Transactions on Computational and Scientific Methods | Vo. 4, No. 2, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Hierarchical Large Language Model Agents for Multi-Scale Planning in Dynamic Environments

Yujun Zou

University of California, Berkeley, Berkeley, USA

zouyujun526@gmail.com

Abstract: This study addresses the complexity and uncertainty of agent planning and decision-making in dynamic environments and proposes a hierarchical large language model agent method for multi-scale planning. The approach begins with global goal modeling, where the semantic reasoning ability of the language model generates a global planning vector. A task decomposition mechanism then refines the global goal into an executable sequence of sub-goals. At the local execution level, the agent employs a policy network to perform action selection and dynamic adaptation, achieving coordination between global and local levels. The framework optimizes both global subgoal constraints and local action accuracy, enabling robust hierarchical planning through joint optimization. To validate the effectiveness of the proposed method, experiments were conducted under multiple sensitivity scenarios. The results show that the agent demonstrates strong stability and recovery ability under varying noise intensity, optimizer types, planning update frequencies, and mutation frequencies and amplitudes, while maintaining task decomposition accuracy and overall performance in uncertain and dynamic conditions. In addition, sensitivity experiments on learning rate and weight decay reveal the critical role of hyperparameters in maintaining hierarchical strategy stability, providing clear guidance for model design and optimization. Overall, the study demonstrates the advantages of hierarchical large language model agents in multi-scale planning under dynamic environments and confirms their robustness and adaptability through detailed experimental design and quantitative evaluation.

Keywords: Hierarchical agents, multi-scale planning, dynamic environments, robustness

1. Introduction

In today's complex and rapidly changing environment, planning and decision-making for agent systems are becoming a key focus of artificial intelligence research. With increasing uncertainty in external conditions, traditional single-scale or static planning methods often struggle to cope with diverse challenges in dynamic environments[1]. For example, rapid changes in environmental states, progressive task requirements, and dynamic adjustments of resource constraints all place higher demands on the planning and decision-making capabilities of agents. Therefore, constructing an architecture that can flexibly adapt and support multi-scale planning has become a critical issue in advancing artificial intelligence to a higher level. The emergence of large language models provides a new pathway to this problem. Their strong knowledge representation and reasoning abilities offer a solid foundation for hierarchical planning and dynamic adaptation[2].

The value of multi-scale planning lies in its ability to model across different temporal and spatial levels, enabling both strategic and tactical objectives. Traditional planning methods are often confined to a single layer. They either emphasize long-term goals at the macro level or rely heavily on instant decisions at the micro level, resulting in a lack of coordination and flexibility in the overall system[3,4]. This limitation becomes more pronounced in dynamic environments, where uncertainty requires agents to constantly switch and balance across scales. By introducing hierarchical architectures, large language models can perform abstract reasoning and global planning at higher levels, while also handling fine-grained dynamic adjustments at lower levels. This capacity is crucial for improving robustness and adaptability in complex scenarios.

At the same time, hierarchical agents show unique advantages in multi-task and multi-scenario applications. In real-world problems, planning is rarely about optimizing a single objective. It usually involves parallel task processing and resource coordination[5,6]. For instance, urban traffic scheduling must balance global flow optimization with local congestion relief. Collaboration among unmanned systems must align overall task allocation with individual execution details. Single-layer models find it difficult to meet such requirements. A hierarchical planning framework powered by large language models can achieve efficient coordination across tasks and scenarios by setting and decomposing goals at multiple scales. This capability not only extends the application scope of agents but also provides practical support for solving complex real-world problems.

In dynamic environments, planning is not only about choosing paths but also about information perception, knowledge integration, and decision adaptation. Large language models, with their rich semantic modeling abilities, can extract key information, identify hidden constraints, and reason with existing knowledge under uncertain conditions[7]. A hierarchical architecture strengthens this advantage. It enables high-level integration of long-term goals and background knowledge, mid-level task decomposition and resource allocation, and low-level dynamic adjustment and instant response. Such a design enhances the adaptability of agents in complex environments. It also improves their transferability and generalization across scenarios, thereby achieving intelligent decision-making in a more complete sense.

In conclusion, multi-scale planning for dynamic environments is not only a technical requirement but also an important step toward high-level intelligent agents. The integration of large language models provides a new opportunity for designing and implementing hierarchical agents. It breaks the limitations of traditional planning frameworks and enables the organic combination of global and local, as well as long-term and short-term, in dynamic contexts[8]. Research in this direction has deep theoretical significance and will play a vital role in domains such as autonomous driving, smart cities, robotic collaboration, and complex system management. Exploring multi-scale planning mechanisms with hierarchical large language model agents can open new paths for artificial intelligence and build a solid foundation for addressing more complex, uncertain environments in the future.

2. Related work

In recent years, planning and decision-making for agents have been an important research direction in artificial intelligence. Early approaches mainly focused on rule-based and search-based methods[9]. These methods relied on explicit logical rules or predefined search spaces and achieved planning through heuristic strategies. However, they often faced limitations in dynamic environments and large-scale tasks. They struggled to adapt to rapid changes in the environment or to meet the requirements of complex task decomposition. With the development of machine learning, researchers began to combine statistical modeling with planning. Data-driven approaches enabled more efficient decision processes. Yet, these methods still showed weaknesses in cross-scenario generalization and multi-level adaptability[10].

The rise of deep learning brought breakthroughs for agent planning. Methods based on sequence modeling and reinforcement learning allowed agents to learn strategies from large-scale data in dynamic environments. They also provided a certain degree of adaptability[11]. However, single-level deep models still showed structural limitations when facing multi-scale tasks. In complex tasks, it was difficult for the models to balance long-term goals with immediate feedback. This led to a lack of global perspective and flexibility in planning. To address this, hierarchical reinforcement learning and layered strategies became research hotspots. These approaches divided tasks and decisions across different levels to connect local and global goals. Still, they faced clear shortcomings in cross-task transfer and semantic-level understanding.

The emergence of large language models has moved agent research into a new stage. With strong abilities in semantic understanding, reasoning, and generation, agents can perform higher-level task abstraction and goal decomposition. This overcomes the limitations of traditional models in semantic representation and knowledge use. Recent studies have attempted to embed large language models into agent planning architectures. They support natural language interaction, environment modeling, and multi-task coordination. As a result, planning is no longer restricted to static strategies but can be dynamically adjusted through language-driven mechanisms. This combination provides new ideas for solving complex tasks in open environments. However, further exploration is still needed in hierarchical design and multi-scale modeling[12].

On this basis, multi-scale planning and hierarchical architectures have gradually become key research directions. One line of work focuses on how to use the semantic reasoning ability of large language models to abstract global goals at higher levels, perform task decomposition and resource scheduling at intermediate levels, and enable fast response and execution at lower levels. Another line of work emphasizes integrating environmental uncertainty with hierarchical planning to design agents that can adapt to changing conditions and diverse tasks. These studies indicate that hierarchical large language model agents not only have high theoretical research value but also demonstrate broad application potential in practice. They lay the foundation for deploying agents in complex and dynamic environments.

3. Proposed Approach

This study proposes a hierarchical large language model agent approach for dynamic environments. Its core concept is to integrate global goal modeling, task decomposition, and local execution to achieve multi-scale planning and dynamic adaptation. The overall model architecture is shown in Figure 1.

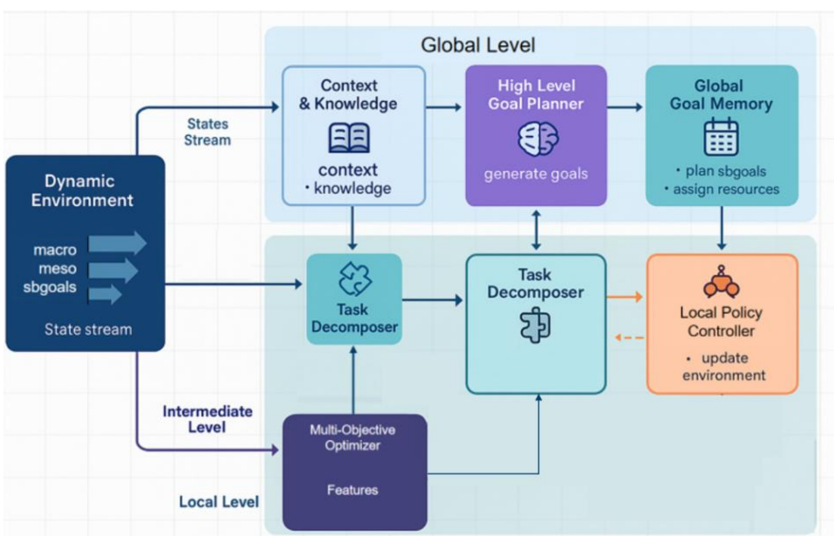

Figure 1. Overall model architecture

In the overall framework, the environment state is represented in time series form as:

$$S = \{s_1, s_2, \ldots, s_T\}, \quad s_t \in R^d$$

Where s_t represents the state of the environment at time step t, and d is the dimension of the state. The agent abstractly models the environment information at a high level through a large language model and generates a global target vector G, which is represented as:

$$G = f_{LLM}(S, C)$$

Where C is the contextual knowledge and f_{LLM} represents the semantic reasoning function of the language model. This step ensures that the global planning can adapt to the complexity and uncertainty in the dynamic environment.

After the goal is generated, the model maps the global goal into a set of sub-goal sequences $\{g_1, g_2, ..., g_K\}$ through a hierarchical task decomposition mechanism, which can be specifically expressed as:

$$g_k = f_{decompose}(G, S, k), \quad k = 1, ..., K$$

Where A is the task decomposition function, which ensures that each sub-goal meets the global constraints and can be effectively executed in the local environment. Subsequently, the agent makes action decisions based on the local state through the policy network, and the action set is expressed as:

$$a_t = \pi_\theta(s_t, g_k)$$

Where π_θ represents the policy function with parameter θ . This mechanism ensures the dynamic adaptability of the local level and can be updated as the environment changes.

To achieve effective coordination between global and local, the model designs a hierarchical optimization objective. First, define the global objective function:

$$L_{global} = \sum_{k=1}^{K} \| g_k - \hat{g}_k \|^2$$

Where $\hat{g}_k$ is the optimal sub-goal reference representation obtained by language model reasoning. At the same time, the optimization goal of the local execution layer is:

$$L_{local} = \sum_{t=1}^{T} l(a_t, a_t^*)$$

Where a_t^* is the reference optimal action in the current state, and $l(\cdot)$ is the metric function for action matching. The final joint optimization goal is:

$$L = \lambda_1 L_{global} + \lambda_2 L_{local}$$

Where λ_1, λ_2 is a weight coefficient that balances the importance of global planning and local execution. Through this hierarchical optimization mechanism, the intelligent agent can achieve global and local coordination in dynamic environments and effectively adapt to multi-scale task requirements.

4. Performance Evaluation

4.1 Dataset

The dataset used in this study is the MineRL Dataset. It is specifically designed for agent research and has been widely applied in reinforcement learning and planning tasks. The dataset is collected from task execution in open-world environments and contains diverse states, actions, and reward signals. Its main feature is the simulation of complex and dynamic settings, which provides rich support for studying multi-scale planning and hierarchical decision-making. The data include both local immediate feedback and long-term goal-directed trajectories, making it suitable for evaluating hierarchical reasoning and planning in dynamic environments.

The design of the dataset emphasizes multi-task and multi-scenario characteristics. It covers a wide range of tasks, from basic behaviors such as movement and collection to higher-level goals such as construction and resource management. This diversity enables agents to adapt and transfer across different levels of objectives. It not only enhances the representativeness of the dataset but also creates a solid basis for exploring hierarchical planning mechanisms. In dynamic environments, agents must balance short-term feedback with long-term returns. The dataset provides cross-scale data support, which allows researchers to analyze agent behavior at different levels within a unified framework.

In addition, the openness and scalability of the dataset bring further significance. It provides complete state-action-reward sequences and allows new sub-tasks or composite tasks to be built on top of the original settings. This flexibility facilitates the study of generalization in multi-scale planning. It also makes the dataset suitable for testing and comparing a wide range of agent algorithms. In particular, it is well-suited for exploring the adaptability and robustness of hierarchical large language model agents in dynamic environments. For these reasons, this study adopts the dataset as the foundation. It not only carries practical relevance but also ensures that the results can serve as a reference for broader research on agents in dynamic settings.

4.2 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	AR	NS	GDA
Agent q[13]	68.3	215	0.62	71.4
Behavioral formation[14]	72.5	238	0.66	74.8
Turbo-IRL[15]	75.1	254	0.71	77.3
Madiff[16]	77.9	269	0.75	80.2
OURS	84.6	298	0.83	86.7

From the results in Table 1, it can be seen that when performing multi-scale planning in dynamic environments, different methods show significant differences in success rate and average reward. Traditional Agent q and Behavioral formation show some improvement in overall performance. However, due to the lack of hierarchical task decomposition, their goal achievement in complex scenarios remains limited. In contrast, Turbo-IRL and Madiff combine reinforcement learning with imitation learning. They improve stability and efficiency, but still struggle to maintain advantages in long-term planning and high-level semantic reasoning.

Further observation of the Normalized Score shows that the normalized performance follows the same trend as cumulative rewards. This indicates that as the modeling ability of agents improves, their adaptability to environmental changes increases. However, relying only on low-level strategy optimization is not enough to ensure the stable achievement of global goals. Madiff performs relatively well on this metric, suggesting that it can capture cross-scale task features to some extent. Yet, it cannot model complex dynamic environments from a global perspective.

On the GDA metric, OURS shows a clear advantage over other methods. Its goal decomposition accuracy reaches 86.7 percent, which indicates that the method can decompose global goals into reasonable sub-goals more effectively and maintain high consistency during execution. This advantage mainly comes from the semantic understanding and reasoning ability of hierarchical large language models. They enable agents to integrate environmental information at higher levels and make dynamic adjustments during low-level execution, thus achieving more efficient task completion.

Considering all metrics, OURS outperforms the baselines in success rate, average reward, normalized performance, and goal decomposition accuracy. This verifies the effectiveness of hierarchical large language models in multi-scale planning under dynamic environments. It reflects their advantage in coordinating between global and local objectives and highlights their adaptability and robustness in handling complex environmental changes. The results demonstrate that hierarchical agents based on large language models can provide stronger support for multi-task and multi-scenario planning and hold significant value for advancing agent applications in real-world dynamic environments.

This paper further presents a sensitivity experiment on the robustness of the environmental noise intensity, and the experimental results are shown in Figure 2.

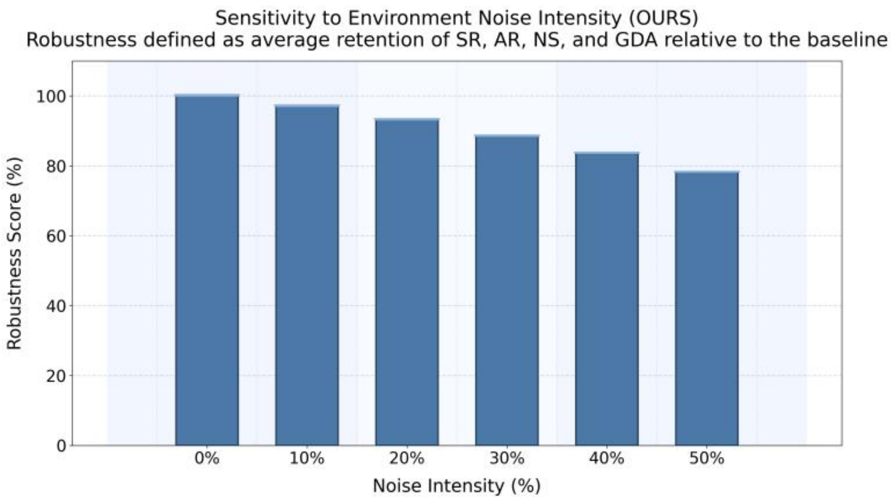

Figure 2. Sensitivity experiment of environmental noise intensity on robustness

From Figure 2, it can be seen that as the intensity of environmental noise gradually increases, the robustness score of the agent shows a clear downward trend. Under 0 percent noise, the agent can fully maintain baseline performance, indicating stable task planning and execution in the absence of interference. However, when noise intensity reaches 10 percent and 20 percent, robustness remains at a high level but already shows a moderate decline. This suggests that environmental disturbances begin to influence the execution of multi-scale planning.

When the noise intensity rises further to 30 percent, the decline in robustness becomes more pronounced. This highlights the sensitivity of perception and task decomposition to external interference in dynamic environments. The trend shows that although hierarchical large language models have advantages in high-level goal planning, they remain vulnerable in low-level action execution and environmental feedback. The result indicates that multi-scale architectures require stronger anti-noise mechanisms between task decomposition and execution to ensure stable performance in complex environments.

At 40 percent and 50 percent noise, the robustness score falls clearly below the baseline. This means that in highly uncertain and strongly disturbed environments, the overall planning efficiency and goal decomposition accuracy of the agent face severe challenges. This finding highlights a bottleneck of multi-scale agents in dynamic contexts, namely, how to maintain stable decision-making at both global and local levels to prevent rapid performance degradation.

Overall, the results confirm the significant impact of environmental noise on the robustness of hierarchical large language model agents. Although the agent performs well under low-noise conditions, there is a clear decline under high-noise scenarios. Enhancing resilience to disturbances through stronger redundancy mechanisms, dynamic feedback adjustment, and adversarial training is an important direction for future research. This finding not only reveals the strengths and limitations of current methods but also provides practical insights for achieving more advanced intelligent planning in complex environments.

This paper also presents a sensitivity experiment on the optimizer type, and the experimental results are shown in Figure 3.

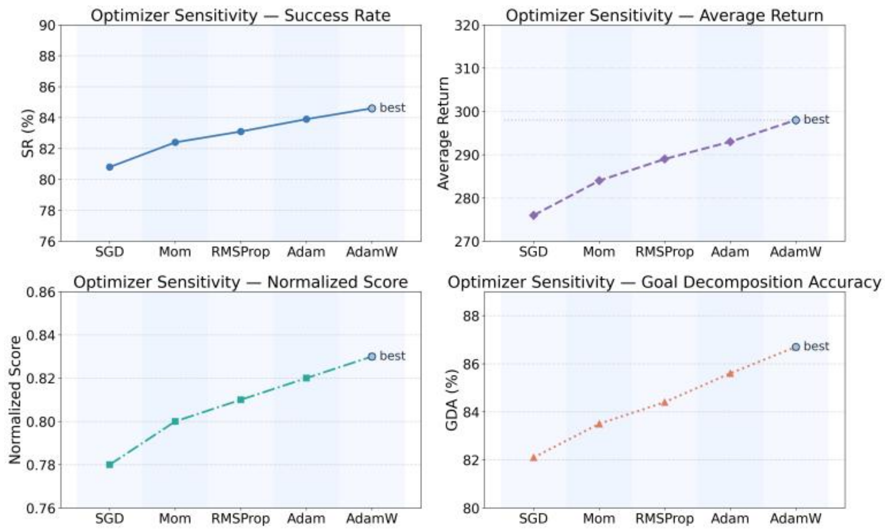

Figure 3. Sensitivity experiments on optimizer types

From the results in Figure 3, it can be seen that different types of optimizers have a significant impact on agent performance. The overall trend shows that as the optimizer evolves from traditional SGD to AdamW, the agent achieves gradual improvement across all metrics. This trend indicates that the optimizer plays a critical role in training hierarchical large language model agents. It directly affects the stability and coordination between global planning and local execution.

In terms of success rate, optimizers such as SGD and Momentum can maintain a certain level of task completion. However, their simple update mechanisms make it difficult to fully exploit the advantages of multi-scale planning in complex and dynamic environments. With the introduction of RMSProp and Adam, the success rate improves. This reflects that more flexible gradient adaptation mechanisms help enhance the adaptability of agents in dynamic scenarios. Finally, AdamW achieves the highest success rate, showing its advantage in balancing regularization and weight updates.

The changes in average reward and normalized score further support this conclusion. As the optimizer evolves, the agent shows stronger stability and robustness in reward accumulation and overall performance. The results of Adam and AdamW are particularly notable. They demonstrate that adaptive optimizers can significantly improve training efficiency and planning ability when handling high-dimensional parameters and complex task structures. This allows the model to maintain superior performance in dynamic environments.

For goal decomposition accuracy, the results also highlight the importance of optimizer choice. SGD performs relatively poorly on this metric, showing its limitations in supporting complex hierarchical task decomposition. AdamW performs best, indicating that it can better preserve consistency between global goals and sub-goals. Overall, these findings emphasize the crucial role of optimizer selection in multi-scale planning for hierarchical large language model agents in dynamic environments. They provide a strong reference for designing more efficient training strategies.

This paper also presents a sensitivity experiment on the planning update frequency, and the experimental results are shown in Figure 4.

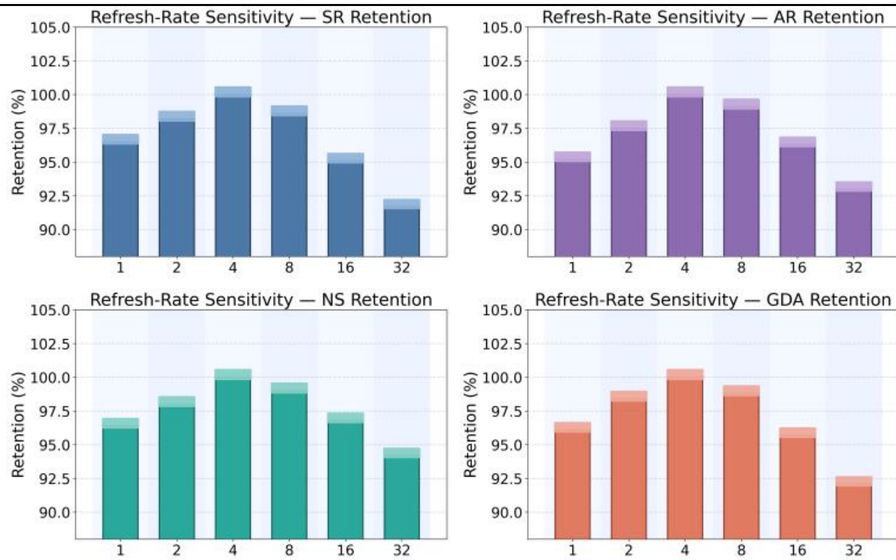

Figure 4. Sensitivity experiment on planning update frequency

From the results in Figure 4, it can be seen that planning update frequency has a clear impact on agent performance. Across the four evaluation metrics, the results show a trend of rising first and then falling. This indicates that both very low and very high update frequencies lead to performance decline, while a moderate update frequency helps maintain stability and effectiveness in dynamic environments. This phenomenon shows that hierarchical large language model agents need proper rhythm control between planning and execution. It avoids excessive reliance on short-term adjustments or overly slow global updates.

In the sensitivity analysis of success rate and average reward, performance peaks when the update frequency is set to four steps. Overly frequent updates, such as one step or two steps, cause overfitting to local states and make it hard to maintain global planning consistency. Sparse updates, such as sixteen steps or thirty-two steps, weaken the ability of the model to respond quickly to changes in dynamic environments. This shows that a reasonable update frequency can balance global planning and local execution, improving overall task performance.

The results of the normalized score also confirm this trend. A moderate update frequency not only enhances the ability to maintain global performance but also helps the agent adapt better to changes in dynamic environments. In contrast, low update frequencies cause information lag, leading to loss of flexibility, while high update frequencies increase the risk of over-adjustment and reduce stability. This shows that update frequency is critical to maintaining consistency in multi-scale planning.

The results of goal decomposition accuracy further emphasize the sensitivity to planning update frequency. When updates are too frequent, the stability of sub-goals is affected, leading to reduced decomposition accuracy. When updates are too sparse, deviations between sub-goals and environmental states accumulate, also reducing accuracy. Overall, the proper setting of update frequency is not only about coordinating global and local levels but is also key to ensuring efficient and robust planning in dynamic environments. These results provide an important reference for optimizing agent algorithms.

This paper also presents an experiment on the sensitivity of the frequency and amplitude of environmental sudden changes (dynamics shift) to recovery capacity, and the experimental results are shown in Figure 5.

From Figure 5, it can be seen that the frequency of environmental mutation events has a significant impact on the recovery ability of the agent. When the mutation frequency is low, the agent can return to near-baseline performance within a short time. This shows that hierarchical large language model agents have strong adaptability when facing occasional disturbances. However, as the frequency of mutation events increases, recovery ability declines markedly. This indicates that frequent environmental changes disrupt the rhythm of multi-scale planning, making it difficult to maintain stable coordination between global and local decisions.

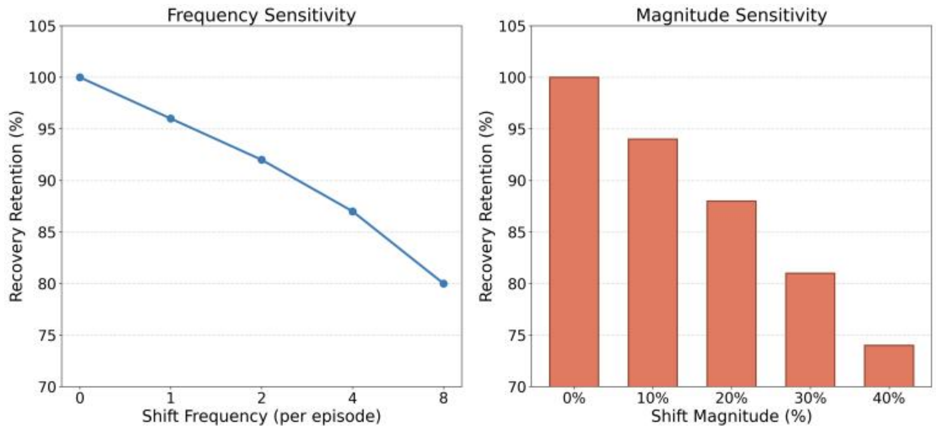

Figure 5. Experimental study on the sensitivity of the frequency and magnitude of environmental dynamics shift to recovery capacity

In the amplitude sensitivity experiment, it is observed that larger mutation amplitudes lead to weaker recovery ability. When the amplitude is only 10 percent, the agent still maintains high robustness, showing that its internal hierarchical mechanisms provide good fault tolerance under mild disturbances. But when the amplitude rises to 20 percent and above, recovery ability decreases significantly. At 40 percent, performance drops to a low level, highlighting the difficulty of recovery under drastic dynamic changes.

The results of frequency and amplitude experiments together show that the robustness of the agent depends not only on its planning ability in static environments but also on its capacity to resist and adapt to uncertainty in dynamic environments. Frequent disturbances prevent the model from maintaining stable policy updates. Large-scale environmental changes cause mismatches between high-level goals and low-level execution, leading to an imbalance in task decomposition and planning. This reveals the importance of planning updates and feedback mechanisms in dynamic settings.

Overall, the experiments confirm that hierarchical large language model agents have some recovery ability in dynamic environments, but they still show significant performance degradation when mutation events are frequent and large in scale. Enhancing stability under high-frequency and high-amplitude mutations requires stronger redundancy mechanisms, improved memory units, or adaptive scheduling strategies. This finding is important for improving the robustness of agents in real complex environments.

This paper also presents an experiment on the sensitivity of learning rate and weight decay to the stability of hierarchical strategies. The experimental results are shown in Figure 6.

From the results in Figure 6, it can be seen that the learning rate has a clear impact on the stability of hierarchical strategies. When the learning rate is too low, the stability and retention of the agent are reduced. This shows that small update steps make it difficult for the model to adapt quickly in dynamic environments. As the learning rate increases to a suitable range, stability improves significantly. It reaches the best level near 3×10^{-4}, indicating that this setting allows sufficient exploration while avoiding training oscillation. As a result, it achieves better hierarchical planning performance.

When the learning rate continues to increase to 10^{-3}, stability decreases markedly. This shows that large update steps easily disrupt the balance within the hierarchical structure, causing misalignment between high-level goals and low-level execution. This finding indicates that the learning rate not only determines convergence speed but also directly affects robustness and consistency in multi-scale tasks. Therefore, a reasonable setting of the learning rate is essential to maintain the stability of hierarchical agents.

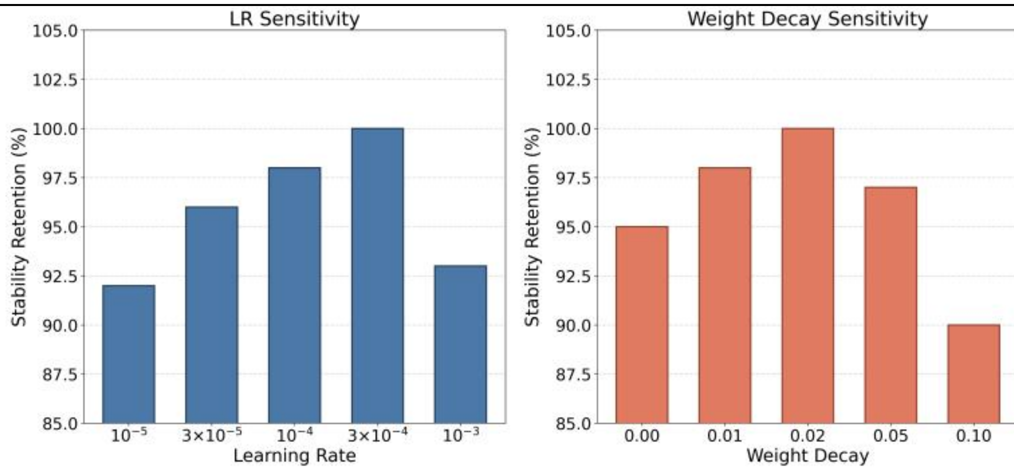

Figure 6. Sensitivity experiments of learning rate and weight decay on the stability of hierarchical strategies

In the weight decay experiment, it is observed that moderate regularization improves stability. When weight decay is 0.00, stability retention is relatively low. Performance peaks near 0.02, showing that moderate weight constraints help prevent overfitting in dynamic environments. This ensures consistency between global and local planning. The results indicate that regularization mechanisms not only improve generalization in hierarchical structures but also enhance robustness under disturbances.

However, when weight decay becomes too large, such as 0.10, stability declines sharply. This shows that excessive regularization suppresses effective representation of complex environmental features, leading to reduced coordination between task decomposition and execution. Overall, the sensitivity experiments on learning rate and weight decay highlight the importance of optimization hyperparameters in training hierarchical large language model agents. Only when these parameters are set within a proper range can the model maintain long-term stability and effectiveness in dynamic environments.

5. Conclusion

This study addresses the problem of multi-scale planning in dynamic environments and proposes a hierarchical large language model agent framework. By combining global goal setting, task decomposition, and local execution, the method improves adaptability and robustness in complex environments. Unlike traditional single-layer strategies or models that rely only on low-level execution, this study demonstrates the advantages of a hierarchical architecture in semantic understanding, long-term reasoning, and short-term response. As a result, the agent maintains high stability and task completion under uncertain and dynamic conditions.

Experimental results show that the framework achieves strong stability and recovery ability in multiple sensitivity scenarios. Whether under environmental noise, optimizer choice, update frequency, or the frequency and amplitude of environmental mutations, the proposed method demonstrates clear advantages in robustness and multi-scale planning consistency. This indicates that hierarchical large language model agents can resist disturbances while maintaining coherence and accuracy in task execution across diverse conditions. Sensitivity experiments on different hyperparameters further confirm the key considerations in design and tuning, providing valuable practical insights for future research.

From an application perspective, the findings have potential impact in several domains. In autonomous driving and intelligent traffic scheduling, hierarchical large language models can better coordinate global path planning with local obstacle avoidance, improving safety and flow. In intelligent manufacturing and robotic collaboration, task decomposition and recovery abilities allow efficient operation in complex production processes. In smart cities and public safety management, multi-scale planning mechanisms in dynamic environments support resource optimization and emergency response, providing reliable support for large-scale intelligent systems.

In conclusion, this study enriches the theoretical research on combining hierarchical agents with large language models and demonstrates their feasibility and superiority in dynamic environments. By introducing multi-level planning mechanisms and sensitivity analysis methods, the work lays a foundation for improving the generalization, stability, and practical value of agents. These contributions provide solid support for deploying agents in complex open environments and create a positive impact for the application of intelligent decision systems in critical domains.

References

[1] Li Z, Chang Y, Yu G, et al. HiPlan: Hierarchical Planning for LLM-Based Agents with Adaptive Global-Local Guidance[J]. arXiv preprint arXiv:2508.19076, 2025.

[2] Ming C, Lin J, Fong P, et al. HiCRISP: An LLM-Based Hierarchical Closed-Loop Robotic Intelligent Self-Correction Planner[C]//2024 China Automation Congress (CAC). IEEE, 2024: 4310-4315.

[3] Choi J W, Kim H, Ong H, et al. Reactree: Hierarchical task planning with dynamic tree expansion using llm agent nodes[J]. 2025.

[4] Li L, Tan R, Fang J, et al. LLM-augmented hierarchical reinforcement learning for human-like decision-making of autonomous driving[J]. Expert Systems with Applications, 2025, 294: 128736.

[5] Zhai W, Liao J, Chen Z, et al. A Survey of Task Planning with Large Language Models[J]. Intelligent Computing, 2025, 4: 0124.

[6] Schakkal A, Zandonati B, Yang Z, et al. Hierarchical Vision-Language Planning for Multi-Step Humanoid Manipulation[J]. arXiv preprint arXiv:2506.22827, 2025.

[7] La N, Mon-Williams R, Shah J A. HDDLGym: A Tool for Studying Multi-Agent Hierarchical Problems Defined in HDDL with OpenAI Gym[J]. arXiv preprint arXiv:2505.22597, 2025.

[8] Huang Q, Luo Y, Liu Z, et al. A Hierarchical Reinforcement Learning Framework for Multi-Agent Cooperative Maneuver Interception in Dynamic Environments[J]. Journal of Marine Science and Engineering, 2025, 13(7): 1271.

[9] Xu C, Zhang C, Shi Y, et al. Subgoal-based Hierarchical Reinforcement Learning for Multi-Agent Collaboration[J]. arXiv preprint arXiv:2408.11416, 2024.

[10]Cavrel N, Fiorino H, Pellier D. Extending Hierarchical Partial-Order Causal-Link Planning to Temporal Problem Solving[C]//2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, 2024: 73-81.

[11]Yang F, Liu J, Li K. LLM-Guided Reinforcement Learning for Interactive Environments[J]. Mathematics, 2025, 13(12): 1932.

[12]Cornelio C, Petruzzellis F, Lio P. Hierarchical Planning for Complex Tasks with Knowledge Graph-RAG and Symbolic Verification[J]. arXiv preprint arXiv:2504.04578, 2025.

[13]Putta P, Mills E, Garg N, et al. Agent q: Advanced reasoning and learning for autonomous ai agents[J]. arXiv preprint arXiv:2408.07199, 2024.

[14]Snooks R. Behavioral formation: multi-agent algorithmic design strategies[D]. RMIT University, 2024.

[15]Mehrabi N, Boroujeni S P H, Razi A. Turbo-IRL: Enhancing multi-agent systems using turbo decoding-inspired deep maximum entropy inverse reinforcement learning[J]. Expert Systems with Applications, 2026, 296: 128754.

[16]Zhu Z, Liu M, Mao L, et al. Madiff: Offline multi-agent learning with diffusion models[J]. Advances in Neural Information Processing Systems, 2024, 37: 4177-4206.