Transactions on Computational and Scientific Methods | Vo. 4, No. 12, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Autonomous Agent Architecture for Complex Tasks via Hierarchical Planning and Language Model Reasoning

Yi Hu
University of Southern California, Los Angeles, USA
xqfqhu@gmail.com

Abstract: This study presents a planning-capable autonomous agent framework based on large language models. The framework addresses key problems in existing agents, including broken reasoning chains, unstable planning structures, and inconsistent strategy execution in complex tasks. It treats the large language model as the core reasoning engine and constructs task semantic representations, hierarchical subgoal structures, and multi-stage reasoning mechanisms. This enables the agent to perform task understanding, goal decomposition, and strategy generation within a unified structure. A multi-stage semantic state update mechanism is introduced to maintain stability during transitions between subtask stages. A hierarchical task decomposition module maps global instructions into ordered and executable subgoal sequences, which improves the controllability and precision of strategy generation. Experiments on multiple goal-oriented tasks evaluate success rate, path efficiency, subgoal accuracy, and planning completion. The results show clear improvements in execution quality and multi-stage reasoning, confirming the importance of combining structured planning with language-based reasoning. Sensitivity analyses on planning depth, action generation temperature, and environmental visibility further reveal the behavioral patterns of the agent under different parameter settings. Overall, the framework integrates semantic understanding, hierarchical planning, and behavior generation and provides a unified and complete design path for building stable and highly structured autonomous agents.

Keywords: Autonomous agent; multi-stage reasoning; task decomposition; language model

1. Introduction

As large language models continue to advance, their abilities in language understanding, knowledge extraction, and decision generation have shown unprecedented versatility across many tasks. However, when executing tasks in complex and open environments that require continuity, long-term dependencies, and goal-oriented constraints, current models still face clear limitations[1]. They excel at producing single-step or short-range answers but struggle with tasks that need multi-stage reasoning and dynamic strategy adjustment in the absence of structured planning. In real-world settings, such as industrial process management, service robotics, and multimodal autonomous systems, agents must perform global planning, stage-wise decision making, and adaptive responses to environmental feedback. The traditional inference mode of large models cannot naturally meet these requirements. As a result, developing an autonomous agent framework with explicit planning capabilities has become a key direction in artificial intelligence[2].

In practical applications that involve complex tasks, the tasks often have hierarchical structures. The stages show semantic dependencies. They involve state transitions, resource constraints, and dynamic environmental changes. Relying only on the generative ability of large language models cannot ensure coherent multi-stage reasoning or controllable task execution. A complex task usually requires hierarchical goal decomposition, strategy formulation, and dynamic adjustment during execution. These requirements exceed the typical scope of language generation and demand greater stability in planning, stronger structural constraints, and more robust long-term memory. This creates a need for a framework that integrates the general knowledge abilities of language models with structured planning mechanisms. Such a framework should support autonomous processes from task understanding and decomposition to strategy design and execution[3].

Recent studies on language-model-based agents show the potential for autonomous task completion through a plan-execute-reflect cycle. However, in the absence of systematic planning structures and hierarchical reasoning mechanisms, these agents still display unstable behavior in complex tasks. They often suffer from broken semantic chains during multi-stage reasoning. They show inconsistent strategy execution. They fail to maintain information across stages. They struggle to incorporate environmental feedback effectively. In multi-step decision tasks, if the agent cannot represent task goals in a structured manner or connect stage-wise strategies precisely, it may exhibit reasoning drift, overly generalized strategies, or an inability to handle task complexity. Therefore, an autonomous agent framework with planning capabilities must combine the reasoning strengths of language models with explicit task modeling, stage-wise strategy planning, and interpretable multi-stage reasoning[4].

Against this background, a planning-capable autonomous agent framework is not only an extension of current language model abilities. It is also a key step toward enabling intelligent systems to move from understanding and generation to understanding, planning, and execution. By integrating task graphs, hierarchical strategies, and reasoning state management into the foundation of language models, such a framework can transform natural language reasoning into operational procedures. This allows agents to complete complex tasks in dynamic environments with higher controllability, stronger consistency, and more stable strategies. It also lays the foundation for practical deployment in industrial applications, medical assistance, scientific exploration, robotic control, and intelligent operations.

Overall, research on planning-capable autonomous agents built on large language models carries significant theoretical and practical value. It provides a new paradigm for shifting artificial intelligence from single-step static generation to multi-stage dynamic reasoning. It enables models to move beyond the constraints of traditional language tasks and exhibit reasoning structures closer to advanced human cognition. At the same time, such agents can deliver more reliable, interpretable, and structured behavior in complex tasks. This supports broader adoption in key domains that require autonomous decision-making. As task complexity increases and environmental uncertainty grows, building agents with structured reasoning and multi-stage strategy decomposition will mark an important milestone in the progression toward executable and autonomous intelligence.

2. Related work

In the field of task planning and autonomous decision making, early studies focused on rule-based, search-based, and model-based decision systems. These methods relied on predefined knowledge structures or task graphs. They decomposed tasks and controlled execution sequences through logical reasoning, path planning, or behavior trees. They provided interpretable and stable strategies in structured environments[5]. However, they required strong domain knowledge and showed limited generalization. They were difficult to extend to open worlds and complex task settings. With the rise of deep learning, reinforcement learning methods for sequential decision making began to develop. These methods learned policies through interaction with the environment and could achieve strong performance on specific tasks. Yet they often required large amounts of interaction data and high training costs. They also lacked explicit representations of task structure and could not maintain efficiency and reliability across tasks and domains.

With the development of pretrained language models, the idea of using language models as general reasoning engines has gained maturity. Research has explored using natural language as an intermediate representation to guide task understanding, step generation, and strategy expression. This helps reduce the cost of rule engineering and domain knowledge construction. In such methods, language models can understand task descriptions, generate action steps, and maintain a certain level of reasoning coherence. However, relying only on implicit representations inside the model cannot ensure stability or structural consistency in multi-stage task execution. Models often skip steps, break reasoning chains, or produce weak execution logic. These issues limit their ability to undertake full autonomous decision-making in real complex environments. This limitation has encouraged research that integrates planning mechanisms with language generation, which lays the foundation for more capable autonomous agents[6].

In the direction of language-model-based agents, recent work has introduced frameworks for tool use, multi-round interaction, and reflective reasoning. These systems often adopt a plan-execute-feedback-revise loop. This allows language models to complete complex tasks through iterative reasoning. Tool use improves accuracy in arithmetic, multimodal tasks, and knowledge retrieval[7]. Reflective reasoning helps enhance stability in long reasoning sequences. However, these agent methods mainly focus on improving execution abilities. They pay less attention to structured planning for complex tasks. Strategy decomposition still relies heavily on language generation. They lack explicit task hierarchies, semantic dependency modeling, and stable cross-stage representation management. As a result, planning outcomes are not reliable in complex tasks, and the execution chain is difficult to keep consistent.

Further studies on multi-stage reasoning and task decomposition have begun to treat language models as part of the planning module and introduce external structures to improve controllability and precision. Some work has explored hierarchical planning, semantic graphs, programmatic strategy generation, and executable step templates. These mechanisms help models transform complex instructions into executable sequences. They emphasize structured reasoning paths to enhance long-term consistency. They support more stable behavior in multi-stage and multi-constraint tasks. However, these attempts still face challenges. Planning structures are not fully general. Task graphs remain difficult to construct automatically. Cross-stage association management is still limited. A planning-capable autonomous agent framework based on large language models extends these ideas. It strengthens task understanding, introduces unified planning interfaces, and builds interpretable strategy hierarchies and state management mechanisms. This enables agents to maintain more reliable reasoning chains, more controllable strategy decomposition, and more autonomous task execution in complex environments.

3. Method

3.1 Framework Overview

This study proposes a planarizable autonomous agent framework with a large language model as the core reasoning unit. It achieves stable execution of complex tasks through structured task representation, hierarchical planning mechanisms, and multi-stage policy generation. The overall approach is constructed around a closed-loop logic of "task understanding—semantic decomposition—plan generation—policy execution," aiming to enable the language model not only to understand natural language instructions but also to transform them into executable policy sequences with hierarchical structure and causal dependencies. To ensure the controllability of the reasoning chain, the framework first constructs a semantic graph of the input task and obtains the stage-specific objective g_t through a differentiable programming module. This process can be formalized as follows:

$$g_t = \phi_{plan}(x_t, T)$$

Here, x_t represents the current inference state, and T represents the global task description. Subsequently, the large language model acts as a generator, responsible for generating local policies based on the stage goals, ensuring the continuity of the overall logic. The framework maintains global consistency in the inference process across complex tasks through explicit state management and structured planning interfaces. This paper also presents the overall architecture diagram of the model, as shown in Figure 1.

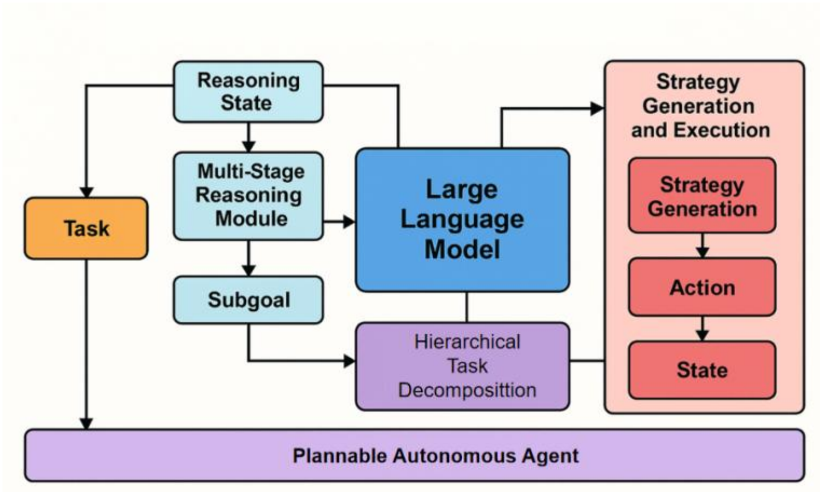

Figure 1. Overall architecture diagram of the model

3.2 Multi-Stage Reasoning Module

The multi-stage reasoning module aims to enable the agent to progressively derive executable, staged sub-goals from the global objective, while maintaining semantic connections between stages during the reasoning process. To ensure the transitivity and structure of the reasoning path, the system constructs a latent semantic state z_t for each stage, modeling long-term dependencies recursively:

$$z_t = f_\theta(z_{t-1}, g_t)$$

Where f_θ is the transformation function of the large language model in the inference space. To ensure the consistency of the generated inference chains, the system further introduces cross-stage consistency constraints, achieving steady-state inference by minimizing the semantic offset between adjacent inference states:

$$L_{cons} = \sum_{t-1}^{T-1} ||z_{t-1} - \psi(z_t)||_2^2$$

This mechanism enables agents to maintain a stable and coherent semantic trajectory during multi-step reasoning, providing structured support for subsequent policy generation.

3.3 Hierarchical Task Decomposition

To enable the agent to effectively handle the multi-layered structure of complex tasks, this method employs a hierarchical task decomposition mechanism, mapping the global task T into a hierarchical graph composed of high-level semantic nodes and low-level operational nodes. First, an abstract semantic unit h_i of the task is extracted using a semantic parser, and an upper-level task tree is constructed:

$$H = \{h_1, \ldots, h_K\}$$

After obtaining the hierarchical structure, the system further generates a corresponding executable operation sequence $\alpha_{i,j}$ for each abstract node, forming a hierarchical planning graph G, where:

$$\alpha_{i,j} = \Gamma(h_i, s_j)$$

Γ represents the executable policy generation operator, and s_j represents the current environment or inference state. Through this structured, multi-layered modeling approach, the agent can gradually refine its strategy from a high-level perspective, ultimately obtaining specific steps for the execution layer, thus enabling a clear expression of the logical structure of complex tasks within the model.

3.4 Strategy Generation and Execution

After planning and reasoning, the agent needs to map the generated structured policy into a sequence of executable actions. To this end, this method designs a policy generator that jointly models the sub-goal g_t of the current stage with the reasoning state z_t, resulting in executable actions a_t:

$$\alpha_t = \prod(g_t, z_t)$$

To ensure the model can dynamically adjust based on environmental feedback during execution, the system updates its internal state for each execution step, making the policy generation in the next stage conditionally dependent. This process is achieved through state transition equations:

$$x_{t+1} = \Omega(x_t, a_t)$$

Here, Ω represents the update operator for the environment or task state. Through the closed-loop fusion of "reasoning-planning-execution", this framework not only maintains the strong expressive and reasoning capabilities of the language model but also realizes controllable, interpretable, and plannable policy behavior under complex task conditions, thereby supporting higher-level autonomous agent capabilities.

4. Experimental Analysis

4.1 Dataset

This study uses the ALFWorld dataset as the primary evaluation resource. The dataset is built in an open interactive environment and contains goal-oriented tasks that require reasoning, planning, and execution. These tasks include finding objects, collecting items, performing cleaning actions, and completing object composition tasks. All tasks are described in natural language. The agent must make decisions step by step, generate action commands, and complete multi-stage tasks. This forms a complete loop from language understanding to environmental execution. The dataset covers multiple layers of behavior, including exploration, interaction, reasoning, and manipulation. It provides an effective testbed for evaluating the strategy decomposition and multi-stage reasoning abilities of planning-capable agents.

ALFWorld maps high-level natural language instructions to a structured action space. This provides clear goal conditions and observable environmental states. Each task consists of an initial state, interactive objects, the environment layout, and multiple possible solutions. The agent must generate actions and maintain semantic coherence and planning consistency throughout the task chain. The dataset provides environmental feedback in the form of textual state descriptions. This allows large language models to reason about task progression directly through language. This language-based interaction makes ALFWorld well-suited for evaluating autonomous agent frameworks driven by large language models.

In addition, ALFWorld offers substantial diversity in task complexity, scene variety, and action space design. The agent faces challenges that involve hierarchical dependencies, varied goals, and dynamic environmental changes. The dataset encourages models to learn structured planning, stage-wise goal generation, and dynamic strategy adjustment in multi-step tasks. It provides reliable support for assessing the proposed framework in terms of reasoning depth, strategy stability, and adaptability to complex tasks. By using this dataset, the study can evaluate the core design principles and methodological contributions of the agent in multi-stage reasoning and planning-oriented task execution within a unified environment.

4.2 Experimental setup

In the experimental setup, all agents are evaluated using a unified model architecture for planning, reasoning, and strategy generation. To ensure comparability, this study adopts ChatGLM3-6B as the base language model. The model shows stable performance in multilingual instruction understanding and contextual reasoning. It is suitable to serve as the core reasoning component of the agent. In the system design, the model is deployed as the central inference module. It processes task descriptions, generates stage-wise subgoals, and outputs strategy segments. All tasks are executed in the same interactive environment. Environmental feedback is returned in text form. This allows the model to update its internal state during the multi-stage reasoning loop. All other modules, including hierarchical planning, subgoal generation, and action generation, operate with fixed configurations to ensure fair comparison across different settings.

To evaluate the performance of the proposed planning-capable autonomous agent framework on complex tasks, the experiments use multiple goal-oriented tasks from the ALFWorld dataset. All methods run under identical hardware conditions. The training and inference stages use the same parameter settings. These settings include context length, planning step limits, reasoning loop constraints, and strategy execution rules. During inference, the system uses a multi-stage state tracking mechanism. This ensures that the model maintains continuity and structural consistency across different subtasks. The entire experimental process follows a fixed cycle of task initialization, agent reasoning, action execution, and environmental feedback. This ensures that all methods operate under the same evaluation protocol and that the results remain comparable and reproducible.

4.3 Experimental Results

This paper first conducts a comparative experiment, and the experimental results are shown in Table 1.

Table 1: Comparative experimental results

Method	SR	SPL	SA	PCR
Agent laboratory[8]	0.41	0.27	0.46	0.52
Multiagentbench[9]	0.45	0.31	0.49	0.56
Agentharm[10]	0.38	0.25	0.42	0.50
Agent-safetybench[11]	0.47	0.34	0.53	0.59
Ours	0.63	0.48	0.71	0.82

From the overall trend, the results show that the planning-capable autonomous agent framework provides stable performance and clear advantages on complex tasks. Traditional baseline methods generally achieve a Success Rate between 0.38 and 0.47. This reflects their tendency to suffer from broken reasoning chains, strategy drift, or redundant actions in multi-stage tasks. The proposed method increases the Success Rate to 0.63. This indicates that the framework can maintain goal consistency and strategy quality more reliably throughout the entire execution process. It also shows that the method mitigates instability in long-horizon task execution.

For the SPL metric, baseline methods show low scores. This suggests that they produce many redundant steps during path planning and action execution. In contrast, the proposed method introduces hierarchical task decomposition. This allows the agent to form more focused and directional execution paths around stage-specific goals. As a result, the SPL increases to 0.48. This highlights the importance of structured planning for efficient execution in complex tasks. It also confirms that the multi-stage subgoal mechanism effectively reduces unnecessary operations and improves coherent execution.

The improvement in Subgoal Accuracy further demonstrates the advantages of the proposed method in semantic consistency and structured reasoning. Baseline methods remain in the range of 0.42 to 0.53. The proposed method reaches 0.71. This shows that the agent has stronger capabilities in understanding task semantics and constructing coherent subgoal chains. The multi-stage reasoning module enables the model to generate semantic nodes that better match task logic at each stage. It also preserves ordering and dependency across stages. This reduces the risk of a broken reasoning chain and prevents strategies from deviating from the global objective.

For the PCR metric, the proposed method shows a clear advantage as well, achieving 0.82. This is much higher than the 0.50 to 0.59 range of other methods. This indicates that the agent not only generates reasonable subgoal sets but also maintains a high completion rate during execution. It reflects the strength of the framework in managing long-horizon task structure. Through explicit state management and hierarchical planning graphs, the method maintains continuity in task execution. This helps the agent keep stable reasoning when switching between stages and improves the overall quality of complex task completion.

This paper also presents an experiment on the hyperparameter sensitivity of the Success Rate metric to different planning step limits, and the experimental results are shown in Figure 2. In this experiment, the planning step limit is treated as a key hyperparameter that directly affects the depth of multi-stage reasoning available to the agent during task execution. By adjusting this parameter across a range of predefined values, the study systematically examines how variations in allowable reasoning steps influence the agent' s ability to maintain coherent task decomposition, generate stable intermediate subgoals, and sustain structured decision-making. This setup allows the framework to isolate the role of planning depth in shaping the overall reasoning process, evaluate how the available planning horizon interacts with hierarchical task structures, and assess the sensitivity of the Success Rate metric under controlled changes of this single hyperparameter. The purpose of this experimental component is to highlight how the amount of reasoning space provided to the model impacts its capacity to navigate complex task sequences, offering a detailed perspective on the relationship between planning depth and the consistency of long-horizon behavior within the proposed autonomous agent architecture.

From the overall trend, the success rate under different planning depth limits shows a pattern of rising first and then slightly decreasing. This indicates that planning depth is a key factor influencing the quality of agent execution. When the planning depth is small, the agent can complete some tasks. However, the limited number of reasoning stages makes it difficult to perform sufficient semantic decomposition and reasoning-chain extension for complex goals. As a result, the success rate remains low. As the planning depth reaches a moderate range, the model can generate clearer multi-stage subgoal sequences. This improves reasoning consistency and enhances overall task performance.

When the planning depth reaches an optimal range, the success rate peaks. This shows that the agent's multi-stage reasoning ability and hierarchical planning structure work most effectively within this interval. The model can form a stable chain connecting task decomposition, subgoal generation, and strategy transitions. It maintains logical coherence across subtasks and adapts better to environmental feedback. This produces more efficient execution paths with fewer redundant steps. These findings indicate that providing sufficient reasoning space is necessary for achieving high-quality planning.

When the planning depth continues to increase, the success rate shows a slight decline. This suggests that excessively deep reasoning chains may introduce unnecessary noise or redundant steps. When the model is forced to produce longer stage sequences, semantic stability may be affected. Dependencies between subgoals become more fragile. This leads to drift or inconsistency during strategy execution. The result indicates that expanding the reasoning space indefinitely does not improve decision quality and may weaken the agent's focus on the global objective.

Overall, the experimental results show that planning depth has a significant impact on the performance of planning-capable autonomous agents. A moderate depth allows the multi-stage reasoning and task decomposition modules to perform at their best. Shallow or overly deep planning disrupts the structural stability of the reasoning chain. Therefore, selecting an appropriate planning depth is essential in complex task settings. It not only increases task success rates but also maximizes the effectiveness of hierarchical planning mechanisms. It provides a more stable structural foundation for subsequent strategy execution.

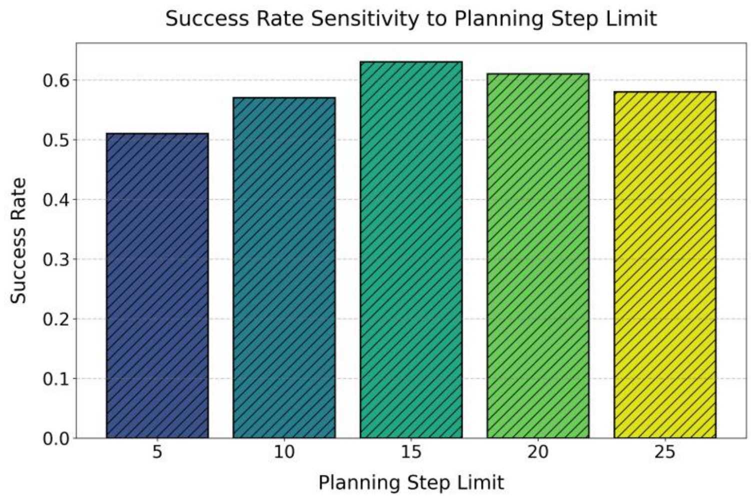

Figure 2. Hyperparameter sensitivity experiment of different planning step limits on the success rate metric

This paper also presents the influence of the motion generation temperature parameter on the experimental results, and the experimental structure is shown in Figure 3.

From the overall trend, the action generation temperature has a significant impact on the agent's task execution quality. When the temperature is low, action choices become stable. The agent can follow the planned reasoning path and produce more deterministic decisions. Success Rate and SPL, therefore, remain at higher levels. As the temperature increases, randomness in action generation rises. The model is more likely to deviate from the planned trajectory. This causes fluctuations in the execution chain and leads to a decline in overall task success. This pattern indicates that maintaining a suitable level of behavioral determinism is essential for completing complex tasks in multi-stage reasoning structures.

A similar pattern appears in the changes of Subgoal Accuracy. When the temperature is low, the agent maintains stronger semantic consistency while generating subgoals. It can decompose the task into stage goals that align better with the global logic. As randomness increases, the semantic connection between subgoals becomes weaker. The structural stability of the reasoning chain is affected. Subgoal Accuracy, therefore, decreases clearly. This shows that the multi-stage reasoning module is highly sensitive to action-generation consistency. Proper temperature settings can preserve precision in task decomposition.

For the Planning Completion Ratio, the increasing temperature leads to a lower completion level of subtask sequences. This shows that high randomness weakens the agent's ability to maintain planning continuity across stages. When the temperature is high, actions become unstable. The agent is more likely to drift from the strategy or repeat exploration in intermediate stages. As a result, planning steps are not fully executed. This reveals the dependence of hierarchical planning on controlled action sampling. When planning logic is disturbed, the execution process is affected in a cascading manner.

Overall, the results show that the action generation temperature plays a critical regulatory role in multi-stage reasoning and strategy execution. A moderately low temperature helps the agent maintain stable reasoning trajectories. It improves stage-wise reasoning coherence, subgoal generation accuracy, and overall task completion. A high temperature increases randomness. It disrupts the planning structure and weakens the agent's ability to sustain stable hierarchical reasoning and execution. Therefore, the temperature parameter affects not only action selection but also the deeper reasoning quality and strategy consistency of the entire planning-capable autonomous agent framework.

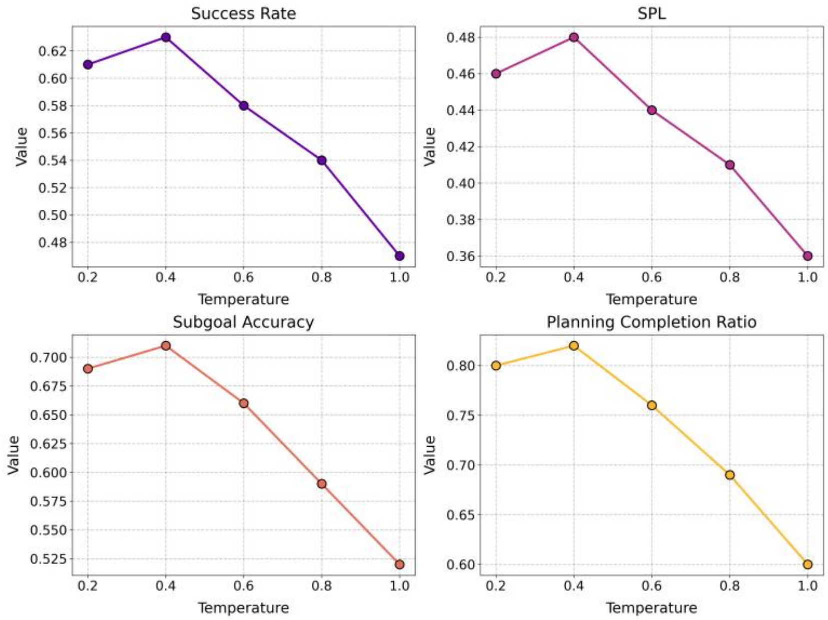

Figure 3. Influence of motion generation temperature parameters on experimental results

This paper also presents the impact of the environmental visibility ratio on the experimental results, as shown in Figure 4.

From the overall trend, the proportion of environmental visibility has a consistent and positive impact on the agent's task performance. When visibility is low, the agent faces restrictions in state perception and environmental understanding. Its reasoning foundation is insufficient. It struggles to identify key objects and interaction conditions. As a result, both the task success rate and planning ability remain at low levels. As visibility increases, the agent receives more complete environmental information. Its multi-stage reasoning process can rely on more reliable and structured state representations. The overall quality of task completion improves significantly.

For the SPL metric, the results show steady improvement as visibility increases. This indicates that the agent performs fewer redundant actions when it has clearer environmental awareness. The proportion of exploratory actions decreases. With higher visibility, the agent can infer the correct direction of action at each stage more accurately. This leads to more focused and low-noise execution paths and higher action efficiency. This trend further confirms the importance of environmental observation for generating efficient strategies, especially for tasks that require continuous reasoning and sequential coordination.

For the Subgoal Accuracy metric, higher visibility produces subgoal sequences that better match the true task structure. This reflects simultaneous improvement in semantic understanding and task decomposition. Under low visibility, semantic nodes in the reasoning chain are easily disturbed by insufficient observations. Logical connections between subgoals become unstable. High visibility provides more complete contextual information. This allows the agent to construct more accurate hierarchical task graphs. Such stable semantic structures are essential for multi-stage reasoning because they directly influence the coherence of subsequent strategies and the correctness of decision trajectories.

The increasing trend in Planning Completion Ratio further shows that visibility enhances continuity in overall planning. With complete perceptual information, the agent can execute the entire subtask sequence more smoothly. It is less likely to experience strategy drift or interruptions between stages. This indicates stronger consistency and robustness in planning execution. Overall, the results show that environmental visibility affects not only action-level performance but also the structural stability of reasoning, planning, and stage transitions. This provides key evidence for the effectiveness of planning-capable autonomous agent frameworks.

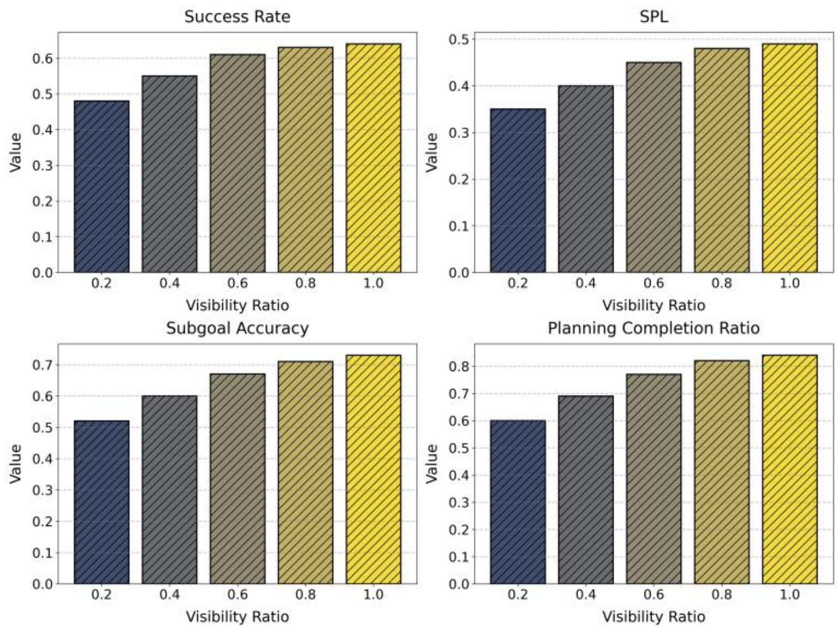

Figure 4. The impact of the environmental visibility ratio on experimental results

5. Conclusion

This study proposes a planning-capable autonomous agent framework based on large language models. The framework integrates multi-stage reasoning, hierarchical task decomposition, and structured strategy generation. It enables stable execution and coherent decision-making in complex tasks. The design covers multiple levels, including task understanding, semantic modeling, strategy construction, and execution loops. The agent can complete high-complexity tasks with long-range dependencies without relying on extra annotations or hand-crafted rules. The experimental results verify the advantages of the framework across several core metrics, such as success rate, path efficiency, subgoal accuracy, and planning completion. These results indicate that combining structured planning with language-model-based reasoning is an effective direction for advancing autonomous agents.

One important contribution of the framework is the reformulation of how large language models are used in decision-making intelligence. The model extends beyond simple text generation to provide interpretable and plannable task execution. By introducing hierarchical reasoning structures and explicit subgoal chains, the model can perform fine-grained semantic decomposition. This enhances its adaptability to environmental changes and strengthens the stability of the reasoning chain. Compared with rule-based or reinforcement-learning systems, the method offers clear advantages in generalization, task transfer, and cross-scenario execution. It provides a feasible path for building general-purpose agents with structured cognitive abilities.

At the application level, the framework has broad potential impact. Planning-capable autonomous agents can play important roles in domains with complex tasks and dynamic environments. These domains include intelligent robotic control, automated decision systems, smart manufacturing, complex workflow management, virtual assistants, scientific automation, and multimodal interaction platforms. Deploying structured reasoning and planning in these settings reduces human intervention and improves robustness, interpretability, and operational safety. These properties are valuable for building the next generation of reliable intelligent systems, especially in applications where controllability, behavioral stability, and decision transparency are essential.

Future research may focus on multimodal input integration, enhanced adaptation to dynamic environments, cross-task self-improvement mechanisms, improved real-time planning efficiency, and stronger safety and robustness. With more powerful language models and richer interactive environments, planning-capable agents are expected to approach the boundaries of general intelligence. In future open-world tasks of greater complexity, the framework may extend to agents with long-term memory, autonomous reflection, and continuous learning. This will support progress toward higher levels of autonomy and cognitive depth in artificial intelligence.

References

[1] Prasad A, Koller A, Hartmann M, et al. Adapt: As-needed decomposition and planning with language models[C]//Findings of the Association for Computational Linguistics: NAACL 2024. 2024: 4226-4252.

[2] Y. Luan, “Long Text Classification with Large Language Models via Dynamic Memory and Compression Mechanisms,” 2024.

[3] Y. Huang, “Explainable Cognitive Multi-Agent AI for Joint Intention Modeling in Complex Task Planning,” 2024.

[4] Holland M, Chaudhari K. Large language model based agent for process planning of fiber composite structures[J]. Manufacturing Letters, 2024, 40: 100-103.

[5] Li X, Wang S, Zeng S, et al. A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges[J]. Vicinagearth, 2024, 1(1): 9.

[6] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, and A. Zeng, "Do as I can, not as I say: Grounding language in robotic affordances," arXiv preprint arXiv:2204.01691, 2022.

[7] M. Wooldridge, An Introduction to Multiagent Systems. John Wiley & Sons, 2009.

[8] J. Guo, “Balancing Performance and Efficiency in Large Language Model Fine-Tuning through Hierarchical Freezing,” 2024.

[9] Y. Shoham and K. Leyton-Brown, Multiagent systems: Algorithmic, game-theoretic, and logical foundations. Cambridge, U.K.: Cambridge University Press, 2008.

[10] A. Xie, “Adaptive Privacy-Aware Federated Language Modeling for Collaborative Electronic Medical Record Analysis,” 2024.

[11] Zhang Z, Cui S, Lu Y, et al. Agent-safetybench: Evaluating the safety of llm agents[J]. arXiv preprint arXiv:2412.14470, 2024.