinci, and code-davinci revealed that these models achieved low accuracy scores, averaging below 0.16, and struggled to effectively solve planning problems. Further comparative analysis between purely text-based models and code-oriented models indicated that models specifically trained for code generation performed better at solving planning tasks. The authors hypothesized that this was due to the similarity between PDDL and programming languages, as both share formal syntax and common concepts such as variables, functions, and control structures. Pallagani et al. (2023) significantly contributed by systematically comparing various models, clearly highlighting both the limitations and potentials of LLMs, and suggesting the future applicability of coding-trained LLMs in solving planning tasks.

The three studies conducted in 2023 explicitly demonstrated the low success rates of LLMs in classical planning tasks. However, these studies have the limitation of assessing the abilities of LLMs exclusively within classical planning, which involves theoretical and simplified planning environments.

A.2 Practical Planning

Practical planning extends classical planning to more realistic and applicable scenarios. While classical planning considers 'what to do' and 'in which order,' it does not incorporate time considerations, such as when to start an action and how long it should last (Russell and Norvig, 2016). In contrast, practical planning inherently requires timing information, substantially increasing the complexity of planning problems. Due to this complexity, research in practical planning has been relatively sparse compared to classical planning, gaining significant academic interest only after 2024.

Xie et al. (2024) proposed a novel benchmark called TravelPlanner to evaluate the capability of LLM-based agent systems in formulating complex plans in realistic environments. Focusing on the travel domain, TravelPlanner provides a sandbox

consisting of 1,225 user queries and six tools covering four million real-world data records. The benchmark aims to assess the accuracy of plans generated by LLMs in response to input queries under various constraints. Evaluations of GPT, Gemini, and Mixtral using this benchmark showed a planning success rate below 1%. These results highlight substantial limitations of current LLMs in achieving human-level performance for complex planning tasks. The study underscores the need for future research to focus on enhancing LLMs' capabilities to handle complex constraints and make multi-step decisions effectively in realistic settings.

Zheng et al. (2024) introduced the NATURAL PLAN benchmark to evaluate how effectively LLMs perform natural language-based planning. The benchmark includes three primary tasks—travel planning, meeting planning, and schedule management—and provides context using outputs from services such as Google Flights, Maps, and Calendar. Unlike other studies, this benchmark provides all necessary planning information entirely in natural language, thus eliminating the need for additional tool environments. Evaluations using NATURAL PLAN revealed that even the most advanced models, including GPT-4 and Gemini 1.5 Pro, achieved task-solving rates between 31.1% and 48.9%. These findings highlight intrinsic limitations in LLMs' abilities to handle realistic planning tasks expressed in natural language. The research emphasizes the complexity of natural language planning as a challenge for LLMs and calls for methodological developments in future research to address these limitations.

Zhang et al. (2024) proposed a new dataset named 'Ask-before-Plan' to assess the ability of LLM-based agents to handle unclear user instructions and formulate plans in realistic settings. The dataset comprises 2,000 travel planning scenarios, each featuring ambiguous or infeasible user requests, clarifying questions, and final travel plans. This distinguishes Ask-before-Plan from earlier datasets, which typically contained clear instructions. Evaluations using GPT-3.5 showed only 0.1% of generated plans ultimately satisfied the given constraints, highlighting significant limitations in current LLMs' capacity to handle unclear instructions and perform practical planning. These findings indicate specific areas requiring improvement for LLM-based agents to operate effectively in real-world environments and emphasize the need for future research to focus on enhancing practical applicability.

All the previously mentioned studies share a common feature in evaluating the planning capabilities of LLMs within practical contexts. From a practical standpoint, research focused on enabling LLMs to consider complex constraints, set long-term goals, and effectively handle dynamic real-world problems should be prioritized over classical planning studies. In this regard, recent research moving beyond theoretical, classical planning towards practical planning that acknowledges the complexities of real-world environments significantly contributes to complementing human decision-making processes, marking a meaningful advancement for future research directions.

B Datasets Construction Process

When constructing Multi-Plan, templates provided to users were originally in Korean to facilitate data collection. However, in this section, system prompts are presented in English to aid readers' understanding.

City Selection
Choose $N \in [3, 6]$ cities within one of the regions Europe, Asia, or the Americas. All N cities must lie on the same continent.

Duration Assignment
For each city, randomly assign a stay length $D \in [2, 7]$ days.

Constraints
– Add at least one date–city constraint (e.g. visiting a relative's home, attending a local soccer match), choosing scenarios that fit each city's cultural context (e.g. Songkran festival for Bangkok).
– You may include multiple date–city constraints, but each itinerary must admit exactly one valid solution.
– Specify which city-to-city routes have direct flights; assume any direct flight takes at most one day and does not count against the stay durations.
– Do *not* allow travel between cities without a direct flight, and ensure that any unnecessary direct-flight edges may also be included.
– All itineraries must respect these flight constraints when constructing the schedule.

Figure 2: Multi-Plan generation rules template

Plan Request (Q)
You plan to visit * cities in Europe/Asia/Americas over a period of * days. You will only take direct flights between cities. Travel between cities is possible within a single day, and flights do not affect the stay durations in each city.

The cities with direct flights are: * and *, * and *, ...

Using only these direct flights, plan a trip to visit * cities in * days.

Answer Plan (A)
- days: Stay in * for * days.
Day *: Fly from * to *.
- days: Stay in * for * days.
Day *: Fly from * to *.
- days: Stay in * for * days.
...

Figure 3: Multi-Plan generation format template

To construct the Multi-Plan dataset, explicit data generation rules were first established. Figure 2 presents the template specifying these Multi-Plan generation rules, and Figure 3 illustrates the template for the desired format of each data instance. We provided these two prompts to three human annotators, who subsequently created a total of 204 natural language Multi-Plan instances following the specified rules and format. Afterwards, the generated Korean Multi-Plan dataset was translated into English using the DeepL translator, followed by additional human review.

Math Structuring Prompt
Below is an example of a natural-language plan request converted into a mathematically structured question. Refer to the examples and convert the following natural-language Q into a mathematically structured Q using exactly the same format. Output only the converted question.

Example
Natural Q1: {example_tasks[0]}
Structured Q1: {example_solutions[0]}
Natural Q2: {example_tasks[1]}
Structured Q2: {example_solutions[1]}
Natural Q3: {example_tasks[2]}
Structured Q3: {example_solutions[2]}
Natural Q4: {example_tasks[3]}
Structured Q4: {example_solutions[3]}

Task: {task}
Structured Q:

Figure 4: Math structuring prompt template with four examples and task

The natural language-based Multi-Plan dataset generated through the aforementioned process was then provided to Claude-3.5-Sonnet for conversion into mathematical form. To clearly define the transformation format, we included four manually constructed examples along with the natural language queries targeted for conversion in a prompt (Figure 4), thereby creating a mathematically structured dataset. Subsequently, human reviewers directly inspected the structured mathematical data, finalizing the construction of the Multi-Plan dataset. Details about the human annotators who constructed and reviewed the data are provided in Table 5.

C Datasets Examples

Category	Human Annotator 1	Human Annotator 2	Human Annotator 3
Birth Year	Born in 1996	Born in 1998	Born in 1997
Major	Computational Linguistics	Computational Linguistics	Social Welfare
Education Level	Master's graduate	Enrolled in Master's program	Master's graduate
Primary Language	Korean	Korean	Korean

Table 5: Demographic and academic background of the three human annotators who participated in the evaluation. All annotators are native Korean speakers with at least a bachelor's degree. Annotators 1 and 2 majored in computational linguistics, while Annotator 3 majored in social welfare. Two annotators have completed a master's program, and one is currently enrolled.

English	**Korean**	**Math**
You plan to visit 5 cities in Europe for a total of 17 days. You will only take direct flights between cities. Travel between cities is possible within a day, and flights do not affect the itinerary for staying in each city. You plan to visit Athens for 6 days. Between the 15th and 16th, You've signed up for a historical tour of Athens. You'll be visiting Liverpool for 3 days. You'll be visiting Interlaken for 4 days. You'll spend 5 days in Saint-Germain. You'll watch a Saint-Germain soccer game on day 6 of your trip. You'll spend 3 days in Naples. You've made reservations for a famous pizzeria in Naples on day 11 of your trip. Here are the cities that have direct flights: Liverpool and Saint-Germain, Interlaken and Athens, Saint-Germain and Interlaken, Interlaken and Naples, Naples and Athens, Liverpool and Naples. Plan a trip to visit five cities in 17 days using direct flights.	총 17일 동안 유럽 5개 도시를 방문할 계획입니다. 도시 간을 이동할 때는 직항 항공편만 이용합니다. 도시 간 이동은 하루 안에 가능하며, 항공편은 각 도시에 머무르는 일정에 영향을 끼치지 않습니다. 6일 동안 아테네를 방문하려고 합니다. 15일과 16일 사이에 아테네 역사 투어를 신청해두었습니다. 3일간 리버풀을 방문할 것입니다. 4일간 인터라켄을 방문할 예정입니다. 5일 동안 생제르망에서 시간을 보낼 것입니다. 여행 6일차에 셍제르망팀의 축구 경기를 직관할 것입니다. 3일 동안 나폴리에서 시간을 보낼 것입니다. 여행 11일차에 나폴리에서 유명한 피자집을 예약해두었습니다. 직항편이 있는 도시는 다음과 같습니다: 리버풀과 생제르망, 인터라켄과 아테네, 생제르망과 인터라켄, 인터라켄과 나폴리, 나폴리와 아테네, 리버풀과 나폴리. 직항 항공편을 이용하여 17일 동안 다섯 도시를 방문하는 여행 계획을 세워보세요.	**Variable Definitions**: A: Athens, L: Liverpool, I: Interlaken, S: Saint-Germain, N: Naples; a_A: Arrival date in Athens, d_A: Departure date from Athens; a_L: Arrival date in Liverpool, d_L: Departure date from Liverpool; a_I: Arrival date in Interlaken, d_I: Departure date from Interlaken; a_S: Arrival date in Saint-Germain, d_S: Departure date from Saint-Germain; a_N: Arrival date in Naples, d_N: Departure date from Naples. **Constraints**: $d_A - a_A + 1 = 6$, $d_L - a_L + 1 = 3$, $d_I - a_I + 1 = 4$, $d_S - a_S + 1 = 5$, $d_N - a_N + 1 = 3$, $\max(d_A, d_L, d_I, d_S, d_N) - \min(a_A, a_L, a_I, a_S, a_N) + 1 = 17$, $[15, 16] \cap [a_A, d_A] \neq \emptyset$, $a_S \leq 6 \leq d_S$, $a_N \leq 11 \leq d_N$. **Direct Flights**: $(L \leftrightarrow S)$, $(I \leftrightarrow A)$, $(S \leftrightarrow I)$, $(I \leftrightarrow N)$, $(N \leftrightarrow A)$, $(L \leftrightarrow N)$; travel between cities takes at most one day and does not affect stay durations. **Task**: Plan a 17-day itinerary visiting A, L, I, S, and N exactly once using only the above direct flights while satisfying all constraints.
Solution Plan (English): Days 1-3: Visit Liverpool for 3 days.;Day 3: Fly from Liverpool to Saint-Germain.;Days 3-7: Visit Saint-Germain for 5 days.;Day 7: Fly from St. Germain to Interlaken. ;Days 7-10: Visit Interlaken for 4 days.;Day 10: Fly from Interlaken to Naples.;Days 10-12: Visit Naples for 3 days.;Day 12: Fly from Naples to Athens.;Days 12-17: Visit Athens for 6 days.		
Solution Plan (Korean): 1-3일차: 3일간 리버풀 방문.; 3일차: 리버풀에서 생제르망으로 비행.;3-7일차: 5일간 생제르망 방문. 7일차: 생제르망에서 인터라켄으로 비행. ;7-10일차: 4일간 인터라켄 방문.;10일차: 인터라켄에서 나폴리로 비행.;10-12일차: 3일간 나폴리 방문.;12일차: 나폴리에서 아테네로 비행.;12-17일차: 6일간 아테네 방문.		

Table 6: Example Multi-Plan evaluation for a European five-city itinerary

English	Korean	Math
You plan to visit 5 cities in Asia for a total of 19 days. You will only take direct flights between cities. Travel between cities is possible within a day, and flights do not affect the itinerary for staying in each city. You will be in Danang for 3 days for a vacation. You have booked a 5-star hotel to stay in Danang between the 12th and 14th. You will be in Bali for 5 days. You have a conference in Bali between the 5th and 7th. You will be in Bangkok for 6 days. You will be visiting Manila for 4 days. You will be spending 5 days in Singapore. Here are the cities that have direct flights: Danang and Manila, Bangkok and Danang, Bangkok and Bali, Bali and Singapore, Manila and Singapore, Manila and Bali. Plan a trip to visit five cities in 19 days using direct flights.	총 19일 동안 아시아 5개 도시를 방문할 계획입니다. 도시 간을 이동할 때는 직항 항공편만 이용합니다. 도시 간 이동은 하루 안에 가능하며, 항공편은 각 도시에 머무르는 일정에 영향을 끼치지 않습니다. 3일간 다낭에서 휴양을 즐길 것입니다. 12일과 14일 사이에 다낭에서 머무를 5성급 호텔을 예약하였습니다. 5일간 발리에 있을 것입니다. 5일과 7일 사이에 발리에서 학회가 있습니다. 6일간 방콕을 방문할 예정입니다. 4일간 마닐라를 방문할 것입니다. 싱가포르에서 5일을 머무를 것입니다. 직항편이 있는 도시는 다음과 같습니다: 다낭과 마닐라, 방콕과 다낭, 방콕과 발리, 발리와 싱가포르, 마닐라와 싱가포르, 마닐라와 발리. 직항 항공편을 이용하여 19일 동안 다섯 도시를 방문하는 여행 계획을 세워보세요.	**Variable Definitions:** D: Danang, B: Bali, T: Bangkok, M: Manila, S: Singapore; a_D: arrival date in Danang, d_D: departure date from Danang; a_B: arrival date in Bali, d_B: departure date from Bali; a_T: arrival date in Bangkok, d_T: departure date from Bangkok; a_M: arrival date in Manila, d_M: departure date from Manila; a_S: arrival date in Singapore, d_S: departure date from Singapore. **Constraints:** d_D - a_D + 1 = 3 (stay in Danang for 3 days) d_B - a_B + 1 = 5 (stay in Bali for 5 days) d_T - a_T + 1 = 6 (stay in Bangkok for 6 days) d_M - a_M + 1 = 4 (stay in Manila for 4 days) d_S - a_S + 1 = 5 (stay in Singapore for 5 days) **Overall Trip Length:** $\max(d\_D, d\_B, d\_T, d\_M, d\_S) - \min(a\_D, a\_B, a\_T, a\_M, a\_S) + 1 = 19$ **Specific Date Constraints:** $[12, 14] \cap [a\_D, d\_D] \neq \emptyset$ (hotel booking window in Danang) $[5, 7] \cap [a\_B, d\_B] \neq \emptyset$ (conference window in Bali) **Direct Flights Allowed:** (D ↔ M), (T ↔ D), (T ↔ B), (B ↔ S), (M ↔ S), (M ↔ B); travel between cities takes at most one day and does not affect stay durations. **Task:** Plan a 19-day itinerary visiting D, B, T, M, and S exactly once using only the above direct flights while satisfying all constraints.
Solution Plan (English): Days 1–5: Visit Singapore for 5 days. Day 5: Fly from Singapore to Bali. Days 5–9: Visit Bali for 5 days. Day 9: Fly from Bali to Manila. Days 9–12: Visit Manila for 4 days. Day 12: Fly from Manila to Danang. Days 12–14: Visit Danang for 3 days. Day 14: Fly from Danang to Bangkok. Days 14–19: Visit Bangkok for 6 days.		
Solution Plan (Korean): 1–5일차: 5일간 싱가포르 방문. 5일차: 싱가포르에서 발리로 비행. 5–9일차: 5일간 발리 방문. 9일차: 발리에서 마닐라로 비행. 9–12일차: 4일간 마닐라 방문. 12일차: 마닐라에서 다낭으로 비행. 12–14일차: 3일간 다낭 방문. 14일차: 다낭에서 방콕으로 비행. 14–19일차: 6일간 방콕 방문.		

Table 7: Multi-Plan evaluation for an Asian five-city itinerary

English	Korean	Math
You plan to visit 6 cities in the Americas for a total of 24 days. You will only take direct flights between cities. Travel between cities is possible within a day, and flights do not affect the itinerary for staying in each city. You want to spend 7 days in Toronto. You will stay at your family home in Toronto starting on day 18. You will visit St. George's for 6 days, meeting friends there on the 10th and staying until the 15th. You will stay in Castries for 5 days, with dates between the 4th and 6th. You will spend 4 days in Fort-de-France. You will visit Bridgetown for 3 days. You will travel from Miami for 4 days. Here are the direct flights available: Miami ↔ St. George's, Toronto ↔ Miami, Bridgetown ↔ Toronto, Bridgetown ↔ St. George's, Castries ↔ Bridgetown, Castries ↔ Miami, Fort-de-France ↔ Bridgetown, Fort-de-France ↔ Castries. Plan a trip to visit six cities in 24 days using only direct flights.	총 24일 동안 아메리카 6개 도시를 방문할 계획입니다. 도시 간을 이동할 때는 직항 항공편만 이용합니다. 도시 간 이동은 하루 안에 가능하며, 항공편은 각 도시에 머무르는 일정에 영향을 끼치지 않습니다. 토론토에서 7일을 머물고 싶습니다. 토론토에 있는 본가에서는 18일부터 머물 예정입니다. 6일간 세인트조지스를 방문할 예정이며, 10일에 친구들을 만나 15일까지 머물기로 했습니다. 5일 동안 캐스트리스에서 머물 예정이며, 4일과 6일 사이에 있어야 합니다. 4일간 포르드프랑스에서 시간을 보낼 것입니다. 브리지타운을 3일 동안 방문할 것입니다. 마이애미에서 4일간 여행을 할 것입니다. 직항편이 있는 도시는 다음과 같습니다: 마이애미와 세인트조지스, 토론토와 마이애미, 브리지타운과 토론토, 브리지타운과 세인트조지스, 캐스트리스와 브리지타운, 캐스트리스와 마이애미, 포르드프랑스와 브리지타운, 포르드프랑스와 캐스트리스. 직항 항공편만 이용하여 24일 동안 여섯 도시를 방문하는 여행 계획을 세워보세요.	**Variable Definitions:** T: Toronto, S: St. George's, C: Castries, F: Fort-de-France, B: Bridgetown, M: Miami; a_T: arrival date in Toronto, d_T: departure date from Toronto; a_S: arrival date in St. George's, d_S: departure date from St. George's; a_C: arrival date in Castries, d_C: departure date from Castries; a_F: arrival date in Fort-de-France, d_F: departure date from Fort-de-France; a_B: arrival date in Bridgetown, d_B: departure date from Bridgetown; a_M: arrival date in Miami, d_M: departure date from Miami. **Constraints:** d_T - a_T + 1 = 7 (stay in Toronto for 7 days) d_S - a_S + 1 = 6 (stay in St. George's for 6 days) d_C - a_C + 1 = 5 (stay in Castries for 5 days) d_F - a_F + 1 = 4 (stay in Fort-de-France for 4 days) d_B - a_B + 1 = 3 (stay in Bridgetown for 3 days) d_M - a_M + 1 = 4 (stay in Miami for 4 days) **Overall Trip Length:** $\max(d\_T, d\_S, d\_C, d\_F, d\_B, d\_M) - \min(a\_T, a\_S, a\_C, a\_F, a\_B, a\_M) + 1 = 24$ **Specific Date Constraints:** $18 \leq a\_T \leq d\_T$ (home stay window in Toronto) $10 \leq a\_S \leq 15 \leq d\_S$ (friend meeting window in St. George's) $[4, 6] \cap [a\_C, d\_C] \neq \emptyset$ (dates in Castries) **Direct Flights Allowed:** (M ↔ S), (T ↔ M), (B ↔ T), (B ↔ S), (C ↔ B), (C ↔ M), (F ↔ B), (F ↔ C); travel between cities takes at most one day and does not affect stay durations.
Solution Plan (English): Days 1–4: Visit Fort-de-France for 4 days. Day 4: Fly from Fort-de-France to Castries. Days 4–8: Visit Castries for 5 days. Day 8: Fly from Castries to Bridgetown. Days 8–10: Visit Bridgetown for 3 days. Day 10: Fly from Bridgetown to St. George's. Days 10–15: Visit St. George's for 6 days. Day 15: Fly from St. George's to Miami. Days 15–18: Visit Miami for 4 days. Day 18: Fly from Miami to Toronto. Days 18–24: Visit Toronto for 7 days.		
Solution Plan (Korean): 1–4일차: 4일간 포르드프랑스 방문. 4일차: 포르드프랑스에서 캐스트리스로 비행. 4–8일차: 5일간 캐스트리스 방문. 8일차: 캐스트리스에서 브리지타운으로 비행. 8–10일차: 3일간 브리지타운 방문. 10일차: 브리지타운에서 세인트조지스로 비행. 10–15일차: 6일간 세인트조지스 방문. 15일차: 세인트조지스에서 마이애미로 비행. 15–18일차: 4일간 마이애미 방문. 18일차: 마이애미에서 토론토로 비행. 18–24일차: 7일간 토론토 방문.		

Table 8: Multi-Plan evaluation for an americas six-city itinerary

D Datasets Analysis

In this section, we analyze the Multi-Plan benchmark dataset from various perspectives to demonstrate that it is a high-quality dataset.

D.1 Query Analysis

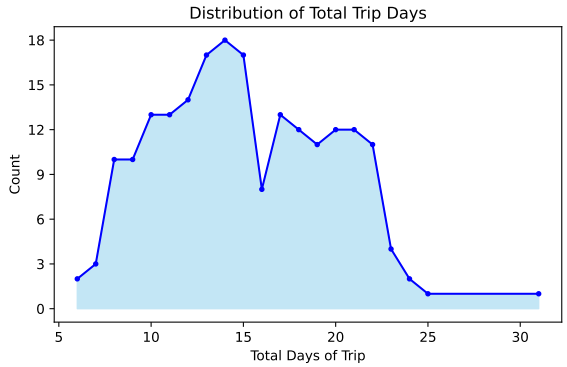

Figure 5: Distribution of total trip days among all travel plans.

Distribution of Travel Days Figure 5 shows the distribution of the total number of travel days across all travel planning queries in the Multi-Plan dataset. The travel plans range from 6 to 31 days, with the highest proportion being queries for 15-day trips, consisting of 18 instances.

Category	Value
Total number of questions	204
Average travel duration	15.1 days
Minimum travel duration	6 days
Maximum travel duration	31 days
Range of number of cities visited	3–6
Number of questions per city count	51 each
Number of questions per continent	Europe (68), Asia (68), Americas (68)

Table 9: Statistics of the Multi-Plan question Set

Table 9 summarizes key statistics of the Multi-Plan query set. The average number of travel days across the 204 queries is 15.1, with the range spanning from 6 to 31 days. Queries evenly represent trips visiting between 3 and 6 cities, with

City	Frequency
Prague	8
Amsterdam	8
Havana	8
La Paz	8
Bucharest	7
Budapest	7
Copenhagen	7
Ljubljana	7
Singapore	7
Santiago	7
Quito	7
Buenos Aires	7
Montevideo	7
Vienna	6
Paris	6
...	...
Total unique cities	**410**

Table 10: Top 15 most frequently mentioned cities and the total number of unique cities.

51 queries for each city count. Additionally, the dataset covers three continents—Europe, Asia, and America—with 68 queries per continent.

Table 10 presents the 15 most frequently mentioned cities and the total number of unique cities (410) in the dataset. Prague, Amsterdam, Havana, and La Paz appeared most frequently, each being mentioned 8 times, while Bucharest, Singapore, and Buenos Aires were mentioned consistently between 6 to 7 times. The extensive variety of 410 unique cities highlights broad geographic diversity without excessive focus on specific hub cities.

This distribution demonstrates that the Multi-Plan dataset is systematically designed, incorporating balanced considerations of travel duration, the number of visited cities, and geographic representation, thus providing a realistic experimental environment where models can learn diverse destinations without bias.

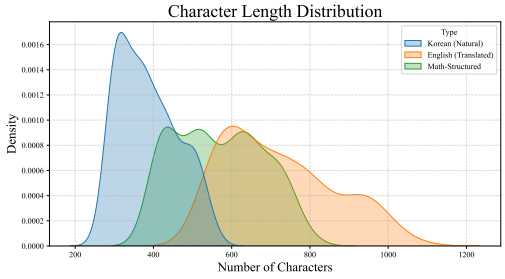

Figure 6: Distribution of question character lengths across three types: Korean (Natural), Math-Structured, and English (Translated).

Question Length Figure 6 analyzes the distribution of question lengths in the Multi-Plan dataset. Natural language-based questions (Korean (Natural)) and mathematically structured questions (Math-Structured) generally exhibited longer character lengths compared to the English-translated questions (English (Translated)).

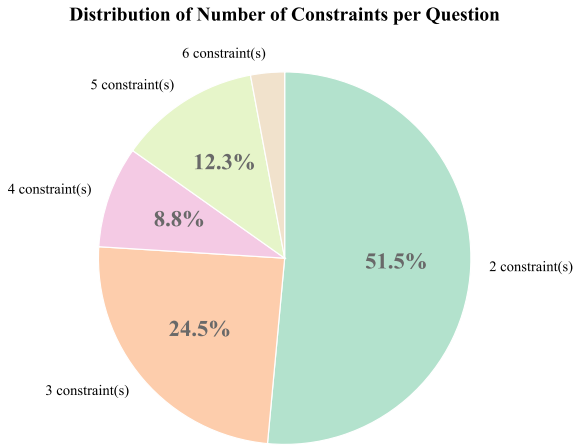

Figure 7: Distribution of the number of constraints per question in the Multi-Plan dataset.

D.2 Question Difficulty Analysis

Figure 7 illustrates the number of constraints included in each query within the Multi-Plan dataset. The constraints are categorized into two types: (1) availability of direct flights only and (2) requirement to stay in a specific city on a specific date. Each query was designed to include at least two constraints—one direct flight constraint and one date-specific stay constraint. Approximately 51.5% of queries have exactly two constraints, around 24.5% include three constraints, and the remaining approximately 24% include four to six constraints. This distribution indicates that Multi-Plan incorporates questions of varying difficulty levels in a balanced manner.

E Details of Model

E.1 Closed Models

In this study, we focused our experiments primarily on closed models developed by three organizations (OpenAI, Anthropic, and Google Deepmind). Detailed information about the closed models used in our experiments can be found in Table 11.

E.2 Open Models

E.2.1 Non-Reasoning Models

OLMo-2-7B-Instruct AllenAI released the Open Language Models (OLMo) series to advance language model science, covering model sizes from 1B to 32B parameters. OLMo-2-7B-Instruct is a 7B parameter model pretrained on 4T tokens and further refined through Supervised Fine-Tuning (SFT), Direct Preference Optimization (DPO), and Reinforcement Learning with Verifiable Rewards (RLVR).

SmolLM2-1.7B-Instruction HuggingFaceTB introduced this model trained on 11T tokens from datasets such as FineWeb-Edu, DCLM, and The Stack. It underwent SFT using curated mathematical and coding datasets combined with publicly available data, followed by Direct Preference Optimization (DPO) utilizing UltraFeedback.

Llama3.1-8B-Instruction/Llama3.3-70B-Instruction Released by Meta, these multilingual language models trained on over 15T tokens from publicly available online data. The Llama3.1 model underwent SFT and Reinforcement Learning with Human Feedback (RLHF).

Qwen2.5-72B-Instruction Alibaba's Qwen2.5 Instruction model supports over 29 languages and was pretrained on approximately 18T tokens. It was further enhanced through SFT, Reinforcement Learning (RL), and Long Context Fine-tuning.

E.2.2 Reasoning Models

Llama-3.1-Nemotron-Nano-8B Developed by Nvidia, based on Meta's Llama-3.1-8B-Instruct, this reasoning model is optimized for tasks such as enhanced reasoning, conversational preference learning, RAG, and tool calling. It underwent Supervised Fine-Tuning (SFT) and reinforcement learning methods including RLOO and Online Reward-aware Preference Optimization (RPO).

Phi-4-reasoning-plus Microsoft's approximately 14.7B parameter Phi-4 derived model underwent supervised fine-tuning using Chain-of-Thought (CoT) reasoning examples and specialized synthetic prompts for mathematics, science, and coding, alongside filtered high-quality public data. It subsequently incorporated reinforcement learning (RL) for enhanced safety and Responsible AI alignment.

Developer	Model Name	Detailed Model Name	Release Year	Context Length	Remarks
OpenAI	o3-mini	o3-mini-2025-01-31	2025	200K	Reasoning Model
	GPT-4o	gpt-4o-2024-08-06	2024	128K	General Model
	GPT-4-turbo	gpt-4-turbo-2024-04-09	2023	128K	General Model
	GPT-4	gpt-4-0613	2023	8K	General Model
Anthropic	Claude-3.7-Sonnet thinking	claude-3-7-sonnet-20250219	2025	200K	Reasoning Model
	Claude-3.7-Sonnet standard	claude-3-7-sonnet-20250219	2025	200K	General Model
	Claude-3.5-Haiku	claude-3-5-haiku-20241022	2024	200K	General Model
	Claude-3-Opus	claude-3-opus-20240229	2024	200K	General Model
Google Deepmind	Gemini-2.5-Pro	gemini-2.5-pro-preview-03-25	2025	1M	Reasoning Model
	Gemini-2.0-Flash	gemini-2.0-flash	2024	1M	General Model
	Gemini-1.5-Pro	gemini-1.5-pro-exp-0827	2024	2M	General Model

Table 11: Information on closed models used in experiments. Each row provides metadata for a closed-source language model used in the evaluation. **Developer** indicates the organization that released the model. **Model Name** is the abbreviated or commonly used name of the model. **Detailed Model Name** refers to the official version name or identifier. **Release Year** is the year the model was publicly released or last updated. **Context Length** denotes the maximum number of tokens the model can process in a single input. **Remarks** describes the model's general purpose, distinguishing between reasoning-oriented models and general-purpose models.

DeepSeek-R1-Distill-Llama-8B/DeepSeek-R1-Distill-Qwen-7B These models were fine-tuned from Llama-3.1-8B and Qwen2.5-Math-7B respectively using samples generated by DeepSeek-R1. The fine-tuning employed CoT reasoning and domain-specific SFT data for high-quality knowledge distillation.

Qwen3-8B, 32B Alibaba's Qwen3 series supports both reasoning and non-reasoning modes, handling 119 languages and pretrained on approximately 36T tokens. It underwent optimization through processes including long chain-of-thought (CoT) cold start, reasoning-based reinforcement learning (RL), thinking mode fusion, and general RL.

F Details of Experiment

F.1 Details of Evaluation Method

To evaluate the structural consistency between the model-generated outputs and the ground truth, we designed a regular-expression-based comparison method. Since each plan follows a consistent sentence pattern, we utilized two regular expressions to parse them into structured events:

```
# visit
Days? (\d+)(?:-(\d+))?:\s*Visit
    ↪ (\w+)\s*for (\d+)\s*days?\.

# fly
Day (\d+):\s*Fly from (\w+)\s*to
    ↪ (\w+)\.
```

After parsing each text into `visit` and `fly` events using the aforementioned regular expressions, we compared the sequences' order, count, and content with the ground truth. Any discrepancy resulted in marking the instance as incorrect, and the overall accuracy was calculated as the ratio of correctly classified instances.

For example, if both the model prediction and the ground truth sentences are "Days 1–3: Visit Paris for 3 days. Day 4: Fly from Paris to Rome," the first sentence is recognized as a "visit" event and parsed into `type='visit', start='1', end='3', location='Paris', duration='3'`, while the second sentence is identified as a "fly" event and parsed into `type='fly', day='4', from='Paris', to='Rome'`.

The instance is marked as `correct` if the event sequence (`visit → fly`), event count, and each event's field values match exactly with the ground truth. Conversely, if the destination in the second event changes from `Rome` to `Milan`, resulting in `type='fly', day='4', from='Paris', to='Milan'`, the instance would be classified as `wrong` due to the differing field values.

F.2 Impact of Few-Shot Examples

The example impact evaluation utilized only the Korean Multi-Plan dataset to investigate how increasing the number of examples in the prompt could enhance the planning performance of LLMs. Few-Shot Learning (FSL) refers to the ability to learn and perform new tasks through simple text interactions without fine-tuning (Brown et al., 2020). Typically, FSL is achieved via in-context learning, and it has already been established that the model's performance improves as the number of provided examples increases (Agarwal et al., 2024). Based

Model	Reasoning	1-shot	3-shot	5-shot
o3-mini	✓	82.84%	82.84%	84.80%
GPT-4o	✗	3.43%	2.94%	2.94%
GPT-4-turbo	✗	10.78%	20.59%	25.00%
GPT-4	✗	15.20%	21.57%	28.92%
Claude-3.7-Sonnet thinking	✓	46.57%	50.98%	52.94%
Claude-3.7-Sonnet standard	✗	29.41%	44.12%	45.10%
Claude-3.5-Haiku	✗	8.33%	10.78%	21.57%
Claude-3-Opus	✗	27.45%	36.27%	32.84%
Gemini-2.5-Pro	✓	82.35%	85.78%	86.76%
Gemini-2.0-Flash	✗	11.76%	18.14%	25.49%
Gemini-1.5-Pro	✗	6.37%	15.20%	18.14%

Table 12: Experimental results by number of few-shot examples. Each column shows the model's accuracy under different few-shot settings.'1-shot', '3-shot', and '5-shot' indicate the number of examples provided in the prompt before the test query. Higher shot counts generally offer more context, potentially improving performance.

on these prior findings, we hypothesized that increasing the number of examples would lead to improved planning capabilities. To verify this hypothesis, we systematically analyzed changes in planning accuracy across different models by applying 1-shot, 3-shot, and 5-shot settings within the prompts.

The experimental results showed a positive correlation between the number of examples provided and the accuracy of plan generation for most models. Except for Claude-3-Opus and GPT-4o, all models consistently improved their performance with increasing examples, indicating that in-context learning can significantly enhance performance even in complex tasks such as planning. Detailed planning accuracy results for the 1-shot, 3-shot, and 5-shot conditions are provided in Table 12.

Claude-3-Opus displayed an exceptional pattern, showing significant performance improvement when moving from 1-shot to 3-shot, but experiencing a decrease when moving from 3-shot to 5-shot. However, since the performance for both 3-shot and 5-shot remained higher than 1-shot, the positive impact of examples on performance cannot be entirely disregarded, thereby still supporting the effectiveness of in-context learning. Conversely, GPT-4o unexpectedly showed a decrease in performance with increased examples, achieving a notably low average accuracy of 2–3% across all settings, the lowest among the 11 models tested. These results suggest that GPT-4o may have difficulty properly interpreting the planning requests, raising questions about its suitability for example impact assessments.

One of the models demonstrating the most prominent improvement through example-based in-context learning was Claude-3.7-Sonnet standard, which showed a substantial accuracy increase of 15.69% from the 1-shot to 5-shot environments. Similarly, other general-purpose models like GPT-4-turbo, Gemini-2.0-Flash, and GPT-4 showed significant accuracy improvements exceeding 10% as the number of examples increased. Notably, while general-purpose LLMs exhibited marked sensitivity to example count increases, reasoning-specialized models displayed relatively limited responsiveness. This observation might indicate that reasoning-focused models either quickly grasped the essential requirements with fewer examples or initially possessed high task-solving capabilities, limiting additional gains from extra examples compared to general models.

In conclusion, although the extent of performance improvement varied with the increase in examples, this experiment clearly demonstrates that few-shot learning is an effective strategy to enhance the planning capabilities of LLMs. Despite the complexity of planning tasks requiring advanced comprehension and reasoning, meaningful improvements can be achieved simply by providing relevant examples in prompts without structural model changes or fine-tuning, highlighting the robust potential of in-context learning. These findings reaffirm the effectiveness of in-context learning beyond simple pattern recognition, underscoring its applicability in complex cognitive tasks. Furthermore, these experimental results provide critical insights for efficient prompt design and model utilization strategies, contributing broadly to the advancement of AI research related to enhancing LLM capabilities.