Journal of Computer Technology and Software
ISSN:2998-2383
Vol. 4, No. 7, 2025

Autonomous Planning and Execution Method Based on Task Graph Construction for Large Language Model Agents

Qingfan Yang[1*], Lin Kuan-Yu[2]
[1]Cornell University, Ithaca, USA
[2]Columbia University, New York, USA
*Corresponding author: Qingfan Yang; angus313074221@outlook.com

Abstract: To address the issues of insufficient goal parsing, loose stage connections, insufficient consistency in the execution chain, and weak constraint satisfaction in the autonomous planning and execution process of large language model agents in complex task scenarios, this paper proposes an autonomous planning and execution method based on task graph construction and stage dependency constraint modeling of large language model agents. This method first jointly encodes user goals, contextual environment, and historical interaction information to form a unified semantic representation for complex tasks, and then generates and structures multi-stage sub-tasks based on this representation. Subsequently, by constructing a task graph representation containing stage nodes, dependencies, and constraint semantics, the hierarchical structure and logical connections within the complex task are explicitly depicted. Furthermore, a stage dependency constraint modeling mechanism is introduced to uniformly constrain preconditions, state transmission, and execution feasibility, thereby enhancing the overall coherence of the planning process and the stability of the execution process. During the execution phase, a stage-aware strategy is used to dynamically select and control the sequence of currently executable nodes, enabling the model to maintain strong structural consistency and decision rationality in complex task processing. The comparative results show that the proposed method achieves superior performance in terms of task completion capability, planning integrity, execution consistency, and constraint satisfaction. This demonstrates that the proposed framework can effectively improve the autonomous planning and execution quality of large language model agents in complex task scenarios, providing new methodological support for the construction of highly reliable intelligent agents for real-world application environments.

Keywords: Semantic goal parsing; stage-aware decision-making; dependency-constrained reasoning; structured execution control.

1. Introduction

As large language models demonstrate continuously improving comprehensive capabilities in natural language understanding, knowledge reasoning, and complex decision generation, autonomous task execution in open environments is gradually becoming an important development direction in intelligent systems research[1], [2]. Compared to traditional automation methods that rely on fixed processes and predefined rules, agents based on large language models possess stronger goal parsing, context awareness, and dynamic adjustment capabilities, enabling them to transform abstract requirements into executable action sequences in complex contexts. This technological evolution allows intelligent agents to move beyond single-turn question answering or local inference, extending to higher-level capabilities such as long-link task decomposition, cross-stage state tracking, and multi-step collaborative execution. Therefore, systematic research on the autonomous planning and execution mechanisms of large language model agents has become a crucial foundation for promoting the practical application of general artificial intelligence[3].

However, in real-world task scenarios, autonomous planning and execution is not a simple step generation problem, but involves joint modeling across multiple levels, including task goal parsing, sub-task organization, execution sequence arrangement, resource condition matching, and stage state constraint coordination[4]. Current large language model agents often face challenges when handling complex, multi-stage tasks, including ambiguous task boundaries, unstable step granularity, insufficient representation of dependencies, and susceptibility to local errors in the execution chain. Particularly in long-flowing, multi-condition, and strongly constrained tasks, the lack of clear structured task representation and explicit stage dependency modeling mechanisms can lead to redundancy in planning paths, conflicts between steps, distorted state propagation, and local decisions deviating from the global objective. These problems not only weaken the stability of autonomous execution but also limit the scalability and reliability of large language model agents in highly complex task scenarios.

Task graph construction provides an effective path for the structured representation of complex tasks. By mapping the overall goal to a graph structure composed of nodes, paths, and dependencies, the hierarchical relationships, sequential logic, and collaborative mechanisms between subtasks can be more clearly depicted, providing an interpretable organizational framework for planning generation.

Simultaneously, stage dependency constraint modeling can further reveal the preconditions, state inheritance relationships, and result feedback mechanisms between different execution stages, shifting task planning from static step arrangement to dynamic organization oriented towards process consistency[5]. Combining task graph construction with stage-dependent constraint modeling not only improves the decomposition accuracy and execution coherence of large language model agents for complex goals, but also enhances their adaptive adjustment capabilities in uncertain environments, providing theoretical support for building an autonomous execution framework that combines flexibility, structure, and controllability.

Therefore, research on autonomous planning and execution methods focusing on task graph construction and stage-dependent constraint modeling for large language model agents has significant theoretical and practical value[6]. From a theoretical perspective, this direction helps propel large language model agents from shallow planning based on language generation to deeper autonomous intelligence oriented towards structural understanding, process control, and global collaboration, providing new research perspectives for representation learning, constraint reasoning, and action organization in complex task execution[7]. From an application perspective, this research can provide a more robust technical foundation for scenarios such as intelligent office work, complex information processing, interactive decision support, automated process management, and multi-step service collaboration, improving the execution efficiency, logical consistency, and result reliability of intelligent agents when facing real-world task requirements. Therefore, exploring structured planning and stage-constrained collaborative mechanisms for complex tasks has become an important research direction for promoting high-quality autonomous execution of large language model agents.

2. Methodology

The proposed method formulates autonomous planning and execution as a structured process in which a large language model based Agent converts an open ended user objective into a task graph with explicit stage dependencies, thereby reducing semantic ambiguity in long horizon decision chains and improving the consistency between planning intent and executable actions. Rather than treating the target task as a flat sequence, the overall objective is first represented as a semantic goal vector derived from the instruction context, external state, and historical interaction trace, so that the subsequent decomposition step is guided by both current intent and process memory. This paper presents the overall model architecture, as shown in Figure 1.

Formally, let the input objective be denoted by , and let the contextualized goal embedding be written as:$\mathcal{O}$

$$\boldsymbol{g} = \Phi_g(\mathcal{O}, \mathcal{C}, \mathcal{H})$$

where denotes the environmental context, denotes the interaction history, and denotes the semantic encoder that maps heterogeneous task signals into a unified planning space. Guided by , the Agent generates a set of candidate stages and organizes them into a graph structured representation:$\mathcal{C}\mathcal{H}\Phi_g\boldsymbol{g}$

$$\mathcal{T} = (\mathcal{V}, \mathcal{E}, \boldsymbol{S}, \boldsymbol{D})$$

in which is the stage node set, is the dependency edge set, stores stage level semantic states, and encodes constraint descriptors associated with inter stage transitions. This formulation is essential because complex tasks rarely fail at the level of local action generation alone, but more often break when prerequisite states are omitted or when later operations are scheduled before the required conditions become valid. To preserve the semantic completeness of each stage, the representation of node is produced by jointly encoding its subgoal, expected output, and operational condition through:$\mathcal{V} = \{v_1, v_2, \ldots, v_N\}\mathcal{E}\boldsymbol{S}\boldsymbol{D}v_i$

Graph-based knowledge representation and semantic alignment offer a complementary basis for constructing stage nodes whose meanings remain comparable across heterogeneous instructions and environmental descriptions[8]. Knowledge graph-augmented semantic fusion further supports the integration of long-context task evidence, allowing the Agent to preserve relations among goals, tools, and intermediate outputs while reducing semantic fragmentation during decomposition[9]. Instruction-driven language modeling through cross-attention fusion similarly motivates coupling the user objective with contextual signals when forming stage-level semantic states[10].

$$\mathbf{s}_i = \Phi_s(u_i, y_i, c_i)$$

where denotes the local subgoal, denotes the expected stage outcome, and denotes the executable condition associated with that stage. Dependency construction is then carried out by measuring whether the output state of one stage can validly support the precondition of another, which yields:$u_i y_i c_i$

$$e_{ij} = \mathbb{I}(\Psi(\boldsymbol{s}_i, \boldsymbol{s}_j) > \tau)$$

where denotes a dependency scorer, denotes the activation threshold, and returns a binary edge decision. Such an edge induction mechanism is meaningful because it transforms implicit procedural knowledge into explicit structural relations, allowing the Agent to reason over feasibility before actual execution begins.$\Psi\tau\mathbb{I}(\cdot)$

Once the task graph has been constructed, stage dependency constraint modeling is introduced to regulate the legal transition space between adjacent or nonadjacent stages, which prevents the execution trajectory from deviating from the global objective when the environment changes or when intermediate outputs become uncertain. In this design, each dependency is not merely a static link but a typed constraint carrying temporal precedence, resource admissibility, and state compatibility. The aggregated constraint intensity from stage to stage is defined as:$v_i v_j$

$$d_{ij} = \alpha d_{ij}^{\text{pre}} + \beta d_{ij}^{\text{res}} + \gamma d_{ij}^{\text{sta}}$$

where measures prerequisite satisfaction, measures resource consistency, measures state continuity, and , , and balance the relative importance of the three factors. Since long chain tasks often contain hidden coupling across distant stages, a masked dependency propagation mechanism is further used

to refine stage representations under structural constraints through:$d_{ij}^{\text{pre}} d_{ij}^{\text{res}} d_{ij}^{\text{sta}} \alpha \beta \gamma$

$$\boldsymbol{h}_i^{'} = \sigma\left(\sum_{j \in \mathcal{N}(i)} a_{ij} \boldsymbol{W} \boldsymbol{h}_j + \boldsymbol{b}\right)$$

where denotes the dependency neighborhood of stage , denotes the normalized influence weight derived from , denotes the projection matrix, denotes the bias term, and denotes a nonlinear activation. By updating stage states in this manner, the planning process becomes sensitive to structural feasibility rather than relying only on token level continuation, which is particularly important for avoiding locally plausible but globally inconsistent action proposals.$\mathcal{N}(i) i a_{ij} d_{ij} \boldsymbol{W} \boldsymbol{b} \sigma(\cdot)$

Frequent subgraph mining and structural prompt injection provide a useful mechanism for identifying recurring dependency motifs and making their constraints explicit during graph-guided reasoning[11]. Adaptive fusion of low-rank and soft-gated sparse updates further suggests an efficient adaptation strategy for retaining shared planning behavior while selectively adjusting the model to task-specific constraint patterns[12].

To bridge planning and execution, a stage aware policy is designed so that the Agent can choose the next executable stage according to both graph topology and real time state feedback, thus maintaining continuity between symbolic planning intentions and grounded operational decisions. At each decision step, the executable candidate set is filtered by the currently satisfied constraints, after which the policy assigns a priority score to every legal stage and selects the one with the highest utility under the present context. The corresponding decision rule is written as:

$$\pi\left(a_t | z_t\right) = softmax\left(\Phi_p\left(z_t, \mathcal{T}, \boldsymbol{M}_t\right)\right)$$

where denotes the current execution state, denotes the dynamic memory recording previously completed stages and observed feedback, and denotes the policy scorer that integrates local state, graph structure, and historical evidence. This stage aware policy has a clear methodological value because autonomous execution is not only a matter of selecting the next action, but also a matter of ensuring that the selected action preserves the validity of all downstream dependencies. When execution feedback indicates that some prerequisite is violated or that a generated outcome fails to match the expected semantic target, the graph is adaptively revised and the priority distribution is recalibrated, which enables the framework to recover from partial planning errors without discarding the entire reasoning chain.$z_t \boldsymbol{M}_t \Phi_p$

Joint optimization of dual retrieval and adaptive re-ranking strengthens the policy module by supporting the retrieval of relevant execution traces and the context-sensitive prioritization of candidate stages[13]. Interpretable feedback principles from vision-language-model-based emotional understanding also motivate exposing the evidence behind stage selection, so that a planner can communicate why a transition is admissible and how feedback changes subsequent execution choices[14].

Finally, the whole framework is optimized through a unified objective that jointly encourages semantic correctness in task decomposition, structural consistency in dependency construction, and decision reliability in execution selection, thereby aligning representation learning with process level control. Instead of learning stage semantics, dependency relations, and execution policy in isolation, the model performs coupled optimization so that graph formation and action generation can mutually reinforce each other during training. The overall loss is defined as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{plan}} + \lambda_2 \mathcal{L}_{\text{dep}} + \lambda_3 \mathcal{L}_{\text{exe}}$$

where constrains the quality of stage decomposition, supervises dependency prediction, regularizes execution decisions, and , , and control their relative contributions. Through this joint formulation, the Agent acquires the ability to interpret abstract goals as structured execution graphs, propagate stage constraints across the planning horizon, and perform adaptive decision making under evolving task states. Consequently, the proposed method provides a coherent route for improving the controllability, interpretability, and robustness of autonomous planning and execution in large language model based Agent systems.$\mathcal{L}_{\text{plan}} \mathcal{L}_{\text{dep}} \mathcal{L}_{\text{exe}} \lambda_1 \lambda_2 \lambda_3$

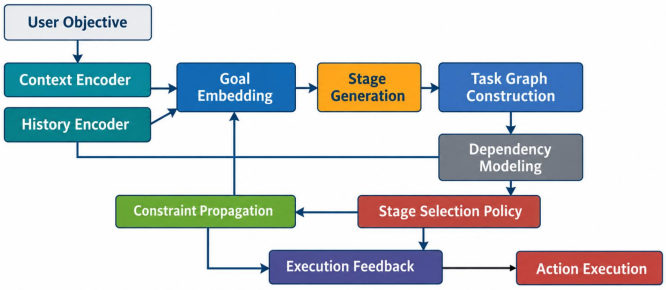

Figure 1. Overall model architecture

3. Experimental Results and Analysis

3.1 Dataset introduction

This paper selects WebArena as the experimental dataset and task environment. This dataset is an open-source, real-world web page task benchmark built for autonomous agents using large language models. It employs a self-hosted, interactive web page environment, unifying various website scenarios such as e-commerce, forums, content management, and collaborative development into an executable task space. Natural language commands drive the agent to complete multi-step web page operations, state tracking, and goal achievement. Compared to datasets that only contain static question answering or single-round decision-making, WebArena emphasizes stage decomposition, dependency linkage, long-term planning, and execution feedback in complex tasks. Therefore, it better meets the research needs of autonomous planning and execution based on large language model agent task graph construction and stage dependency constraint modeling. Furthermore, WebArena's code and benchmark environment are publicly available, demonstrating good openness, reproducibility, and task diversity, making it suitable as a unified data foundation for validating the methods presented in this paper.

3.2 Experimental Results and Analysis

To make a unified comparison between the proposed method and existing related research from multiple dimensions, representative works closely related to large language model agent autonomous planning, subtask decomposition, graph structure reasoning, and collaborative planning execution in recent years were selected as reference methods. The comparison results were compiled under a consistent evaluation standard, and the experimental results are shown in Table 1.

Table 1. Experimental results compared with other models

Methods	SR	PC	EC	CS
Ruan et al.[15]	74.62	71.48	69.35	70.91
Wu et al.[16]	78.34	76.12	73.89	75.43
Zhu et al.[17]	80.17	78.46	76.58	79.04
Xie et al.[18]	77.29	74.83	72.64	73.95
Verma et al.[19]	81.06	79.25	78.31	80.14
Li et al.[20]	82.48	80.72	79.68	81.33
Chen et al.[21]	83.11	81.36	80.27	82.05
Ours	86.92	84.71	83.64	85.48

Table 1 uses four evaluation metrics to measure the method's performance. SR stands for Success Rate, measuring the overall ability to successfully complete task objectives during autonomous planning and execution; PC stands for Plan Completeness, reflecting the completeness of the task planning chain and the sufficiency of key step coverage; EC stands for Execution Consistency, assessing the consistency of the execution process in terms of stage transitions, state transfer, and operational implementation; and CS stands for Constraint Satisfaction, characterizing the method's ability to satisfy stage dependencies, preconditions, and execution constraints in complex tasks. These four metrics correspond to core dimensions such as task completion, planning quality, execution stability, and constraint compliance, and can comprehensively reflect the actual performance of the autonomous planning and execution method.

From an overall performance perspective, the algorithm proposed in this paper demonstrates superior performance across multiple evaluation dimensions, indicating that the constructed method can form a more coordinated linkage between task decomposition, stage organization, and execution control. On the one hand, the task graph construction mechanism enhances the structured expression of complex goals, enabling the planning process to possess clearer hierarchical relationships and semantic organization capabilities. On the other hand, stage-dependent constraint modeling further strengthens the ability to perceive conditional constraints and maintain state consistency during execution, thereby improving the overall planning execution quality. Therefore, the method presented in this paper not only generates more reasonable task arrangements but also maintains high stability and constraint compliance during execution, demonstrating strong advantages in autonomous planning and execution.

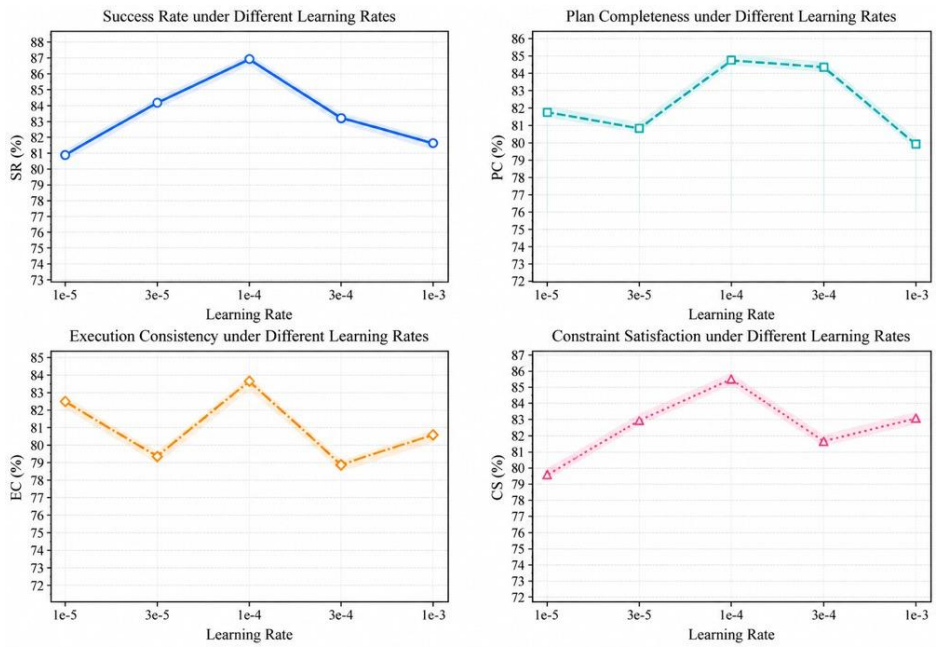

Figure 2. The impact of learning rate on experimental results

As shown in Figure 2, the proposed algorithm maintains good task planning and execution quality under different learning rate settings, indicating that the method has strong parameter adaptability and overall stability. Thanks to the effective characterization of complex target structures by the task graph construction mechanism and the continuous constraints on logical consistency and condition satisfaction in the execution process by stage dependency constraint

modeling, the model exhibits high levels of performance in task completion capability, planning completeness, execution consistency, and constraint satisfaction. This demonstrates that the constructed method can form a more coordinated stage organization and decision control mechanism during autonomous planning and execution, thereby ensuring a more reasonable, stable, and reliable overall execution process.

4. Conclusion

This paper addresses the challenges faced by large language model agents in complex task scenarios, including loose planning chains, insufficient stage connections, and weak controllability in the execution process. It proposes an autonomous planning and execution method based on task graph construction and stage dependency constraint modeling. Starting with the structured representation of complex goals, this method integrates task decomposition, stage organization, dependency modeling, and execution decision-making into a unified framework. This enables the autonomous agent to form clearer task hierarchies, more rational execution sequences, and more stable state transmission relationships in an open environment. By using the task graph as the planning carrier and explicitly integrating stage dependencies into the execution control process, this method demonstrates strong advantages in improving the depth of task understanding, enhancing the coherence of the planning process, and ensuring the consistency of execution logic. This research further illustrates that large language model agents for complex tasks should not only remain at the level of generative step inference but should continuously advance towards structured, constrained, and procedural autonomous decision-making mechanisms, thereby providing a more solid methodological foundation for high-quality intelligent execution.

From a research value perspective, the significance of this work lies not only in the optimization of the methodological level but also in the expansion of the application paradigm of large language model agents. Traditional language-based agents often suffer from localized rationality but overall mismatch when faced with long-chain, multi-constraint, and highly dependent tasks. The framework proposed in this paper effectively enhances the agent's overall grasp of complex processes by strengthening the structural relationships within tasks and the conditional linkages between stages. This capability is significant for applications such as intelligent office work, automated task management, complex information processing, interactive service orchestration, digital process collaboration, and multi-step decision support. As large language models gradually shift from knowledge-based question answering to task execution, the requirements for interpretability, stability, and reliability in related systems are constantly increasing. This research provides a valuable technical path for building agent systems with greater engineering usability and scenario adaptability, and lays the theoretical and methodological foundation for the subsequent construction of intelligent execution platforms for complex real-world tasks.

Further research is still possible. On the one hand, it can enhance the adaptive update capability of task graphs in a more dynamic and open environment, enabling agents to complete higher-quality online replanning and continuous execution under conditions of changing task objectives, external state disturbances, or adjustments to intermediate conditions. On the other hand, it can further enhance the hierarchical expression capability of stage-dependent constraints, incorporating time constraints, resource constraints, semantic constraints, and multi-agent collaborative constraints into a unified modeling framework to improve the method's coverage depth and generalization ability for complex scenarios. Furthermore, as large language model agents continue to penetrate fields such as smart government, enterprise process management, intelligent operation and maintenance, educational assistance, medical service collaboration, and complex human-computer interaction, how to ensure execution efficiency while also considering security, transparency, and traceability will become an important research topic in this direction. Overall, research on autonomous planning and execution based on task graph construction and stage-dependent constraint modeling has the necessity for continued deepening and broad application prospects, and is of positive and far-reaching significance for promoting large language model agents from language understanding to highly reliable autonomous action.

References

[1] J. Xie, K. Zhang, J. Chen, et al., "Travelplanner: A benchmark for real-world planning with language agents", arXiv preprint arXiv:2402.01622, 2024.

[2] X. Zhang, Y. Deng, Z. Ren, et al., "Ask-before-plan: Proactive language agents for real-world planning", Findings of the Association for Computational Linguistics: 2024 Conference on Empirical Methods in Natural Language Processing, pp. 10836-10863, 2024.

[3] L. E. Erdogan, N. Lee, S. Kim, et al., "Plan-and-act: Improving planning of agents for long-horizon tasks", arXiv preprint arXiv:2503.09572, 2025.

[4] R. Xiao, W. Ma, K. Wang, et al., "Flowbench: Revisiting and benchmarking workflow-guided planning for LLM-based agents", Findings of the Association for Computational Linguistics: 2024 Conference on Empirical Methods in Natural Language Processing, pp. 10883-10900, 2024.

[5] M. Hu, P. Zhao, C. Xu, et al., "Agentgen: Enhancing planning abilities for large language model based agent via environment and task generation", Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Volume 1, pp. 496-507, 2025.

[6] G. Gonzalez-Pumariega, L. S. Yean, N. Sunkara, et al., "Robotouille: An asynchronous planning benchmark for LLM agents", arXiv preprint arXiv:2502.05227, 2025.

[7] J. Liu, W. Hao, K. Cheng, et al., "Large language model-based planning agent with generative memory strengthens performance in textualized world", Engineering Applications of Artificial Intelligence, vol. 148, p. 110319, 2025.

[8] J. Zhou, "Graph-Based Knowledge Representation and Semantic Alignment for Large Language Model-Driven Entity Recognition", 2025.

[9] Z. Zhu, "Knowledge Graph-Augmented Semantic Fusion for Large Language Model-Based Long-Text Classification", 2024.

[10] Y. Jiang and Y. L. Peng, "Instruction-Driven Language Model for Text Classification via Cross-Attention Fusion and Semantic Alignment", 2024.

[11] Z. Xiao, C. Moretti and Q. Li, "Frequent Subgraph Mining and Structural Prompt Injection for Large Language Model Fine-Tuning", 2025.

[12] B. Chen, "Adaptive Fusion of Low-Rank and Soft-Gated Sparse Updates for Parameter-Efficient Instruction Tuning", 2024.

[13] L. Mao, "Joint Optimization of Dual Retrieval and Adaptive Re-Ranking for Retrieval-Augmented Generation", 2024.

[14] J. Y. Su, "Vision-Language Model-Based Emotional Understanding and Interpretable Feedback for Art Therapy Support", 2025.

[15] J. Ruan, Y. Chen, B. Zhang, et al., "Tptu: Task planning and tool usage of large language model-based AI agents", NeurIPS 2023 Foundation Models for Decision Making Workshop, 2023.

[16] X. Wu, Y. Shen, C. Shan, et al., "Can graph learning improve planning in LLM-based agents?", Advances in Neural Information Processing Systems, vol. 37, pp. 5338-5383, 2024.

[17] Y. Zhu, S. Qiao, Y. Ou, et al., "Knowagent: Knowledge-augmented planning for LLM-based agents", Findings of the Association for Computational Linguistics: 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics, pp. 3709-3732, 2025.

[18] J. Xie, K. Zhang, J. Chen, et al., "Revealing the barriers of language agents in planning", Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1: Long Papers, pp. 1872-1888, 2025.

[19] N. Verma and M. Bharadwaj, "LEAP & LEAN: Look-ahead Planning and Agile Navigation for LLM Agents", Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, Volume 6: Industry Track, pp. 896-933, 2025.

[20] A. Li, Y. Xie, S. Li, et al., "Agent-oriented planning in multi-agent systems", arXiv preprint arXiv:2410.02189, 2024.

[21] J. Chen, H. Li, J. Yang, et al., "Enhancing LLM-based agents via global planning and hierarchical execution", arXiv preprint arXiv:2504.16563, 2025.