
A Human-Like Reasoning Framework for Multi-Phases Planning Task with Large Language Models

Chengxing Xie¹ Difan Zou¹

Abstract

Recent studies have shown that LLM agents excel in simple tasks like writing and coding using various reasoning strategies. However, they struggle with tasks requiring comprehensive planning, a critical research issue. This study focuses on travel planning, a Multi-Phase problem involving outlining, information gathering, and planning, which requires managing various constraints and uncertainties. Existing approaches have struggled with this complexity (Xie et al., 2024b). Our research aims to address this challenge by developing a human-like planning framework for LLM agents, guiding them through the steps humans take to solve Multi-Phase problems. We implement strategies enabling LLM agents to generate coherent outlines for travel queries, mirroring human planning patterns. The framework integrates a Strategy Block for information collection and a Knowledge Block for detailed planning. Extensive experiments demonstrate that our framework significantly enhances the planning capabilities of LLM agents, improving efficiency and effectiveness in travel planning tasks. Our experimental results show exceptional performance, with GPT-4-Turbo achieving 10× the performance gains compared to the baseline framework.

1. Introduction

Recently, large language models (LLMs) like GPTs (Achiam et al., 2023; Ouyang et al., 2022) and LLaMAs (Touvron et al., 2023) have shown remarkable potential across various domains, demonstrating impressive generalization capabilities. Studies (Wang et al., 2023b; Mukobi et al., 2023; , FAIR) have shown LLM agents can play card

¹Department of Computer Science, The University of Hong Kong. Correspondence to: Chengxing Xie <xiechengxing34@gmail.com>, Difan Zou <dzou@cs.hku.hk>.

games with human-like proficiency, sometimes outperforming human players. Moreover, research (Zhang et al., 2023; Dasgupta et al., 2023; Fu et al., 2024) revealed LLM agents can handle daily tasks such as cooking and door-opening, and establish complex software systems through collaborative roles (Qian et al., 2023; Huang et al., 2023a).

Nevertheless, LLM agents still struggle with certain tasks (Wang et al., 2023a; Wu et al., 2023b; Jimenez et al., 2023), which are relatively solvable to humans. We categorize these challenges as **Multi-Phases Planning Tasks**. These tasks are difficult because their solutions can be divided into multiple interconnected phases, each requiring shared information and having dependencies with each other.

We categorize the stages of solving such tasks into three main phases, detailed as follows: **Outline Generation Phase**: When confronted with a Multi-Phase Problem, individuals typically begin by sketching a preliminary outline outlining the path toward a solution. **Information Collection Phase**: Generally, to solve this type of task, individuals must proactively identify the required information and ascertain how to obtain it effectively. **Plan Making Phase**: Armed with the necessary information, individuals proceed to formulate a plan. Given the challenges posed by Multi-Phase Problems, we have chosen **travel planning** as our focal point for several reasons. Firstly, it is a time-consuming and often challenging problem for many people. Secondly, travel planning inherently involves a multitude of constraints, requires long-term strategizing, and demands significant travel-related information gathering.

Previous reasoning strategies (Yao et al., 2022; Wei et al., 2022; Shinn et al., 2024) have shown limited effectiveness in addressing the complexities of travel planning (Xie et al., 2024b). However, by understanding and emulating the processes humans use to tackle Multi-Phases Problems, we can develop more robust planning frameworks to enable LLM agents to tackle such challenges effectively. Our proposed framework aims to capture the essence of human reasoning processes, particularly in the context of Multi-Phase Problems, to enhance the performance of LLM agents.

Our framework consists of three main components. First, **Outline Generation**: we generate a travel planning out-

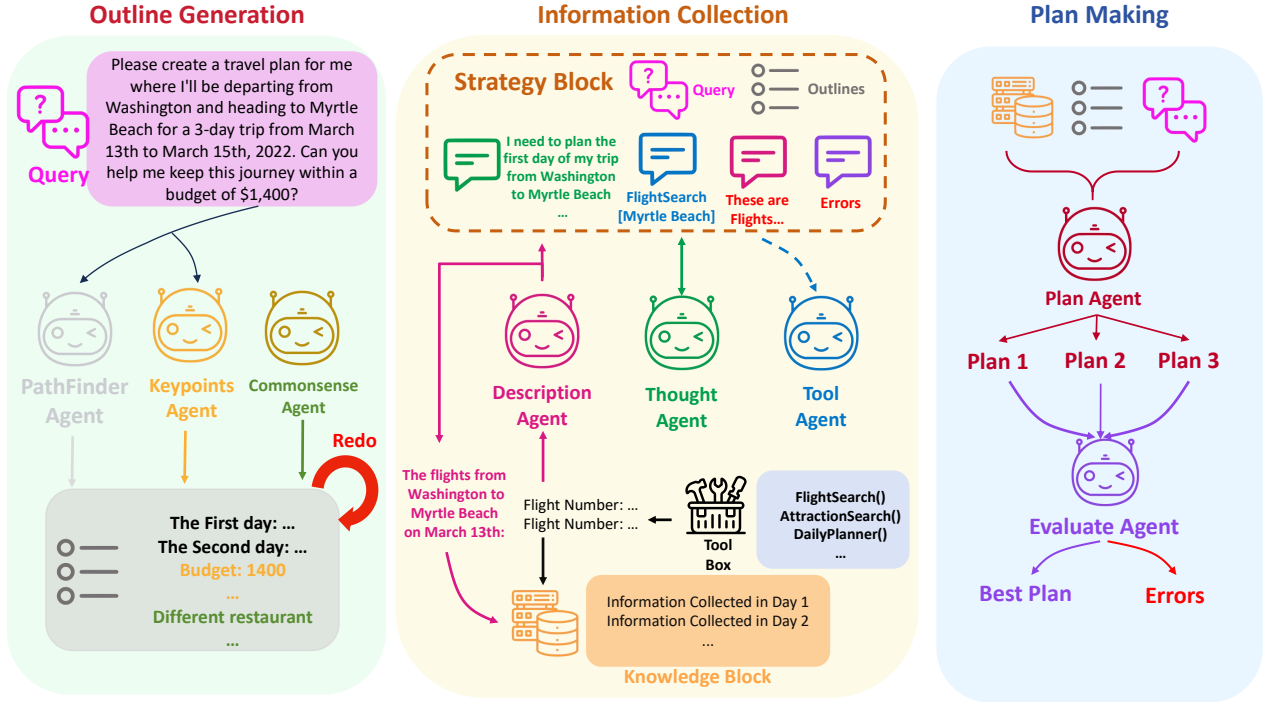


Figure 1. Our human-like planning framework has three key parts: **Outline Generation**, where LLM agents produce rough plans and identify key information; **Information Collection**, where agents gather essential data for comprehensive planning; and **Plan Making**, where agents explore potential plans and return a well-structured, reasonable plan.

line using multi-agent collaboration, establishing a foundational guide. Second, **Information Collection**: LLM agents gather relevant travel data to ensure they have the necessary details for making informed decisions. Third, **Plan Making**: after collecting sufficient information, the LLM agent creates detailed daily plans using a plan search and evaluation method. Multiple plans are generated, and an evaluation agent selects the best plan and flags errors. This structured approach significantly improves LLM agents’ ability to handle complex travel planning tasks. Our major findings are summarized as follows:

- Our work underscores the challenges LLM agents face with Multi-Phase Planning tasks, highlighting the necessity of employing a novel reasoning framework for effective problem-solving.
- We found that utilizing a human reasoning framework for complex tasks is a key factor in enhancing LLM agents’ performance on these tasks.
- By identifying key factors contributing to effective problem-solving and integrating them into a human-like planning framework, we enable LLM agents to exhibit human-like reasoning, resulting in impressive performance improvements.
- Through extensive experimentation, we validate the effectiveness of our approach, demonstrating its capability to tackle complex planning problems with remarkable success.

2. Method

2.1. Task Description:

We introduce the TravelPlanner task setting (Xie et al., 2024b), where LLM agents must create reasonable travel plans based on given queries, adhering to constraints (See Appendix A.9 for examples). An initial analysis of GPT-4-Turbo’s performance revealed struggles with efficient information collection and valid plan making. To address these issues, we employ the **Outline Generation** phase (Section 2.3) to create logical outlines, enhance the **Information Collection** phase (Section 2.4) to ensure accurate data gathering, and utilize the **Plan Making** phase (Section 2.5) to develop correct and comprehensive plans.

2.2. Framework Overview

Our framework, as shown in Fig 1, employs a human-like reasoning approach to address travel planning problems. Based on the specific query, we first generate an outline for the query (Section 2.3). Following the *Outline Generation*, LLM agents equipped with *Strategy block* and *Knowledge Block* proceed with *Information Collection* (Section 2.4). During this stage, once sufficient information is gathered for a single day’s plan, the agent creates the daily plan in *Plan Making* phase (Section 2.5). The final travel plan is composed of each daily plan.

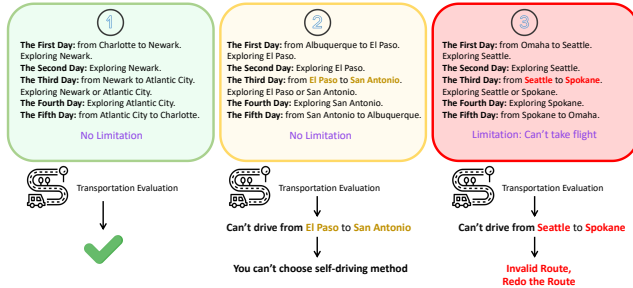


Figure 2. This is our transportation evaluation process for each route generated by the PathFinder Agent. We assess if the route is reasonable under the specific travel planning query’s constraints (e.g., no flights allowed). If unreasonable, we provide feedback and redo the route. If specific transportation methods are unreasonable, we add constraints accordingly.

2.3. Outline Generation

When humans tackle such tasks, they start with a rough outline, including the route, transportation options, and key points (see Appendix A.10). To replicate this, we introduce the Route Generation phase, which incorporates transportation evaluation and additional agents to generate key points, enhancing the outline’s usefulness.

Route Generation: We’ve developed the Route Generation component of Outline Generation. Firstly, the query is first passed to the **PathFinder Agent**. This agent generates a rough route for the entire trip, including city transfers and exploratory travel information. This preliminary route serves as a guide for the subsequent *Information Collection* and *Plan Making* phases, providing a clear roadmap for the journey and making the planning process more structured and transparent.

Transportation Evaluation: We’ve observed that LLM agents may generate invalid routes due to a lack of detailed transportation information between cities. To address this issue, we propose adding an evaluation stage after route generation. As shown in Fig 2, during this stage, each route is evaluated to determine its rationality from a transportation perspective.

Keypoints Generation: We introduced the **Keypoints Agent** (Fig 1 left) to identify the critical points in the query that need to be considered by the Plan Agent. Moreover, as (Xie et al., 2024b) demonstrated, LLM agents tend to overlook common sense when planning, such as navigating the same restaurant or attraction. To address this issue, the **Commonsense Agent** (Fig 1 left) generates basic guides related to travel planning.

2.4. Information Collection

After the **Outline Generation** phase, humans proceed to collect necessary travel-related information, such as specific attractions and restaurants. In our framework, the Infor-

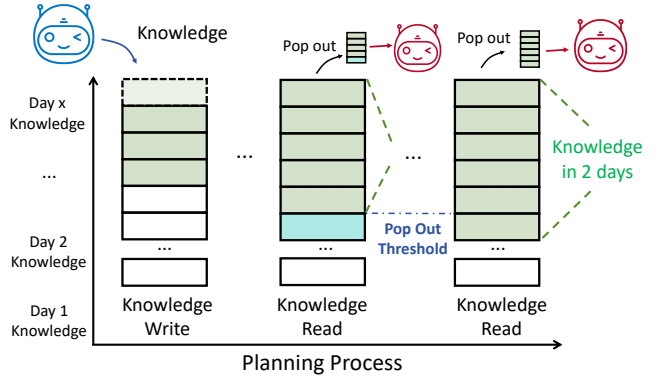


Figure 3. The left part is the **Knowledge Write** process, where a function’s result is written to the top of the Knowledge Block. About **Knowledge Read**, when the knowledge needs to be popped out but is below the threshold, the Knowledge Block pops out enough items to meet the threshold. If the number exceeds the threshold, the items in past two days are popped out.

mation Collection process begins with the **Thought Agent** generating the next steps based on the Strategy Block. Subsequently, the **Tool Agent** utilizes the output of the Thought Agent to generate a function expression. The result of this function is then recorded in the Knowledge Block with a description from the **Description Agent**.

Strategy Block: In the information collection process, humans typically remember the types of information they have gathered, which guides their subsequent steps. To emulate this procedure, we introduce a component called the **Strategy Block** (Fig 1), primarily used by the **Thought Agent**. First, the outline is stored in the Strategy Block to guide information collection. The Strategy Block also informs the Thought Agent which day it is in the travel plan and short descriptions of the collected data. Besides, we streamline the prompts for better workflow management.

Tool Agent: We’ve observed that as the planning process progresses, the context length can become too long, leading to potential oversight of tool documents placed at the beginning of input messages. Thus, we introduce the Tool Agent tasked with generating the correct function expression format messages.

Knowledge Block: We introduce the **Knowledge Block**, primarily used by the **Plan Agent**, to obtain detailed information (Fig 1). This block automatically records all information and descriptions generated by the **Description Agent** (Fig 3, left). For long-term travel plans, extracting pertinent details can be challenging. To address this, the block "pops out" information collected over the past two days, restricted by a minimal threshold (Fig 3).

2.5. Plan Making

When humans make plans, they typically do not plan everything at once; instead, they approach it step by step. There-

Table 1. Our experimental results on the validation dataset of the TravelPlanner benchmark demonstrate that our framework enables GPT-3.5 to surpass previous GPT-4’s performance.

Model	Delivery Rate		Commonsense Pass Rate		Hard Constraint Pass Rate		Final Pass Rate
	Micro	Macro	Micro	Macro	Micro	Macro	
Mistral-7B-32K (Jiang et al., 2023)	8.9	5.9	0	0	0	0	0
Mixtral-8x7B-MoE (Jiang et al., 2024)	49.4	30.0	0	1.2	0.6	0	0
GPT-3.5-Turbo (Ouyang et al., 2022)	86.7	54.0	0	0	0	0	0
GPT-4-Turbo (Achiam et al., 2023)	89.4	61.1	2.8	15.2	10.6	0.6	0.6
Our Results							
Mistral-7B-32K (Jiang et al., 2023)	39.4	24.0	1.1	0.5	0.6	0.6	0.6
Mixtral-8x7B-MoE (Jiang et al., 2024)	67.8	40.2	0	0	0	0	0
GPT-3.5-Turbo (Ouyang et al., 2022)	100.0	75.1	15.6	15.5	4.4	2.2	2.2
GPT-4-Turbo (Achiam et al., 2023)	91.7	74.6	24.4	35.7	16.7	6.7	6.7

fore, unlike the approach in (Xie et al., 2024b) that plans the entire trip at once, we adopt a daily planning strategy. This method requires less information at each step, making the planning process easier. Each time the Thought Agent determines that sufficient information has been collected for a specific day, the Tool Agent calls the *DailyPlanner* tool, and the Plan Agent creates the daily plan.

Plan Search: When examining the results of Plan Agent, we observe that creating a plan often introduces various errors. These errors can stem from insufficient information collected during the information collection stage or from oversights during the planning stage. To address this issue, we propose a plan search method. Each time the Plan Agent generates a daily plan, it creates several plans. An **Evaluate Agent** reviews each plan, converting them into JSON format and using code to identify and rank errors. Based on these evaluations, we select the best plan.

3. Experiment

3.1. Experiment Setup

We utilize the TravelPlanner benchmark proposed by (Xie et al., 2024b), which includes a variety of travel queries with different travel lengths and difficulty levels. We use gpt-3.5-turbo-1106, gpt-4-1106-preview, mixtral, mistral-7B-32K in our experiments on validation set (Achiam et al., 2023; Jiang et al., 2024; 2023; Ouyang et al., 2022). **Metrics:** We use the same metrics proposed in TravelPlanner (Xie et al., 2024b).

- **Delivery Rate:** Measures if agents can successfully deliver a final plan within a set number of steps (max 45). Failure includes dead loops, numerous failed attempts, or exceeding the step limit.
- **Commonsense Constraint Pass Rate:** Assesses if agents can incorporate commonsense into their plans without explicit instructions, across eight dimensions.
- **Hard Constraint Pass Rate:** Evaluates if a plan meets all explicitly given hard constraints, testing agents’ adaptability to diverse user queries.
- **Final Pass Rate:** Indicates the proportion of plans that

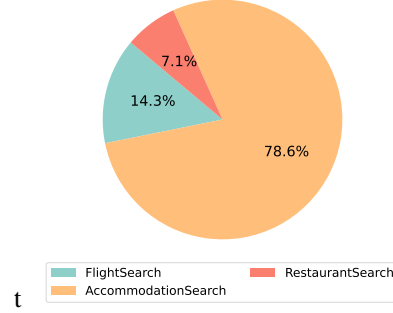


Figure 4. The error distribution about GPT-4-Turbo’s Delivery Rate Failure.

meet all constraints (delivery, commonsense, and hard constraints), reflecting agents’ overall proficiency in producing practical plans.

- **Micro Pass Rate:** The Micro Pass Rate evaluates the proportion of constraints that are successfully passed by an agent’s plans, as shown in Formula 2. It is calculated by taking the total number of successfully met constraints across all plans and dividing it by the total number of constraints applied to all plans.
- **Macro Pass Rate:** The Macro Pass Rate assesses the proportion of plans that satisfy all of their constraints. It calculates the ratio of the number of plans that meet all applicable commonsense or hard constraints to the total number of plans evaluated.

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

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

P is the set of all plans. C_p is the set of constraints applicable to a specific plan p . $1_{\text{passed}(c,p)}$ is an indicator function that returns 1 if constraint c is passed in plan p , and 0 otherwise. $1_{\text{passed}(C_p,p)}$ is an indicator function that returns 1 if all constraints in C_p are passed in plan p , and 0 otherwise.

3.2. Experiment Result

As shown in Table 1, we tested four LLMs on the dataset to verify our framework’s effectiveness. All models exhibited increased Delivery Rate, GPT-3.5-Turbo achieving 100% Delivery Rate. This indicates that GPT-3.5-Turbo can generate a travel plan for every query, strongly demonstrating that our framework enables better planning for the task.

However, we observed limited improvement for GPT-4-Turbo on Delivery Rate. To understand the reason, we visualized the error distribution from GPT-4-Turbo in Fig 4. We found that the primary issue was the repeated use of

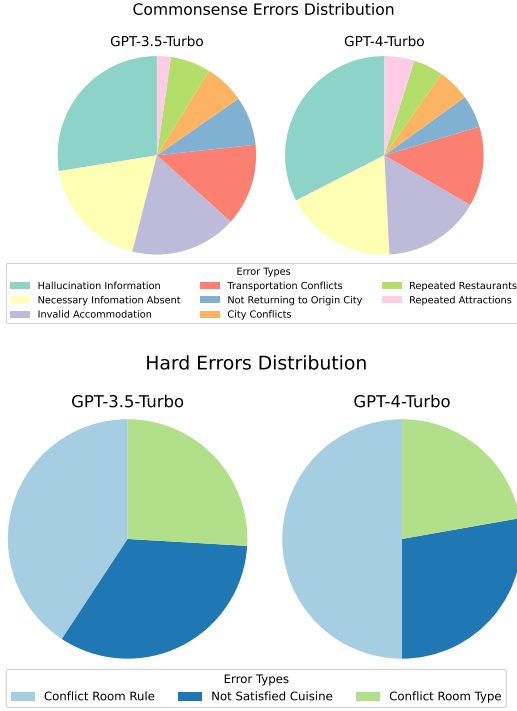


Figure 5. The error distribution of GPT-3.5-Turbo and GPT-4-Turbo.

the same function three times, with the most frequent being AccommodationSearch. The agent repeatedly attempted to find the 'correct' room type, often overlooking some information, which led to redundancy. Additionally, GPT-4-Turbo aimed to find accommodations that could fit all travelers in one room, ignoring the possibility of booking multiple rooms. The lower Delivery Rate will cause a lower Micro Commonsense pass rate. We conjecture that GPT-4-Turbo's stronger reasoning capabilities hindered its progress by overcomplicating the task.

Commonsense Pass Rate: Our framework demonstrates impressive improvements in the Commonsense Pass Rate. For GPT-3.5-Turbo, the Micro Pass Rate increased from 54.0% to 75.1%, and the Macro Pass Rate increased from 0% to 15.6%. For GPT-4-Turbo, the Micro Pass Rate increased from 61.1% to 74.6%, while the Macro Pass Rate saw a significant boost from 2.8% to 24.4%. Although Mistral and Mixtral showed improvements in their Micro Pass Rates, their Macro Pass Rates did not see significant increases.

Analyzing the commonsense error distribution in Fig 5, we find that GPT-3.5-Turbo and GPT-4-Turbo exhibit very similar error patterns. The top three errors are Hallucinated Information, Necessary Information Absent, and Invalid Accommodation, accounting for nearly 60% of all errors. Hallucinated Information indicates that the LLM agents still generate unreal information. Necessary Information Absent suggests that the Plan Agent might not strictly follow

Table 2. Ablation study results on GPT-3.5-Turbo.

	Delivery Rate	Commonsense Pass Rate		Hard Constraint Pass Rate		Final Pass Rate
		Micro	Macro	Micro	Macro	
Original Result (Xie et al., 2024b)	86.7	54.0	0	0	0	0
No Outlines	93.3	59.2	1.1	3.8	1.1	0
No Strategy	100	74.9	13.9	12.1	2.2	0.6
No Knowledge	99.4	66.0	8.3	10.2	1.1	0
No Plan Search	99.4	67.4	0.6	1.9	0	0
Our Framework	100.0	75.1	15.6	15.5	4.4	2.2

instructions or the information collection phase misses key details. Invalid Accommodation implies that some points in the accommodation data are overlooked.

Hard Constraints Pass Rate: Additionally, our framework enhances the Pass Rate for Hard Constraints. For GPT-4-Turbo, the Micro Pass Rate increased from 15.2% to 35.7%, and the Macro Pass Rate increased from 10.6% to 16.7%. For GPT-3.5-Turbo, the improvements were slightly lower, with increases from 0% to 15.5% for the Micro Pass Rate and from 0% to 4.4% for the Macro Pass Rate.

Regarding the Final Pass Rate, GPT-3.5-Turbo's result using our framework surpasses the GPT-4-Turbo's result in the previous algorithm. Notably, GPT-4-Turbo's Final Pass Rate increased 10× compared to the previous algorithm.

3.3. Ablation Study

We analyzed the impact of each component of our framework by removing them one at a time, as shown in Table 2. Removing the outline generation component resulted in a significant drop in all metrics, underscoring its importance. For the Strategy Block, we removed specific elements such as daytime guidance, Knowledge Block information, budget requirements, and key query points. Although the remaining components are necessary for the information collection process, we observed decreases in all metrics, particularly in plan quality rather than feasibility. The Knowledge Block's removal, managed through a maximum length limitation to avoid exceeding token limits, caused notable declines in all metrics except Delivery Rate, emphasizing its critical role. Similarly, removing the Plan Search component led to a dramatic drop in all metrics, especially the Macro Commonsense Pass Rate, highlighting its significance. Our ablation study results confirm that each block is essential for managing information and enhancing plan quality, ensuring a robust and efficient planning process.

4. Conclusion

In this paper, we presented a novel human-like reasoning framework for travel planning using LLM Agents. Our approach integrates several key components: Outline Generation, Information Collection with blocks, and Plan Search. These components synergistically mimic human problem-solving strategies, enabling LLM agents to handle multi-phase tasks more effectively. Extensive experiments on the TravelPlanner benchmark demonstrated significant improvements across multiple models with our framework.

References

- Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. [arXiv preprint arXiv:2303.08774](#), 2023.
- Aher, G. V., Arriaga, R. I., and Kalai, A. T. Using large language models to simulate multiple humans and replicate human subject studies. In [International Conference on Machine Learning](#), pp. 337–371. PMLR, 2023.
- Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., Gajda, J., Lehmann, T., Niewiadomski, H., Nyczyk, P., et al. Graph of thoughts: Solving elaborate problems with large language models. In [Proceedings of the AAAI Conference on Artificial Intelligence](#), volume 38, pp. 17682–17690, 2024.
- Boiko, D. A., MacKnight, R., and Gomes, G. Emergent autonomous scientific research capabilities of large language models. [arXiv preprint arXiv:2304.05332](#), 2023.
- Bran, A. M., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. Chemcrow: Augmenting large-language models with chemistry tools. [arXiv preprint arXiv:2304.05376](#), 2023.
- Chen, G., Dong, S., Shu, Y., Zhang, G., Sesay, J., Karlsson, B. F., Fu, J., and Shi, Y. Autoagents: A framework for automatic agent generation. [arXiv preprint arXiv:2309.17288](#), 2023.
- Cheng, Z., Xie, T., Shi, P., Li, C., Nadkarni, R., Hu, Y., Xiong, C., Radev, D., Ostendorf, M., Zettlemoyer, L., et al. Binding language models in symbolic languages. [arXiv preprint arXiv:2210.02875](#), 2022.
- Dagan, G., Keller, F., and Lascarides, A. Dynamic planning with a llm. [arXiv preprint arXiv:2308.06391](#), 2023.
- Dasgupta, I., Kaeser-Chen, C., Marino, K., Ahuja, A., Babayan, S., Hill, F., and Fergus, R. Collaborating with language models for embodied reasoning. [arXiv preprint arXiv:2302.00763](#), 2023.
- (FAIR)†, M. F. A. R. D. T., Bakhtin, A., Brown, N., Dinan, E., Farina, G., Flaherty, C., Fried, D., Goff, A., Gray, J., Hu, H., et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. [Science](#), 378(6624):1067–1074, 2022.
- Fu, Z., Zhao, T. Z., and Finn, C. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. [arXiv preprint arXiv:2401.02117](#), 2024.
- Gao, C., Lan, X., Lu, Z., Mao, J., Piao, J., Wang, H., Jin, D., and Li, Y. S3: Social-network simulation system with large language model-empowered agents. [arXiv preprint arXiv:2307.14984](#), 2023.
- Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Duan, N., and Chen, W. Critic: Large language models can self-correct with tool-interactive critiquing. [arXiv preprint arXiv:2305.11738](#), 2023.
- Guan, L., Valmeekam, K., Sreedharan, S., and Kambhampati, S. Leveraging pre-trained large language models to construct and utilize world models for model-based task planning. [Advances in Neural Information Processing Systems](#), 36:79081–79094, 2023.
- Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., and Zhang, X. Large language model based multi-agents: A survey of progress and challenges. [arXiv preprint arXiv:2402.01680](#), 2024.
- Hao, S., Gu, Y., Ma, H., Hong, J. J., Wang, Z., Wang, D. Z., and Hu, Z. Reasoning with language model is planning with world model. [arXiv preprint arXiv:2305.14992](#), 2023.
- Hong, S., Zheng, X., Chen, J., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S. K. S., Lin, Z., Zhou, L., et al. Metagpt: Meta programming for multi-agent collaborative framework. [arXiv preprint arXiv:2308.00352](#), 2023.
- Huang, D., Bu, Q., Zhang, J. M., Luck, M., and Cui, H. Agentcoder: Multi-agent-based code generation with iterative testing and optimisation. [arXiv preprint arXiv:2312.13010](#), 2023a.
- Huang, X., Lian, J., Lei, Y., Yao, J., Lian, D., and Xie, X. Recommender ai agent: Integrating large language models for interactive recommendations. [arXiv preprint arXiv:2308.16505](#), 2023b.
- Huang, X., Liu, W., Chen, X., Wang, X., Wang, H., Lian, D., Wang, Y., Tang, R., and Chen, E. Understanding the planning of llm agents: A survey. [arXiv preprint arXiv:2402.02716](#), 2024.
- Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. [arXiv preprint arXiv:2310.06825](#), 2023.
- Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., Casas, D. d. l., Hanna, E. B., Bressand, F., et al. Mixtral of experts. [arXiv preprint arXiv:2401.04088](#), 2024.
- Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. Swe-bench: Can language models resolve real-world github issues? [arXiv preprint arXiv:2310.06770](#), 2023.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel,

- T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.
- Li, G., Hammoud, H. A. A. K., Itani, H., Khizbullin, D., and Ghanem, B. Camel: Communicative agents for "mind" exploration of large scale language model society. 2023a.
- Li, N., Gao, C., Li, Y., and Liao, Q. Large language model-empowered agents for simulating macroeconomic activities. *arXiv preprint arXiv:2310.10436*, 2023b.
- Li, Y., Yu, Y., Li, H., Chen, Z., and Khashanah, K. Tradinggpt: Multi-agent system with layered memory and distinct characters for enhanced financial trading performance. *arXiv preprint arXiv:2309.03736*, 2023c.
- Liu, B., Jiang, Y., Zhang, X., Liu, Q., Zhang, S., Biswas, J., and Stone, P. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023a.
- Liu, L., Yang, X., Shen, Y., Hu, B., Zhang, Z., Gu, J., and Zhang, G. Think-in-memory: Recalling and post-thinking enable llms with long-term memory. *arXiv preprint arXiv:2311.08719*, 2023b.
- Liu, S., Cheng, H., Liu, H., Zhang, H., Li, F., Ren, T., Zou, X., Yang, J., Su, H., Zhu, J., et al. Llava-plus: Learning to use tools for creating multimodal agents. *arXiv preprint arXiv:2311.05437*, 2023c.
- Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegrefe, S., Alon, U., Dziri, N., Prabhunoye, S., Yang, Y., et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024.
- Mao, Y., He, P., Liu, X., Shen, Y., Gao, J., Han, J., and Chen, W. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553*, 2020.
- Mukobi, G., Erlebach, H., Lauffer, N., Hammond, L., Chan, A., and Clifton, J. Welfare diplomacy: Benchmarking language model cooperation. *arXiv preprint arXiv:2310.08901*, 2023.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Parisi, A., Zhao, Y., and Fiedel, N. Talm: Tool augmented language models. *arXiv preprint arXiv:2205.12255*, 2022.
- Park, J. S., O’Brien, J., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023.
- Patil, S. G., Zhang, T., Wang, X., and Gonzalez, J. E. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*, 2023.
- Qian, C., Cong, X., Yang, C., Chen, W., Su, Y., Xu, J., Liu, Z., and Sun, M. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2023.
- Qin, Y., Hu, S., Lin, Y., Chen, W., Ding, N., Cui, G., Zeng, Z., Huang, Y., Xiao, C., Han, C., et al. Tool learning with foundation models. *arXiv preprint arXiv:2304.08354*, 2023.
- Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2024.
- Shen, Y., Song, K., Tan, X., Li, D., Lu, W., and Zhuang, Y. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36, 2024.
- Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.
- Surfís, D., Menon, S., and Vondrick, C. ViperGPT: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11888–11898, 2023.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023a.
- Wang, S., Liu, C., Zheng, Z., Qi, S., Chen, S., Yang, Q., Zhao, A., Wang, C., Song, S., and Huang, G. Avalon’s game of thoughts: Battle against deception through recursive contemplation. *arXiv preprint arXiv:2310.01320*, 2023b.

- Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. [arXiv preprint arXiv:2203.11171](#), 2022.
- Wang, Z., Cheng, Z., Zhu, H., Fried, D., and Neubig, G. What are tools anyway? a survey from the language model perspective. [arXiv preprint arXiv:2403.15452](#), 2024.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., and Duan, N. Visual chatgpt: Talking, drawing and editing with visual foundation models. [arXiv preprint arXiv:2303.04671](#), 2023a.
- Wu, Y., Tang, X., Mitchell, T. M., and Li, Y. Smartplay: A benchmark for llms as intelligent agents. [arXiv preprint arXiv:2310.01557](#), 2023b.
- Xiao, H. and Wang, P. Llm a*: Human in the loop large language models enabled a* search for robotics. [arXiv preprint arXiv:2312.01797](#), 2023.
- Xie, C., Chen, C., Jia, F., Ye, Z., Shu, K., Bibi, A., Hu, Z., Torr, P., Ghanem, B., and Li, G. Can large language model agents simulate human trust behaviors? [arXiv preprint arXiv:2402.04559](#), 2024a.
- Xie, J., Zhang, K., Chen, J., Zhu, T., Lou, R., Tian, Y., Xiao, Y., and Su, Y. Travelplanner: A benchmark for real-world planning with language agents. [arXiv preprint arXiv:2402.01622](#), 2024b.
- Xu, Z., Yu, C., Fang, F., Wang, Y., and Wu, Y. Language agents with reinforcement learning for strategic play in the werewolf game. [arXiv preprint arXiv:2310.18940](#), 2023.
- Yang, R., Song, L., Li, Y., Zhao, S., Ge, Y., Li, X., and Shan, Y. Gpt4tools: Teaching large language model to use tools via self-instruction. *Advances in Neural Information Processing Systems*, 36, 2024.
- Yang, Z., Ishay, A., and Lee, J. Coupling large language models with logic programming for robust and general reasoning from text. [arXiv preprint arXiv:2307.07696](#), 2023.
- Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models. [arXiv preprint arXiv:2210.03629](#), 2022.
- Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.
- Yuan, S., Song, K., Chen, J., Tan, X., Shen, Y., Kan, R., Li, D., and Yang, D. Easytool: Enhancing llm-based agents with concise tool instruction. [arXiv preprint arXiv:2401.06201](#), 2024.
- Zhang, H., Du, W., Shan, J., Zhou, Q., Du, Y., Tenenbaum, J. B., Shu, T., and Gan, C. Building cooperative embodied agents modularly with large language models. [arXiv preprint arXiv:2307.02485](#), 2023.
- Zhao, Q., Wang, J., Zhang, Y., Jin, Y., Zhu, K., Chen, H., and Xie, X. Competeai: Understanding the competition behaviors in large language model-based agents. [arXiv preprint arXiv:2310.17512](#), 2023.
- Zhao, Z., Lee, W. S., and Hsu, D. Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems*, 36, 2024.

A. Related Works

A.1. Reasoning Strategy for LLM Agents

Planning without Additional Components: In recent times, the development of large language models (LLMs) agents has enabled numerous tasks to be addressed by directly inputting questions into these models. However, LLM agents often struggle with some problem-solving, particularly in areas like mathematics and other intricate planning tasks. To address this limitation, recent research has developed various reasoning strategies aimed at enhancing the problem-solving abilities of LLM agents (Huang et al., 2024). One common strategy, employed by works like (Wei et al., 2022; Yao et al., 2022; Shen et al., 2024), leverages the divide-and-conquer approach. This involves breaking down a complex task into simpler subtasks and allowing the LLM agents to tackle each subtask sequentially. This method has shown improved performance across a range of tasks. Other works (Wang et al., 2022; Yao et al., 2024; Besta et al., 2024; Zhao et al., 2024; Xiao & Wang, 2023; Hao et al., 2023) employ similar strategies using tree searches, such as Monte Carlo Tree Search (MCTS), A*, Breadth-First Search (BFS), and Depth-First Search (DFS). They generate multiple alternative plans via various sampling methods and select the optimal plan through different selection techniques. Reflecting on and refining plans based on prior experiences also significantly improves the planning process of LLM agents (Shinn et al., 2024; Gou et al., 2023; Madaan et al., 2024; Huang et al., 2023b). By reevaluating their plans, LLM agents can avoid recurring errors, thereby enhancing their problem-solving capabilities.

Planning with Addition Components: The planning capabilities of LLM agents can also be improved through interactions with other components. For instance, (Liu et al., 2023a), (Guan et al., 2023), (Dagan et al., 2023), and (Cheng et al., 2022) combine symbolic planners with LLM agents, utilizing the natural language generation capabilities of LLMs to create formalized task descriptions. Other works (Liu et al., 2023b; Lewis et al., 2020; Yang et al., 2023; Mao et al., 2020) enhance the problem-solving ability of LLMs by integrating additional memory modules. A particularly effective approach involves enabling LLM agents to access various APIs and tools (Parisi et al., 2022; Qin et al., 2023; Schick et al., 2024; Wang et al., 2024; Yang et al., 2024; Liu et al., 2023c; Patil et al., 2023; Yuan et al., 2024). Some tools can be directly included in the context (Surfs et al., 2023; Shen et al., 2024; Wu et al., 2023a) as prompt. However, improving an LLM agent’s tool-using abilities often requires generating appropriate datasets and fine-tuning the models (Schick et al., 2024; Patil et al., 2023; Yang et al., 2024). By providing access to external tools, LLM agents can tackle tasks that are otherwise challenging, such as computational problems and formula verification. Our planning framework integrates these diverse reasoning strategies to improve overall planning capabilities, allowing for more sophisticated problem-solving and strengthening the effectiveness of LLM agents in handling complex tasks.

A.2. Multi-Agents Framework

Multi-agent frameworks (Guo et al., 2024) have garnered significant interest from researchers due to their flexibility across a broad range of tasks. These frameworks can be utilized in simulations of various kinds, including game simulations (Xu et al., 2023; Wang et al., 2023b; Mukobi et al., 2023), economic simulations (Li et al., 2023b;c), and societal simulations (Gao et al., 2023; Xie et al., 2024a; Park et al., 2023; Zhao et al., 2023; Aher et al., 2023). Moreover, multi-agent frameworks demonstrate superior performance compared to single-agent systems in solving diverse tasks. For instance, (Li et al., 2023a), (Hong et al., 2023), and (Chen et al., 2023) have developed general multi-agent frameworks that enhance task performance. For specific tasks, works like (Qian et al., 2023; Hong et al., 2023; Huang et al., 2023a) propose specialized frameworks that enable agents to autonomously develop software. In addition, multi-agent systems have been effectively employed for reasoning in embodied environments (Dasgupta et al., 2023; Zhang et al., 2023) and have demonstrated their potential in scientific research (Boiko et al., 2023; Bran et al., 2023). Our framework also incorporates a multi-agent system to enhance overall planning capabilities, leveraging the collaborative and complementary strengths of multiple agents for more effective problem-solving.

A.3. Limitation

Although our framework has significantly improved performance compared to previous studies, achieving a Final Pass Rate of 100% akin to human performance remains elusive. One limitation we observed is that even when provided with detailed information, LLM Agents still tend to overlook certain key points. Addressing this challenge may require the integration of alternative planning strategies. Additionally, effectively utilizing extensive accommodation-related information to generate reasonable plans remains a significant challenge, even for advanced models like GPT-4-Turbo. Furthermore, the occurrence of hallucinations during the planning process presents another obstacle that requires attention.

In future work, addressing these limitations could involve exploring novel planning strategies that complement the capabilities of LLM Agents, such as incorporating heuristics or domain-specific rules. Moreover, developing more sophisticated models that can better comprehend and utilize complex accommodation-related information could lead to more accurate and reliable travel plans. Additionally, investigating techniques to mitigate the occurrence of hallucinations and improve the overall robustness of LLM-based planning systems could further enhance their performance. Overall, overcoming these challenges will be crucial for advancing the state-of-the-art in LLM-driven travel planning systems.

A.4. Impact

The social impact of our work is profound, as it pioneers a novel methodology by emulating human reasoning processes to address Multi-Phases planning challenges. Equipping LLM agents with human-like problem-solving abilities, our framework enhances their efficacy in applications, such as travel planning, thereby potentially revolutionizing various industries reliant on complex decision-making. However, the advent of LLM agents with enhanced planning capabilities also raises concerns about potential job displacement and ethical implications surrounding data privacy and algorithmic bias, necessitating careful consideration and regulation to mitigate adverse consequences.

A.5. Ethics Statement

In our study, we introduce a novel human-like reasoning framework to enhance LLM agents' proficiency in Multi-Phases planning tasks, especially on traveling planning problems. While our work aims to advance the capabilities of LLM agents for beneficial purposes, we recognize the potential for misuse or unintended consequences, such as the propagation of inaccurate information. To address these ethical considerations, we advocate for ongoing research into methods for detecting and mitigating biases in LLMs, as well as the promotion of responsible use and transparency in the deployment of such frameworks. We urge users and researchers to remain vigilant of these ethical risks and prioritize the ethical considerations in the development and application of our framework.

A.6. Metrics:

- **Delivery Rate:** Measures if agents can successfully deliver a final plan within a set number of steps (max 45). Failure includes dead loops, numerous failed attempts, or exceeding the step limit.
- **Commonsense Constraint Pass Rate:** Assesses if agents can incorporate commonsense into their plans without explicit instructions, across eight dimensions.
- **Hard Constraint Pass Rate:** Evaluates if a plan meets all explicitly given hard constraints, testing agents' adaptability to diverse user queries.
- **Final Pass Rate:** Indicates the proportion of plans that meet all constraints (delivery, commonsense, and hard constraints), reflecting agents' overall proficiency in producing practical plans.
- **Micro Pass Rate:** The Micro Pass Rate evaluates the proportion of constraints that are successfully passed by an agent's plans, as shown in Formula 2. It is calculated by taking the total number of successfully met constraints across all plans and dividing it by the total number of constraints applied to all plans.
- **Macro Pass Rate:** The Macro Pass Rate assesses the proportion of plans that satisfy all of their constraints. It calculates the ratio of the number of plans that meet all applicable commonsense or hard constraints to the total number of plans evaluated.

$$\begin{aligned} \text{Micro Pass Rate} &= \frac{\sum_{p \in P} \sum_{c \in C_p} 1_{\text{passed}(c,p)}}{\sum_{p \in P} |C_p|}, \\ \text{Macro Pass Rate} &= \frac{\sum_{p \in P} 1_{\text{passed}(C_p,p)}}{|P|} \end{aligned} \tag{2}$$

P is the set of all plans. C_p is the set of constraints applicable to a specific plan p . $1_{\text{passed}(c,p)}$ is an indicator function that returns 1 if constraint c is passed in plan p , and 0 otherwise. $1_{\text{passed}(C_p,p)}$ is an indicator function that returns 1 if all constraints in C_p are passed in plan p , and 0 otherwise.

A.7. Example

A.8. Travel Query Example:

1. Could you please create a 5-day travel itinerary for one person, starting in Albuquerque and visiting 2 cities in Texas from March 25th to March 29th, 2022? The travel plan should work within a budget of \$2,100.
2. Could you help create a 7-day travel plan for a group of 3, departing from Greensboro and touring 3 different cities in Georgia from March 10th to March 16th, 2022? We have a new budget of \$4,000 for this trip. We'd also appreciate if our accommodations have smoking areas.
3. Could you help create a 5-day itinerary for a travel plan departing from Grand Junction and heading to 2 cities in Arizona from March 19th to March 23rd, 2022? It's a plan for two people with a budget of \$2,100. Our accommodations should allow visitors and our preference is for private rooms. Additionally, we do not require any flight transportation.
4. Could you create a 7-day travel itinerary for 2 people, departing from Albuquerque and visiting 3 cities in Texas from March 8th to March 14th, 2022? Our budget is set at \$5,000. We require accommodations that allow smoking and are preferably not shared rooms. We would prefer to avoid any flights for our transportation.

Figure 6. Few examples of the travel query

A.9. Travel Plan Example:

```
"day": 1,
"current_city": "from Buffalo to Atlanta",
"transportation": "Flight Number: F3502691, from Buffalo to Atlanta, Departure
Time: 18:48, Arrival Time: 20:51",
"breakfast": "-",
"attraction": "Georgia Aquarium, Atlanta;World of Coca-Cola, Atlanta;",
"lunch": "-",
"dinner": "Chaina Ram Sindhi Confectioners, Atlanta",
"accommodation": "Spacious private room close St. Barnabas Hospital, Atlanta"
"day": 2,
"current_city": "Atlanta",
"transportation": "-",
"breakfast": "Baba Au Rhum, Atlanta",
"attraction": "Atlanta Botanical Garden, Atlanta;High Museum of Art, Atlanta;",
"lunch": "Barkat, Atlanta",
"dinner": "Taste of Vishal, Atlanta",
"accommodation": "Spacious private room close St. Barnabas Hospital, Atlanta"
"day": 3,
"current_city": "from Atlanta to Buffalo",
"transportation": "Flight Number: F3502694, from Atlanta to Buffalo, Departure
Time: 15:47, Arrival Time: 17:42",
"breakfast": "Asian Bistro, Atlanta",
"attraction": "Piedmont Park, Atlanta;",
"lunch": "Beliram Degchiwala, Atlanta",
"dinner": "-",
"accommodation": "-"
```

Figure 7. An example of the travel plan

A.10. Outline Example:

The First Day: from Buffalo to Atlanta. Exploring Atlanta.
The Second Day: Exploring Atlanta.
The Third Day: from Atlanta to Buffalo.

1. **Departure and Return Dates:** The travel must commence on March 2nd, 2022, and conclude with a return to Buffalo on March 4th, 2022.
2. **Solo Travel:** The plan should be tailored for a single traveler, ensuring accommodations and activities are suitable for one person.
3. **Budget Limit:** The total cost of the trip, including transportation, accommodation, meals, and activities, must not exceed \$1,100.
4. **Destination Specific:** All travel arrangements must be made with Atlanta as the sole destination for the duration of the trip.
5. **Time Frame:** The entire travel plan, including all experiences and bookings, must fit within the 3-day window specified.

1. Accommodations should be booked for the correct dates corresponding to the traveler's stay in each city, ensuring no gaps or overlaps in lodging.
2. Dining options for breakfast, lunch, and dinner should be reasonably spaced throughout the day and geographically convenient to either the accommodation or planned attractions.
3. Transportation arrangements must align with the traveler's schedule, allowing sufficient time for check-in, security clearance at airports, and transfers between different modes of transport.
4. Attractions should be planned with consideration of their operating hours and any required advance reservations or tickets, avoiding scheduling conflicts.
5. The sequence of visiting attractions should be logical and efficient, minimizing backtracking and excessive travel time between locations.
6. Ensure that the transportation method selected is appropriate for the distance and time required to reach the scheduled activities, such as not planning long-distance road trips between breakfast and lunch.
7. Dining reservations should be made with consideration of dietary restrictions, restaurant capacity, and the expected duration of the meal to maintain the day's itinerary flow.
8. Verify that the accommodation has the necessary amenities and services required for the traveler's comfort and convenience, such as Wi-Fi if needed for planning or work purposes.

Figure 8. An example of the outline

A.10.1. STRATEGY BLOCK CONTENT:

Below is a preliminary outline of your trip, which can serve as a reference for collecting information:
 The First Day: from Ontario to Honolulu. Exploring Honolulu.
 The Second Day: Exploring Honolulu.
 The Third Day: from Honolulu to Ontario.
 You now need to gather relevant information to specify the travel plan for the First day. Remember that the total daily expenses of your trip (the sum of expenses for each person) do not exceed 1066.
 Hard Constraints. Total Budget: The trip must not exceed a total cost of \$3,200, including all transportation, accommodation, meals, and activities.
 {Detail Omit}
 Thought 1: To begin planning your first day, I will collect information on a flight from Ontario to Honolulu for March 4th, 2022. After that, I will look for dining options, attractions to visit, and accommodation in Honolulu. Let's start with the flight information.
 Action 1: FlightSearch[Ontario, Honolulu, 2022-03-04]
 Observation 1: Masked due to limited length. Make sure the data has been written in Notebook. Successfully recorded in Notebook: Flight Information for Ontario to Honolulu on March 4, 2022
 Thought 2: ...
 {Omit Other Process}
 Action 5: DailyPlanner[Please outline the plan for the first day of a 3-day trip from Ontario to Honolulu on March 4th, 2022, including flight, dining, attractions, and accommodation.] Observation 5: Travel Plan:
 Day 1:
 Current City: from Ontario to Honolulu
 Transportation: Flight Number: F3584294, from Ontario to Honolulu, Departure Time: 10:11, Arrival Time: 14:46
 Breakfast: - (Assuming breakfast is eaten before departure or on the flight)
 Attraction 1: Aloha Tower, Honolulu (Estimated visit time: 1 hour)
 Lunch: Bait El Khetyar, Honolulu (Cuisines: Cafe, Seafood, Average Cost: 28)
 Attraction 2: Iolani Palace, Honolulu (Estimated visit time: 1.5 hours)
 Attraction 3: Honolulu Museum of Art (HoMA), Honolulu (Estimated visit time: 2 hours)
 Dinner: Crystal Restaurant, Honolulu (Cuisines: Tea, Pizza, Bakery, Fast Food, Chinese, Average Cost: 57)
 Accommodation: Park, Subway & All Conveniences; Room type: Entire home/apt; Price: 754;
 City: Honolulu
 Outline:
 The First Day: from Ontario to Honolulu. Exploring Honolulu.
 The Second Day: Exploring Honolulu.
 The Third Day: from Honolulu to Ontario.
 Remember that the total daily expenses of your trip (the sum of expenses for each person) do not exceed 1066.
 You should gather the necessary information to plan your trip for the Second day. Each time you use the DailyPlanner tool, the planner can only access information queried during the previous 2 days. However, if the number of queries in the past 2 days is less than 5, then it will return the last 5 queried pieces of information.
 Thought 6: ...
 {Omit Future Steps}

Figure 9. An example of the content of strategy block

A.10.2. KNOWLEDGE BLOCK POP OUT CONTENT:

```
1: Flight Number: F3502691; Price: 240; DepTime: 18:48; ArrTime: 20:51;
OriginCityName: Buffalo; DestCityName: Atlanta
2: Flight Number: F3514187; Price: 322; DepTime: 06:51; ArrTime: 08:40;
OriginCityName: Buffalo; DestCityName: Atlanta
3: Flight Number: F3555201; Price: 265; DepTime: 12:44; ArrTime: 14:33;
OriginCityName: Buffalo; DestCityName: Atlanta
...
-----
1: Attraction Name: Atlanta Botanical Garden; City: Atlanta
2: Attraction Name: World of Coca-Cola; City: Atlanta
3: Attraction Name: Georgia Aquarium; City: Atlanta
...
-----
1: Restaurant: Adda; City: Atlanta; Cuisines: Tea, Mexican, Bakery;
Average Cost: 22; Rating: 3.8
2: Restaurant: Baba Au Rhum; City: Atlanta; Cuisines: Desserts, Pizza,
Mexican, BBQ, Fast Food; Average Cost: 27; Rating: 4.5
3: Restaurant: Barkat; City: Atlanta; Cuisines: Bakery, Indian,
Mediterranean, Desserts; Average Cost: 78; Rating: 3.4
...
-----
1: Accommodation: Fantastic Room in Bushwick; Room type: Private room;
Price: 1069.0; Minimum number of nights stay: 2.0; review rate number:
3.0; House rules: No children under 10; One room can accommodate how many
people: 2; City: Atlanta
2: Accommodation: Sunny, Friendly, Brooklyn Apartment; Room type:
Private room; Price: 874.0; Minimum number of nights stay: 1.0; review
rate number: 4.0; House rules: No pets; One room can accommodate how many
people: 2; City: Atlanta
3: Accommodation: 1bd in a sunny 2 bd Ft. Greene Apt; Room type: Private
room; Price: 1056.0; Minimum number of nights stay: 1.0; review rate
number: 4.0; House rules: No visitors & No pets; One room can accommodate
how many people: 1; City: Atlanta
...
```

Figure 10. An example of format information in Knowledge Block.

A.11. Prompt

A.11.1. PATHFINDER AGENT PROMPT:

You are a proficient planner. Based on the provided information and query, please give me an outline for my whole trip. Please help me generate an outline for each day of this query, primarily including which city it involves, whether there's a need to travel from one city to another. Don't include any specific details like flight numbers, restaurant names, or attraction names. You should use 'The First', 'The Second', 'The Third', etc., to indicate the order of the days. You shouldn't include any city or state names which are not in the query. In your outline, only the cities mentioned in the query can appear; absolutely no other cities or state names are allowed.

Here are some examples:

* Example 1 *

Query: Could you create a travel plan for 4 people from Boston to San Francisco spanning 3 days, from September 21st to September 23rd, 2023, with a budget of \$18,500?

Your Outline:

The First Day: from Boston to San Francisco. Exploring San Francisco.

The Second Day: Exploring San Francisco.

The Third Day: from San Francisco to Boston.

STOP

* Example 2 *

Query: Could you craft a 5-day journey itinerary for a party of 6? We'll be kicking off in Seattle and aim to explore 2 cities in California from April 25th to April 29th, 2023. Our budget is roughly \$16,500, and we prefer to book whole places for our lodging.

The cities in California are here: (Omitting the detail list of cities in California in this example.)

Your Outline:

The First Day: from Seattle to San Diego. Exploring San Diego.

The Second Day: Exploring San Diego.

The Third Day: from San Diego to Butte. Exploring San Diego or Butte.

The Fourth Day: Exploring Butte

The Fifth Day: from Butte to Seattle.

STOP

* Example 3 *

Query: Could you create a 7-day travel itinerary for two, starting from Seattle and traveling to New York, where we will explore 3 distinct cities? The journey is planned from July 10th to July 20th, 2023. Our updated budget is \$6,000. For our stays, we're looking for private accommodations. We plan to avoid any air travel for moving between locations. When it comes to dining, we're eager to try a range of food styles, including Italian, Japanese, Indian, and Thai.

The cities in New York are here: (Omitting the detail list of cities in New York in this example.)

Your Outline:

The First Day: from Seattle to Buffalo. Exploring Buffalo.

The Second Day: Exploring Buffalo.

The Third Day: from Buffalo to Niagara Falls. Exploring Buffalo or Niagara Falls.

The Fourth Day: Exploring Niagara Falls.

The Fifth Day: from Niagara Falls to Albany. Exploring Niagara Falls or Albany.

The Sixth Day: Exploring Albany.

The Seventh Day: from Albany to Seattle.

STOP

* Example Ends *

You must plan an outline that matches the number of days mentioned in the query.

If the trip is for three days, you must include The First day, The Second day, The Third day. If the duration is longer, for example, six days, then your outline must cover the content for The First Day, The Second day, through to the Sixth day. The transition between cities must follow the sentence pattern "from ... to ...".

{scratchpad}

Query:{query}

Your Outline:

Figure 11. PathFinder Agent Prompt

A.11.2. THOUGHT AGENT PROMPT:

Planning Query Plan Guidelines

General Instructions:

- Each day's plan needs the information of transportation, dining, attractions, and accommodation. You should collect the FOUR parts in the order of transportation, dining, attractions, and accommodation for the target city.
- Information will be recorded in a Notebook, and used later in the DailyPlanner tool to create a detailed daily plan.
- Collect information on a day-by-day basis, ensuring each day's plan includes transportation, dining, attractions, and accommodation information.
- Transformation information is only needed if there is a transfer between cities on that day. Otherwise, you should not collect it.
- Only you collecting information of the first day's plan, you can use the DailyPlanner tool to plan the first day. Don't collect the information of the second day's plan before you finish the first day's plan.

Transportation Information:

- Transportation choice: Flight, Taxi, Self-driving.
- Specify the starting and destination locations.
- Transfer Mode Rules:
 - If the initial transportation involves a flight, subsequent city transfers must also be via flight or taxi (no self-driving).
 - If initially self-driving, all subsequent city transfers should remain via self-driving.
- Please ensure that transportation information is collected only if there is a transfer between cities on that day; otherwise, it is not needed.
- Your query should clearly specify which kind of Transportation you want to know. (Flight from A to B on DATE or Self-driving from A to B or Taxi from A to B). You can't query for all transportation methods at once. You can only query for one transportation method at a time.

Dining, Attractions, and Accommodation:

- Clearly specify which type of information required, as well as the location.
- Collect dining attraction, accommodation information for the target city each day.
- You should not collect repeated information. If you have already collected information about restaurants, you should not collect it again.

Data Recording

- You should record the data into the Notebook immediately after FlightSearch, AccommodationSearch, AttractionSearch, RestaurantSearch or GoogleDistanceMatrix. Only the data stored in Notebook can be seen by DailyPlanner.
- You should express you need to record the data in your response after you collect information, like "I need to record the data into the Notebook" or "I need to write the data into the Notebook". If you don't express, the data will not be recorded.

Completion and Next Steps:

- Once enough information is collected, clearly state the need to proceed with the day's plan.
- You should collect dining, attractions, and accommodation information for the target city each day.
- You should explicitly state which kind of information you need to collect in each step before you use "\n". Gather the necessary information to plan the trip for {date} day.

Query: {query}
{scratchpad}

Figure 12. Thought Agent Prompt

A.11.3. TOOL AGENT PROMPT:

You are an experienced tool user. You are currently assisting in planning a trip. Then, I will provide you with the original travel query as well as which day's plan we are now planning. Besides, I will provide you with a 'Thought', and then you need to generate an 'Action' based on this 'Thought'. 'Action' can have 6 different types:

FlightSearch[Departure City, Destination City, Date]:

Description: A flight information retrieval tool.

Parameters:

Departure City: The city you'll be flying out from.

Destination City: The city you aim to reach.

Date: The date of your travel in YYYY-MM-DD format.

Example: FlightSearch[New York, London, 2022-10-01] would fetch flights from New York to London on October 1, 2022.

GoogleDistanceMatrix[Origin, Destination, Mode]:

Description: Estimate the distance, time and cost between two cities.

Parameters:

Origin: The departure city of your journey.

Destination: The destination city of your journey.

Mode: The method of transportation. Choices include 'self-driving' and 'taxi'.

Example: GoogleDistanceMatrix[Paris, Lyon, self-driving] would provide driving distance, time and cost between Paris and Lyon.

AccommodationSearch[City]:

Description: Discover accommodations in your desired city.

Parameter: City - The name of the city where you're seeking accommodation.

Example: AccommodationSearch[Rome] would present a list of hotel rooms in Rome.

RestaurantSearch[City]:

Description: Explore dining options in a city of your choice.

Parameter: City - The name of the city where you're seeking restaurants.

Example: RestaurantSearch[Tokyo] would show a curated list of restaurants in Tokyo.

AttractionSearch[City]:

Description: Find attractions in a city of your choice.

Parameter: City - The name of the city where you're seeking attractions.

Example: AttractionSearch[London] would return attractions in London.

DailyPlanner[Query]

Description: A smart planning tool that crafts detailed Daily Travel plans. You should use this when the user wants to plan a travel plan.

Parameters:

Query: The query from user.

Example 1: DailyPlanner[Please outline the plan for the first day of a 3-day trip from Seattle to New York.] This request would yield a comprehensive plan specifically for the initial day of the journey.

Example 2: DailyPlanner[Can you detail the itinerary for the second day of a 3-day journey from Seattle to New York, with this day being in New York?] This prompt asks for a detailed plan for the trip's second day, with an emphasis on the fact that this day is spent in New York City.

Each action only calls one function once. Do not add any description in the action. Your actions shouldn't violate the constraints provided in the query.

Please make sure your action does not start with ['\n', 'Thought', 'Action', 'Observation'] and assume all the actions are permitted in this environment.

Original Travel Plan Query:{query}

We are now planning {date} day of the trip.

Some constraints are provided for the plan: {constraints}

{outlines}

Previous Thought, Action and Observation:

{previous_react}

Thought: {thought}

Action:

Figure 13. ToolAgent Prompt

A.11.4. PLAN AGENT PROMPT:

You are a proficient planner. Based on the provided information and query, please give me a detailed plan, including specifics such as flight numbers (e.g., F0123456), restaurant names, and accommodation names. Note that all the information in your plan should be derived from the provided data. You must adhere to the format given in the example. Additionally, all details should align with commonsense. The symbol '-' indicates that information is unnecessary. For example, in the provided sample, you do not need to plan after returning to the departure city. When you travel to two cities in one day, you should note it in the 'Current City' section as in the example (i.e., from A to B).

***** Example *****

Query: Could you create a travel plan for 4 people from Boston to San Francisco spanning 3 days, from September 21st to September 23rd, 2023, with a budget of \$18,500?

Requirement: You need to plan for the third day of the trip.

Travel Plan:

Day 1:

Current City: from Boston to San Francisco

Transportation: Flight Number: B549321, from Boston to San Francisco, Departure Time: 07:00, Arrival Time: 10:22

Breakfast: The Bay View Cafe, Boston

Attraction: The San Francisco Museum of Modern Art, San Francisco

Lunch: Golden Gate Grille, San Francisco

Dinner: Alcatraz Eatery, San Francisco

Accommodation: Luxurious City View Apartment in Nob Hill!, San Francisco

Day 2:

Current City: San Francisco

Transportation: -

Breakfast: Sunrise Deli, San Francisco

Attraction: The Exploratorium, San Francisco; Golden Gate Park, San Francisco.

Lunch: Pier 39 Seafood Palace, San Francisco

Dinner: The Fog City Diner, San Francisco

Accommodation: Luxurious City View Apartment in Nob Hill!, San Francisco

Day 3:

Current City: from San Francisco to Boston

Transportation: Flight Number: S623478, from San Francisco to Boston, Departure Time: 17:37, Arrival Time: 21:09

Breakfast: The Golden Coffee Shop, San Francisco

Attraction: Coit Tower, San Francisco.

Lunch: Lombard Street Bistro, San Francisco

Dinner: The Bay Bridge Restaurant, San Francisco

Accommodation: -

Apart from travel by Flight, there are other modes of transportation available:

Transportation: Self-driving, from Key West to San Antonio, Departure Time: 09:00, Arrival Time: 12:26

Transportation: Taxi, from Key West to San Antonio, Duration: 2 hours 22 mins, Distance: 254 km, Cost: 254

Please choose the appropriate mode based on the specifics of your plan.

***** Example Ends *****

Given information: {text}

Rules:

Rule1: On the final day of the trip, accommodation information is not required; use '-' to indicate this is unnecessary.

Rule2: Diversify the travel plans as much as possible within the context of activities.

Rule3: Transportation methods must be consistent with the earlier part of the trip. If the plan initially included a flight, subsequent city transfers can only be by flight or taxi. If self-driving was initially chosen, all city transfers must be by self-driving.

Rule4: Transportation details are only required during city transfers. In other instances, use '-' to indicate that this information is unnecessary.

Rule5: The minimum night stay of accommodation should be considered when planning the trip.

Constraints: {hard_constraints}\n{common_constraints}

Budget requirement: The total budget for the plan you make for each day cannot exceed {avg_budget}.

Query: {query}

Detail Query: {detail_query}

Travel Outlines: {outlines}

Requirement: You need to plan for {date} day of the trip.

{inform}

Travel Plan:

{previous_plan}

Rule6: You can not add any additional information beyond the specific plan. Pay attention to the format provided in the example.