DISENTANGLING EXPLORATION OF LARGE LANGUAGE MODELS BY OPTIMAL EXPLOITATION

Tim Grams[1]* **Patrick Betz**[2] **Christian Bartelt**[1]
[1]Technical University of Clausthal [2]University of Mannheim

ABSTRACT

Exploration is a crucial skill for self-improvement and open-ended problem-solving. However, it remains unclear if large language models can effectively explore the state-space within an unknown environment. This work isolates exploration as the sole objective, tasking the agent with delivering information that enhances future returns. Within this framework, we argue that measuring agent returns is not sufficient for a fair evaluation and decompose missing rewards into exploration and exploitation components based on the optimal achievable return. Comprehensive experiments with various models reveal that most struggle to sufficiently explore the state-space and weak exploration is insufficient. We found a positive correlation of exploration performance with language comprehension and reasoning capabilities. Furthermore, we show that our decomposition can provide insights into differences in behaviors driven by prompt engineering, offering a valuable tool for refining performance in exploratory tasks.

1 INTRODUCTION

Large language models in decision making. Recently, large language models (LLMs) have demonstrated promising results in various decision making tasks such as web browsing Yao et al. (2022b); Shinn et al. (2024); Ma et al. (2023), game-playing Paglieri et al. (2024), and simulated households Yao et al. (2022b); Shinn et al. (2024). Hereby, LLMs act as agents that observe states and take actions in environments. Through their vast internal knowledge-base and autoregressive in-context reasoning capabilities, the models are supposed to quickly adapt to new tasks. However, previous work has shown that LLMs struggle with solving complex problems due to several shortcomings: For example, the ability to learn from mistakes is often limited Huang et al. (2023) and LLMs have difficulties with planning over long horizons Kambhampati et al. (2024).

Exploration is essential for self-improvement. Intelligent exploration is an important skill for improvement in many tasks. Hereby, exploration is a critical process for systematically seeking novel information to reduce uncertainty and improve decision-making over time. Traditional approaches often rely on stochastic noise Mnih (2013); Lillicrap (2015) or reward-shaping Bellemare et al. (2016). Such methods induce redundancy and differ from human decision making which is mainly driven by reasoning about current information, past steps, and the unknown potential of yet unknown actions. Therefore, recent works proposed several approaches towards more intelligent exploration Huang et al. (2024); Lu et al. (2024); Nie et al. (2024). Hereby, LLMs could have the potential to assist discovery by identifying patterns, leveraging prior knowledge, and intelligently navigating the state-space. LLMs may transform exploration into a targeted and efficient process.

Limited focus on exploration as an independent ability. While exploration is acknowledged as a crucial component for learning and LLM decision making, prior work has predominantly investigated it in conjunction with exploitation Krishnamurthy et al. (2024); Nie et al. (2024); Paglieri et al. (2024); Ke et al. (2024); Chen et al. (2024). In this context, exploration is often measured indirectly through cumulative rewards or success rates. However, these approaches conflate the ability with overall performance. It is difficult to isolate an agent's exploration, as true progress is independent of the agent's return. While seeking information, agents are supposed to sacrifice short-term rewards in favor of long-term understanding. Also, strong exploration does not always translate into high

*Corresponding author: `tim.nico.grams@tu-clausthal.de`

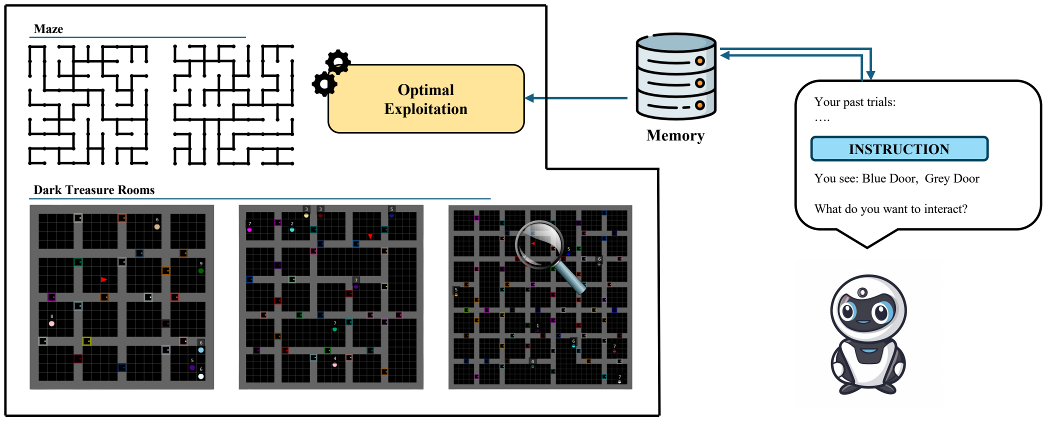


Figure 1: **An overview of our evaluation procedure based on an optimal exploitation**. The agent navigates a maze or grid of rooms by interacting with doors and balls. Overall, the goal is to collect information about high rewarding balls. After each episode, we calculate the optimal sequence of actions to measure exploration progress based on the agent's current memory.

performance if the agent lacks the ability to exploit the knowledge it has gathered. An agent may thoroughly explore an environment but fail to leverage its findings. Therefore, it is essential to develop methods that accurately measure exploration of LLMs, as it helps to foster our understanding of their learning and adaptability in sequential decision making.

Contribution. In this paper, we claim that exploration will be an essential skill for future agents. Recent studies Krishnamurthy et al. (2024); Nie et al. (2024) investigated LLMs balancing exploration and exploitation where they found that they often struggle to achieve high rewards. Conversely, we disentangle exploration for an independent evaluation. In this controlled setting, we argue that conventional metrics, such as agent returns, LLM exploitation, or the convergence towards the best sequence of actions Huang et al. (2024); Krishnamurthy et al. (2024) are insufficient. To fill this gap, we make the following contributions:

- We propose to measure the *optimal exploitation return* (see Figure 1) and decompose missing rewards into their exploration and exploitation components. This decomposition measures exploration progress accurately and enables insights into reward sacrification.
- To the best of our knowledge, we are the first to conduct a comprehensive evaluation of various popular open-source and closed LLMs with the exploration being disentangled.

Our evaluation framework is designed to be environment-agnostic, making it applicable to a broader range of tasks and scenarios in future research.

Results. Our findings indicate that most LLMs face challenges with state-space exploration, when the task is isolated. These results align with prior work Krishnamurthy et al. (2024); Paglieri et al. (2024), which highlight the trade-off between exploration and exploitation. We observe that state-space coverage diminishes significantly when agents are required to plan over extended horizons, indicating a potential limitation in long-term exploratory capabilities. Contrary to earlier claims Huang et al. (2024), our analysis demonstrates that weak LLMs are insufficient. Notably, we identified a significant trend wherein stronger exploration capabilities correlate with language understanding and reasoning abilities, suggesting that future, more advanced LLMs may possess enhanced exploration potential. Our method further uncovered model-specific differences in reward sacrification and we demonstrated how the impact of agent instructions on exploratory behavior can be assessed, providing insights into the design of prompts during prompt-engineering.

2 Related Work

Recently, exploring the capabilities of LLMs gained significant attention and has been investigated from different perspectives. Prior works include, but are not limited to aspects such as planning

Kambhampati et al. (2024); Song et al. (2023); Valmeekam et al. (2023), world models Hao et al. (2023); Guan et al. (2023), logical- Xu et al. (2023); Parmar et al. (2024), commonsense- Li et al. (2021), and mathematical reasoning Imani et al. (2023); Yuan et al. (2023). Our research focuses on intelligent exploration of the state-space in sequential decision making. Generally, we summarize existing methods into three streams of work:

Large language models in decision making. A broad line of works has applied LLMs in decision making. Relatively few addressed the challenge of exploration. The majority of existing research focuses on multi-armed bandit problems Park et al. (2024); Krishnamurthy et al. (2024); Nie et al. (2024); Chen et al. (2024). For example, Krishnamurthy et al. (2024) analyze scenarios involving multiple buttons with underlying stochastic rewards. The LLM is presented past actions with their rewards and must maximize the return over multiple trials. However, multi-armed bandits lack the rich sequential state-space of many real-world applications. Other research examined exploration in hypothesis testing Piriyakulkij et al. (2024); Ke et al. (2024). Only a limited number of works Ma et al. (2023); Paul (2023); Huang et al. (2024); Lu et al. (2024) proposed methods for sequential state-spaces. Lu et al. (2024) suggest returning to interesting states. Huang et al. (2024) argue that smaller-scale models can handle exploration, reserving larger models for exploitation. Generally, evaluations mostly focus on agent returns. While exploration LLMs are used as system components, an isolated evaluation is missing. Therefore, the focus of our study is specifically on LLMs that gather information about their environment.

Exploration in reinforcement learning. Our evaluation is closely related to exploration in RL. Early approaches predominantly relied on random methods, such as epsilon-greedy strategies Mnih (2013); Lillicrap (2015). More recent research has shifted toward developing intelligent exploration techniques Zhang et al. (2021); Ecoffet et al. (2019); Norman & Clune (2023); Mazoure et al. (2023). Methods that decouple exploration and exploitation are particularly relevant to our work Liu et al. (2021); Avner et al. (2012); Norman & Clune (2023); Zhang et al. (2021); Duan et al. (2016); Schäfer et al. (2021); Whitney et al. (2021). In methods, such as First-Explore Norman & Clune (2023) exploration and exploitation are handled by separate policies, with the exploration policy informing subsequent exploitation decisions. Generally, RL focuses on stabilizing the decoupled policies. Evaluations use non-optimal exploitation returns as optimal policies can be hard to obtain in complex environments. Contrary, we measure the optimal exploitation return for the evaluation of LLMs. Agent returns and LLM exploitation do not provide reliable estimates.

Evaluation environments. A variety of environments have been developed to assess the different capabilities of LLMs. Notable examples that require exploration include TextWorld Côté et al. (2019), Minecraft Fan et al. (2022), Crafter Hafner (2021), NetHack Küttler et al. (2020), MiniHack Samvelyan et al. (2021), MiniGrid Chevalier-Boisvert et al. (2024), and BabyAI Chevalier-Boisvert et al. (2018). There is also growing interest in environments for computer and web agents Zhou et al. (2023); Yao et al. (2022a); Xie et al. (2024). Comprehensive benchmarks, including SmartPlay Wu et al. (2023), AgentBench Liu et al. (2023), and BALROG Paglieri et al. (2024), aggregate multiple environments to evaluate agents across a wide range of capabilities. In this context, BALROG reported aimless movements and redundant behaviors. The difference in our work lies in the isolation of exploration. Our experiments are based on mazes and an adapted version of Dark Treasure Room Norman & Clune (2023). Our environments offer limited prior knowledge, provide non-binary, relatively dense rewards, and maintain a high-level action space.

3 Evaluating with Optimal Exploitation

In the following section, we present our evaluation method. We introduce the formal background, the necessity of deriving the optimal exploitation for a fair comparison, and our return decomposition. Additionally, we introduce Maze and Dark Treasure Rooms environments which provide dense rewards and an unsolved challenge for LLM exploration in sequential decision-making.

3.1 Theoretical Framework

Markov Decision Process. We study a deterministic Markov decision process $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$ where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is the finite action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ defines the transition dynamics of the environment, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function and γ the discount factor.

At each time step t, an agent samples an action from policy $\pi : \mathcal{S} \to \mathcal{A}$ based on the current observation $s_t \in \mathcal{S}$ and executes it in the environment. The environment transitions and the agent receives a reward r_t. In an episodic MDP, the procedure repeats until a terminal state s_T is reached or the maximum number of steps exceeded. The sequence of state, action, and reward triples until s_T is called a trajectory and describes an episode $\tau_i = (s_0, a_0, r_0, s_1, a_1, r_1, s_2, \dots)$. In our setting, the agent has access to the current triples and the history of prior trajectories $h_i = (\tau_1, \dots, \tau_i)$ at each time step t in episode i. The goal is to maximize the cumulative return $R_i = \sum_{t=1}^{T} r_t$.

Exploitation versus exploration. The cumulative return R_i^{LLM} of an agent is influenced by its ability to balance exploitation and exploration. Exploitation leverages past high-rewarding actions to maximize the immediate return based on the agent's current knowledge. In contrast, exploration emphasizes interacting with the environment to gather new information, which can enhance future decision-making and rewards. An agent fully dedicated to exploitation acts according to policy $\pi^*_{\text{exploit}}(h_i) = \arg\max_\pi \sum_{t=0}^{T} r(s_t, \pi(s_t))$ with $s_t, \pi(s_t) \in h_i$ that utilizes past trajectories h_i to estimate the optimal action sequence. Conversely, an agent focused on exploration seeks to augment history h_i by incorporating new trajectories that maximize the potential future exploitation return. Under the assumption of an optimal exploitation return R_i^{exploit}, which ignores redundant or negatively impactful trajectories and whose return monotonically increases over episodes, the objective of the exploration policy is to maximize the exploitation return at the end of exploration: $\pi^*_{\text{explore}} = \arg\max_{\pi_{\text{explore}}} \mathbb{E}\left[R_{\text{final}}^{\text{exploit}} \mid h_i = h_{i-1} \cup \tau_{\text{explore}}\right]$.

The necessity of optimal exploitation. When an agent does both, exploitation and exploration, progress can be evaluated by temporarily disabling exploration and measuring the cumulative rewards of the agent. However, for an agent fully dedicated to exploration, the return alone does not accurately reflect success (see Figure 2), as successful exploration does not directly result in a higher return. Also measuring exploitation returns with an LLM is similarly hard to exploration and, therefore, does not give reliable estimates (appendix A.7). Nevertheless, in our environments, we can measure the optimal exploitation return R_i^{exploit}. Note that this is possible in most evaluation environments by using, for example, dynamic programming methods which guarantee the optimal sequence of actions based on history h_i. By calculating the optimal exploitation return, we can assess an agent's exploration capabilities accurately without conflating them with exploitation.

Decomposing the total return gap. Based on the optimal exploitation return, we can decompose the total return gap, i.e., the difference of the maximal achievable return of the environment and the actual achieved return. Let R^{max} denote the maximal achievable reward of the environment; it is independent of the agent and can be predefined for evaluation purposes. As previously, let R_i^{LLM} be the agent reward and $R_i^{exploit}$ the optimal exploitation reward by acting according to policy π^*_{exploit}. Contrary to prior evaluations, π^*_{exploit} is not determined by an agent. Note that it follows $R^{max} \geq R^{exploit} \geq R^{LLM}$. We let Δ^{total} denote the total reward gap, which is calculated as the difference of maximal achievable and the actual achieved return by the agent:

$$\Delta^{total} = R^{max} - R^{LLM}. \tag{1}$$

Next, we define the exploration gap $\Delta^{explore}$ as the difference of the maximal achievable and the optimal exploitation return. **Why?** If the return under optimal exploitation is not as high as the maximal achievable return, the only reason can be lacking exploration. Therefore, $\Delta^{explore}$ evaluates the true progress in exploration:

$$\Delta^{explore} = R^{max} - R^{exploit}. \tag{2}$$

Finally, we define the exploitation gap $\Delta^{exploit}$ as the difference between the optimal exploitation and the agent reward. It can determine the rewards the agent sacrifices for potential exploration:

$$\Delta^{exploit} = R^{exploit} - R^{LLM}. \tag{3}$$

It is easy to see that it holds by construction that

$$\Delta^{total} = \Delta^{explore} + \Delta^{exploit}. \tag{4}$$

The total return gap decomposes into the exploration and exploitation gap.

3.2 ENVIRONMENTS

Decision making in room worlds. Our experiments are conducted in interconnected rooms. To mitigate the impact of low-level spatial navigation, we use a higher-level action space that is conceptually similar to TextWorld Côté et al. (2019). Conversely, our domain features only minimal narrative and semantic associations. Each environment has a fixed layout and starting position. Episodes terminate when the agent collects three objects or exceeds a maximum number of door traversals. The latter is calibrated to the distance between the starting position and the furthest room, ensuring that the exploration budget scales with complexity. At the beginning of each episode, the agent is reset to its starting position, requiring it to strategically plan exploration over extended horizons. In our case, the optimal exploitation is calculated by solving an instance of the orienteering optimization problem. More details about the calculation and our environments can be found in appendix A.

Dark Treasure Rooms. The first task follows a grid layout and is an adapted version of Dark Treasure Room Norman & Clune (2023) as we replace the low-level navigation with high-level object interactions. Each map contains balls which are placed randomly on the grid and associated with a fixed reward sampled from a uniform distribution $U(1, 10)$. The agent's goal is to maximize the return by collecting at most three balls within the step budget. As reward locations and their values are randomly generated, the optimal exploration strategy, across multiple layouts, requires full coverage of the state-space. Inadequate exploration risks the exploitation collecting suboptimal rewards due to missed balls. The environments are illustrated in appendix A.1.

Maze. Finally, we compare exploration in randomly generated mazes to scale beyond simple grid layouts. Mazes are similarly common in the RL literature to test in-context learning abilities Laskin et al. (2022); Grigsby et al. (2023); Schmied et al. (2024). Again, we take inspiration from Norman & Clune (2023) and randomly place multiple rewards in the environment. In contrast, to Dark Treasure Rooms, rewards are more sparsely distributed. Further, mazes demand increased long-horizon planning as decisions have more impact on possible future paths. The layouts used for our experiments can be found in appendix A.2 and an exemplary interaction is depicted in Figure 1.

4 EXPERIMENTAL EVALUATION

Subsequently, we describe our experimental setup and results. The experiments aim to demonstrate the utility of our framework by addressing the following research questions: **1.)** ***Effectiveness:*** *Can LLMs explore and reduce the exploration gap?* and **2.)** ***Prompt-engineering:*** *How do agent instructions impact exploration and exploitation gaps?*

4.1 SETUP

Agent instruction. Our experimental setup directs agents to explore in order to solve tasks. The specific instruction tells to *"collect information that helps to become better at maximizing the return"*. We reference this prompt as *task-oriented* exploration. The exact template is detailed in appendix A.4. Each agent is equipped with a simple memory system that stores the history of past interactions. Notably, unlike many prior studies, our approach does not provide agents with explicit hints about the domain.

Statistical robustness. In terms of statistical robustness, we test each open-source with ten and each closed model with five runs on every environment. This results in a total of 30 runs per data point for open-source models and 15 runs for closed models. In the following Figures, we report statistical significance using the standard error. Our return decomposition is measured at the end of exploration and as the mean over all episodes. The values are normalized to account for environments with different maximum returns. Other statistics are evaluated after all environment interactions. Full results are listed in appendix B and C.

Evaluation procedure. The **1.)** ***Effectiveness*** is evaluated across three grids of Dark Treasure Rooms (4×4, 5×5, and $7{\times}7$) and mazes. The configurations test the scalability of the planning horizon. To assess **2.)** ***Prompt-engineering***, we compare our task-oriented agent instruction, with alternative prompts in a $7{\times}7$ grid. We set soft lower and upper bounds for the exploration capabilities by directing the agents to strictly exploit or aimlessly explore without referencing the task. See appendix D for more information and the alternative instructions.

4.2 Other Statistics

We also describe exploration using the following statistics:

Agent return represents the total return collected by the agent during its interactions with the environment. Although it is not a direct indicator of effective exploration, this metric reflects the ability to engage in ways that yield cumulative benefits. A higher agent return implies that the agent is navigating towards rewarding states and is possibly close to exploitation. Contrary to our decomposition, it does not necessarily imply that the agent is effectively exploring.

State-space coverage reflects the ability to discover new information and avoid being trapped in local optima. In our task, effective exploration requires covering the entire state space, as balls with high rewards are randomly spread. However, complete exploration may not always be necessary or could be infeasible in large, infinite, or continuous spaces due to natural and resource constraints. In contrast, our decomposition also covers cases in which full exploration is not essential.

Memory redundancy measures the fraction of redundant state-action pairs in the memory. The objective should be to minimize repetition. In our environments, the agent should avoid revisiting states or actions as they do not provide new information. Reducing redundancy conserves resources and ensures that the history can be utilized efficiently. Otherwise, information has to be retrieved from longer histories to find novel or informative states to explore.

Sample efficiency evaluates the number of environment interactions required to converge to the 90% equilibrium of the maximum achieved exploitation return. This means minimizing memory redundancy and adding trajectories to the history such that the exploitation return increase is maximized. Note that sample efficiency is decoupled from the quality of the final policy. Being constantly stuck with the same low reward balls is equally as fast convergence to the maximum exploitation return.

4.3 Models

We evaluate the exploration ability of the popular open-source LLMs Mistral (7B) Jiang et al. (2023), Gemma2 (9B and 27B) Team et al. (2024b), Llama 3.1 (7B) Dubey et al. (2024), Llama3.3 (70B), Mistral Small (22B) AI (2024a), and Mistral Large 2 (123B) AI (2024b). We further include closed-models GPT-4o Achiam et al. (2023), o1-mini OpenAI (2024), Gemini 1.5 Pro Team et al. (2024a), and Claude 3.5 Sonnet Anthrophic (2024). The LLMs are compared to a random baseline.

4.4 Can Large Language Models Explore?

Most agents fail to minimize their exploration gap. Our results indicate that LLMs struggle to explore the state-space although their sole objective is to collect information rather than exploit it. The results, averaged across all grid sizes of Dark Treasure Rooms, are presented in Table 1 and 2. In this case, almost all models demonstrated significantly larger or indifferent exploration gaps in their final episodes compared to a random baseline, with the exceptions of Mistral Large, o1-mini, and Gemini 1.5 Pro. However, also these models failed to achieve complete state-space coverage, exhibiting non-zero exploration gaps and exploring less than 90% of the state space on average. Performance was additionally worse in the maze environment, where the increased structural complexity and sparse reward distribution led to lower coverage and larger exploration gaps, further emphasizing the models' limitations. At closer inspection, models showed redundant exploration of previously visited states, with exploration disproportionately focused on initial high-reward regions.

LLMs are limited in targeted long-horizon exploration. Our evaluation reveals a statistically significant decrease ($p \leq 0.01$) in the fraction of the explored state-space by most models as the grid gets larger. Figure 2 contains a plot showing the explored state-space coverage for the different sizes grid sizes. Notably, Claude 3.5 Sonnet was the only model to maintain high coverage as the environment size increased, with no significant drop observed ($p \geq 0.05$). Larger parameter models, including GPT-4o, Mistral Large, and Claude 3.5 Sonnet, only maintained consistent coverage when the grid size increased slightly from 4×4 to 5×5 ($p \geq 0.05$). Despite achieving superior results and demonstrating strong average state-space coverage in smaller environments, the test-time reasoning model o1-mini similarly experienced a significant performance drop in larger grids. Likewise, all models exhibited the largest exploration gaps in the maze environments, underscoring the limitations of careful long-horizon planning in more complex settings.

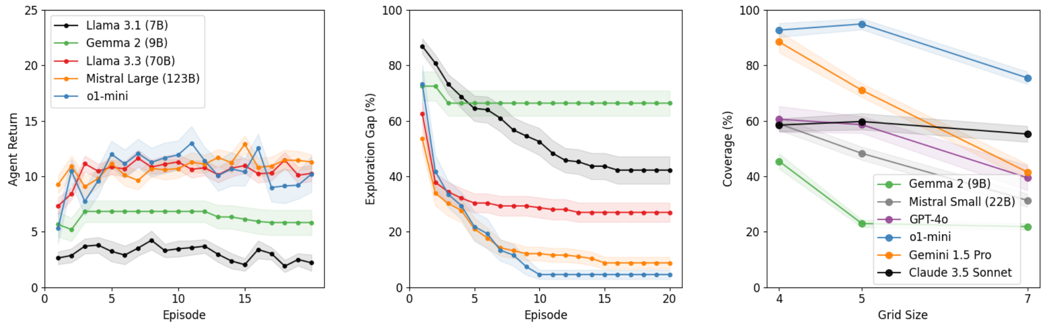


Figure 2: **The agent return, exploration gap, and grid coverage in 5×5 Dark Treasure Rooms.** Our exploration gap (middle, lower is better) measures progress in exploration accurately in contrast to the agent return. Note, for example, that Llama3.1 (7B) performs better exploration while achieving lower agent return than Gemma 2 (9B). Also Mistral Large (123B), Llama 3.3 (70B), and o1-mini do not show significant differences in the agent return although they are significantly different ($p \leq 0.01$) in their exploration gap. Our experiments further demonstrate that, coverage shrinks with grid size and, therefore, increased long horizon planning.

Table 1: **Total return gap decomposition averaged over the 4×4, 5×5, and 7×7 Dark Treasure Rooms.** The exploitation and exploration gaps are normalized to account for varying maximum rewards in different environments. The numbers in parentheses show the fraction that exploitation and exploration gaps make up of the total return gap, respectively.

		Last Episode			Mean		
		Total Gap	**Exploitation Gap**	**Exploration Gap**	**Total Gap**	**Exploitation Gap**	**Exploration Gap**
<30B	Mistral (7B)	0.80	0.25 (0.31)	0.55 (0.69)	0.78	0.20 (0.26)	0.57 (0.74)
	Llama 3.1 (7B)	0.84	0.35 (0.41)	0.49 (0.59)	0.82	0.24 (0.28)	0.57 (0.72)
	Gemma 2 (9B)	0.75	**0.15** (0.17)	0.60 (0.83)	0.69	**0.08** (0.10)	0.60 (0.90)
	Mistral Small (22B)	0.68	0.28 (0.41)	0.40 (0.59)	0.69	0.20 (0.29)	0.49 (0.71)
	Gemma 2 (27B)	**0.57**	0.18 (0.32)	**0.38** (0.68)	**0.52**	0.12 (0.22)	**0.40** (0.78)
>30B	Llama 3.3 (70B)	0.50	**0.23** (0.40)	0.27 (0.60)	0.50	**0.18** (0.32)	0.32 (0.68)
	Mistral Large (123B)	**0.33**	0.26 (0.75)	**0.07** (0.25)	**0.38**	0.25 (0.62)	**0.13** (0.38)
Closed	GPT-4o	0.44	0.14 (0.31)	0.30 (0.69)	0.49	0.14 (0.27)	0.35 (0.73)
	GPT-o1-mini	0.43	0.38 (0.79)	**0.06** (0.21)	0.47	0.32 (0.66)	**0.16** (0.34)
	Gemini 1.5 Pro	0.52	0.30 (0.58)	0.22 (0.42)	0.53	0.23 (0.41)	0.30 (0.59)
	Claude 3.5 Sonnet	**0.38**	**0.09** (0.29)	0.29 (0.71)	**0.41**	**0.09** (0.26)	0.32 (0.74)
Baseline	Random Walk	0.85	0.53 (0.62)	0.32 (0.38)	0.86	0.37 (0.42)	0.49 (0.58)

Weak exploration is not enough. Unlike the claims of previous research Huang et al. (2024), we found that LLMs with limited language understanding and reasoning abilities struggle to effectively explore their environment. This contrast is probably due to the accurate evaluation and increased exploration complexity compared to prior experiments in domains such as ALFWorld. We further test this hypothesis by regressing our exploration gap with scores on an enhanced version of the Massive Multilingual Language Understanding (MMLU-Pro) (Wang et al., 2024a) benchmark in Figure 3. Our results reveal a clear trend of improved exploration capabilities with MMLU-Pro score, as evidenced by the statistically significant slope of a linear regression ($p \leq 0.01$). Interestingly, closed-source models do not necessarily outperform open-source counterparts. For instance, GPT-4o and Llama 3.3 exhibit comparable exploration gaps at the end of training.

Optimality gaps measure exploration accurately. Our results support the decomposition of the total return gap into its optimal exploitation and exploration components. Reconsider in this context the agent returns and the corresponding exploration gaps, which are illustrated in Figure 2. We find that the agent return does often not correlate with true exploration progress. For instance, models such as Mistral Large (123B), Llama 3.3 (70B), and o1-mini exhibit no significant differences in their agent returns, despite showing statistically significant differences in their exploration gaps ($p \leq 0.01$). To further investigate this disparity, we conducted an ablation study in which we substituted our optimal exploration with an alternative LLM. The results for this follow-up experiment are listed in appendix A.7. The additional ablation demonstrates that the exploitation return cannot be reliably measured in this setting. These findings underscore our hypothesis that exploitation and exploration are equally challenging and should be assessed independently to ensure an accurate evaluation.

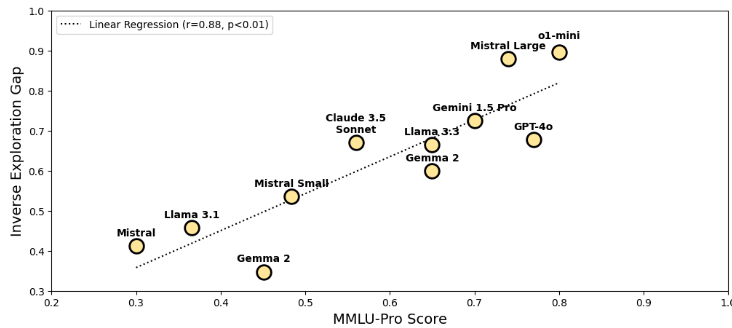


Figure 3: **Exploration gaps strongly correlate with knowledge and reasoning capabilities.** We found that models achieving high scores in MMLU-Pro are more competent in exploring their environment than their lower scoring counterparts. We test this hypothesis by regressing the average exploration gaps of Dark Treasure Rooms and the mazes with the results of the MMLU-Pro benchmark. A linear regression confirmed a significant correlation ($p \leq 0.01$). These outcomes indicate that less capable models are not more uncertain or associated with sampling different paths. Instead, they do not only tend to produce lower quality solutions but also stick to the same trajectories.

Table 2: **The other statistics averaged over the 4×4, 5×5, and 7×7 Dark Treasure Rooms.** Some LLMs achieve lower agent returns but are higher in their optimal exploitation return (see Gemini 1.5 Pro and Claude 3.5 Sonnet). We find that all agents converge quickly to their individual maximum optimal exploitation return, although more rewarding balls are not explored yet.

		Exploitation Return	Agent Return	Coverage	Redundancy	Sample Efficiency
<30B	Mistral (7B)	9.47	4.13	38.88	0.92	13.82
	Llama 3.1 (7B)	10.56	3.27	37.03	0.92	31.25
	Gemma 2 (9B)	8.53	5.18	30.10	0.95	**7.25**
	Mistral Small (22B)	12.54	6.67	46.19	**0.91**	35.44
	Gemma 2 (27B)	**12.82**	**8.99**	**49.99**	0.92	14.13
>30B	Llama 3.3 (70B)	14.94	10.32	56.93	0.91	**19.31**
	Mistral Large (123B)	**19.20**	**13.86**	**78.59**	**0.88**	19.56
Closed	GPT-4o	14.22	11.38	54.33	0.92	**20.73**
	GPT-o1-mini	**19.40**	11.76	**87.73**	**0.86**	27.88
	Gemini 1.5 Pro	16.09	9.91	67.02	0.89	23.25
	Claude 3.5 Sonnet	15.03	**13.08**	58.00	0.91	12.08
Baseline	Random Walk	14.18	3.13	69.11	0.89	51.25

4.5 What is the Impact of Agent Instructions?

Exploration stays hard for low parameter models. In the following, we compare our task-oriented exploration prompt with the alternative instructions as described in Section 4.1. The results show that models with ≤ 9 billion parameters still maintain significant exploration gaps when prompted to disregard the task and focus solely on exploring any unvisited states. For instance, Mistral (7B) and Gemma 2 (9B) exhibit exploration gaps of 57% and 26%, respectively, in their last episode. These findings may suggest that smaller models face intrinsic limitations in their ability to explore, even when tasked to prioritize exploration exclusively. Interestingly, also more competent models, such as o1-mini, do not achieve full coverage. The full list of results for the soft upper bound prompt are detailed in appendix D.2. The soft lower bound outcomes can be found in appendix D.1.

Prompts can heavily impact reward collection. Our results indicate that prompts can impact reward collection during exploration. The exploitation gap increased significantly when agents were prompted with the soft upper bound instruction that is tailored towards strictly performing any exploration. Figure 4 contains the decomposed total return gaps of Mistral Large. While the model maintained a similar task-oriented exploration gap, the manner in which this gap was minimized appeared to differ in terms of agent returns. At closer inspection, the upper bound instruction leads to more independence across episodes, whereas task-oriented exploration tends to focus on previously visited regions. These findings highlight the impact of prompt design on exploration and return.

LLMs interpret the agent instructions differently. Our results suggest that LLMs interpret the same task-oriented exploration instruction with varying behaviors. Lower parameter models predominantly exhibit behavior closer to strict exploitation when prompted to collect information to

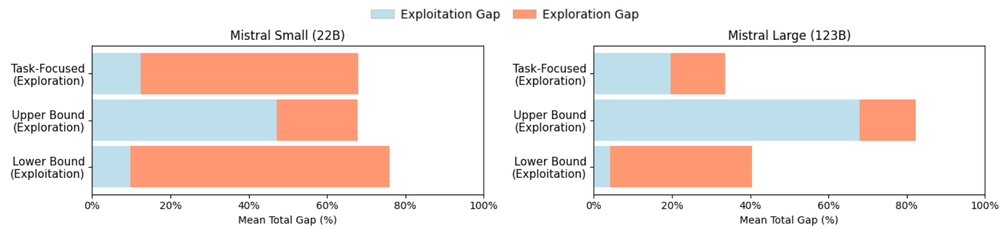


Figure 4: **The normalized mean total return gaps decomposed into their exploration and exploitation components.** Larger LLMs employing task-oriented exploration tend to sacrifice more rewards compared to smaller models. Furthermore, LLMs interpret our exploration instruction differently: for Mistral Small, the gap proportions align more closely with the soft lower bound, whereas for Mistral Large, they are closer to the upper bound. These findings support our hypothesis that our evaluation can be utilized to analyze agent instructions during prompt engineering.

solve the task. In contrast, other models perform task-oriented exploration more in line with the soft upper bound. Surprisingly, o1-mini achieved lower exploration gaps under task-oriented instructions compared to arbitrary strict exploration. However, it remains an open question how effective aimless exploration is in environments governed by rules or requiring semantic understanding.

Our framework assists prompt-engineering. The results demonstrate that our evaluation can effectively analyze various prompting strategies within simulated environments. We therefore believe it is a valuable tool for refining agent behavior prior to deployment in real-world scenarios. This approach represents an initial step toward methods that assess the impact of prompt engineering on the exploration capabilities of LLMs in sequential decision-making.

5 Discussion and Future Implications

Consistent with prior findings Krishnamurthy et al. (2024); Nie et al. (2024), we observe that most of the popular LLMs struggle with state-space exploration when exploration is isolated from exploitation. Contrary to earlier claims Huang et al. (2024), our results indicate that low parameter models exhibit significant weaknesses. This discrepancy may arise from the increased complexity to find objects in our setting while the small number of rooms is easily covered in household domains. Moreover, our experiments have shown that comparing models based on agent returns or an LLM for the exploitation is not reliable. Our evaluation procedure fills this gap and allows a fair evaluation of the following promising avenues for future research:

Reasoning and self-improvement. Various reasoning and self-improvement strategies have been proposed recently. Examples include Reflection Shinn et al. (2024), Tree-of-Thoughts Yao et al. (2024), and RL Guo et al. (2025). Evaluating how these approaches influence exploration may be a promising research direction. Our results indicate that test-time reasoning has a positive impact. In this regard, it would also be interesting to see, if LLMs can self-improve their own exploration skills. Developing strategies specifically designed with exploration in mind could be necessary.

Training for intelligent exploration. Training an agent to systematically gather information may be required to develop foundational agents. Hereby, LLMs should be investigated from a meta-RL perspective. Nie et al. (2024) show first evidence that finetuning an agent on exploration trajectories can help in multi-armed bandits. Our results demonstrate that exploration capabilities vary with prompt-engineering. Future agents should be less dependent on prompting and implicitly explore.

Small exploration models for real time applications. Our results have revealed new research directions for low parameter LLMs. We believe improving the ability of weak models to explore is a necessary steps towards many real life applications, requiring embedded and real time agents. In this context, it would also be interesting to develop LLMs that are generally less capable in other domains but can perform strong exploration. As larger LLMs have shown to be more proficient at exploration, distillation methods could possibly be developed to teach exploration to smaller models.

We discuss additional future implications in appendix E.

6 Conclusion

In conclusion, we proposed a novel evaluation framework for LLM-based agents that isolates state-space exploration through a decomposition based on the optimal exploitation return. Our approach incorporates adaptations of popular environments from the RL domain, moving beyond multi-armed bandits. We argue that developing systems capable of both intelligent exploitation and exploration is a pivotal step toward creating truly generalist agents. During the presentation of our framework, we highlighted various research directions for which it offers a robust testbed.

Future work could extend our framework with more complex environments, featuring, for example, randomization and stochasticity. A limitation of our domain is that rewards are randomly spread and high return correlates with high state-space coverage. By developing environments that combine complex exploration with underlying rules, domain knowledge and different tasks, our framework could be used to test more intelligent exploration. Hereby, benchmarks such as WebArena Zhou et al. (2023) could serve as a starting point. Exploitation and exploration are both difficult problems. Therefore, it is essential to continue evaluating them in isolation.

Acknowledgments

This research was supported in part by the German Federal Ministry for Economic Affairs and Climate Action of Germany (BMWK), and in part by the German Federal Ministry of Education and Research (BMBF).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Mistral AI. Mistral small. `https://mistral.ai/news/september-24-release/`, 2024a. [Accessed 10-01-2025].

Mistral AI. Mistral large 2i. `https://mistral.ai/news/mistral-large-2407/`, 2024b. [Accessed 10-01-2025].

Petr Anokhin, Nikita Semenov, Artyom Sorokin, Dmitry Evseev, Mikhail Burtsev, and Evgeny Burnaev. Arigraph: Learning knowledge graph world models with episodic memory for llm agents. *arXiv preprint arXiv:2407.04363*, 2024.

Anthrophic. Claude 3.5 Sonnet. `https://www.anthropic.com/news/claude-3-5-sonnet`, 2024. [Accessed 10-01-2025].

Orly Avner, Shie Mannor, and Ohad Shamir. Decoupling exploration and exploitation in multi-armed bandits. *arXiv preprint arXiv:1205.2874*, 2012.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Dingyang Chen, Qi Zhang, and Yinglun Zhu. Efficient sequential decision making with large language models. *arXiv preprint arXiv:2406.12125*, 2024.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. *arXiv preprint arXiv:1810.08272*, 2018.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *Advances in Neural Information Processing Systems*, 36, 2024.

Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. Textworld: A learning environment for text-based games. In *Computer Games: 7th Workshop, CGW 2018, Held in Conjunction with the 27th International Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, July 13, 2018, Revised Selected Papers 7*, pp. 41–75. Springer, 2019.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl2: Fast reinforcement learning via slow reinforcement learning. 2016. *URL http://arxiv.org/abs/1611.02779*, 2016.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35: 18343–18362, 2022.

Jake Grigsby, Linxi Fan, and Yuke Zhu. Amago: Scalable in-context reinforcement learning for adaptive agents. *arXiv preprint arXiv:2310.09971*, 2023.

Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. Leveraging pre-trained large language models to construct and utilize world models for model-based task planning. *Advances in Neural Information Processing Systems*, 36:79081–79094, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv:2109.06780*, 2021.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. Wese: Weak exploration to strong exploitation for llm agents. *arXiv preprint arXiv:2404.07456*, 2024.

Shima Imani, Liang Du, and Harsh Shrivastava. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*, 2023.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. Llms can't plan, but can help planning in llm-modulo frameworks. *arXiv preprint arXiv:2402.01817*, 2024.

Nan Rosemary Ke, Danny P Sawyer, Hubert Soyer, Martin Engelcke, David P Reichert, Drew A Hudson, John Reid, Alexander Lerchner, Danilo Jimenez Rezende, Timothy P Lillicrap, et al. Can foundation models actively gather information in interactive environments to test hypotheses? *arXiv preprint arXiv:2412.06438*, 2024.

Byoungjip Kim, Youngsoo Jang, Lajanugen Logeswaran, Geon-Hyeong Kim, Yu Jin Kim, Honglak Lee, and Moontae Lee. Prospector: Improving llm agents with self-asking and trajectory ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 14958–14976, 2024.

Akshay Krishnamurthy, Keegan Harris, Dylan J Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context? *arXiv preprint arXiv:2403.15371*, 2024.

Heinrich Küttler, Nantas Nardelli, Alexander Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The nethack learning environment. *Advances in Neural Information Processing Systems*, 33:7671–7684, 2020.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, et al. In-context reinforcement learning with algorithm distillation. *arXiv preprint arXiv:2210.14215*, 2022.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. A systematic investigation of commonsense knowledge in large language models. *arXiv preprint arXiv:2111.00607*, 2021.

TP Lillicrap. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Evan Z Liu, Aditi Raghunathan, Percy Liang, and Chelsea Finn. Decoupling exploration and exploitation for meta-reinforcement learning without sacrifices. In *International conference on machine learning*, pp. 6925–6935. PMLR, 2021.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023.

Cong Lu, Shengran Hu, and Jeff Clune. Intelligent go-explore: Standing on the shoulders of giant foundation models. *arXiv preprint arXiv:2405.15143*, 2024.

Kaixin Ma, Hongming Zhang, Hongwei Wang, Xiaoman Pan, Wenhao Yu, and Dong Yu. Laser: Llm agent with state-space exploration for web navigation. *arXiv preprint arXiv:2309.08172*, 2023.

Bogdan Mazoure, Jake Bruce, Doina Precup, Rob Fergus, and Ankit Anand. Accelerating exploration and representation learning with offline pre-training. *arXiv preprint arXiv:2304.00046*, 2023.

Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Allen Nie, Yi Su, Bo Chang, Jonathan N Lee, Ed H Chi, Quoc V Le, and Minmin Chen. Evolve: Evaluating and optimizing llms for exploration. *arXiv preprint arXiv:2410.06238*, 2024.

Ben Norman and Jeff Clune. First-explore, then exploit: Meta-learning intelligent exploration. *arXiv preprint arXiv:2307.02276*, 2023.

OpenAI. OpenAI o1-mini. `https://platform.openai.com/docs/models#o1`, 2024. [Accessed 10-01-2025].

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, et al. Balrog: Benchmarking agentic llm and vlm reasoning on games. *arXiv preprint arXiv:2411.13543*, 2024.

Chanwoo Park, Xiangyu Liu, Asuman Ozdaglar, and Kaiqing Zhang. Do llm agents have regret? a case study in online learning and games. *arXiv preprint arXiv:2403.16843*, 2024.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. Logicbench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13679–13707, 2024.

Swarna Kamal Paul. Sequential planning in large partially observable environments guided by llms. *arXiv preprint arXiv:2312.07368*, 2023.

Wasu Top Piriyakulkij, Cassidy Langenfeld, Tuan Anh Le, and Kevin Ellis. Doing experiments and revising rules with natural language and probabilistic reasoning. *arXiv preprint arXiv:2402.06025*, 2024.

Mikayel Samvelyan, Robert Kirk, Vitaly Kurin, Jack Parker-Holder, Minqi Jiang, Eric Hambro, Fabio Petroni, Heinrich Küttler, Edward Grefenstette, and Tim Rocktäschel. Minihack the planet: A sandbox for open-ended reinforcement learning research. *arXiv preprint arXiv:2109.13202*, 2021.

Lukas Schäfer, Filippos Christianos, Josiah Hanna, and Stefano V Albrecht. Decoupling exploration and exploitation in reinforcement learning. In *ICML 2021 Workshop on Unsupervised Reinforcement Learning*, 2021.

Thomas Schmied, Fabian Paischer, Vihang Patil, Markus Hofmarcher, Razvan Pascanu, and Sepp Hochreiter. Retrieval-augmented decision transformer: External memory for in-context rl. *arXiv preprint arXiv:2410.07071*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 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, 2023.

Yifan Song, Weimin Xiong, Xiutian Zhao, Dawei Zhu, Wenhao Wu, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. Agentbank: Towards generalized llm agents via fine-tuning on 50000+ interaction trajectories. *arXiv preprint arXiv:2410.07706*, 2024.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024a.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size, 2024. *URL https://arxiv.org/abs/2408.00118*, 1(3), 2024b.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005, 2023.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024a.

Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, et al. Jarvis-1: Open-world multi-task agents with memory-augmented multimodal language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024b.

William F Whitney, Michael Bloesch, Jost Tobias Springenberg, Abbas Abdolmaleki, Kyunghyun Cho, and Martin Riedmiller. Decoupled exploration and exploitation policies for sample-efficient reinforcement learning. *arXiv preprint arXiv:2101.09458*, 2021.

Yue Wu, Xuan Tang, Tom M Mitchell, and Yuanzhi Li. Smartplay: A benchmark for llms as intelligent agents. *arXiv preprint arXiv:2310.01557*, 2023.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *arXiv preprint arXiv:2404.07972*, 2024.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. Are large language models really good logical reasoners? a comprehensive evaluation from deductive, inductive and abductive views. *arXiv preprint arXiv:2306.09841*, 2023.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022b.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agenttuning: Enabling generalized agent abilities for llms. *arXiv preprint arXiv:2310.12823*, 2023.

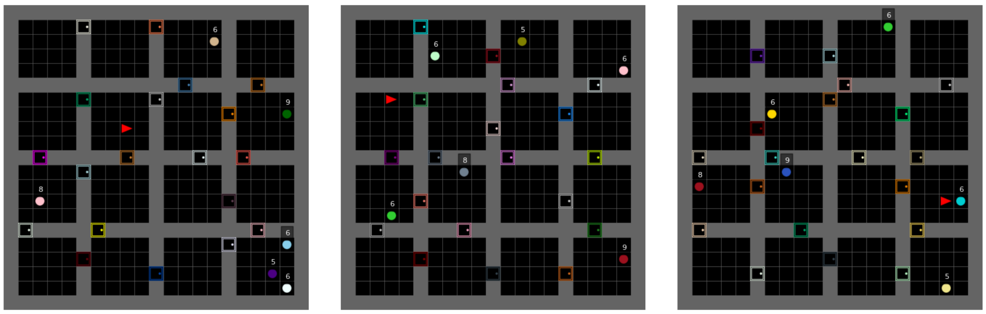
Jin Zhang, Jianhao Wang, Hao Hu, Tong Chen, Yingfeng Chen, Changjie Fan, and Chongjie Zhang. Metacure: Meta reinforcement learning with empowerment-driven exploration. In *International Conference on Machine Learning*, pp. 12600–12610. PMLR, 2021.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

A Evaluation Framework

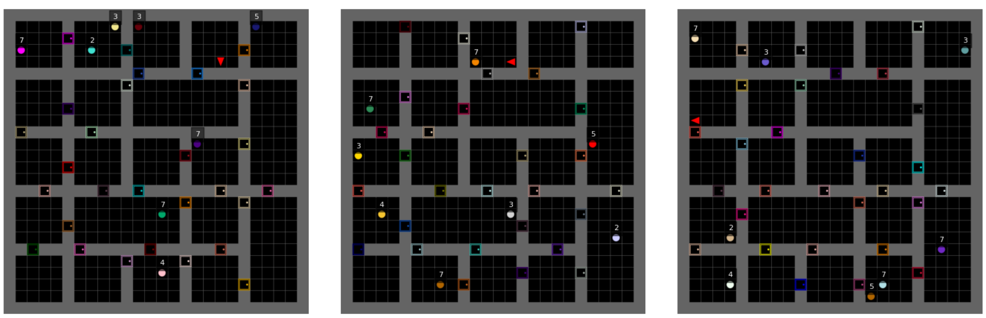
In this section, we provide further information about the environments used in in the main paper.

A.1 Dark Treasure Rooms



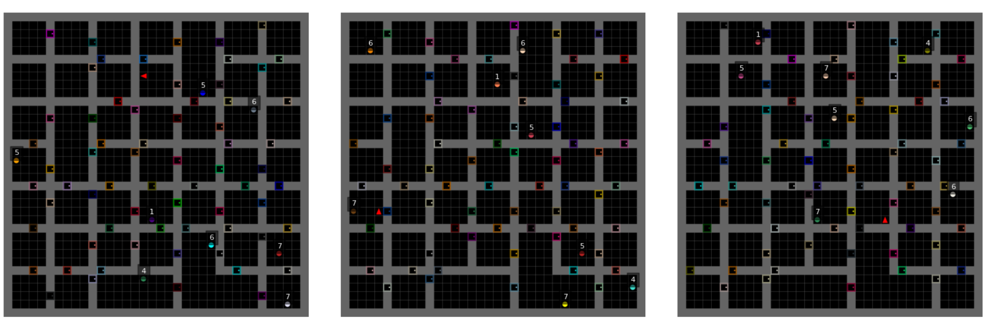
(a) (b) (c)

Figure 5: 4×4 variations of Dark Treasure Rooms.



(a) (b) (c)

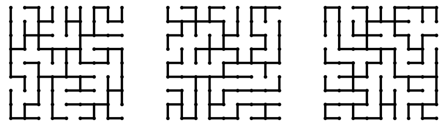
Figure 6: 5×5 variations of Dark Treasure Rooms.



(a) (b) (c)

Figure 7: 7×7 variations of Dark Treasure Rooms.

A.2 Maze



(a) (b) (c)

Figure 8: **The maze layouts used in the experiments.** Our variations are created using Kruskal's algorithm. The starting position is the center. Rewards are sampled uniformly from $U[1, 10]$ and distributed randomly across the maze. The agent can take a maximum of 15 steps in each episode.

A.3 Additional Environment Details

Implementation. Our environments are built upon the minigrid framework. To address the limitations of the low-level action and observation spaces, we use an abstraction wrapper function. This function transforms raw observations and actions into higher-level abstractions, allowing agents to reason more effectively about interactions with objects and doors within a room. By abstracting low-level details, the evaluation focuses more on the agent's exploration strategy. The abstraction wrapper defines the state as a discrete set of available objects and doors within the current room. Each room is uniquely configured to ensure that the agent can differentiate between them. The action space is simplified to interactions with specific objects or doors (e.g., choosing to open a door or collect an object). Once the agent selects an action at this abstract level, the wrapper translates it into a sequence of low-level actions using a predefined low-level policy to execute the selected behavior. Our wrapper can be seen as bridging the minigrid framework to a representation similar to text-environments like TextWorld, but minimizes narrative understanding and semantic associations. During the environment creation, we randomly drop doors and walls for Dark Treasure Rooms with probability $p_{\text{drop}} = 0.01$ to increase the difficulty. We experimented with different model temperatures settings but did not find significant differences. Therefore, we set the hyperparameter, due to technical constraints, to 1.0 for o1-mini and 0.1 for the other models.

Source code. We open-source the code for our evaluation and the minigrid-wrapper after the peer-review. We also publish our framework on Huggingface Spaces to facilitate experimentation.

A.4 Prompt Template

In-Context Learning Agent

Your past episodes: <HISTORY>

You are controlling an agent in an unknown world.
Over a total of 20 episodes, you can interact with objects in the environment.
<INSTRUCTION>
You have <MAX STEPS> door interactions per episode but can pick up three balls, keys, or boxes.

Current episode: <EPISODIC HISTORY>
Which object do you want to interact with next?
Reply with one object enclosed with <and >, e.g. <door>.
What is your next action?

Figure 9: **The template used to prompt the LLMs.** `<History>` is replaced with past episodes. `<MAX STEPS>` is set to 15 for the mazes and to the number of steps required to reach the furthest room in Dark Treasure Rooms. `<EPISODIC HISTORY>` shows the current agent-environment interactions. `<INSTRUCTION>` is where we command the agent to perform a specific task. In total we compare three different instructions. See appendix D for the specific details.

A.5 EXEMPLARY ENVIRONMENT INTERACTION

Targeted-Exploration Prompting

Your past episodes:
Episode 1:
You see:
dodger_blue door, tangerine door
>dodger_blue door
Reward: 0
You walk through the door. You see: apricot door, dodger_blue door, cerulean door, honeydew door
>cerulean door
Reward: 0
You walk through the door. You see: rosewood ball, teal door, cerulean door
>rosewood ball
Reward: 3
You see: teal door, cerulean door
>teal door
Reward: 0
You walk through the door. You see: turquoise ball, magenta door, teal door, khaki ball
>turquoise ball
Reward: 2
You see: magenta door, teal door, khaki ball
>khaki ball
Reward: 3
You see: magenta door, teal door

You are controlling an agent in an unknown world.
Over a total of 20 episodes, you can interact with objects in the environment.
You have 8 door interactions per episode but can pick up three balls, keys, or boxes.
Based the current and past trials, explore the environment to collect information that may help to become better at maximizing the reward.
Current episode:
You see:
dodger_blue door, tangerine door
>tangerine door
Reward: 0
You walk through the door. You see: midnight_blue ball, tangerine door
>midnight_blue ball
Reward: 5
You see: tangerine door
>tangerine door
Reward: 0
You walk through the door. You see: tangerine door, dodger_blue door
Which object do you want to interact with next?
Reply with one object enclosed with <and >,e.g. <door>. What is your next action?

Figure 10: **An exemplary prompt after multiple agent-environment interactions.** In the first episode, the agent agent traveled through multiple doors and collected three balls. In the current trial, the agent takes a different path and discovers another ball which leads to more reward than the others.

A.6 Optimal Exploitation

We formulate the computation of the optimal exploitation return R^{opt} as an instance of the Orienteering Problem. We represent the explored balls in the state-space of history h_t as a graph $G = (V, E)$, where V denotes the set of balls and E the paths between them. Each ball $i \in V$ is assigned a reward p_i, and each edge $(i, j) \in E$ has an associated travel cost c_{ij}. Starting at a designated initial state $s \in V$, the goal is to determine a path T through a subset of nodes $S \subseteq V$ that maximizes the cumulative rewards $\sum_{i \in S} p_i$, while ensuring that the total travel cost $\sum_{(i,j) \in T} c_{ij}$ does not exceed a predefined budget C_{max} and a maximum of C_{balls} balls is collected. Formally, this optimization problem is expressed as:

$$\max_{x_{ij}, y_i} \quad \sum_{i \in V} p_i y_i \tag{5a}$$

$$\text{subject to} \quad \sum_{(i,j) \in E} c_{ij} x_{ij} \leq C_{\text{max}}, \tag{5b}$$

$$\sum_{i \in V} y_i \leq C_{\text{balls}}, s \tag{5c}$$

$$\sum_{j \in V} x_{ij} - \sum_{j \in V} x_{ji} = \begin{cases} 1 & \text{if } i = s, \\ 0 & \text{otherwise}, \end{cases} \quad \forall i \in V, \tag{5d}$$

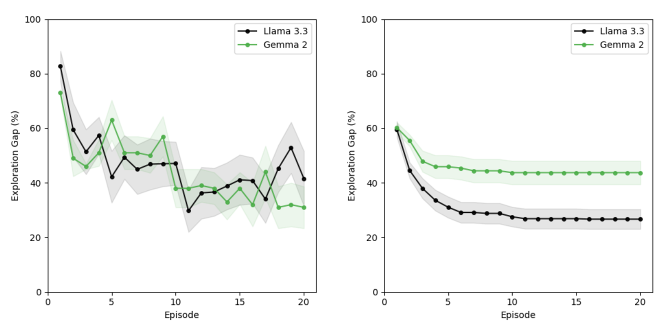
$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in E, \tag{5e}$$

$$y_i \in \{0, 1\}, \quad \forall i \in V. \tag{5f}$$

where x_{ij} and y_i are binary decision variables indicating whether edge (i, j) and ball i is visited, respectively. Exact solvers, such as integer linear programming methods, or heuristic algorithms can be used to solve the optimization problem. We use a brute force approach. Note that our framework is not bound to any specific exploitation method.

A.7 Measuring Exploration with Large Language Model Exploitation

In the following, we build an agent, in which Llama 3.3 conducts the exploitation of the current history. We tested if LLMs, similarly to optimal returns, can find reliable exploitation steps from a history of trajectories. We found in Figure 11 that LLMs struggle to achieve reliable and differentiable exploitation returns. These additional findings show that there is also a large research gap when it comes to understanding multiple sequential trajectories and finding the optimal sequence of actions with LLMs. We exemplary illustrate and hypothesize that, when exploration becomes more demanding, then LLM exploitation returns are similarly not reliable. For a proper assessment of LLM exploration, it is necessary to rely on exact methods, as proposed in our paper.

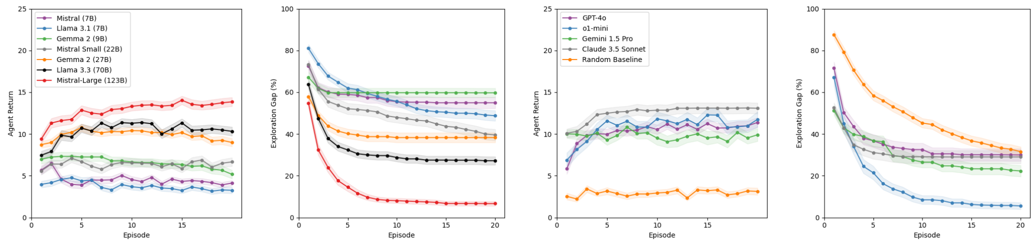


(a) LLM Exploitation (b) Optimal Exploitation

Figure 11: **Comparison of LLM and our optimal exploitation.** We find that LLM exploitation (left) is similarly hard to exploration and cannot reliably measure exploration progress. Instead, we propose to rely on exact methods (right) to fairly evaluate exploration.

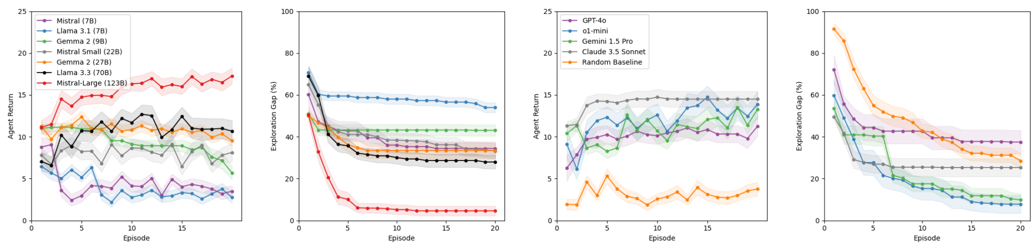
B Results - Dark Treasure Rooms

B.1 Learning Curves



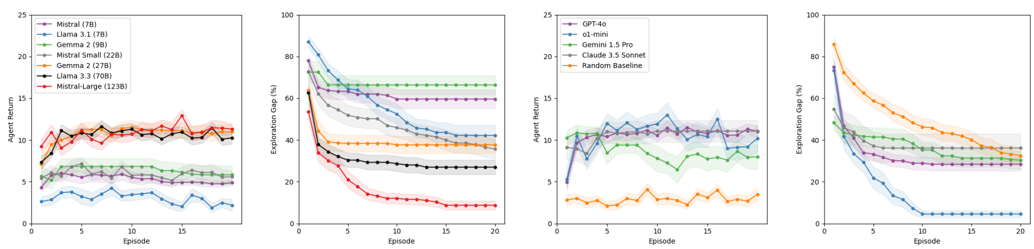
(a) Agent Return (b) Exploration Gap (c) Agent Return (d) Exploration Gap

Figure 12: Averaged over Dark Treasure Rooms 4×4, 5×5, and 7×7.



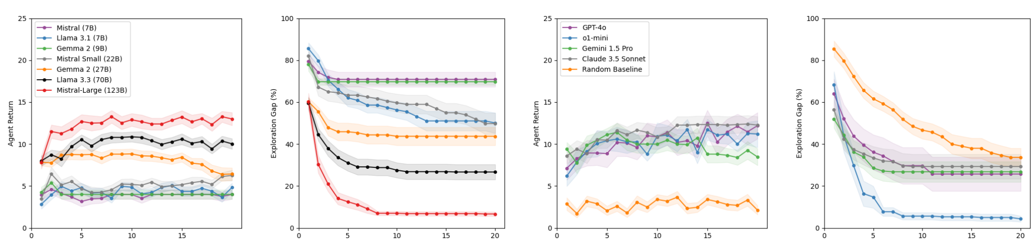
(a) Agent Return (b) Exploration Gap (c) Agent Return (d) Exploration Gap

Figure 13: 4×4 Dark Treasure Rooms.



(a) Agent Return (b) Exploration Gap (c) Agent Return (d) Exploration Gap

Figure 14: 5×5 Dark Treasure Rooms.



(a) Agent Return (b) Exploration Gap (c) Agent Return (d) Exploration Gap

Figure 15: 7×7 Dark Treasure Rooms.

B.2 TOTAL RETURN GAP DECOMPOSITION

Table 3: **Normalized total gap decomposition averaged over 4×4, 5×5, and 7×7 Dark Treasure Rooms.** The numbers in parentheses show the fraction that exploitation and exploration gaps make up of the total return gap.

		Last Episode			Mean		
		Total Gap	**Exploitation Gap**	**Exploration Gap**	**Total Gap**	**Exploitation Gap**	**Exploration Gap**
<30B	Mistral (7B)	0.80	0.25 (0.31)	0.55 (0.69)	0.78	0.20 (0.26)	0.57 (0.74)
	Llama 3.1 (7B)	0.84	0.35 (0.41)	0.49 (0.59)	0.82	0.24 (0.28)	0.57 (0.72)
	Gemma 2 (9B)	0.75	**0.15** (0.17)	0.60 (0.83)	0.69	**0.08** (0.10)	0.60 (0.90)
	Mistral Small (22B)	0.68	0.28 (0.41)	0.40 (0.59)	0.69	0.20 (0.29)	0.49 (0.71)
	Gemma 2 (27B)	**0.57**	0.18 (0.32)	**0.38** (0.68)	**0.52**	0.12 (0.22)	**0.40** (0.78)
>30B	Llama 3.3 (70B)	0.50	**0.23** (0.40)	0.27 (0.60)	0.50	**0.18** (0.32)	0.32 (0.68)
	Mistral Large (123B)	**0.33**	0.26 (0.75)	**0.07** (0.25)	**0.38**	0.25 (0.62)	**0.13** (0.38)
Closed	GPT-4o	0.44	0.14 (0.31)	0.30 (0.69)	0.49	0.14 (0.27)	0.35 (0.73)
	GPT-o1-mini	0.43	0.38 (0.79)	**0.06** (0.21)	0.47	0.32 (0.66)	**0.16** (0.34)
	Gemini 1.5 Pro	0.52	0.30 (0.58)	0.22 (0.42)	0.53	0.23 (0.41)	0.30 (0.59)
	Claude 3.5 Sonnet	**0.38**	**0.09** (0.29)	0.29 (0.71)	**0.41**	**0.09** (0.26)	0.32 (0.74)
Baseline	Random Walk	0.85	0.53 (0.62)	0.32 (0.38)	0.86	0.37 (0.42)	0.49 (0.58)

Table 4: **Normalized total gap decomposition of 4×4 Dark Treasure Rooms.** The numbers in parentheses show the fraction that exploitation and exploration gaps make up of the total return gap.

		Last Episode			Mean		
		Total Gap	**Exploitation Gap**	**Exploration Gap**	**Total Gap**	**Exploitation Gap**	**Exploration Gap**
<30B	Mistral (7B)	0.84	0.50 (0.59)	0.34 (0.41)	0.80	0.41 (0.51)	0.39 (0.49)
	Llama 3.1 (7B)	0.88	0.34 (0.36)	0.54 (0.64)	0.83	0.24 (0.26)	0.58 (0.74)
	Gemma 2 (9B)	0.74	0.31 (0.35)	0.43 (0.65)	0.58	**0.14** (0.20)	0.44 (0.80)
	Mistral Small (22B)	0.64	0.30 (0.46)	**0.33** (0.54)	0.64	0.24 (0.34)	0.40 (0.66)
	Gemma 2 (27B)	**0.57**	**0.24** (0.43)	**0.33** (0.57)	**0.52**	0.16 (0.30)	**0.36** (0.70)
>30B	Llama 3.3 (70B)	0.53	0.25 (0.35)	0.28 (0.65)	0.53	**0.18** (0.26)	0.34 (0.74)
	Mistral Large (123B)	**0.23**	**0.18** (0.79)	**0.05** (0.21)	**0.31**	0.21 (0.68)	**0.10** (0.32)
Closed	GPT-4o	0.50	0.12 (0.20)	0.37 (0.80)	0.55	0.12 (0.20)	0.43 (0.80)
	GPT-o1-mini	0.39	0.31 (0.77)	**0.08** (0.23)	0.47	0.27 (0.59)	**0.20** (0.41)
	Gemini 1.5 Pro	0.40	0.30 (0.82)	**0.10** (0.18)	0.51	0.27 (0.55)	0.23 (0.45)
	Claude 3.5 Sonnet	**0.35**	**0.10** (0.36)	0.25 (0.64)	**0.37**	**0.09** (0.29)	0.28 (0.71)
Baseline	Random Walk	0.83	0.55 (0.64)	0.28 (0.36)	0.86	0.39 (0.44)	0.47 (0.56)

Table 5: **Normalized total gap decomposition of 5×5 Dark Treasure Rooms.** The numbers in parentheses show the fraction that exploitation and exploration gaps make up of the total return gap.

		Last Episode			Mean		
		Total Gap	**Exploitation Gap**	**Exploration Gap**	**Total Gap**	**Exploitation Gap**	**Exploration Gap**
<30B	Mistral (7B)	0.75	0.16 (0.23)	0.60 (0.77)	0.73	0.12 (0.17)	0.62 (0.83)
	Llama 3.1 (7B)	0.89	0.47 (0.53)	0.42 (0.47)	0.85	0.30 (0.35)	0.55 (0.65)
	Gemma 2 (9B)	0.72	**0.05** (0.05)	0.66 (0.95)	0.69	0.02 (0.02)	0.67 (0.98)
	Mistral Small (22B)	0.72	0.36 (0.51)	**0.36** (0.49)	0.70	0.23 (0.34)	0.47 (0.66)
	Gemma 2 (27B)	**0.45**	0.07 (0.15)	0.38 (0.85)	**0.46**	**0.07** (0.13)	**0.40** (0.87)
>30B	Llama 3.3 (70B)	0.48	**0.21** (0.41)	0.27 (0.59)	0.47	**0.16** (0.33)	0.31 (0.67)
	Mistral Large (123B)	**0.43**	0.34 (0.79)	**0.09** (0.21)	**0.46**	0.29 (0.61)	**0.17** (0.39)
Closed	GPT-4o	**0.44**	0.16 (0.33)	0.28 (0.67)	**0.47**	0.14 (0.26)	0.33 (0.74)
	GPT-o1-mini	0.49	0.44 (0.77)	**0.05** (0.23)	**0.47**	0.32 (0.65)	**0.15** (0.35)
	Gemini 1.5 Pro	0.60	0.30 (0.47)	0.30 (0.53)	0.57	0.20 (0.35)	0.37 (0.65)
	Claude 3.5 Sonnet	0.45	**0.08** (0.25)	0.36 (0.75)	**0.47**	**0.09** (0.24)	0.38 (0.76)
Baseline	Random Walk	0.82	0.50 (0.60)	0.32 (0.40)	0.85	0.36 (0.41)	0.50 (0.59)

Table 6: **Normalized total gap decomposition of 7×7 Dark Treasure Rooms.** The numbers in parentheses show the fraction that exploitation and exploration gaps make up of the total return gap.

		Last Episode			Mean		
		Total Gap	**Exploitation Gap**	**Exploration Gap**	**Total Gap**	**Exploitation Gap**	**Exploration Gap**
<30B	Mistral (7B)	0.79	**0.09** (0.10)	0.71 (0.90)	0.80	0.08 (0.10)	0.71 (0.90)
	Llama 3.1 (7B)	0.75	0.25 (0.33)	0.50 (0.67)	0.78	0.19 (0.23)	0.59 (0.77)
	Gemma 2 (9B)	0.79	0.10 (0.10)	0.70 (0.90)	0.79	**0.09** (0.09)	0.70 (0.91)
	Mistral Small (22B)	0.68	0.18 (0.26)	0.50 (0.74)	0.74	0.14 (0.18)	0.60 (0.82)
	Gemma 2 (27B)	**0.67**	0.23 (0.36)	**0.44** (0.64)	**0.59**	0.13 (0.24)	**0.46** (0.76)
>30B	Llama 3.3 (70B)	0.49	**0.22** (0.45)	0.27 (0.55)	0.49	**0.18** (0.36)	0.31 (0.64)
	Mistral Large (123B)	**0.34**	0.27 (0.67)	**0.07** (0.33)	**0.37**	0.25 (0.58)	**0.12** (0.42)
Closed	GPT-4o	**0.38**	0.12 (0.38)	0.26 (0.62)	0.47	0.15 (0.36)	0.32 (0.64)
	GPT-o1-mini	0.43	0.38 (0.85)	**0.04** (0.15)	0.48	0.35 (0.73)	**0.13** (0.27)
	Gemini 1.5 Pro	0.57	0.30 (0.45)	0.27 (0.55)	0.51	0.21 (0.33)	0.30 (0.67)
	Claude 3.5 Sonnet	**0.38**	**0.08** (0.20)	0.29 (0.80)	**0.42**	**0.10** (0.21)	0.33 (0.79)
Baseline	Random Walk	0.89	0.55 (0.61)	0.34 (0.39)	0.86	0.35 (0.40)	0.50 (0.60)

B.3 OTHER STATISTICS

Table 7: Other statistics averaged over 4×4, 5×5, and 7×7 Dark Treasure Rooms.

		Exploitation Return	Agent Return	Coverage	Redundancy	Sample Efficiency
<30B	Mistral (7B)	9.47	4.13	38.88	0.92	13.82
	Llama 3.1 (7B)	10.56	3.27	37.03	0.92	31.25
	Gemma 2 (9B)	8.53	5.18	30.10	0.95	**7.25**
	Mistral Small (22B)	12.54	6.67	46.19	**0.91**	35.44
	Gemma 2 (27B)	**12.82**	**8.99**	**49.99**	0.92	14.13
>30B	Llama 3.3 (70B)	14.94	10.32	56.93	0.91	**19.31**
	Mistral Large (123B)	**19.20**	**13.86**	**78.59**	**0.88**	19.56
Closed	GPT-4o	14.22	11.38	54.33	0.92	20.73
	GPT-o1-mini	**19.40**	11.76	**87.73**	**0.86**	27.88
	Gemini 1.5 Pro	16.09	9.91	67.02	0.89	23.25
	Claude 3.5 Sonnet	15.03	**13.08**	58.00	0.91	**12.08**
Baseline	Random Walk	14.18	3.13	69.11	0.89	51.25

Table 8: Other statistics of 4×4 Dark Treasure Rooms.

		Exploitation Return	Agent Return	Coverage	Redundancy	Sample Efficiency
<30B	Mistral (7B)	14.57	3.50	50.00	0.92	19.67
	Llama 3.1 (7B)	10.30	2.77	44.39	**0.91**	21.50
	Gemma 2 (9B)	12.83	5.70	45.45	0.94	**9.33**
	Mistral Small (22B)	**15.00**	8.13	58.94	**0.91**	32.00
	Gemma 2 (27B)	**15.00**	**9.53**	**63.64**	0.92	15.33
>30B	Llama 3.3 (70B)	16.03	10.67	63.79	0.91	20.83
	Mistral Large (123B)	**21.33**	**17.27**	**83.48**	**0.89**	**18.83**
Closed	GPT-4o	14.07	11.27	60.61	0.92	19.00
	GPT-o1-mini	**20.67**	13.87	**92.73**	**0.88**	33.33
	Gemini 1.5 Pro	20.07	13.27	88.48	**0.88**	35.00
	Claude 3.5 Sonnet	16.73	**14.50**	58.48	0.92	**11.17**
Baseline	Random Walk	16.10	3.77	78.79	0.88	49.67

Table 9: Other statistics of the 5×5 Dark Treasure Rooms.

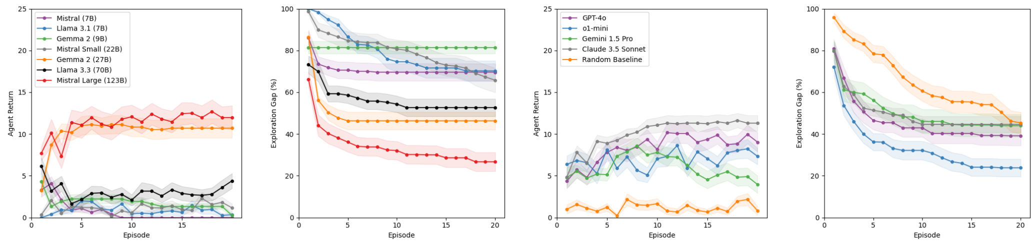
		Exploitation Return	Agent Return	Coverage	Redundancy	Sample Efficiency
<30B	Mistral (7B)	8.13	4.90	40.34	0.92	11.50
	Llama 3.1 (7B)	11.53	2.20	41.84	0.93	39.67
	Gemma 2 (9B)	6.83	5.83	22.99	0.96	**7.33**
	Mistral Small (22B)	**12.73**	5.60	48.28	**0.91**	41.00
	Gemma 2 (27B)	12.37	**11.00**	**50.23**	0.93	12.17
>30B	Llama 3.3 (70B)	14.40	10.27	60.34	0.91	**18.67**
	Mistral Large (123B)	**17.93**	**11.30**	**86.90**	**0.87**	26.67
Closed	GPT-4o	14.13	11.03	58.62	0.91	18.83
	GPT-o1-mini	**18.73**	10.20	**94.94**	**0.86**	31.00
	Gemini 1.5 Pro	13.80	8.00	71.03	0.89	25.33
	Claude 3.5 Sonnet	12.73	**11.07**	59.77	0.91	**13.00**
Baseline	Random Walk	13.37	3.50	72.53	0.89	57.67

Table 10: Other statistics of 7×7 Dark Treasure Rooms.

		Exploitation Return	Agent Return	Coverage	Redundancy	Sample Efficiency
<30B	Mistral (7B)	5.70	4.00	26.28	0.93	6.88
	Llama 3.1 (7B)	9.83	4.83	24.87	0.93	23.83
	Gemma 2 (9B)	5.93	4.00	21.86	0.95	**6.67**
	Mistral Small (22B)	9.90	6.27	31.35	**0.92**	25.83
	Gemma 2 (27B)	**11.10**	**6.43**	**36.09**	0.93	13.83
>30B	Llama 3.3 (70B)	14.40	10.03	46.67	0.91	**15.33**
	Mistral Large (123B)	**18.33**	**13.00**	**65.38**	**0.87**	16.50
Closed	GPT-4o	14.53	12.20	39.49	0.92	21.33
	GPT-o1-mini	**18.80**	11.20	**75.51**	**0.85**	25.33
	Gemini 1.5 Pro	14.40	8.47	41.54	0.91	14.67
	Claude 3.5 Sonnet	13.93	**12.27**	55.26	0.89	**13.85**
Baseline	Random Walk	13.07	2.13	56.03	0.88	42.17

C Results - Maze

C.1 Learning Curves



(a) Agent Return (b) Exploration Gap (c) Agent Return (d) Exploration Gap

Figure 16: Learning curves averaged over all maze variations.

C.2 Total Return Gap Decomposition

Table 11: **Normalized total gap decomposition averaged over all maze variations.** The numbers in parentheses show the fraction that exploitation and exploration gaps make up of the total gap.

		Last Episode			Mean		
		Total Gap	**Exploitation Gap**	**Exploration Gap**	**Total Gap**	**Exploitation Gap**	**Exploration Gap**
<30B	Mistral (7B)	1.00	0.30 (0.30)	0.70 (0.70)	0.97	0.26 (0.27)	0.71 (0.73)
	Llama 3.1 (7B)	0.99	0.28 (0.29)	0.70 (0.71)	0.96	0.17 (0.18)	0.79 (0.82)
	Gemma 2 (9B)	0.99	0.17 (0.17)	0.81 (0.83)	0.92	0.11 (0.11)	0.81 (0.89)
	Mistral Small (22B)	0.95	0.29 (0.31)	0.66 (0.69)	0.95	0.16 (0.17)	0.80 (0.83)
	Gemma 2 (27B)	**0.52**	**0.06** (0.09)	**0.46** (0.91)	**0.54**	**0.05** (0.08)	**0.49** (0.92)
>30B	Llama 3.3 (70B)	0.80	0.28 (0.29)	0.53 (0.71)	0.86	0.30 (0.31)	0.56 (0.69)
	Mistral Large (123B)	**0.48**	**0.21** (0.41)	**0.27** (0.59)	**0.51**	**0.17** (0.32)	**0.34** (0.68)
Closed	GPT-4o	0.61	0.22 (0.42)	0.39 (0.58)	0.64	0.18 (0.32)	0.46 (0.68)
	GPT-o1-mini	0.68	0.44 (0.63)	**0.24** (0.37)	0.70	0.36 (0.52)	**0.34** (0.48)
	Gemini 1.5 Pro	0.83	0.39 (0.54)	0.44 (0.46)	0.74	0.24 (0.35)	0.50 (0.65)
	Claude 3.5 Sonnet	**0.50**	**0.06** (0.10)	0.45 (0.90)	**0.56**	**0.06** (0.10)	0.50 (0.90)
Baseline	Random Walk	0.96	0.51 (0.53)	0.45 (0.47)	0.95	0.29 (0.31)	0.65 (0.69)

C.3 Other Statistics

Table 12: Other statistics averaged over all maze variations.

		Exploitation Return	**Agent Return**	**Coverage**	**Redundancy**	**Sample Efficiency**
<30B	Mistral (7B)	7.13	0.00	12.17	0.97	9.81
	Llama 3.1 (7B)	6.83	0.33	19.78	0.94	26.11
	Gemma 2 (9B)	4.33	0.27	10.75	0.97	**5.00**
	Mistral Small (22B)	8.03	1.17	18.24	**0.93**	39.00
	Gemma 2 (27B)	**12.17**	**10.70**	**25.06**	**0.93**	10.17
>30B	Llama 3.3 (70B)	10.90	4.40	34.27	0.91	**17.00**
	Mistral Large (123B)	**16.73**	**11.97**	**48.09**	**0.87**	21.04
Closed	GPT-4o	13.93	9.03	38.24	0.90	23.85
	GPT-o1-mini	**17.40**	7.33	**57.34**	**0.85**	31.67
	Gemini 1.5 Pro	12.90	3.97	28.76	0.90	25.36
	Claude 3.5 Sonnet	12.60	**11.30**	35.09	0.91	**17.31**
Baseline	Random Walk	12.60	0.83	42.77	0.88	54.67

D WHAT IMPACT DO AGENT INSTRUCTIONS HAVE ON EXPLORATION?

Our experiments aim to demonstrate how our evaluation framework can be used to analyze the impact of prompt-engineering on the exploration capability. In this context, we designed two agent instructions to test the limits of our agents. We compare their total return decomposition to our task-oriented exploration. All prompts used to evaluate the impact on the exploration are illustrated in Figure 17.

Soft Lower Bound: In order to create a lower exploration bound, we instruct the agent to strictly stick to known rewards. This way, we investigate if agents can perform constant exploitation. The prompt is shown on the left of Figure 17. We expect a continuously low exploitation and a high exploration gap.

Soft Upper Bound: We designed another prompt to measure the upper exploration performance. The instruction is supposed to direct the agent towards an aimless exploration as we do not mention a task. This also allows us to investigate if goal orientation hinders exploration. As high optimal exploitation returns correlate with state-space coverage in our environments, we expect this setting to achieve the lowest exploration gap.

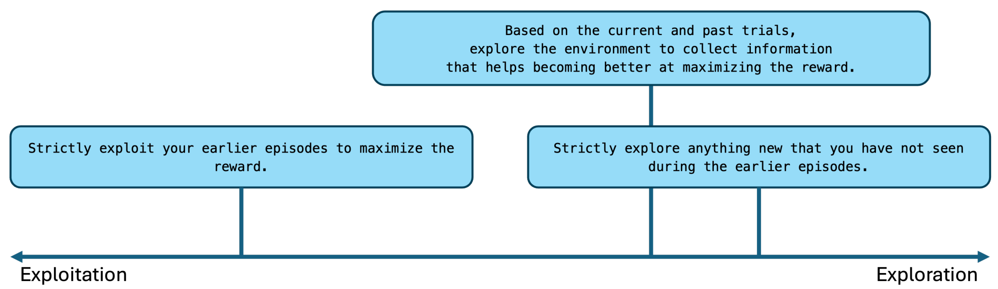
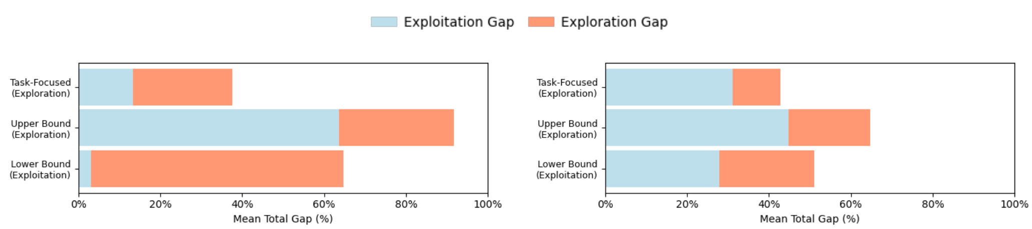
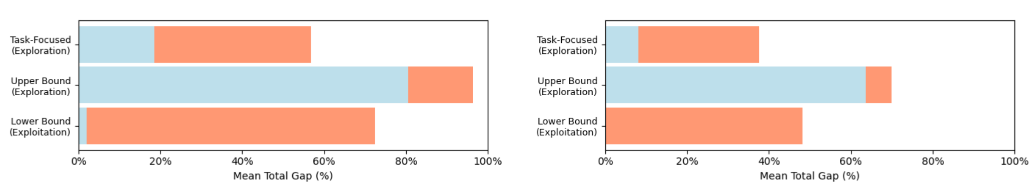


Figure 17: **The agent instructions used for the task-oriented exploration as well as our soft lower and upper bounds.** The Figure displays the prompts on a hypothetical and subjective continuum which ranges from strict exploitation to exploration.

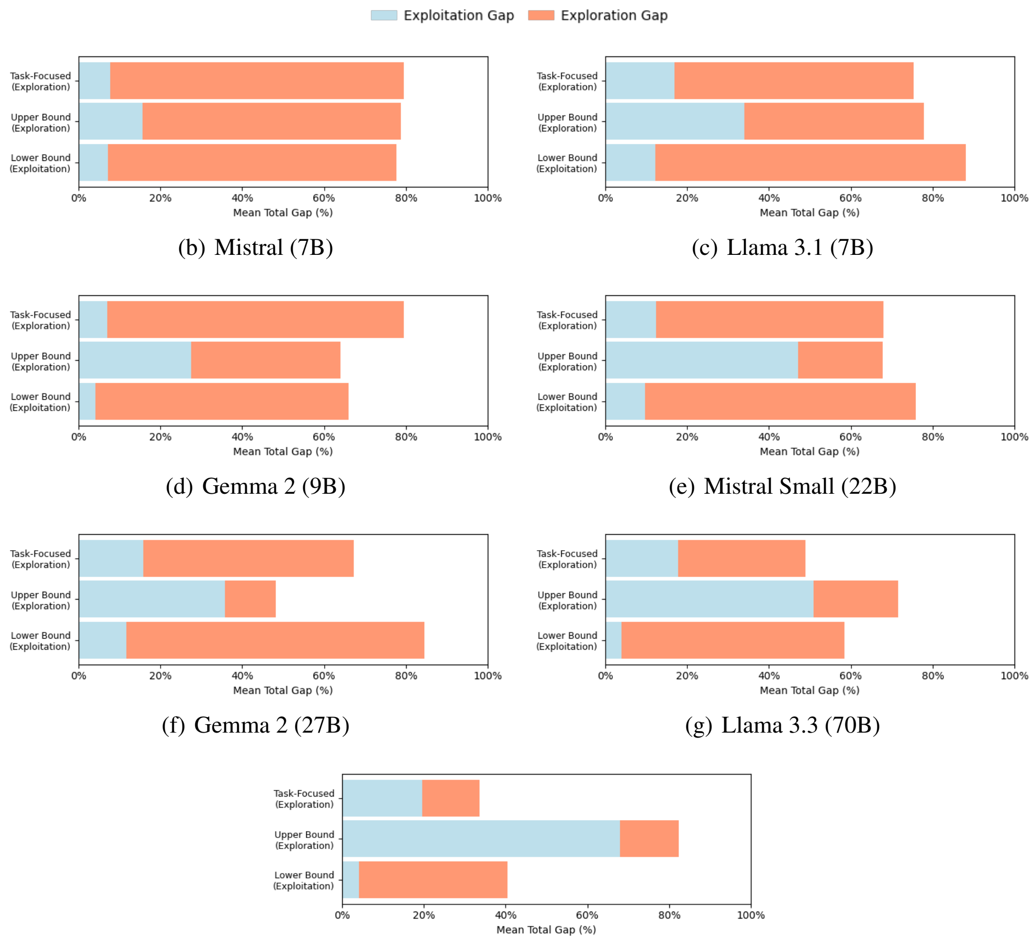


(b) GPT-4o (c) o1-mini



(d) Gemini 1.5 Pro (e) Claude 3.5 Sonnet

Figure 18: Impact of the agent instruction on the total return gap decomposition for closed models.



(b) Mistral (7B)

(c) Llama 3.1 (7B)

(d) Gemma 2 (9B)

(e) Mistral Small (22B)

(f) Gemma 2 (27B)

(g) Llama 3.3 (70B)

(h) Mistral Large (123B)

Figure 19: Impact of the agent instruction on the total return gap decomposition for closed models.

D.1 SOFT EXPLORATION LOWER BOUND

Table 13: **Normalized total return gap decomposition of** 7×7 **Dark Treasure Rooms.** The numbers in parentheses show the fraction that exploitation and exploration gaps make up of the total return gap.

		Last Episode			Mean		
		Total Gap	**Exploitation Gap**	**Exploration Gap**	**Total Gap**	**Exploitation Gap**	**Exploration Gap**
<30B	Mistral (7B)	0.78	0.09 (0.10)	0.69 (0.90)	0.79	0.08 (0.09)	0.71 (0.91)
	Llama 3.1 (7B)	0.88	0.25 (0.25)	0.63 (0.75)	0.81	0.13 (0.14)	0.68 (0.86)
	Gemma 2 (9B)	**0.66**	**0.00** (0.00)	0.66 (1.00)	**0.72**	**0.06** (0.06)	0.66 (0.94)
	Mistral Small (22B)	0.76	0.21 (0.26)	**0.55** (0.74)	0.74	0.10 (0.13)	**0.65** (0.87)
	Gemma 2 (27B)	0.85	0.21 (0.22)	0.64 (0.78)	0.79	0.12 (0.14)	0.67 (0.86)
>30B	Llama 3.3 (70B)	0.58	**0.06** (0.08)	0.53 (0.92)	0.60	0.05 (0.07)	0.55 (0.93)
	Mistral Large (123B)	**0.40**	**0.06** (0.13)	**0.34** (0.87)	**0.44**	**0.04** (0.10)	**0.40** (0.90)
Closed	GPT-4o	0.65	0.04 (0.06)	0.61 (0.94)	0.66	0.03 (0.05)	0.63 (0.95)
	o1-mini	0.51	0.43 (0.82)	**0.08** (0.18)	0.50	0.28 (0.55)	**0.21** (0.45)
	Gemini 1.5 Pro	0.73	0.06 (0.07)	0.66 (0.93)	0.69	0.03 (0.03)	0.66 (0.97)
	Claude 3.5 Sonnet	**0.48**	**0.00** (0.00)	0.48 (1.00)	**0.48**	**0.00** (0.00)	0.48 (1.00)
Baseline	Random Walk	0.89	0.55 (0.61)	0.34 (0.39)	0.86	0.35 (0.40)	0.50 (0.60)

Table 14: Percentage change in the total return gap decomposition of 7×7 Dark Treasure Rooms.

		Last Episode			Mean		
		Total Gap	**Exploitation Gap**	**Exploration Gap**	**Total Gap**	**Exploitation Gap**	**Exploration Gap**
<30B	Mistral (7B)	-2.10	1.23 (-0.85)	-2.50 (0.09)	-0.72	-1.31 (-5.21)	-0.65 (0.56)
	Llama 3.1 (7B)	16.93	-1.91 (-22.82)	26.43 (11.24)	4.71	-31.25 (-38.33)	16.40 (11.17)
	Gemma 2 (9B)	-16.80	-100.00 (-100.00)	-5.26 (10.70)	-8.35	-28.40 (-28.52)	-5.82 (2.77)
	Mistral Small (22B)	11.76	17.04 (-0.54)	9.83 (0.19)	0.78	-30.79 (-30.00)	8.03 (6.69)
	Gemma 2 (27B)	25.70	-11.53 (-38.55)	45.72 (21.83)	34.50	-5.38 (-41.08)	45.76 (12.75)
>30B	Llama 3.3 (70B)	19.50	-73.66 (-81.69)	96.81 (67.30)	22.00	-73.53 (-81.42)	78.20 (46.57)
	Mistral Large (123B)	19.95	-76.79 (-80.06)	409.96 (163.94)	18.54	-82.24 (-82.47)	219.95 (115.75)
Closed	GPT-4o	71.96	-66.32 (-84.95)	135.81 (51.97)	39.42	-77.77 (-86.81)	93.63 (48.09)
	o1-mini	19.24	11.00 (-2.86)	92.31 (15.61)	2.88	-19.92 (-24.88)	65.47 (66.61)
	Gemini 1.5 Pro	27.64	-79.59 (-84.88)	147.64 (69.16)	36.86	-86.76 (-91.17)	122.21 (44.49)
	Claude 3.5 Sonnet	28.13	-100.00 (-100.00)	64.53 (24.60)	13.78	-100.00 (-100.00)	48.43 (27.18)

Table 15: Other statistics of 7×7 Dark Treasure Rooms.

		Exploitation Return	**Agent Return**	**Coverage**	**Redundancy**	**Sample Efficiency**
<30B	Mistral (7B)	6.07	4.33	21.86	0.95	10.43
	Llama 3.1 (7B)	7.20	2.33	16.41	**0.94**	18.50
	Gemma 2 (9B)	6.67	**6.67**	13.33	0.97	**5.00**
	Mistral Small (22B)	**8.90**	4.70	**26.47**	**0.94**	27.67
	Gemma 2 (27B)	7.13	3.00	20.51	0.95	14.33
>30B	Llama 3.3 (70B)	9.30	8.13	26.22	0.95	**12.17**
	Mistral Large (123B)	**12.87**	**11.63**	**38.08**	**0.92**	21.60
Closed	GPT-4o	7.73	6.93	25.64	0.95	12.67
	GPT-o1-mini	**18.00**	9.60	**62.56**	**0.88**	36.33
	Gemini 1.5 Pro	6.60	5.40	14.10	0.96	**5.00**
	Claude 3.5 Sonnet	10.20	**10.20**	23.59	0.96	**5.00**
Baseline	Random Walk	13.07	2.13	56.03	0.88	42.17

Table 16: Percentage change in the total return gap decomposition of 7×7 Dark Treasure Rooms.

		Exploitation Return	**Agent Return**	**Coverage**	**Redundancy**	**Sample Efficiency**
<30B	Mistral (7B)	6.43	8.33	-16.83	2.86	51.78
	Llama 3.1 (7B)	-26.78	-51.72	-34.02	0.61	-22.38
	Gemma 2 (9B)	12.36	66.67	-39.00	1.95	-25.00
	Mistral Small (22B)	-10.10	-25.00	-15.54	1.81	7.10
	Gemma 2 (27B)	-35.74	-53.37	-43.16	2.47	3.61
>30B	Llama 3.3 (70B)	-35.42	-18.94	-43.82	4.01	-20.65
	Mistral Large (123B)	-29.82	-10.51	-41.76	5.94	30.91
Closed	GPT-4o	-46.79	-43.17	-35.06	3.15	-40.63
	o1-mini	-4.26	-14.29	-17.15	3.11	43.42
	Gemini 1.5 Pro	-54.17	-36.22	-66.05	5.78	-65.91
	Claude 3.5 Sonnet	-26.79	-16.85	-57.31	7.00	-63.89

D.2 SOFT EXPLORATION UPPER BOUND

Table 17: **Normalized total return gap decomposition of** 7×7 **Dark Treasure Rooms.** The numbers in parentheses show the fraction that exploitation and exploration gaps make up of the total return gap.

		Last Episode			Mean		
		Total Gap	**Exploitation Gap**	**Exploration Gap**	**Total Gap**	**Exploitation Gap**	**Exploration Gap**
<30B	Mistral (7B)	0.79	**0.21** (0.24)	0.57 (0.76)	0.78	**0.18** (0.20)	0.60 (0.80)
	Llama 3.1 (7B)	0.78	0.45 (0.55)	0.33 (0.45)	0.75	0.35 (0.44)	0.40 (0.56)
	Gemma 2 (9B)	0.64	0.38 (0.58)	0.26 (0.42)	0.61	0.27 (0.43)	0.34 (0.57)
	Mistral Small (22B)	0.68	0.57 (0.85)	0.11 (0.15)	0.63	0.45 (0.70)	0.18 (0.30)
	Gemma 2 (27B)	**0.48**	0.45 (0.91)	**0.03** (0.09)	**0.52**	0.41 (0.74)	**0.12** (0.26)
>30B	Llama 3.3 (70B)	**0.72**	**0.63** (0.88)	0.09 (0.12)	**0.71**	**0.50** (0.71)	0.21 (0.29)
	Mistral Large (123B)	0.82	0.81 (0.99)	**0.01** (0.01)	0.81	0.69 (0.83)	**0.12** (0.17)
Closed	GPT-4o	0.92	0.86 (0.93)	0.06 (0.07)	0.80	0.56 (0.69)	0.24 (0.31)
	o1-mini	**0.65**	**0.59** (0.84)	0.06 (0.16)	**0.70**	**0.48** (0.69)	0.22 (0.31)
	Gemini 1.5 Pro	0.96	0.91 (0.94)	0.05 (0.06)	0.85	0.75 (0.84)	0.11 (0.16)
	Claude 3.5 Sonnet	0.70	0.70 (1.00)	**0.00** (0.00)	0.76	0.71 (0.91)	**0.05** (0.09)
Baseline	Random Walk	0.89	0.55 (0.61)	0.34 (0.39)	0.86	0.35 (0.40)	0.50 (0.60)

Table 18: Percentage change in the total return gap decomposition of 7×7 Dark Treasure Rooms.

		Last Episode			Mean		
		Total Gap	**Exploitation Gap**	**Exploration Gap**	**Total Gap**	**Exploitation Gap**	**Exploration Gap**
<30B	Mistral (7B)	-0.87	148.21 (139.65)	-18.89 (-15.52)	-2.38	107.55 (104.17)	-15.41 (-11.24)
	Llama 3.1 (7B)	3.40	77.68 (65.65)	-34.03 (-32.32)	-3.47	83.17 (93.99)	-31.63 (-27.38)
	Gemma 2 (9B)	-19.31	295.83 (499.07)	-63.01 (-53.41)	-22.40	207.05 (386.99)	-51.30 (-37.56)
	Mistral Small (22B)	-0.37	212.75 (226.00)	-78.14 (-79.81)	-14.77	226.15 (281.42)	-70.10 (-62.79)
	Gemma 2 (27B)	-28.18	91.27 (151.68)	-92.38 (-85.89)	-10.56	215.73 (212.76)	-74.49 (-66.01)
>30B	Llama 3.3 (70B)	46.52	182.46 (95.38)	-66.28 (-78.57)	45.61	177.53 (95.43)	-32.02 (-54.58)
	Mistral Large (123B)	144.41	201.33 (46.88)	-85.06 (-96.00)	117.56	177.52 (41.58)	-2.29 (-58.36)
Closed	GPT-4o	143.82	623.63 (144.12)	-77.76 (-88.18)	68.45	272.84 (94.56)	-26.10 (-52.38)
	o1-mini	51.48	53.81 (-0.33)	30.77 (1.79)	46.32	37.07 (-5.22)	71.72 (13.97)
	Gemini 1.5 Pro	69.58	204.44 (109.78)	-81.35 (-89.45)	69.06	262.88 (154.71)	-64.75 (-75.49)
	Claude 3.5 Sonnet	85.84	740.00 (406.57)	-100.00 (-100.00)	78.58	614.58 (325.05)	-84.62 (-88.36)

Table 19: Other statistics of 7×7 Dark Treasure Rooms.

		Exploitation Return	**Agent Return**	**Coverage**	**Redundancy**	**Sample Efficiency**
<30B	Mistral (7B)	8.36	4.14	35.16	0.92	**15.95**
	Llama 3.1 (7B)	13.17	4.30	34.49	0.86	22.33
	Gemma 2 (9B)	14.70	7.07	52.76	0.89	26.83
	Mistral Small (22B)	17.50	6.30	57.69	0.87	25.50
	Gemma 2 (27B)	**19.00**	**10.13**	**71.15**	**0.85**	25.67
>30B	Llama 3.3 (70B)	17.87	**5.53**	81.60	0.83	30.33
	Mistral Large (123B)	**19.47**	3.50	**90.83**	**0.81**	**32.83**
Closed	GPT-4o	18.53	1.60	81.15	0.83	38.33
	o1-mini	18.53	**6.87**	82.05	0.83	**40.00**
	Gemini 1.5 Pro	18.67	0.73	78.46	**0.73**	19.67
	Claude 3.5 Sonnet	**20.00**	6.00	**86.54**	0.83	15.00
Baseline	Random Walk	13.07	2.13	56.03	0.88	42.17

Table 20: Percentage change in the total return gap decomposition of 7×7 Dark Treasure Rooms.

		Exploitation Return	**Agent Return**	**Coverage**	**Redundancy**	**Sample Efficiency**
<30B	Mistral (7B)	46.62	3.57	33.80	-0.93	132.03
	Llama 3.1 (7B)	33.90	-11.03	38.66	-7.99	-6.29
	Gemma 2 (9B)	147.75	76.67	141.35	-6.31	302.50
	Mistral Small (22B)	76.77	0.53	84.05	-4.83	-1.29
	Gemma 2 (27B)	71.17	57.51	97.16	-8.19	85.54
>30B	Llama 3.3 (70B)	24.07	-44.85	74.86	-8.28	97.83
	Mistral Large (123B)	6.18	-73.08	38.92	-6.69	98.99
Closed	GPT-4o	27.52	-86.89	105.52	-9.42	79.69
	o1-mini	-1.42	-38.69	8.66	-1.91	57.89
	Gemini 1.5 Pro	29.63	-91.34	88.89	-19.41	34.09
	Claude 3.5 Sonnet	43.54	-51.09	56.61	-7.52	8.33

E Additional Future Implications

Besides the research directions in the main paper, we believe the following paths are interesting to investigate in combination with our evaluation framework:

Memory architectures. Many agent systems currently focus on developing memory architectures Wang et al. (2024b); Anokhin et al. (2024) that load the LLM context dynamically. Testing such architectures for systematic exploration with an optimal exploitation, should be an important dimension in the evaluation. Further, building a meta memory that allows to reuse exploration-specific information across tasks and environments may be a fruitful path.

Improving sequence understanding. Similarly to Paglieri et al. (2024), we found that agents increasingly produce invalid actions and misunderstand longer sequences of past actions. Developing agents specifically made for long-sequence understanding is crucial for future LLM-based agents. It would be interesting to see if LLMs finetuned on sequential data Kim et al. (2024); Zeng et al. (2023); Song et al. (2024) also exhibit better exploration capabilities.