Exploration with Foundation Models: Capabilities, Limitations, and Hybrid Approaches

Remo Sasso **Michelangelo Conserva** **Dominik Jeurissen** **Paulo Rauber**
School of Electronic Engineering and Computer Science
Queen Mary University of London, United Kingdom
{r.sasso, m.conserva, d.jeurissen, p.rauber}@qmul.ac.uk

Abstract

Exploration in reinforcement learning (RL) remains challenging, particularly in sparse-reward settings. While foundation models possess strong semantic priors, their capabilities as zero-shot exploration agents in classic RL benchmarks are not well understood.We benchmark LLMs and VLMs on multi-armed bandits, Gridworlds, and sparse-reward Atari to test zero-shot exploration. Our investigation reveals a key limitation: while VLMs can infer high-level objectives from visual input, they consistently fail at precise low-level control—the "knowing–doing gap". To analyze a potential bridge for this gap, we investigate a simple on-policy hybrid framework in a controlled, best-case scenario. Our results in this idealized setting show that VLM guidance can significantly improve early-stage sample efficiency, providing a clear analysis of the potential and constraints of using foundation models to guide exploration rather than for end-to-end control.

1 Introduction

Reinforcement learning (RL) provides a framework for sequential decision-making, where an agent interacts with an environment to maximize cumulative rewards [Sutton and Barto, 2018]. A fundamental challenge in RL is exploration—the need to efficiently discover high-value states rather than prematurely exploiting suboptimal strategies. In sparse-reward settings, where rewards are infrequent, traditional exploration heuristics such as random actions or uncertainty-based methods can be highly inefficient, often requiring millions of interactions to uncover meaningful solutions [Bellemare et al., 2016]. While advanced model-based methods like posterior sampling have been developed to improve this sample efficiency [Sasso et al., 2023], our work explores an alternative direction by leveraging the semantic priors of large foundation models.

Large language models (LLMs) and vision-language models (VLMs) have recently demonstrated strong reasoning, semantic understanding, and in-context learning capabilities [Brown et al., 2020, Team et al., 2023, Achiam et al., 2023, Touvron et al., 2023, Jiang et al., 2024, Team et al., 2024, DeepSeek-AI et al., 2025]. Unlike RL agents, which rely on trial-and-error learning, LLMs can infer objectives, recognize patterns, and generate structured action sequences with minimal experience [Wang et al., 2024b, Du et al., 2023, Wang et al., 2024a]. This raises an important question: *How do LLMs and VLMs perform in traditional hard-exploration settings, and could they be leveraged to improve performance?*

Recent studies have established that LLMs can perform in-context exploration in multi-armed bandits (MABs), though performance is sensitive to complex prompting and history summarization [Krishnamurthy et al., 2024]. Our work complements these findings by analyzing the impact of simple, general-purpose instruction phrasing (implicit vs. explicit). However, as bandits lack state transitions and long-term planning, a broader evaluation is necessary to assess their potential for general RL. To address this gap, we conduct a systematic benchmark of foundation models as

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

zero-shot agents across a progression of classic exploration environments: from Bernoulli MABs, to spatial reasoning in Gridworlds, and finally to high-dimensional, sparse-reward Atari games.

This paper makes the following contributions:

- **A Systematic Benchmark of Foundation Models in RL Exploration.** We evaluate a range of LLMs and VLMs on a progression of classic exploration tasks to provide a clear empirical snapshot of their zero-shot capabilities.
- **A Characterization of VLM Failure Modes.** Through a detailed qualitative analysis of VLM agents in hard-exploration Atari games, we identify and document a persistent "knowing-doing gap," encompassing challenges in motor control, semantic grounding, and contextual reasoning.
- **An Upper-Bound Analysis of a Hybrid Approach.** We provide a quantitative analysis of a simple hybrid framework in a carefully selected, near-ideal environment. This controlled study serves to establish a potential upper bound on the sample efficiency gains achievable with this approach, rather than to propose it as a general-purpose solution.

To investigate these contributions, we conduct a systematic evaluation across a hierarchy of increasingly complex environments. We begin with Multi-Armed Bandits to isolate the exploration-exploitation trade-off, move to Gridworlds to study structured, memory-based exploration, and finally, extend our analysis to hard-exploration Atari games using Vision-Language Models to assess their capabilities with high-dimensional visual input.

The remainder of this paper is organized as follows. Section 2 reviews related literature. Section 3 presents our benchmark of text-based agents in bandits and Gridworlds. Section 4 provides our benchmark of vision-language models in Atari, including a detailed qualitative analysis of their failure modes. Section 5 investigates our hybrid framework as a proof-of-concept. Finally, Section 6 summarizes our findings.

2 Related Work

Recent research has explored the integration of foundation models into reinforcement learning from multiple perspectives. We position our work in relation to two primary research threads: the evaluation of foundation models as zero-shot decision-makers and the development of hybrid frameworks that combine them with traditional RL agents.

2.1 Foundation Models as Zero-Shot Decision-Makers

A growing body of work evaluates the innate capabilities of foundation models as autonomous agents. In the classic exploration-exploitation testbed of multi-armed bandits, studies by Krishnamurthy et al. [2024] and Wu et al. [2023] have shown that LLMs can execute exploration strategies, though performance is highly sensitive to detailed prompting. The trend towards more comprehensive evaluation is exemplified by benchmarks like AgentBench [Liu et al., 2024], which systematically assessed LLM agents across eight distinct environments and found that even capable models exhibit aimless behavior in long-horizon tasks.

In more complex visual domains, recent benchmarks have sought to understand VLM capabilities. The Atari-GPT benchmark [Waytowich et al., 2024] evaluated VLMs on dense-reward Atari games, finding that they struggle with the precise temporal reasoning required for fine-grained control. Concurrently, the BALROG benchmark [Paglieri et al., 2025] identified a significant *knowing-doing gap* in complex procedural games like NetHack, where models can often describe the optimal strategy but fail to execute the necessary low-level actions. To further isolate this issue, the TextAtari benchmark [Li et al., 2025] circumvents pixel-based control by converting Atari states into rich textual descriptions. This work demonstrates that when freed from visual-grounding challenges, text-based LLMs can reason and plan much more effectively over long horizons. Collectively, these studies reinforce the idea that the primary bottleneck for current VLMs is not a lack of high-level understanding, but a failure in low-level, pixel-to-action control. Our work directly builds on these insights by conducting a targeted analysis on a suite of classic *sparse-reward*, hard-exploration Atari games, specifically probing VLM reasoning where extrinsic signals are rare.

Listing 1: The prompt versions used for the bandit settings.

```
// Implicit (v1)
Your goal is to maximize the total reward by pulling the arm with the
    highest probability of success.
// Explicit (v2)
Your goal is to maximize the total reward by finding out which arm has
     the highest probability of success.
```

2.2 Hybrid Frameworks for RL and Foundation Models

To bridge the knowing-doing gap, a prominent research direction integrates foundation models with traditional RL algorithms. These hybrid frameworks leverage the strengths of both paradigms. One popular approach is to use the foundation model as an auxiliary component for shaping intrinsic rewards, as seen in Motif [Klissarov et al., 2024], where an LLM's judgment of "interestingness" guides an RL agent to SOTA performance in NetHack.

Another powerful approach uses FMs to guide the exploration process itself. For example, Intelligent Go-Explore (IGE) [Lu et al., 2025] replaces the hand-crafted heuristics of the Go-Explore algorithm with a GPT-4 model that identifies promising states to return to, solving hard-exploration games where prior LLM agents failed. Other sophisticated frameworks like ACE [Wan et al., 2025] integrate the LLM as an offline planner and critic in an actor-critic loop, enabling improved performance on industrial-scale control problems.

In contrast to these more complex integrations, our work investigates a simpler, direct-control intervention. We propose an on-policy hybrid framework where the VLM acts as a temporary, exploratory guide for a standard PPO agent. The goal is not to engineer a reward function or build a co-evolutionary system, but rather to use the VLM's zero-shot semantic understanding to steer the agent to promising regions of the state space during the earliest stages of training. This allows us to directly measure the impact of VLM guidance on sample efficiency without introducing complex hierarchical structures.

3 LLM Exploration Study

In this section, we evaluate the performance of LLMs in two structured exploration settings: multi-armed bandits (MABs) and Gridworlds. We investigate whether LLMs can infer the need for exploration through prompt phrasing in MABs (Section 3.1) and how they adapt to increasing prompt complexity in Gridworlds (Section 3.2).

3.1 Multi-Armed Bandits

Multi-Armed Bandits (MABs) provide a simplified framework for studying exploration strategies in reinforcement learning (RL). Since bandits do not involve state transitions, they isolate the exploration-exploitation trade-off, making them a useful testbed for evaluating how LLMs reason about exploration. In our experiments, we use Bernoulli bandits, where each arm provides rewards sampled from a Bernoulli distribution. A formal definition of MABs, including the Bernoulli bandits used in our experiments, is provided in Appendix A.

Prior studies have analyzed LLM exploration in bandit settings but primarily focused on complex reasoning techniques, such as external summarization and chain-of-thought prompting, rather than studying the effect of simple prompt phrasing and its impact on inference capabilities [Krishnamurthy et al., 2024]. To address this gap, we investigate whether LLMs can infer the need for exploration from how instructions are phrased.

Prompts. We compare two general-purpose prompting strategies, *implicit prompting (v1)* and *explicit prompting (v2)*, each with a different level of specificity (Listing 1). Implicit prompting (v1) provides a broad objective, requiring LLMs to infer exploration needs, whereas explicit prompting (v2) directly instructs exploration. By contrasting these two conditions, we analyze whether LLMs naturally adopt exploration strategies or require explicit guidance to behave optimally.

Figure 1: Averaged regret of the bandit experiments with $k = 3$ and $k = 5$ for various LLMs, Thompson Sampling, and UCB.

Algorithms. We evaluate four LLMs—GPT-3.5, GPT-4, Gemini 1.0, and Gemini 1.5—spanning different architectures and parameter scales [Achiam et al., 2023, Team et al., 2023]. These models were selected based on API availability and performance comparisons in prior work [Chiang et al., 2024]. As baselines, we use two classical exploration algorithms: Thompson Sampling [Thompson, 1933] and Upper Confidence Bound (UCB) [Auer, 2002], which provide strong theoretical guarantees for exploration efficiency. Further details on these algorithms are provided in Appendix A.

Evaluation. We first conduct a general evaluation of LLM performance in multi-armed bandits by analyzing their behavior in three-armed and five-armed bandit settings. In this setup, the success probabilities of each Bernoulli arm are sampled from $U(0, 1)$, rather than using fixed reward structures. This ensures greater variability across runs, allowing us to test whether LLMs can explore effectively in unstructured reward distributions. To assess performance, we measure regret, a standard metric in bandit problems that quantifies the difference between the cumulative reward an agent could have obtained by always selecting the optimal arm and the actual cumulative reward it accumulates over time. Figure 1 presents the regret curves for this evaluation.

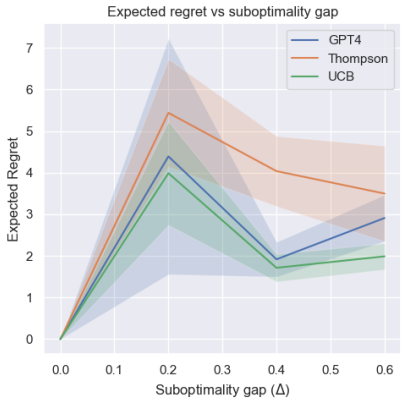

Figure 2: Suboptimality gap experiments for 2-armed bandit settings, comparing GPT-4, UCB, and Thompson Sampling.

Following this, we conduct a finer-grained evaluation using two-armed Bernoulli bandits, introducing suboptimality gap analysis. The suboptimality gap Δ is defined as

$$\Delta := \theta^* - \theta, \tag{1}$$

where θ^* is the success probability of the optimal arm, and θ is the success probability of the next-best arm. We evaluate the best-performing LLM strategy across three suboptimality gaps: $\Delta \in \{0.2, 0.4, 0.6\}$. The $\Delta = 0.2$ condition presents a particularly challenging scenario, requiring models to distinguish between nearly identical reward probabilities. Figure 2 presents the regret curves for this evaluation.

Results. Our evaluation (Figure 1) shows that explicit prompting significantly improves exploration efficiency, with GPT-4 achieving the lowest regret. Implicitly prompted models often commit to suboptimal arms early, suggesting they do not infer the need for exploration without direct instruction. From the suboptimality gap experiments (Figure 2), we find that a prompted GPT-4 performs competitively with classical baselines when reward differences are distinct ($\Delta \geq 0.4$). However, it

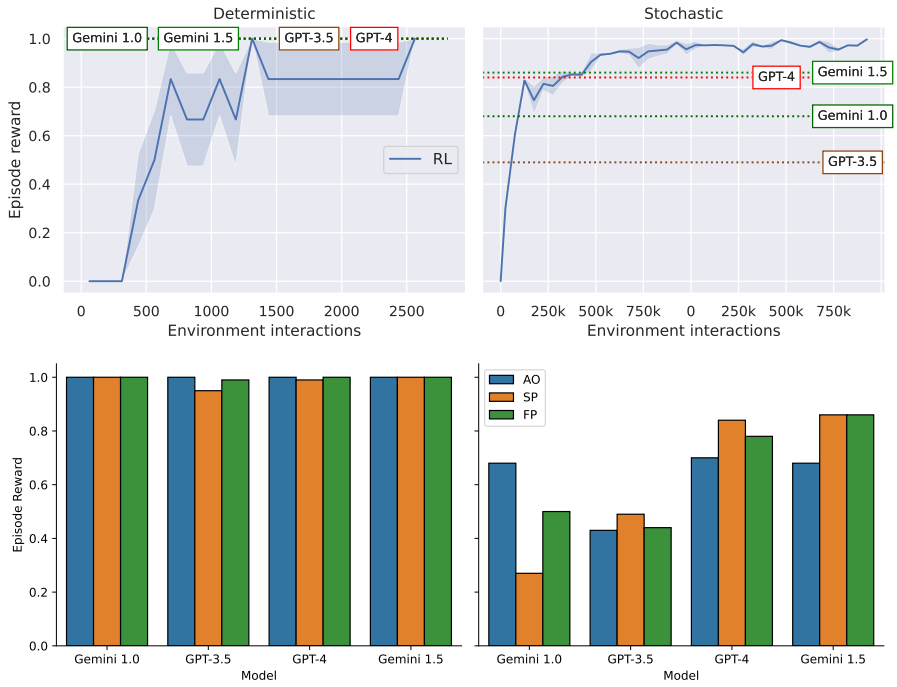

Figure 3: Decision-making results for the deterministic (left) and stochastic (right) setting. In the top row, the learning curves are visible for the RL agents. For each LLM, the performance of the best prompt approach is reported as a horizontal bar. In the bottom row, the performance for each prompt-model approach can be found for both settings.

struggles in the more challenging $\Delta = 0.2$ setting, suggesting that while LLMs can be prompted to explore, their ability to distinguish between subtle statistical differences remains a limitation.

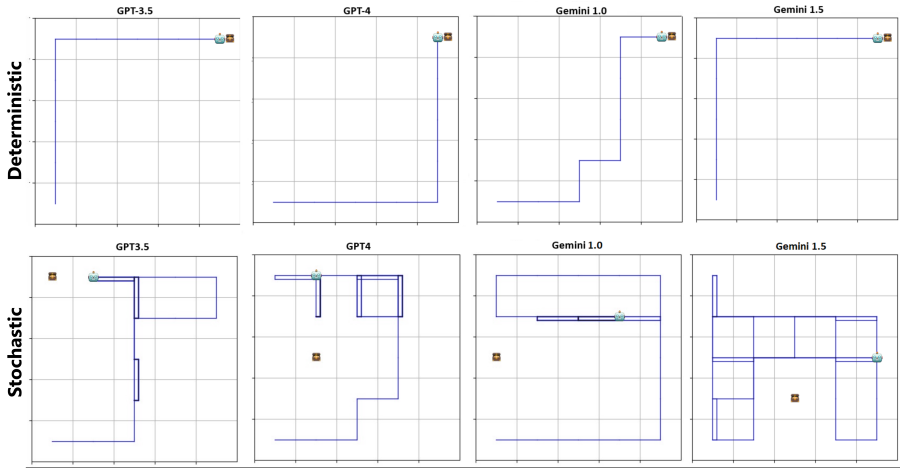

Figure 4: Example trajectories of various foundation model decision-making agents in the deterministic setting (top row) and the stochastic setting (bottom row).

3.2 Gridworld Environments

To further evaluate how LLMs handle structured decision-making, we extend our analysis to Gridworld environments, where agents must explore a spatially structured state space instead of selecting from static actions. Unlike MABs, Gridworlds introduce state transitions, meaning optimal explo-

ration requires both short-term action planning and long-term memory. This setting allows us to assess whether LLMs can reason about spatial navigation and exploration without direct supervision.

Gridworld Setup. We evaluate LLMs in two variations of a 5×5 Gridworld. First, in the deterministic Gridworld, the reward is placed in a fixed location, and the agent observes both its own coordinates and the reward's location. This tests whether LLMs can efficiently navigate to a goal when full information is available. Then, in the stochastic Gridworld, the reward location is uniformly sampled at the start of each episode and is not visible to the agent. The agent only observes its own coordinates, making the problem partially observable and requiring systematic exploration to find the goal.

Prompts. To test whether LLMs can generalize across these settings, we compare three prompting strategies of increasing specificity. First, in Action Only (AO), the prompt provides minimal guidance, simply asking the agent to choose the next action. Second, in Simple Plan (SP), the agent is encouraged to reason about the next steps before selecting an action. Finally, Focused Plan (FP) explicitly instructs the agent to use the available memory of past visited locations to determine unexplored areas and plan the next movement. The complete prompt templates can be found in Appendix B.

Algorithms. We compare the LLM agents to reinforcement learning baselines, using the Proximal Policy Optimization (PPO) algorithm in the deterministic setting and the RecurrentPPO adaptation in the stochastic setting [Schulman et al., 2017]. RecurrentPPO leverages a recurrent neural network to incorporate temporal information, enabling it to handle partial observability effectively. We use the same setup of LLMs as the MABs experiments and average the performance across five random seeds.

Results. Figure 3 presents the performance of each LLM-prompt strategy across both deterministic and stochastic settings, and in Figure 4, example trajectories of each of the foundation models can be found in both the deterministic and stochastic settings. In the deterministic setting, LLMs perform well across all prompting conditions, successfully navigating to the fixed reward location with minimal interactions. However, in the stochastic setting, where systematic exploration is required, performance declines sharply, particularly when using general prompts (Action-Only). Without explicit guidance, LLMs frequently revisit previously explored locations and fail to search the full state space efficiently. While the SP and FP prompts improve performance, they do not fully resolve the challenges LLMs face in handling long-horizon dependencies. Even with explicit instructions, LLMs currently struggle to effectively leverage memory over multiple interactions, leading to redundant exploration patterns. The RL baseline adapts to partial observability over time and eventually solves the environment. These experiments highlight the limitations of applying LLMs to structured exploration tasks requiring long-term memory.

4 Zero-Shot VLM Performance in Atari

Experimental Setup. We evaluate GPT-4o on seven hard-exploration Atari games where traditional RL agents struggle due to sparse rewards: Freeway, Gravitar, Montezuma's Revenge, Pitfall, Private Eye, Solaris, and Venture. To assess whether VLMs can infer objectives directly from visual input, we use a general, minimal prompt that remains the same across all games (Listing 2), providing only the list of available actions. To contextualize performance, we compare its zero-shot cumulative reward against scores from a highly optimized RainbowDQN agent [Castro et al., 2018] at various stages of training (Table 1).

Temporal Information. In RL, agents typically process stacks of consecutive frames to detect motion. However, VLMs must infer motion from visual cues alone. To accommodate this, we modify the standard approach by introducing a lag of $m = 6$ timesteps between each of the four frames in the input stack. This increases temporal diversity, making motion easier to interpret semantically. Unlike other work [Waytowich et al., 2024], our approach balances temporal variation and frame continuity to better capture exploration-relevant motion patterns.

Listing 2: The prompt templates used for the Atari games.

```
Given the following four frames, where the last frame is the current
    game state, to beat the game - what action would you take from the
     actions {actions}?
```

Table 1: Cumulative reward of GPT-4o compared to Rainbow (RB) at different stages of training and human scores across several hard-exploration environments.

Game	GPT-4o	RB 250K	RB 2.5M	RB 25M	Human
Freeway	21	8	32	32	29.6
Gravitar	500	64	199	2405	3351
Montezuma	0	0	50	544	4753
Pitfall	-158	-26	-7	-7	6464
Private Eye	-1000	503	125	1573	69571
Solaris	600	681	1137	2093	12326
Venture	0	8	20	1513	1188

Quantitative Results. The quantitative results are presented in Table 1, comparing the zero-shot VLM agent (GPT-4o) against RainbowDQN at different stages of training, as well as human performance. We find that the VLM agent achieves strong zero-shot exploration in Freeway, Gravitar, and Solaris, obtaining cumulative rewards comparable to or exceeding those of Rainbow agents trained for hundreds of thousands of environment steps. In Freeway, the VLM agent quickly recognizes that moving up the road is the correct strategy, reaching 21 points—substantially outperforming Rainbow at 250K environment steps and approaching human-level performance. Similarly, in Gravitar, the VLM agent successfully navigates multiple levels, engages with enemy objects, and accurately fires its weapon, achieving 500 points, significantly surpassing Rainbow at 250K and 2.5M steps. In Solaris, the VLM agent performs comparably to Rainbow at 250K steps, demonstrating its ability to generalize without prior training.

Qualitative Analysis The divergence in quantitative performance motivates a deeper qualitative analysis to understand the underlying causes of VLM behavior. Our investigation, summarized in Figure 5, reveals a persistent "knowing-doing gap."

In several games, the VLM demonstrates an impressive ability to infer goals and strategies directly from pixels. In **Freeway**, it successfully recognizes the yellow character and understands its objective is to cross the road by moving up. The VLM's performance in **Gravitar** is particularly striking: it identifies the player's ship, its positioning relative to an enemy, and correctly deduces that it should fire, resulting in an immediate 250-point score. Similarly, in **Solaris**, the agent recognizes hostile ships and understands that the 'fire' action is required, even when enemies are not perfectly aligned in its path. These successes highlight a core strength: a powerful, pre-trained semantic understanding of objects and objectives.

This high-level understanding breaks down when precise, low-level control is required. In games like **Montezuma's Revenge** and **Pitfall**, the VLM correctly identifies the goal (e.g., "retrieve the key," "jump over the pit") but consistently fails at execution. It struggles with the precise timing and momentum needed to perform the actions it reasons about, leading to repeated failure. The gap is also evident in grounding and self-recognition. For example, in **Venture**, the VLM fails to identify the player's avatar (the small pink square), undermining any potential for strategic action.

Collectively, these successes and failures paint a clear picture. While VLMs possess a powerful semantic "knowing" of what to do, they often lack the fine-grained procedural "doing" required for execution. This provides strong motivation for investigating hybrid approaches that leverage their semantic guidance while offloading precise control to more robust learning mechanisms.

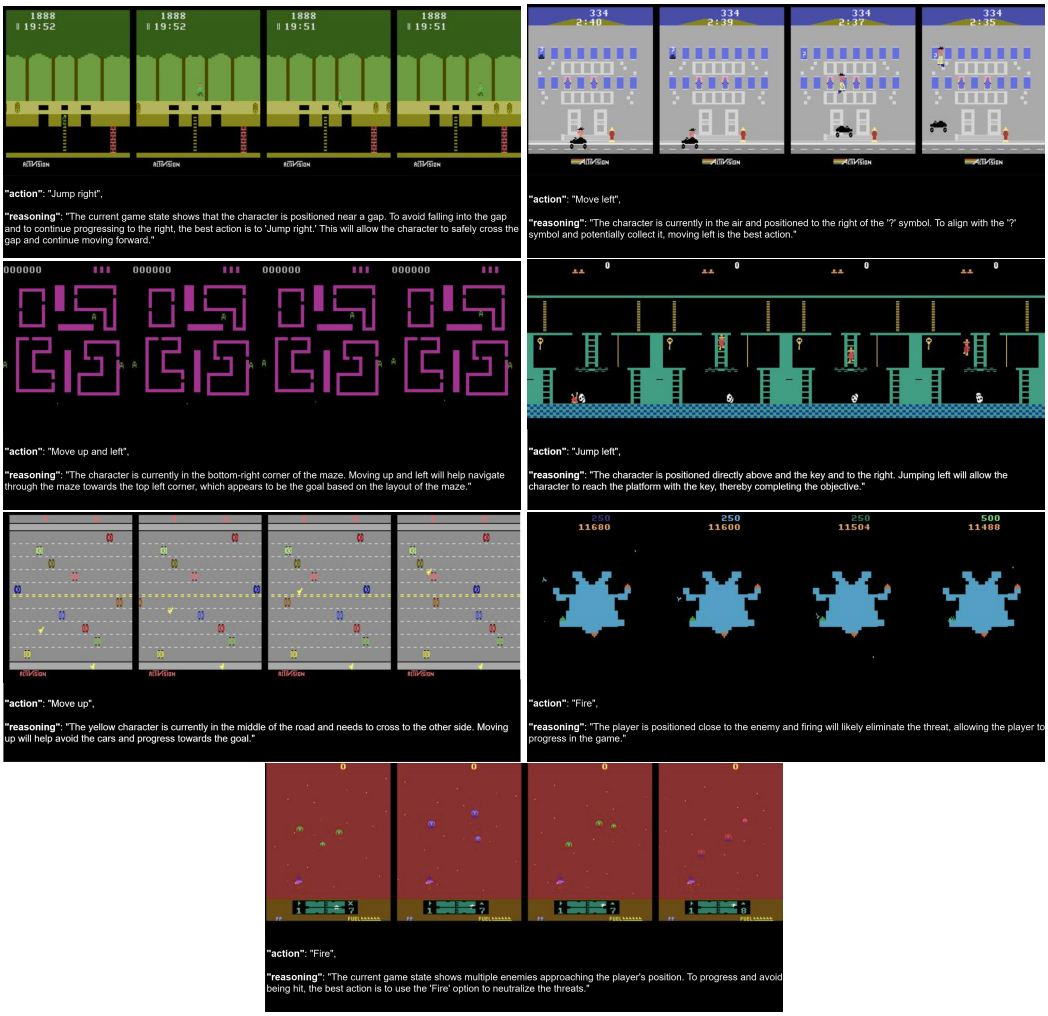

Figure 5: Selected moments from all tested Atari games demonstrating both impressive high-level reasoning and poor low-level execution from the GPT-4o agent.

5 An Upper-Bound Analysis of a Hybrid VLM-RL Agent

Our analysis in Section 4 established that VLMs consistently fail as autonomous agents due to the knowing-doing gap. This motivates investigating hybrid frameworks—not as general solutions, but as tools to understand the theoretical upper bounds of VLM-RL synergies under ideal conditions. .

Hybrid Algorithm. We propose a simple on-policy intervention where an RL agent's trajectory is periodically guided by a VLM, governed by an intervention probability ϵ and a duration T. We chose Proximal Policy Optimization (PPO) [Schulman et al., 2017] as the base algorithm to explicitly test this on-policy intervention. The goal is not to distill the VLM's knowledge into a replay buffer, which would be a natural approach for an off-policy method. Instead, we use the VLM as a semantic explorer to steer the agent to a new state. The PPO agent then resumes its standard on-policy learning process from this more promising starting point.

Experimental Design. To create a clean proof of concept, we selected *Freeway*—an environment where the VLM's high-level strategy is known to be correct and the required control is simple. This allows us to cleanly isolate the potential effect of VLM guidance. We compare three agents: (1) a vanilla PPO baseline, (2) a PPO agent augmented with Random Network Distillation [Burda et al., 2019] (PPO+RND) as a strong exploration baseline, and (3) our PPO-VLM hybrid. All agents were trained for 100,000 environment steps, with results averaged over 5 random seeds.

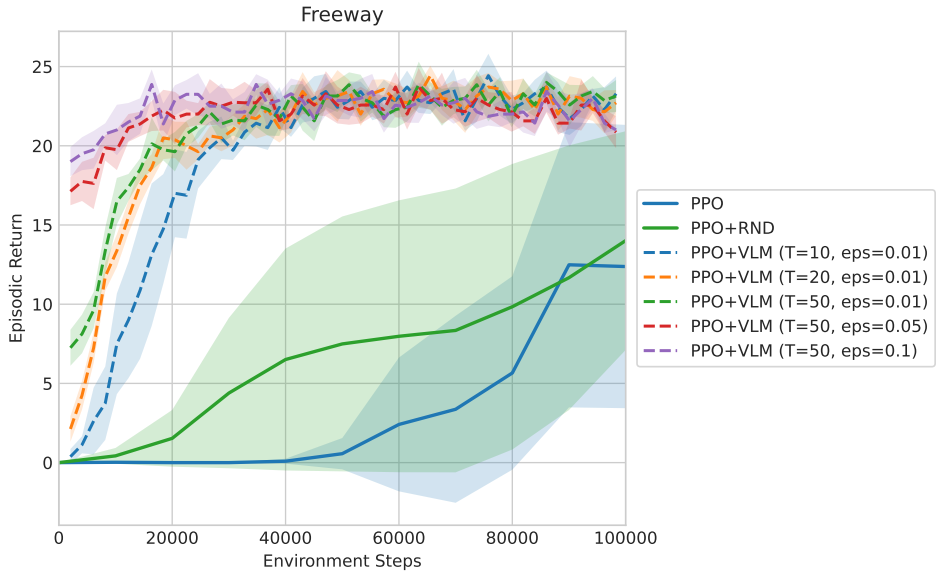

Figure 6: Training curves for various configurations of T and ϵ in Freeway

Analysis and Discussion. The learning curves from our experiment are presented in Figure 6. The data shows that in this specific setting, the PPO-VLM agent learns significantly faster than both the vanilla PPO and the strong PPO+RND baselines. This comes at the cost of increased computation per step due to VLM queries, creating a trade-off between sample efficiency and computational cost. Furthermore, this result should not be interpreted as a general solution, but as an empirical data point. It suggests that under favorable conditions, VLM guidance can act as a potent "semantic accelerator" for an RL policy. In more complex environments where a VLM's guidance is less reliable, we would expect a substantially smaller benefit. This study's contribution is to provide a clear, quantitative data point demonstrating that a synergy is possible under ideal conditions. Future research could extend this hybrid strategy to more complex settings where VLM guidance is less reliable, perhaps by developing adaptive scheduling mechanisms that integrate VLM assistance based on exploration uncertainty or by using policy distillation to mitigate the high inference cost of VLM queries.

6 Conclusion

Our systematic benchmark of foundation models in classic, hard-exploration RL tasks provides a clear characterization of their current capabilities and limitations. In text-based environments, we find that performance remains highly dependent on explicit instruction. In visually complex Atari games, our analysis reveals a persistent "knowing-doing gap," where VLMs consistently fail at the low-level execution and semantic grounding necessary for autonomous control, even when their high-level understanding appears correct.

Our investigation into a simple hybrid framework offers an encouraging data point. The results in Freeway serve as a potential upper bound on the sample efficiency gains, demonstrating that VLM guidance can significantly accelerate learning under favorable conditions. These findings indicate that while foundation models may not be ready to serve as end-to-end agents in complex domains, a promising path forward lies in designing hybrid systems for scenarios where VLMs demonstrate sound high-level reasoning but lack precise execution capabilities. Such systems can strategically leverage the semantic priors of foundation models to guide and bootstrap the learning of more robust, traditional RL policies, a direction that warrants further exploration.

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.

P Auer. Finite-time analysis of the multiarmed bandit problem, 2002.

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. In *7th International Conference on Learning Representations, ICLR 2019*, New Orleans, LA, USA, May 6–9 2019. OpenReview.net. URL https://openreview.net/forum?id=H1lJJnR5Ym.

Pablo Samuel Castro, Subhodeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G Bellemare. Dopamine: A research framework for deep reinforcement learning. *arXiv preprint arXiv:1812.06110*, 2018.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*, 2024.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu

Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning*, pages 8657–8677. PMLR, 2023.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Martin Klissarov, Pierluca D'Oro, Shagun Sodhani, Roberta Raileanu, Pierre-Luc Bacon, Pascal Vincent, Amy Zhang, and Mikael Henaff. Motif: Intrinsic motivation from artificial intelligence feedback. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tmBKIecDE9.

Akshay Krishnamurthy, Keegan Harris, Dylan J. Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context?, 2024. URL https://arxiv.org/abs/2403.15371.

Wenhao Li, Wenwu Li, Chuyun Shen, Junjie Sheng, Zixiao Huang, Di Wu, Yun Hua, Wei Yin, Xiangfeng Wang, Hongyuan Zha, and Bo Jin. Textatari: 100k frames game playing with language agents. *arXiv preprint arXiv:2506.04098*, June 2025. URL https://arxiv.org/abs/2506.04098.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating LLMs as agents. In *International Conference on Learning Representations (ICLR) 2024*, 2024. URL https://openreview.net/forum?id=zAdUB0aCTQ. Poster.

Cong Lu, Shengran Hu, and Jeff Clune. Intelligent go-explore: Standing on the shoulders of giant foundation models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=apErWGzCAA.

Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. BALROG: Benchmarking agentic LLM and VLM reasoning on games. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fp6t3F669F.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Remo Sasso, Michelangelo Conserva, and Paulo Rauber. Posterior sampling for deep reinforcement learning. In *International Conference on Machine Learning*, pages 30042–30061. PMLR, 2023. URL https://openreview.net/forum?id=ZwjSECgl6p.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Xu Wan, Wenyue Xu, Chao Yang, and Mingyang Sun. Think twice, act once: A co-evolution framework of llm and rl for large-scale decision making. *arXiv preprint arXiv:2506.02522*, June 2025. URL https://arxiv.org/abs/2506.02522. Agents Co-Evolution (ACE): LLMs act as both Policy Actor and Value Critic in an actor-critic loop for control with massive action spaces.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856. URL https://openreview.net/forum?id=ehfRiF0R3a.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), Mar 2024b. doi: 10.1007/s11704-024-40231-1.

Nicholas R. Waytowich, Devin White, MD Sunbeam, and Vinicius G. Goecks. Atari-gpt: Benchmarking multimodal large language models as low-level policies in atari games, 2024. URL https://arxiv.org/abs/2408.15950.

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

A Multi-Armed Bandit Experiments

A.1 Background

Reinforcement Learning. Reinforcement learning (RL) is a framework for sequential decision-making where an agent learns to maximize cumulative rewards by interacting with an environment. RL problems are typically modeled as a Markov Decision Process (MDP), defined by the tuple:

$$(\mathcal{S}, \mathcal{A}, P, R, \gamma), \tag{2}$$

where:

- $\mathcal{S}$ is the set of possible states of the environment.
- $\mathcal{A}$ is the set of actions available to the agent.
- $P(s' \mid s, a)$ is the transition probability function, defining the probability of moving to state s' given current state s and action a.
- $R(s, a)$ is the reward function, mapping state-action pairs to scalar rewards.
- $\gamma \in [0, 1]$ is the discount factor, determining the importance of future rewards.

At each timestep t, the agent observes a state s_t, selects an action a_t, and transitions to a new state s_{t+1} based on the environment dynamics P, receiving a reward $R(s_t, a_t)$. The objective is to learn a policy $\pi(a \mid s)$ that maximizes the expected return:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, a_{t+k}). \tag{3}$$

Multi-armed bandits. The multi-armed bandit (MAB) problem is a fundamental decision-making framework in reinforcement learning. It models a scenario where an agent selects from a set of K independent actions, or "arms," each associated with an unknown reward distribution. The goal is to maximize cumulative reward over a given horizon by balancing:

- **Exploration**: Trying different arms to gather information about their reward distributions.
- **Exploitation**: Selecting the arm with the highest expected reward based on current knowledge.

Formally, at each time step t, the agent selects an arm $k \in \{1, ..., K\}$, receiving a reward r_t drawn from an unknown distribution P_k:

$$r_t \sim P_k. \tag{4}$$

In our experiments, we specifically consider *Bernoulli bandits*, where each arm provides rewards sampled from a Bernoulli distribution with an unknown success probability θ_k. That is, for each arm k,

$$r_t \sim \text{Bernoulli}(\theta_k), \tag{5}$$

where θ_k represents the probability of obtaining a reward of 1, while a reward of 0 occurs with probability $1 - \theta_k$.

The agent's objective is to identify the optimal arm k^* with the highest θ_k, while minimizing cumulative regret over time. The suboptimality gap Δ_k for an arm k is defined as:

$$\Delta_k = \theta^* - \theta_k, \tag{6}$$

where $\theta^* = \max_k \theta_k$ is the success probability of the optimal arm.

A.2 Experiments

Algorithms. For the multi-armed bandit experiments, methods like Thompson Sampling and UCB naturally consider previous trials' outcomes by updating their belief distributions. As could be seen in Listing 1, to account for this in the decision-making for the LLMs, we provide a memory in the prompt. In this memory, we append all previous trials as '*Pulled arm {ACTION} resulting in a reward of {REWARD}*'. In the case of Thompson Sampling, we found that the best performing prior was $\alpha = 1$ and $\beta = 1$, and in the case of UCB, we used the UCB1 variant with a tuned constant $c = 0.25$.

Prompt phrasing. In preliminary experiments, we tried slight variations of the prompt presented in Listing 1. For instance, we found that providing the maximum number of trials or encouraging "an efficient exploration approach" did not improve the performance.

Additional Results. Additional results can be found in Figure 7 where the individual seeds for the best-performing LLM prompts can be found in the two bandit settings.

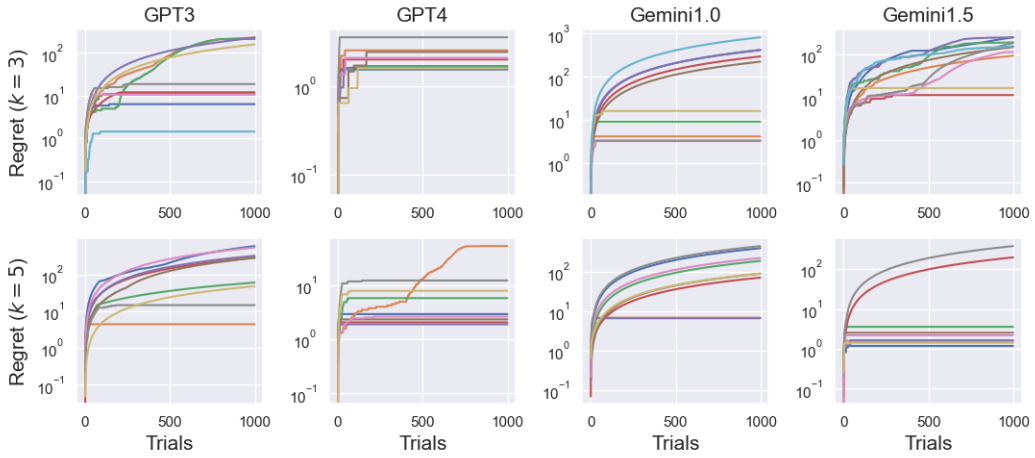

Figure 7: Individual seeds for each of the LLMs using the best-performing prompt in the $k = 3$ (top) and $k = 5$ (bottom) settings, providing insights on the different strategies employed by each model.

B Gridworld Decision-Making Experiments

Reinforcement learning agents. For the reinforcement learning agents in these experiments, we used the default Stable-Baselines3 implementations of PPO and RecurrentPPO [Raffin et al., 2021].

Prompt templates. For the foundation models, the prompt templates used for the deterministic and stochastic LLM agents are almost identical. While the deterministic agent receives the exact position where the reward is located, the stochastic agent is only told the following: *Your goal is to reach the reward located at a random coordinate as quickly as possible.* See Listing 3 for the full prompt used by the deterministic LLM agent. The agent's memory is filled as it interacts with the environment. Whenever an action is executed, we add the following line to the memory: "*Executed {ACTION} at {LOCATION} resulting in {NEW LOCATION} and no reward.*" Additionally, as seen in Listing 5, whenever an agent chooses an action, it outputs a plan representing its thoughts. We add each plan to the memory as well.

Listing 3: The prompt template for the deterministic LLM agent.

```
**Context**
You are an agent in a {n}x{n} grid.
The bottom left corner is at {BOTTOM LEFT}, top left at [0, n-1], top
    right at [{n-1, n-1}], and bottom right at [{n-1}, 0].
The x-axis increases as you move rightward, and the y axis increases
    as you move upwards.
Your goal is to reach the reward located at a coordinate [{n-1},{n-1}]
     (the top-right corner).

**Memory**
[...]

**Observation**
Your current location is {OBSERVATION}

**Available Actions**
up
right
down
left

**Task**
Choose an action from the given list of actions. Output your response
    using the following JSOn format and do not use markdown.
{OUTPUT FORMAT}
```

The three prompting approaches are implemented using different JSON output formats. Note that since the action-only agent outputs no plan, its memory will only contain the results of the executed actions. Note that in the stochastic setting, we do not mention that the reward is located in the top-right corner.

Listing 4: The output format used by the **Action Only** agent.

```
{
    "action": "{The action you want to take}"
}
```

Listing 5: The output format used by the **Simple Plan** agent.

```
{
    "plan": "{Think about what you want to do next to fulfill your
    goal.}",
    "action": "{The action you want to take}"
}
```

Listing 6: The output format used by the **Focused Plan** agent.

```
{
    "plan": "{Use your memory to determine which position you want to
    go next.}",
    "analysis": "{Using your plan analyze which action is best to
    efficiently reach the position.}",
    "action": "{The action you want to take}"
}
```

Results. The full list of numerical results for the FA performances from the experiments in Section 3.2 can be found in Table 2.

Table 2: LLM Agent performances for an empty 5×5 grid with a fixed reward location and random reward location, averaged over 100 episodes.

Model	Fixed Reward	Random Reward
Gemini 1.0 (AO)	100%	68%
Gemini 1.0 (SP)	100%	27%
Gemini 1.0 (FP)	100%	50%
GPT-3.5 (AO)	100%	43%
GPT-3.5 (SP)	95%	49%
GPT-3.5 (FP)	99%	44%
GPT-4 (AO)	100%	70%
GPT-4 (SP)	99%	84%
GPT-4 (FP)	100%	78%
Gemini 1.5 (AO)	100%	68%
Gemini 1.5 (SP)	100%	86%
Gemini 1.5 (FP)	100%	86%

C Atari Experiments

Algorithms. For the Atari experiments, we passed four individual frames with a lag of 6 environment timesteps between using a frameskip of 4. Along with these frames, we provided the prompt and action space to the OpenAI API. The action space was provided as specified by the official ALE action space descriptions. The RainbowDQN implementation and results are from the Dopamine [Castro et al., 2018], while the PPO implementation came from the default visual implementation from Stable-Baselines3 [Raffin et al., 2021]. For the hybrid strategy we implemented the VLM actions in the Stable-Baselines3 implementation in an epsilon-greedy fashion.

D Compute

We use OpenAI's APIs for GPT-3.5, GPT-4, and GPT-4o, and Google Studio's APIs for Gemini 1.0 and Gemini 1.5. For training the PPO, we used a single NVIDIA A100 GPU in all environments. For the GPT models, we used 'GPT-3.5-turbo-0613', 'GPT-4-0613', and 'GPT-4o-2024-08-06' cutoffs, which cost US$ 0,50 / 1M input tokens US$ 1,50 / 1M output tokens, US$ 30,00 / 1M input tokens US$ 60,00 / 1M output tokens, US$ 2,50 / 1M input tokens US$ 10,00 / 1M output tokens, respectively, as of writing. The Gemini models can be used freely as of writing, although the Gemini models have a relatively low limit for queries per minute. For the Gemini models we used 'gemini-1.0-pro-001' and 'gemini-1.5-pro'.