FD-Bench: Fine-Grained Evaluating the Decision-Making Capability of LLM Agents in Dynamic Scenarios

Zhihao Zhu, Yifan Zheng, Yaohui Jin*
{zzh2021,yifanzheng,jinyh}@sjtu.edu.cn
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University, Shanghai, China

Abstract

Large language models(LLMs) exhibit growing potential as autonomous agents, yet their decision-making capabilities in real-world scenarios remain underexplored, particularly in dynamic scenarios where conditions are constantly changing. Most existing benchmarks mainly focus on static environments, which significantly differ from real-world scenarios. Additionally, existing evaluation frameworks lack fine-grained assessments, providing limited insights during evaluation. To address these, we propose `FD-Bench` a benchmark for evaluating the decision-making in dynamic scenarios. `FD-Bench` employs a fire evacuation scenario as a representative dynamic setting and decomposes decision-making into perception, prediction, and action stages, enabling granular evaluation of 8 LLMs and different reasoning frameworks. Our results show that LLMs experience a performance drop of over 50% in dynamic versus static scenarios. Inspired by "chunking" principle in Cognitive Load Theory (CLT), our hierarchical prompting strategy demonstrates improved performance in dynamic decision-making tasks. This work provides insights into LLMs' limitations and pathways toward robust real-world deployment.

Keywords: Large language models; Agents; Decision-making; Benchmark; Dynamic environment

Introduction

Recent advancements in LLMs have shown their potential as autonomous agents (L. Wang et al., 2024; Shinn, Labash, & Gopinath, 2023), enabling them to interact with environments autonomously and complete complex tasks across various domains, such as embodied intelligence (Song et al., 2023), tool agents (Qin et al., 2023), and web agents (S. Zhou et al., 2023). These works highlight the potential of LLM agents in solving real-world problems.

However, there is a significant discrepancy between existing tasks and real-world tasks, primarily because the environments used for interaction are generally static and only exhibiting responsive changes. In contrast, real-world environments are characterized by **spontaneous changes over time or space**, such as fire spread and day-night cycles and these changes can be irregular or follow certain patterns. These challenges resonate with Cognitive Load Theory (CLT) (Sweller, 1988) in human cognition, where dynamic factors introduce cognitive load by requiring LLM agents to perceive changes, infer the underlying dynamic mechanisms, and make decisions based on predictions of the changing environment. This certainly poses significant challenges to the temporal-spatial reasoning and decision-making abilities of LLMs, like the example shown in Figure1. Indeed, our experiments demonstrate that the presence of dynamic factors in the environment significantly diminishes the performance of LLM agents, which validates our initial hypothesis.

To explore the potential applications of LLM agents in real-world scenarios, it is essential to gain a comprehensive understanding of their decision-making capabilities in dynamic environments. Our work aims to address the following questions:

1. *Can LLM agents identify dynamic factors in the environment and understand their patterns or behaviors?*
2. *Can LLM agents predict and model the states of dynamic scenarios over multiple steps?*
3. *Can LLM agents make correct decisions in dynamic scenarios based on predictions?*

To address these, we propose a novel method `FD-Bench` (**F**ine-Grained **D**ynamic Decision **Bench**mark), a progress-based evaluation benchmark for LLM's decision-making capabilities. We constructed a dynamic grid-based fire spread scenario, with the objective of safely escaping from a room on fire. Differing from holistic evaluations in Table 1, we decompose the LLM's decision-making process in dynamic scenarios into three sub-tasks based on the basic ability of LLM: perception of dynamic elements and rules, prediction and modeling of environmental states and decision-making based on predictions. Notably, we design specific evaluation metrics and prompts for each sub-task. Overall, `FD-Bench` provides a total of 500 test scenarios and 3,000 prompts as part of our dataset. Additionally, `FD-Bench` adopts a hybrid evaluation approach that combines LLM-based and metric-based methods to efficiently and accurately evaluate spatial and temporal decision-making abilities.

Through extensive experiments with `FD-Bench`, we conduct an in-depth analysis of the performance of 8 LLMs in dynamic environments, and drive a clear perspective on the performance of LLM agents in dynamic environments: (1) The performance of LLMs significantly declines in dynamic scenarios compared to static scenarios. (2) GPT-4 and GPT-4o-mini perform far better than other models, and other open-source and closed-source models show mixed results across

*Corresponding author.

In D. Barner, N.R. Bramley, A. Ruggeri and C.M. Walker (Eds.), *Proceedings of the 47th Annual Conference of the Cognitive Science Society*

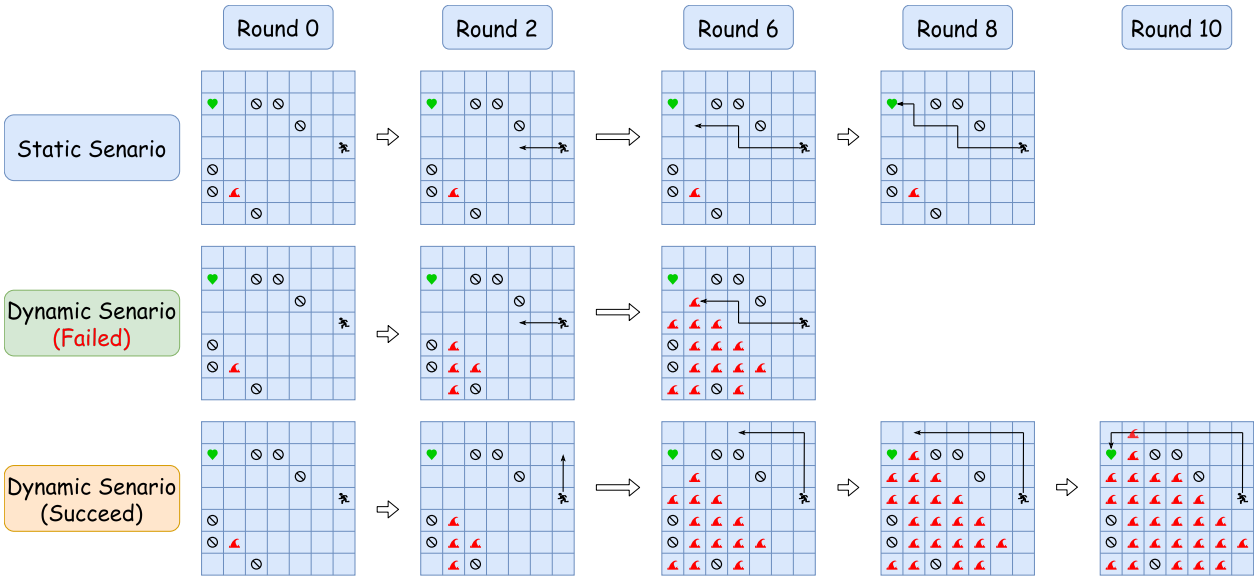

Figure 1: This is a case of scenarios involving dynamic elements that influence decision-making. In the figure, represents the starting point, represents the target point, and represents walls or obstacles. Additionally, there is a spreading , which expands by one grid every two rounds. In the presence of only static elements, the shortest path length is **8**. However, when dynamic elements such as fire are introduced, the previously optimal path becomes unsafe. Therefore, it is essential for large language models to anticipate this situation and select an alternative route in advance, resulting in a new shortest path length of **10**.

different tasks. (3) The main limitations of LLMs lie in the recognition and prediction of dynamic elements with complex patterns, as well as in decision-making based on multi-step predictions. (4) Inspired by the "chunking" principle in CLT, we introduce a hierarchical prompting strategy that reduces cognitive load, which helps improve decision-making processes in dynamic scenarios.

Additionally, the consistency between the performance of individual tasks and end-to-end decision-making validates the rationality of our framework's decision process segmentation. Through `FD-Bench`, we offer a new perspective on defining the capability boundaries of LLMs in dynamic environments. We expect that `FD-Bench` will provide valuable insights for LLM application to real-world environments and advance further development of LLM agents.

Related Work

Decision Making in Dynamic Environments Dynamic environments are characterized by stochasticity and spontaneous change. These environments pose significant and distinct challenges for decision-making (Padakandla, 2021), necessitating that agents detect changes in the environment and adjust their decisions accordingly. Evacuation during fire emergencies is one of the scenarios used to study the decision-making capabilities of agent-based model (ABM) in complex and dynamic environment(Kasereka et al., 2018). To cope with dynamic environments, various methods have been proposed to improve decision-making in dynamic environments (Padakandla, 2021; Zhao et al., 2022; Hu et al., 2022), but traditional approaches often suffer from high exploration costs, limited adaptability, slow convergence, and dependence on expert data (Hospedales, Antoniou, Micaelli, & Storkey, 2021).

LLM agents have recently demonstrated general solutions across diverse tasks, including web browsing (S. Zhou et al., 2023; Yao, Chen, Yang, & Narasimhan, 2022), tool utilization (Qin et al., 2023), and embodied tasks (Shridhar et al., 2021; R. Wang, Jansen, Côté, & Ammanabrolu, 2022; Fan et al., 2022), due to their excellent reasoning (Wei et al., 2022) and zero-shot generalization (Yao, Zhao, et al., 2023). However, most existing works focus on task-specific evaluations, **overlooking the challenges posed by dynamic environments**, as illustrated in Table 1

Recent work by (Q. Zhou et al., 2024b) explores LLM agents' performance in search and rescue (SAR) tasks under dynamic conditions such as fires, floods, and high winds, comparing them with traditional models. The results show that LLMs, utilizing their extensive world knowledge and decision-making capabilities, outperform most baselines. However, the impact of dynamic factors on LLM agents remains underexplored. To better understand the boundaries of LLM agents' decision-making abilities in dynamic environments, we conduct more detailed evaluations.

Fine-grained evaluation of LLM Evaluating LLMs is essential for understanding their capability boundaries and potential applications. Many benchmarks have been established involving reasoning (Valmeekam, Olmo, Sreedharan, & Kambhampati, 2022), code generation (Jimenez et al., 2023; J. Liu,

Table 1: Benchmark comparison. *AgentBench and AgentBoard include scenarios like web browsing where spontaneous changes exist, but these works focus not on decision-making in dynamic scenarios but rather on the general decision-making capabilities of LLMs as agents.

Benchmark	Supports LLM	Spontaneous Environment Changes	Fine-Grained Metrics	Dynamic-Focused Analysis
ALFRED (Shridhar et al., 2020)	✗	✗	✗	✗
VirtualHome (Puig et al., 2018)	✗	✗	✔	✗
FCMs (Nachazel, 2021)	✗	✔	✗	✗
ViZDoom (Kempka et al., 2016)	✗	✔	✔	✗
D4RL (Fu et al., 2021)	✗	✔	✔	✗
FinRL (X.-Y. Liu et al., 2023)	✗	✔	✔	✗
AgentBench (X. Liu et al., 2023)	✔	✗*	✗	✗
WebShop (Yao, Chen, et al., 2023)	✔	✗	✔	✗
PlanBench (Valmeekam et al., 2023)	✔	✗	✔	✗
Auto-gpt (Yang et al., 2023)	✔	✗	✔	✗
AgentBoard (Ma et al., 2024a)	✔	✗*	✔	✗
VOYAGER(G. Wang et al., 2023)	✔	✔	✗	✗
WebArena (S. Zhou et al., 2024)	✔	✔	✗	✗
HAZARD (Q. Zhou et al., 2024a)	✔	✔	✗	✗
FD-Bench	✔	✔	✔	✔

Xia, Wang, & Zhang, 2024), and agent tasks (Qin et al., 2023; S. Zhou et al., 2023).

Compared to most existing benchmarks that use success rates as the primary metric, fine-grained evaluations often provide more detailed information, especially when LLMs perform similarly bad across challenging tasks. (Ma et al., 2024b) introduces a progress rate to assess LLMs as general agents, offering deeper insights into the performance differences between various LLMs. Similarly, (Chen et al., 2023) decomposes the pipeline of tool use into several key tasks, minimizing the impact of external factors. These works utilize reasonable annotations or capability decomposition to advance the understanding and insights of LLMs' boundaries in specific tasks and scenarios.

In this work, we propose a fine-grained evaluation framework based on the decomposition of the decision-making process. This framework aims to provide more precise insights into the capability boundaries of LLMs in dynamic environments, further advancing the development of LLM agents.

FD-Bench - Overview

`FD-Bench` aims to provide a detailed and fine-grained evaluation of LLM agents' decision-making abilities in dynamic environments and explore their decision-making boundaries. `FD-Bench` distinguishes itself through three key features:

- *Fine-grained Evaluation Based on Capability Decomposition:* `FD-Bench` employs a fine-grained evaluation method by decomposing the complex decision-making process into multiple sub-tasks. Additionally, well-designed evaluation metrics are provided for each sub-task, enabling a more detailed assessment of individual capabilities.

- *Generalizability of the Evaluation Framework:* The modular design of `FD-Bench` allows for the integration of different environmental conditions and decision-making challenges, enabling its adaptation to other dynamic environments.

- *Focus on the Impact of Dynamic Factors on Decision-Making:* Each test scenario includes a random number of static and dynamic factors that do not affect decision-making. By including these random factors, `FD-Bench` enhances evaluation robustness and provides a more detailed and realistic assessment of LLM agents' decision-making in dynamic environments.

Preliminaries

In our evaluation of agents in dynamic environments, we extend Markov Decision Processes (MDPs) to accommodate environments where state transitions are influenced by both the agent's actions and the dynamics of the environment itself. To model these interactions, we consider a dynamic variant of MDPs defined by the tuple $\langle g, S, A, \mathcal{T}, \mathcal{D} \rangle$, where g is the goal, S is the state space, A is the valid actions space, $\mathcal{T} : S \times A \times \mathcal{D} \to S$ is the transition function considering both the agent's actions and the environment's dynamics, $\mathcal{D}$ is the set of dynamic rules of the environment.

An agent with policy π makes a prediction at time step t based on goal g, current state s_t, memory m_t, and predicted future states $\hat{s}_{t+1}, \ldots, \hat{s}_{t+k}$, where:

$$m_t = \{o_j, a_j, o_{j+1}, a_{j+1}, \ldots, o_t\}, \quad 0 \leq j < t, \tag{1}$$

which is a sequence of actions and observations. The trajectory of the agent τ is formulated by policy and environmental state transitions, such as:

$$p_\pi(\tau) = p(s_0) \prod_{t=0}^{T} \pi(a_t \mid g, s_t, m_t, \hat{s}_{t+1}, \ldots, \hat{s}_{t+k}) \\ \mathcal{T}(s_{t+1} \mid s_t, a_t, d_t) \tag{2}$$

Where $p(s_0)$ is the initial state distribution, $\pi(a_t \mid g, s_t, m_t, \hat{s}_{t+1}, \ldots, \hat{s}_{t+k})$ is the policy, representing the probability of choosing action a_t given the goal g, current state

s_t, memory m_t, and predicted future states, $\mathcal{T}(s_{t+1} \mid s_t, a_t, d_t)$ is the comprehensive transition function combining both the environment's spontaneous changes and the agent's actions, formulated as:

$$s_{t+1} = \mathcal{T}(s_t, a_t, d_t) = \mathcal{T}_2(\mathcal{T}_1(s_t, d_t), a_t) \tag{3}$$

where $\mathcal{T}_1 : S \times \mathcal{D} \to S$ represents the environment's intrinsic dynamics and $\mathcal{T}_2 : S \times A \to S$ represents the response to the agent's actions.

Dynamic Environment Setting

In the fire spread scenario, both static factors (**SF**) and dynamic factors (**DF**) are set to evaluate the decision-making capabilities of LLM agents in dynamic environments.

Static Factors (**SF**): These factors remain unchanged throughout the scenario but may still influence the decision-making process, such as physical obstacles and available exits. These factors establish a stable and consistent foundation for the environment.

Dynamic Factors (**DF**): These factors change over time and have a direct impact on the decision-making process, such as the spread of fire and the movement of obstacles. For each dynamic factor $i = 0, 1, \ldots, n$, in each environment $j = 0, 1, \ldots, m$ they have corresponding rules $DF_{i,j}$. For example, the rule for fire spread is to expand one block in all directions every second.

To ensure the stability and reliability of the tests, we also randomly added some static and dynamic factors that do not affect decision-making in the environment, such as accumulated dust and scurrying mice. These settings increase the variability and complexity of the environment, evaluating the agent's ability to maintain stable decisions in the face of irrelevant changes. Furthermore, each dynamic environmental factor has multiple rules, but under each task, the rule $DF_{i,j}$ for the dynamic factor is fixed.

Fine-grained Evaluation Protocols

Capability Decomposition In the real scenario, decision-making by LLM agents in dynamic environments encompasses multiple dimensions of capabilities. To gain a deeper understanding the boundaries and limitations of LLM agents' decision-making abilities in dynamic environments, we deconstruct the decision-making process of LLM agents in such environments into the following three key aspects:

Stage 1. Perception of Environmental Dynamics and Modeling of Dynamic Rules

Formal Representation: Given the sequence of initial states to state at time t, $s_{0:t}$ and the task goal g, the LLM agent is required to determine whether there are dynamic factors d_i (belonging to **D**) in the environment and distinguish which dynamic factors d_i might affect decision-making and the corresponding dynamic rules $\hat{d}_t$. Specifically, the agent uses the perception function f_{percept} to estimate the dynamic change rule $\hat{d}_t$:

$$\hat{d}_t = f_{\text{percept}}(s_{0:t}, g) \tag{4}$$

Evaluation Metrics:

- *Recognition Accuracy (%)*: Calculate the F1-score between the set of dynamic elements recognized by the LLM and the gold answer.
- *Decision-Relevant Recognition Accuracy (%)*: Calculate the F1-score between the set of decision-relevant dynamic elements recognized by LLM and the gold answer.
- *Dynamic Rule Induction Accuracy (%)*: Evaluate the consistency between the dynamic transition rules inferred by the LLM and the actual rules, which is evaluated through CLAUDE3-SONNECT.

Stage 2. Environmental Modeling and Prediction

Formal Representation: Given the rule of dynamic factor changes d_t and the sequence of initial states to state at time t, $s_{0:t}$, the LLM agent is required to predict and model the future state of the scenario for k steps. Specifically, the scenario prediction and modeling ability f_{predict} of the LLM agent is represented as:

$$\hat{s}_{t+1}, \ldots, \hat{s}_{t+k} = f_{\text{predict}}(s_{0:t}, d_t) \tag{5}$$

Evaluation Metrics:

- *State Prediction Accuracy (%)*: Calculate the similarity between the predicted scenario sequence $\hat{S}_{t \ldots t+k_{\text{pred}}}$ and the gold answer $S_{t \ldots t+k_{\text{answer}}}$, with k set to 3 and 6.

Stage 3. Action Decision-Making Based on Predictions

Formal Representation: Given the goal g, the sequence of initial states to state at time t, $s_{0:t}$, the memory at time t, and the n-step prediction sequence of the environment $\hat{s}_{t+n}$ (where n is set to the number of steps required for the fire to reach the target location), the agent needs to make the optimal action decision to achieve the goal. In this pathfinding task, the LLM agent needs to output the sequence of movement actions that can reach the target location based on the spatiotemporal decision strategy f_{decision}:

$$a_{\text{pred}} = f_{\text{decision}}(g, s_{0:t}, m_t, \hat{s}_{t+1}, \ldots, \hat{s}_{t+n}) \tag{6}$$

Evaluation Metrics:

- *Success Rate of Decisions (%)*: The percentage of predicted paths that successfully move from the initial position to the target.
- *Optimal Rate of Decisions (%)*: The percentage of predicted paths that successfully move from the initial position to the target and meet the minimum number of steps.

Overall, an in-depth analysis of each dimension is crucial for a comprehensive evaluation of decision-making abilities in dynamic environments. On the one hand, the general capability decomposition approach based on the decision-making process adopted by FD-Bench is also applicable to other dynamic environments. Secondly, independent evaluations based on capability decomposition help better analyze the strengths and weaknesses.

Dataset Construction

The construction of FD-Bench involves three main phases: scenario generation, instruction generation, and solution generation. In the primary experiments, we created 500 fire scenario-instruction-solution trios. Following the fine-grained evaluation protocol, we formulated corresponding subsets, resulting in a total of 3,000 test cases. We also generated additional test sets with varying levels of difficulty.

Fire Scenario Generation

We use Python to develop an automated and standardized pipeline for initializing scenarios and simulating fire spread. We randomly select the map size, start and target points, static elements (walls), and dynamic elements (obstacles, fire, rats, dust). Meanwhile, the patterns of change in dynamic elements are randomized, for example, the speed of fire spread may vary in different directions. Breadth-first search(BFS) is then used to calculate the shortest path with and without subsequent changes to the dynamic elements. To generate meaningful fire scenarios within a dynamic environment, we only include cases where solutions exist both with and without changes to the dynamic elements, and where the lengths of these solutions differ. This selection procedure ensures that the problem is solvable and that the dynamic elements influence decision-making.

Instruction Generation

The prompt is structured into three distinct parts: background, supplied information, and a series of questions. The background section provides a detailed overview of the task, along with a simulation of the testing scenario during the initial six rounds. This allows LLM agents to perceive and observe the environment before making decisions. The supplied information section contains the answers to all preceding subtasks, ensuring that each sub-task can be evaluated independently. Finally, the question section consists of queries designed according to the fine-grained evaluation protocol, following a standardized output format.

Solution Generation

The gold solutions for dynamic element identification and rule induction are determined when the fire scenario is initialized. The gold solutions for environmental modeling and prediction are derived from the rules. The gold solutions for shortest-path decision-making are calculated using the BFS algorithm.

Experiment

Evaluation Setup

We evaluate both proprietary and open-weight LLMs on FD-Bench, aiming to provide a comprehensive benchmark for popular LLMs.

(1) For proprietary LLMs, we select five SOTA models: GPT-3.5, GPT-4(OpenAI, 2023) and GPT-4O-MINI from OpenAI, as well as CLAUDE3-HAIKU and CLAUDE3-OPUS from Anthropic(Anthropic, 2024).

(2) For open-source LLMs, we select three popular models with different sizes: LLAMA3-8B(AI, 2024), MIXTRAL-8X7B(Team, 2024), and llama3-70B.*

Hybrid Evaluation Method

For questions with fixed-form outputs, we use regular expressions to extract answers and then apply rule-based methods, such as string mapping and step-by-step verification, to validate the generated answers. For questions with free-form outputs, we apply a LLM (CLAUDE3-SONNET) to identify the answer and make the judgment, which is a common and effective method (Zheng et al., 2023). Notably, the effectiveness of both the rule-based and LLM-based annotations has been spot-checked by human annotators. Human annotators reviewed a subset of cases to verify the accuracy of the automated evaluations, and any discrepancies were resolved through discussion and consensus. This human-in-the-loop approach ensures that the evaluation results are robust and free from significant biases.

Table 2: Performance comparison between static and dynamic scenario (End-to-End).(%) **Succ.** measures the percentage of scenarios in which the model plans a feasible path, and **Opt.** indicates the rate at which the planned path is optimal.

Model	Static E2E		Dynamic E2E	
	Succ.(↑)	Opt.(↑)	Succ.(↑)	Opt.(↑)
gpt-3.5	1.0	1.0	$0.2_{\downarrow 0.8}$	$0.2_{\downarrow 0.8}$
gpt-4	74.8	63.4	$8.8_{\downarrow 60.0}$	$8.0_{\downarrow 55.4}$
gpt-4o-mini	80.8	67.2	$12.0_{\downarrow 68.8}$	$9.4_{\downarrow 57.8}$
claude-3-haiku	1.2	0.8	$0.4_{\downarrow 0.8}$	$0.4_{\downarrow 0.4}$
claude-3-opus	1.3	0.8	$0.4_{\downarrow 0.9}$	$0.4_{\downarrow 0.4}$
llama3-8B	0.0	0.0	$0.0_{\rightarrow 0.0}$	$0.0_{\rightarrow 0.0}$
Mixtral-8x7B	0.6	0.0	$0.0_{\downarrow 0.6}$	$0.0_{\rightarrow 0.0}$
llama3-70B	0.6	0.6	$0.0_{\downarrow 0.6}$	$0.0_{\downarrow 0.6}$

Main Results

In this section, We aim to explore research questions below.

RQ1: How do LLMs perform in dynamic versus static scenarios? To explore how dynamic scenarios impact the decision-making abilities of LLMs compared to static scenarios, we conducted two sets of experiments.

The results in Table 2 show that all models, including advanced ones like GPT-4 and GPT-4O-MINI, perform significantly worse in dynamic scenarios, with over a 50% drop in success and optimal rates. Additionally, models like LLAMA3 and MIXTRAL-8X7B fail to solve any cases, emphasizing the challenges of decision-making in dynamic environments.

RQ2: How do models perform in dynamic decision-making? To further analyze, the fine-grained evaluation results are presented in Table 3. GPT-4O-MINI performs

*We used API from Groq(Groq, 2024).

Table 3: Main Results of FD-Bench. **bold** denotes the best score among all models. **Red** represents the maximum value within the model and reasoning framework. (%) **Stage 1: RD** represents the F1-score for the recognition of dynamic elements, while **RD-I** indicates the F1-score for recognizing dynamic elements that impact task completion. **IN-S** and **IN-H** refer to the accuracy of rule induction for dynamic elements governed by simple rules (obstacle, rat, dust) and complex rules (fire), respectively. Likewise, **Stage 2: PR-S** and **PR-H** denote the accuracy of predicting future three-round conditions for dynamic elements with simple and complex rules. **Stage 3: PP-S** measures the percentage of scenarios in which the model plans a feasible path, and **PP-O** indicates the rate at which the planned path is optimal.

Model	Stage 1				Stage 2		Stage 3	
	RD(↑)	RD-I(↑)	IN-S(↑)	IN-H(↑)	PR-S(↑)	PR-H(↑)	PP-S(↑)	PP-O(↑)
claude-3-haiku (`claude-3-haiku-20240307`)	76.7	66.1	48.2	3.2	49.3	20.0	0.2	0.2
claude-3-opus (`claude-3-opus-20240229`)	68.0	73.1	52.0	2.8	50.4	23.1	0.3	0.3
gpt-3.5 (`gpt-3.5-turbo-0125`)	75.7	68.5	32.0	0.2	57.8	19.5	0.4	0.4
gpt-4o-mini (`gpt-4o-mini-2024-07-18`)	**86.7**	81.2	56.6	11.8	**68.5**	**40.4**	**14.3**	**11.0**
gpt-4 (`gpt-4-turbo-2024-04-09`)	85.8	**90.0**	**77.3**	**21.0**	67.9	38.3	13.4	9.6
- gpt-4 + CoT	**87.2**	91.0	72.5	18.3	68.1	39.7	13.5	9.6
- gpt-4 + ReAct	86.8	**92.0**	73.4	20.0	**69.5**	40.3	**14.4**	**12.0**
llama3-8B (`Meta-Llama-3-8B-Instruct`)	79.6	74.3	56.6	14.2	47.7	22.5	0.0	0.0
llama3-70B (`Meta-Llama-3-70B-Instruct`)	76.0	71.0	54.5	14.8	49.5	22.7	0.0	0.0
mixtral-8x7B (`Mixtral-8x7B-Instruct-v0.1`)	84.7	85.1	47.7	13.8	31.1	16.0	0.2	0.0

best overall, excelling in state prediction and prediction-based decision-making. GPT-4 also demonstrates impressive performance, particularly excelling in rule induction tasks. Other proprietary models like GPT-3.5 and CLAUDE-3-HAIKU perform well in simpler tasks but struggle with complex rule induction, even performing worse than open-source models due to hallucination. Regarding open-source models, LLAMA3-8B and MIXTRAL-8X7B show competitive performance in recognizing dynamic elements, but they struggle with complex rule induction and prediction-based decision-making.

Overall, these results indicate that proprietary models generally outperform open-source models in dynamic, multi-state decision-making scenarios. However, both categories of models face significant challenges when dealing with more complex, prediction-based decision-making tasks.

RQ3: How to improve capabilities via structured reasoning? Inspired by the "chunking" mechanism(Thalmann, Souza, & Oberauer, 2019) in CLT and positive impact of reasoning frameworks, we propose a hierarchical prompting strategy to mitigate the cognitive load imposed by dynamic environments. This approach aligns the decision-making process with the modular structure of FD-Bench by explicitly guiding agents to decompose tasks into three stages: dynamic element recognition, state prediction, and action planning.

The results in Table 4 across three difficulty levels (Easy (4-6 steps), Mid (7-9 steps), and Hard (10-12 steps)) exhibit marked enhancements in end-to-end performance. The adoption of this structured framework not only enhances success rates but also reduces hallucination by constraining the agent's focus on relevant environmental dynamics at each stage. For instance, explicitly instructing models to first identify fire spread patterns before engaging in path planning, and compelling them to ground their decisions in observed patterns, yields notable benefits. These results underscore the potential of structured reasoning frameworks to expand the capability boundaries of LLMs in dynamic environments, offering a pathway toward more reliable real-world deployment.

Table 4: GPT-4 performance with vs. without hierarchical guidance framework across task difficulty levels

Difficulty Level	End to End w/o Guide		End to End w Guide	
	Succ.(↑)	Opt.(↑)	Succ.(↑)	Opt.(↑)
Easy (4-6 Steps)	8.8	8.0	13.4$_{\uparrow 4.6}$	9.6$_{\uparrow 1.6}$
Mid (7-9 Steps)	4.0	3.4	6.0$_{\uparrow 2.0}$	4.6$_{\uparrow 1.2}$
Hard (10-12 Steps)	1.0	0.5	1.2$_{\uparrow 0.2}$	0.5$_{\rightarrow 0.0}$

Conclusion

In this paper, we present FD-Bench, a fine-grained benchmark for evaluating decision-making in dynamic scenarios. FD-Bench utilizes a fire emergency evacuation scenario as a prototypical dynamic setting, offering a comprehensive evaluation framework to systematically analyze variations in decision-making performance. Our key findings reveal that even state-of-the-art models (e.g., GPT-4 and GPT-4o-mini) experience notable performance declines (50% success rate reduction) in dynamic versus static scenarios, primarily due to challenges in multi-step prediction and complex rule induction. To tackle these limitations, we also propose a hierarchical prompting strategy that improves success rates by up to 4.6% through structured reasoning, demonstrating the potential of task decomposition to mitigate cognitive load in dynamic environments. Overall, FD-Bench not only advances the understanding of decision-making in complex environments but also provides actionable insights for enhancing the practical application of LLM agents in real-world scenarios.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 72432007 and the National Key R&D Program of China under Grant No. 2022YFC3302900.

References

AI, M. (2024). *Introducing meta llama 3: The most capable openly available llm to date.* Retrieved from https://ai.meta.com/blog/introducing-llama-3 (Accessed: 2024-06-15)

Anthropic. (2024). *Claude 3 technical report.* Retrieved from https://www.anthropic.com/claude-3 (Accessed: 2024-06-15)

Chen, Z., Du, W., Zhang, W., Liu, K., Liu, J., Zheng, M., ... others (2023). T-eval: Evaluating the tool utilization capability step by step. *arXiv preprint arXiv:2312.14033*.

Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., ... Anandkumar, A. (2022). Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, *35*, 18343–18362.

Fu, J., Kumar, A., Nachum, O., Tucker, G., & Levine, S. (2021). *D4rl: Datasets for deep data-driven reinforcement learning.*

Groq. (2024). *Groq api documentation.* https://api.groq.com/documentation. (Accessed: 2024-06-15)

Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2021). Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, *44*(9), 5149–5169.

Hu, A., Corrado, G., Griffiths, N., Murez, Z., Gurau, C., Yeo, H., ... Shotton, J. (2022). Model-based imitation learning for urban driving. *Advances in Neural Information Processing Systems*, *35*, 20703–20716.

Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., & Narasimhan, K. (2023). Swe-bench: Can language models resolve real-world github issues? *ArXiv*, *abs/2310.06770*.

Kasereka, S., Kasoro, N., Kyamakya, K., Doungmo Goufo, E.-F., Chokki, A. P., & Yengo, M. V. (2018). Agent-based modelling and simulation for evacuation of people from a building in case of fire. *Procedia Computer Science*, *130*, 10-17. (The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops) doi: https://doi.org/10.1016/j.procs.2018.04.006

Kempka, M., Wydmuch, M., Runc, G., Toczek, J., & Jaśkowski, W. (2016). *Vizdoom: A doom-based ai research platform for visual reinforcement learning.*

Liu, J., Xia, C. S., Wang, Y., & Zhang, L. (2024). Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, *36*.

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., ... Tang, J. (2023). *Agentbench: Evaluating llms as agents.*

Liu, X.-Y., Xia, Z., Yang, H., Gao, J., Zha, D., Zhu, M., ... Guo, J. (2023). *Dynamic datasets and market environments for financial reinforcement learning.*

Ma, C., Zhang, J., Zhu, Z., Yang, C., Yang, Y., Jin, Y., ... He, J. (2024a). *Agentboard: An analytical evaluation board of multi-turn llm agents.*

Ma, C., Zhang, J., Zhu, Z., Yang, C., Yang, Y., Jin, Y., ... He, J. (2024b). Agentboard: An analytical evaluation board of multi-turn llm agents. *ArXiv*, *abs/2401.13178*.

Nachazel, T. (2021). Fuzzy cognitive maps for decision-making in dynamic environments. *Genetic Programming and Evolvable Machines*, *22*(1), 101–135.

OpenAI. (2023). *Gpt-4 technical report.* Retrieved from https://cdn.openai.com/papers/gpt-4.pdf (Accessed: 2024-06-15)

Padakandla, S. (2021). A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Computing Surveys (CSUR)*, *54*(6), 1–25.

Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S., & Torralba, A. (2018). *Virtualhome: Simulating household activities via programs.*

Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., ... others (2023). Toolllm: Facilitating large language models to master 16000+ real-world apis. *ArXiv preprint*, *abs/2307.16789*.

Shinn, N., Labash, B., & Gopinath, A. (2023). Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*.

Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., ... Fox, D. (2020). *Alfred: A benchmark for interpreting grounded instructions for everyday tasks.*

Shridhar, M., Yuan, X., Côté, M., Bisk, Y., Trischler, A., & Hausknecht, M. J. (2021). Alfworld: Aligning text and embodied environments for interactive learning. In *9th international conference on learning representations, ICLR 2021, virtual event, austria, may 3-7, 2021*. OpenReview.net.

Song, C. H., Wu, J., Washington, C., Sadler, B. M., Chao, W.-L., & Su, Y. (2023). Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 2998–3009).

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive science*, *12*(2), 257–285.

Team, M. A. (2024). *Mixtral of experts: A high quality sparse mixture-of-experts.* Retrieved from https://mistral.ai/news/mixtral-of-experts/ (Accessed: 2024-06-15)

Thalmann, M., Souza, A. S., & Oberauer, K. (2019). How does chunking help working memory? *Journal of Experimental Psychology: Learning, Memory, and Cognition*,

45(1), 37.

Valmeekam, K., Marquez, M., Olmo, A., Sreedharan, S., & Kambhampati, S. (2023). *Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change.*

Valmeekam, K., Olmo, A., Sreedharan, S., & Kambhampati, S. (2022). Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *Neural information processing systems.*

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., … Anandkumar, A. (2023). *Voyager: An open-ended embodied agent with large language models.*

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., … others (2024). A survey on large language model based autonomous agents. *Frontiers of Computer Science*, *18*(6), 186345.

Wang, R., Jansen, P., Côté, M.-A., & Ammanabrolu, P. (2022). ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 conference on empirical methods in natural language processing* (pp. 11279–11298). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., … others (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, *35*, 24824–24837.

Yang, H., Yue, S., & He, Y. (2023). *Auto-gpt for online decision making: Benchmarks and additional opinions.*

Yao, S., Chen, H., Yang, J., & Narasimhan, K. (2022). Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, *35*, 20744–20757.

Yao, S., Chen, H., Yang, J., & Narasimhan, K. (2023). *Webshop: Towards scalable real-world web interaction with grounded language agents.*

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations.*

Zhao, C., Mi, F., Wu, X., Jiang, K., Khan, L., & Chen, F. (2022). Adaptive fairness-aware online meta-learning for changing environments. In *Proceedings of the 28th acm sigkdd conference on knowledge discovery and data mining* (pp. 2565–2575).

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., … Stoica, I. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, & S. Levine (Eds.), *Advances in neural information processing systems* (Vol. 36, pp. 46595–46623). Curran Associates, Inc.

Zhou, Q., Chen, S., Wang, Y., Xu, H., Du, W., Zhang, H., … Gan, C. (2024a). *Hazard challenge: Embodied decision making in dynamically changing environments.*

Zhou, Q., Chen, S., Wang, Y., Xu, H., Du, W., Zhang, H., … Gan, C. (2024b). Hazard challenge: Embodied decision making in dynamically changing environments. *ArXiv*, *abs/2401.12975*.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., … others (2023). Webarena: A realistic web environment for building autonomous agents. *ArXiv preprint*, *abs/2307.13854*.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., … Neubig, G. (2024). *Webarena: A realistic web environment for building autonomous agents.*