Exploring exploration with foundation agents in interactive environments

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Foundation models excel at single-turn reasoning, but many real-world challenges, from scientific research to technology development, require multi-turn exploration in dynamic interactive environments. Crucial components of learning from experience in these settings, such as efficiently gathering information to test hypotheses, meta-learning a model of the world's dynamics, and adapting to unexpected changes, remain largely unexplored for these models. We first evaluate foundation models in Feature World, a setting that primarily tests information gathering about a static hidden reward function. In this initial setting, we show that state-of-the-art foundation models come close to optimal efficiency in selecting maximally informative actions in tasks with simple reward functions. We also show a model can gather information efficiently in a 3D embodied version of this task, though errors in vision limit some aspects of performance. In order to test exploration across multiple dependent turns and trials, we implement a custom, text-based version of the Alchemy environment, a benchmark designed for meta-learning. Here, agents must deduce a latent causal structure by integrating information across multiple state-dependent trials. In this more complex setting, we find that recent foundation models struggle to meta-learn strategies that enable improved performance over time. However, prompting the models to summarize their observations at regular intervals enables an emergent meta-learning process, allowing them to improve across trials. Notably, in some models, summarization also enabled adaptive relearning of this information when the environment's rules change unexpectedly. While most models performed reasonably well on simple Feature World tasks, evaluations in Alchemy reveal stark differences in robustness among the models. These results demonstrate that scaling environmental demands is a powerful method for revealing both the capabilities and limitation of current agents, highlighting that the primary challenge is not just selecting informative actions, but integrating knowledge over time. Intriguingly, we find there is likely no intrinsic barrier to future generations of foundation agents more fully mastering these abilities.

1 Introduction

Foundation models have demonstrated remarkable abilities in understanding and generating complex human-like text and multi-modal content [Achiam et al., 2023, Gemini Team et al., 2023, Jiang et al., 2024, Reid et al., 2024, Dubey et al., 2024, Dai et al., 2024, Deitke et al., 2024]. However, this success has largely been measured in static, single-turn settings where information is provided upfront. The next frontier for these models lies in their application as interactive agents, which must operate in dynamic environments where crucial information is not given, but must be actively discovered. To achieve goals in such settings, an agent cannot merely react; it must proactively explore. This contrasts with classic reinforcement learning (RL) paradigms that use undirected exploration [Burda

Submitted to Embodied World Models for Decision Making Workshop @ NeurIPS 2025

et al., 2018, Ecoffet et al., 2019, Badia et al., 2020]. Real-world endeavors often demand a more sophisticated, hypothesis-driven approach. This involves strategically formulating beliefs about the world, designing experiments to test those beliefs, and integrating findings gathered across multiple trials. Such capabilities will become increasingly important as training on human-generated data reaches a limit and we enter the "era of experience", in which models generate their own training data through interaction with their environment [Silver and Sutton, 2025]. Whether, and to what extent, today's foundation models possess this latent capacity for active exploration remains a critical and largely open question.

We evaluate LLMs in three environments: text-based and 3D variants of Feature World, and a text-based version of Alchemy [Wang et al., 2021]. The Feature World is largely stateless and does not necessitate extensive sequential decision-making, allowing us to isolate and analyze efficiency of information gathering. Alchemy, in contrast, demands strategic exploration and reasoning over multiple trials, which allows us to evaluate the foundation models' meta-learning and strategy adaptation abilities.

In this paper, we operationally define and measure three key capabilities involved in exploration: efficient information gathering, meta-learning, and strategy adaptation.

- **Efficient information gathering:** Selecting actions that maximally increase expected information gain. In Feature World, we measure this as success rate in finding a rewarding object within a fixed step budget, compared to a random policy.
- **Meta-learning (learning to learn)**: Improving expected performance on new tasks in a given family through experience of other tasks in that family [Thrun and Pratt, 1998]. In Alchemy, we measure this as a significant improvement in performance over successive trials within an episode.
- **Strategy adaptation**: Detecting when a strategy becomes invalid due to environmental changes and adapting by learning a new one. In Alchemy, we measure this as performance recovery after an uncued change to environment dynamics.

More specifically, this paper investigates the capacity of foundation models to conduct exploratory behavior within interactive environments in the zero-shot setting, using in-context prompting alone and without requiring task-specific training, fine-tuning, or few-shot examples.

We performed experiments using Gemini 1.5 Pro and Flash [Reid et al., 2024], Gemini 2.5 Pro and Flash [Google, 2025], Claude 3.7 Sonnet [Anthropic, 2025], and ChatGPT-4o [OpenAI, 2024] and o4-mini [OpenAI, 2025].

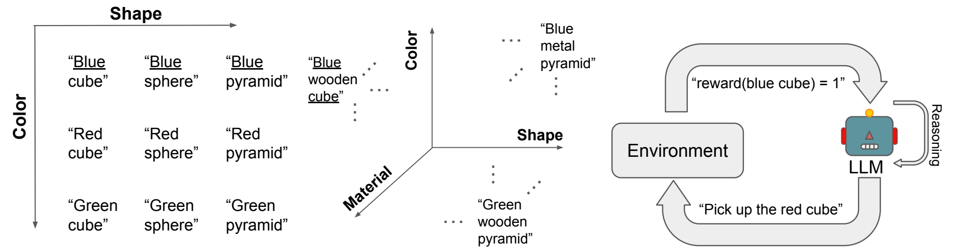
Overall, this work makes the following key contributions and findings:

- We conduct extensive experiments evaluating multi-turn exploration performance of foundation models across a diverse set of interactive environments. We analyzed several foundation models and a range of in-context prompting strategies, including variations in the amount of prior information and the structure of demonstrations.
- Our findings reveal a strong inherent exploratory capacity in foundation models across simple interactive settings. Specifically, all LLMs we evaluated demonstrated near-optimal performance in Feature World with simple reward functions. Likewise, some models outperformed the memoryless heuristic in Alchemy, something that the RL agents benchmarked in the original Alchemy study were unable to do.
- We find that in complex, multi-trial environments, such as Alchemy, foundation models struggle to show meta-learning (improving over trials) and strategy adaptation (re-learning a world model when the effects of actions unexpectedly change). However, both these abilities can emerge when models are prompted to summarize information across trials.
- We find stark differences in the robustness of meta-learning and strategy adaptation in frontier LLMs using Alchemy, demonstrating the utility of this environment as a benchmark for LLM exploration capabilities.

2 Related Work

Exploration has been studied in the context of games for both RL agents [e.g., Burda et al., 2018, Ecoffet et al., 2019, Osband et al., 2019, Guo et al., 2022, Saade et al., 2023] and foundation models [e.g., Wang et al., 2023a,b, Feng et al., 2023, Tan et al., 2024]. All of these works, however, focus more on improving agent performance rather than performing an explicit, systematic investigation of well-defined exploration capabilities with foundation models in controlled, zero-shot settings and in comparison to known optimal policies. See Section A.1 for a more detailed overview of related work.

3 Feature World



(a) (b) (c)

Figure 1: Task structures and experimental setups for Feature World. (a) Example task setup for text environment with single-feature reward function, with "blue" as the rewarding feature. (b) Example task setup for text environment with conjunction reward function, with "blue" and "cube" as the rewarding conjunction. (c) Schematic of text Feature World experiment setup.

We first evaluate models in Feature World: a simple, memory-less, text-based setting. In this environment, actions can be executed repeatedly without altering the underlying state dynamics due to previous actions, and each exploratory action provides immediate feedback.

3.1 The Feature World environments

3.1.1 Text environment

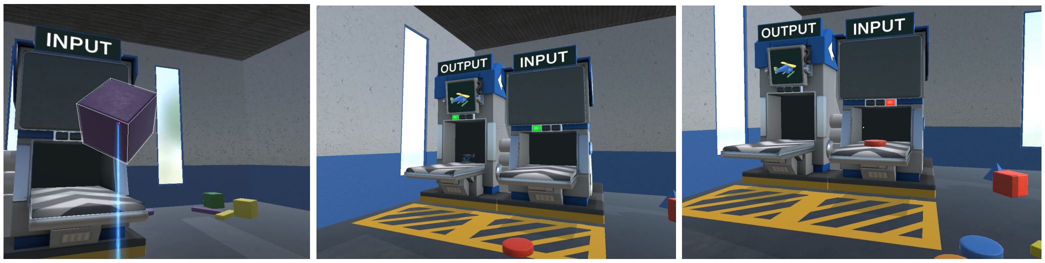
To investigate information gathering, we use a task where models are presented with objects possessing multiple features (e.g., color, shape). A specific feature or conjunction of features determines a reward, mirroring sparse-reward RL settings.

This task isolates multi-step information gathering within a single trial. Actions provide immediate feedback on a static reward function, with no latent dynamics to discover across trials.

To assess the robustness of different models, we modulate task difficulty by adjusting two key aspects: the number of distinct colors (increasing the cognitive load) and the complexity of the reward function. Reward functions can be based on a single property (**single-feature tasks**) or a conjunction of two properties (**conjunction tasks**). See Figure 1 for a visualization of the tasks.

3.1.2 3D Feature World – Construction Lab

To test active exploration in a more realistic, multi-modal setting, we created a 3D version of Feature World in the Construction Lab simulation **[reference-anonymized]**. In this environment, the agent receives video input and outputs action instructions for a human to execute. This setup assesses exploratory behavior while introducing real-world challenges like visual understanding. To manage visual complexity, these 3D tasks use only three colors and a single rewarding feature, mirroring the simplest text-based condition.



(a) (b) (c)

Figure 2: Gameplay in Construction Lab. (a) The agent uses a blue laser beam to pick up objects. (b) Result of a correct object placement. (c) Result of an incorrect object placement.

3.2 Feature World experiments and results

Our experiments in Feature World aim to address the following two questions: (a) How does the complexity of the environment affect the information gathering efficiency of foundation models? (b) What new challenges emerge in an embodied 3D version of the task?

We evaluate on different foundation models, using publicly available APIs with default settings unless noted otherwise. For Feature World, we compared ChatGPT-4o [Achiam et al., 2023, OpenAI, 2024] (200k context), Claude 3.7 Sonnet [Anthropic, 2025] (200k context), Gemini 1.5 Flash and Pro [Reid et al., 2024], and Gemini 2.5 Flash and Pro [Google, 2025] (1M context).

3.2.1 Task setup

Baselines We compare to two baselines: *Optimal Baseline*: This baseline represents an upper bound on exploration performance. It selects actions that maximize information gain at each step. See Section A.3.1 for a more detailed description of the optimal strategy. *Random Baseline*: This baseline establishes a lower bound by choosing objects randomly with replacement. Both baselines are evaluated with 1000 episodes.

See Tables 1, 2, 3, and 4 in the Appendix for the prompts used in Feature World for all models.

Evaluation To evaluate information gathering efficiency, we assess how often models are successful at finding a rewarding object given a fixed budget of exploration steps. We set the step budget as the maximum number of steps that an optimal policy would need before finding at least one rewarding object. This measures the model's active exploration capabilities independent of its ability to draw conclusions from its observations.

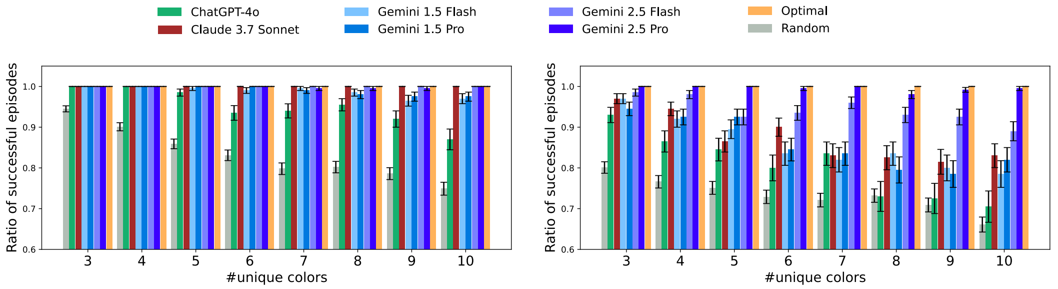
For the 3D exploration task, we measure two key metrics: 1) the number of steps required to gather sufficient information to identify the reward function, and 2) the model's accuracy in correctly identifying that function from its observations. This second metric assesses the model's combined ability to efficiently explore as an actor and to draw conclusions from visual evidence.

3.2.2 Effects of environment complexity on exploration

We examine the effect of two forms of environmental complexity on information gathering efficiency: reward function complexity and object quantity. To measure the former, we designed tasks where reward is determined by either a single feature (like "red" or "square") or a conjunction of two features (e.g., "red" and "square"). The latter requires the agent to reason about multiple properties to identify the rewarding combination. To investigate the impact of cognitive load on model performance, we also vary the number of unique colors in the environment.

In both single-feature and conjunction tasks, almost all models outperform the random baseline in all conditions (Figure 3), showing most LLMs have a robust capacity for efficient information gathering.

In the single-feature task, all models perform better than random and most remain nearly optimal as cognitive load increases (Figure 3a). This shows that, when the reward function is simple, most models are capable of nearly optimal information gathering efficiency even as cognitive load increases.



(a) Single-feature tasks

(b) Conjunction tasks

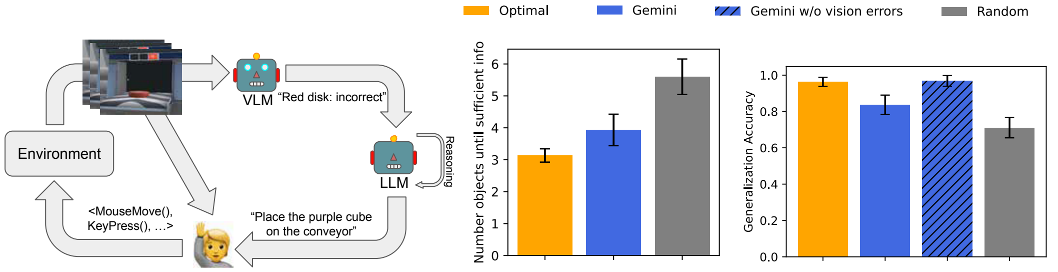
Figure 3: Fraction of Feature World episodes in which models found a rewarding object before reaching the maximum number of exploration steps. (a) Single-feature reward function. (b) Conjunction reward function. Error bars represent standard error of the mean, with 200 episodes per condition for the models and 1000 for the random and optimal baselines.

In the conjunction task, most models show a large drop in performance compared to the single feature task, though Gemini 2.5 Pro maintained strong performance despite the increase in task difficulty (Figure 3b). This demonstrates that reward function complexity exacerbates the impact of cognitive load on performance for most LLMs, but not universally.

Taken together, these results show that LLMs have a robust capacity for gathering information efficiently for simple reward functions. On the other hand, most LLMs degrade with increasing cognitive load when the reward function is more complex. Interestingly, our results show that Gemini 2.5 Pro is an outlier in this trend, achieving nearly optimal performance across all levels of reward function complexity and cognitive load. This shows that robust, near-optimal information gathering efficiency is possible in LLMs, and may eventually become a common capability.

3.2.3 3D embodied Feature World results

As a proof of concept, we also tested the multimodal and visual understanding capabilities of a model in an embodied 3D verion of Feature World (Figure 4a). See Section A.4 for more setup details.



(a) 3D Feature World Setup

(b) Steps to sufficient info

(c) Rewarding property accuracy

Figure 4: Schematic and performance metrics for 3D exploration task, with 15 episodes per condition. (a) Schematic of 3D task setup. (b) Mean number of exploration steps (objects placed on the conveyor) before sufficient information is available to determine the correct factor. (c) Accuracy of the model in determining the correct rewarding feature. Hatched blue bar represents accuracy if episodes with vision errors are removed. Error bars represent standard error of the mean.

We show that Gemini 1.5 Pro is capable of extracting the necessary symbolic information from video, providing real-time instructions to a human player, and reasoning. The Gemini actor gathered information more efficiently than a random actor and was close to optimal (Figure 4c). This shows that a foundation model can gather information efficiently in a 3D embodied setting requiring vision.

However, its accuracy in identifying the correct rewarding feature was hampered by vision errors, which occurred in 8 of 15 episodes. While accuracy improved to near-optimal levels on trajectories without such errors, overall performance was not statistically superior to a random actor ($p = 0.13$, 2-sample t-test) (Figure 4c). This suggests that multi-modal perception, rather than reasoning, is the primary bottleneck for applying foundation models to more complex, real-world exploratory tasks.

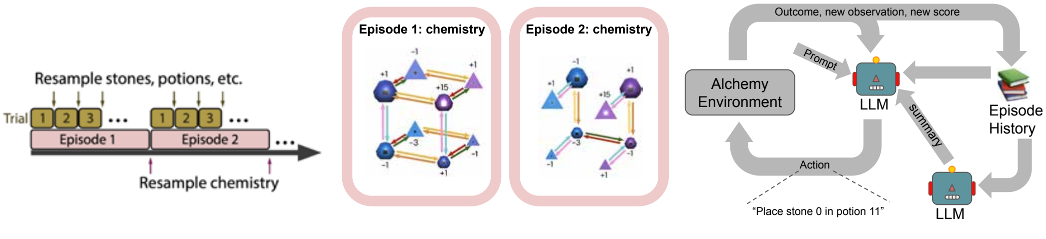
See Section A.4.4 for a more detailed discussion of the 3D Feature World results.

4 Alchemy

We then evaluate agents in Alchemy, a more complex environment with a persistent, hidden task structure across multiple trials. This requires strategic exploration in early trials to inform exploitation in later ones. For this more demanding evaluation, we selected the Alchemy environment [Wang et al., 2021]. Alchemy is notable for its structured task distribution and its design to test reasoning, planning, and, importantly, exploration and meta-learning. In Alchemy, the agent needs to take actions to not only discover rewards but also latent causal dynamics, which are randomly resampled every episode. Additionally, constraints are introduced such that not all actions can be taken repeatedly within a trial (which are themselves multi-step), necessitating planning both within and across trials. These characteristics contribute to a more complex testbed for evaluating components of exploration strategies, such as meta-learning and strategy adaptation, that are measured across multiple timescales (i.e., within trials, across trials, and across episodes).

4.1 The Alchemy environment

Alchemy [Wang et al., 2021] is a procedurally generated environment specifically created to test meta-learning capabilities. The core gameplay involves using a set of potions to transform various visually distinctive stones into more valuable forms, and then depositing them into a central cauldron to score points. Stone appearance varies along three feature dimensions: size, color, and shape, and their value is visually indicated by a marker. Potion effects are determined by color. A central concept in Alchemy is the "chemistry", which represents a latent causal structure that governs the value of stone appearances and the transformative effects of potions on stones. This chemistry is procedurally resampled for each episode, meaning the specific rules linking appearance, value, and potion effects change every time a new episode begins. We define a step as a single action (e.g., applying a potion), a trial as a sequence of steps ending with scoring or resource exhaustion under a fixed chemistry and finite set of objects, and an episode as a sequence of N trials (defaulting to $N = 10$) where the chemistry is constant, resetting only between episodes (Figure 5a-c). Within a single episode, the agent's implicit challenge is to diagnose the current chemistry through repeated observation and experimentation. This involves operating at two timescales: making effective choices *within* each trial and synthesizing the information gathered *across trials* to learn about the latent dynamics, applying this knowledge to maximize scores in subsequent trials. After each episode, the chemistry is reset and all information from the previous episode is cleared from the model's context (with the exception of the strategy adaption experiments: see Section 4.2.3).



(a) (b) (c)

Figure 5: Task structures and experimental setup for Alchemy. (a) The structure of an Alchemy experiment. (b) Example chemistries, represented as graphs determining the effects of potions (edges) on stones of different properties (nodes), that change between episodes. (c) Alchemy exploration setup. An LLM receives feedback from the environment, a prompt, past episode history, and, optionally, a summary. (a) and (b) are adapted, with permission, from figures in Wang et al. [2021].

4.2 Alchemy experiments and results

Our experiments in Alchemy aim to address the following questions: (a) How does the multi-trial setting, which requires long-context and memory, impact the exploration performance and meta-learning of foundation models? (b) How do different prompting strategies and cross-trial summarization methods impact exploration performance and meta-learning? (c) How well can foundation models adapt their learned strategies to uncued changes in environment dynamics?

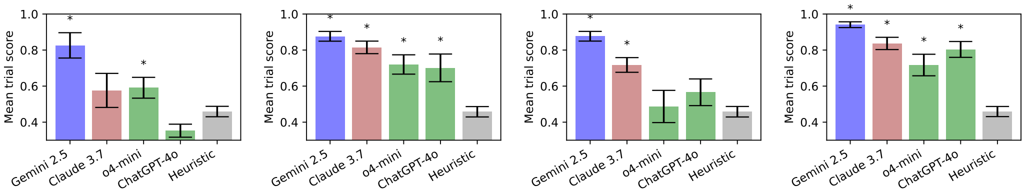
For Alchemy, we compared Gemini 2.5 Pro (1M context), Claude 3.7 Sonnet (200k context), and o4-mini [OpenAI, 2025] (200k context), all of which are reported to employ an explicit thinking step, and ChatGPT-4o (200k context), which does not employ a thinking step.

4.2.1 Task setup

We evaluated the LLMs on a symbolic version of Alchemy in which both actions and observations of the game state are represented as text. We used the same parameters as in Wang et al. [2021].

Baselines We compare LLMs to two baselines: *Optimal Baseline*: the oracle baseline in Wang et al. [2021], which is a baseline that knows the underlying causal structure of the environment and can perform optimal actions. All results shown are normalized to the score of the oracle baseline. *Heuristic Baseline*: To set a baseline for reasonable performance, we use the memoryless heuristic described in Wang et al. [2021], which places random stones in random potions until either a stone reaches the maximum reward (in which case that stone is placed in the cauldron, and random selection then continues) or all stones are used up (in which case all positive-valued stones are placed in the cauldron, and the trial then ends).

Evaluation We assess two variables impacting model performance on the Alchemy task: 1) inclusion of prior information on invariant principles of Alchemy in the prompt (see Section A.5.1 for details), and 2) use of summarization to augment learning across trials (see Section A.5.2 for details).



(a) No summary, no prior information

(b) No summary, prior information

(c) Summary, no prior information

(d) Summary, prior information

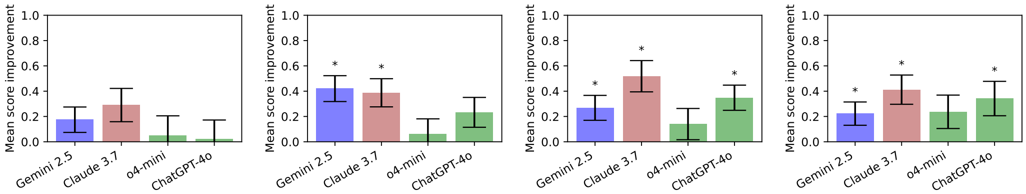
Figure 6: Mean Alchemy episode scores for different models and conditions. (a) No summarization, no prior information. (b) No summarization, prior information. (c) Summarization, no prior information. (d) Summarization, prior information. N=10 replicates of 10-trial episodes. Error bars represent standard error of the mean. Asterisk indicates the mean is significantly different from that of the memoryless heuristic ($p < 0.05$, paired-sample t-test).

We measure model performance primarily through three metrics: 1) performance: mean score over the 10 trials of an episode, 2) improvement: difference of the mean score of the last 5 trials and the score of the first trial (Appendix Figure 10), and 3) adaptation: the mean score for 10 trials following an unexpected change in chemistry. For all metrics, trial score is normalized as a fraction of the score of an oracle that takes the optimal set of actions for the given items. We use an additional two metrics to gain further insight into model decision making: 1) change in the number of potions used between the first trial and the last five trials, and 2) the fraction of trials in which the model places at least one negative-valued stone in the cauldron. See A.5.7 for results on these latter two metrics. For all metrics, we run 10 replicate episodes with randomized chemistries.

4.2.2 Effects of summarization and prior information on model performance and learning

As shown in Figure 6a, Gemini and, to a lesser extent, o4-mini, significantly outperform the memoryless heuristic. Notably, the RL agents evaluated in Wang et al. [2021] did not significantly outperform the memoryless heuristic, despite being trained for 1e9 episodes. This condition with no summarization strategy and no prior information most closely mirrors the setting of the original Alchemy task experienced by the RL agents. This shows that some LLMs can act as powerful agents on tasks requiring exploitation of strategies learned through extended exploration across multiple tasks, even in environments originally designed for RL agents.

However, in Figure 7a, we see that none of the models shows a statistically significant within-episode improvement in score for this setting, suggesting that meta-learning is not operating efficiently.



(a) No summary, no prior information

(b) No summary, prior information

(c) Summary, no prior information

(d) Summary, prior information

Figure 7: Improvement in score over the episode, computed as mean of the last 5 trial scores minus the score of the first trial. (a-d) Same conditions as Figure 6. N=10 replicates of 10-trial episodes. Error bars represent standard error of the mean. Asterisk indicates the mean is significantly different from 0 ($p < 0.05$, single-sample t-test).

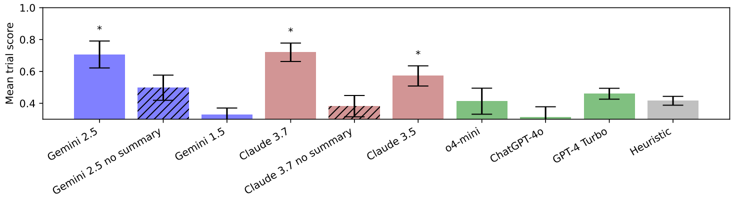
The inclusion of prior information increased the mean scores (Figure 6b), and resulted in score improvements over the episode becoming significant for Gemini and Claude (Figure 7b). This shows that the prior information about invariant principles boosts performance and enables significant improvement over trials, supporting the hypothesis that it provides a framework for the model to generate hypotheses and targeted exploration actions[1].

However, since Alchemy was designed specifically to evaluate models' ability to meta-learn these invariant principles from experience, the same prior information provided in the prompt ought to be present in the model's action and observation history. As such, we hypothesized that prompting the models to summarize their observations and actions after each trial would encourage them to extract equivalent information and lead to a similar boost in performance. To test this hypothesis, we implemented the summarization strategy described in Section A.5.2.

We found that, with summarization, all models showed significant score improvement over the episode except for o4-mini (Figure 7c). Summarization also improved the mean scores for all models (Figure 6c). This supports the hypothesis that summarization enables meta-learning.

To test whether the information gained in the summarization condition is functionally similar to that provided in the prior information condition, we evaluated models with both summarization and prior information. In this setting, mean scores were similar to the prior information condition (Figure 6d). Score improvements were likewise similar to the cases with summarization only or prior information only (Figure 7d). This supports the hypothesis that summarization enabled the acquisition of information similar to or redundant with that provided in the prior information.

4.2.3 Strategy adaptation following uncued change in game rules



(a) Mean scores after chemistry change

Figure 8: Mean normalized model scores for trials 11-20 when an uncued change in chemistry occurs halfway through a 20-trial episode. Hatched pattern represents no summary. Error bars represent standard error of the mean, across 10 replicates.

---

[1] We also performed ablations on the specific types of prior information provided to determine which invariant principles were most useful to the models. See Section A.5.6 for details.

To evaluate strategy adaptation, a human ability associated with dorsolateral prefrontal cortex (dlPFC) function [Mansouri et al., 2007, Donahue et al., 2018], in LLMs, we modified the task setup such that the models are exposed to two consecutive episodes before their observation history is cleared. However, the change in chemistry in the transition to the second episode occurs silently, leaving the model to grapple with unexpected outcomes of previously-predictable actions. We ran experiments with summaries enabled and reward and potion pair information provided, but withheld causal information since it was no longer accurate given the change in chemistry. See Figure 11 in the appendix for full timeseries plots of these results.

Of our four thinking models studied, Gemini 2.5 and Claude 3.7 were able to regain full performance after an initial drop following the change in chemistry (Figure 8, Figure 11a,b). However, o4-mini and ChatGPT-4o were indistinguishable from the heuristic policy following the change (Fig 11c,f). Unlike o4-mini, ChatGPT-4o had strong performance and improvement over trials prior to the change, showing that the ability to improve through learning of an initial strategy (meta-learning) does not necessarily predict the ability to learn a new strategy when the world changes (strategy adaptation).

To test whether Gemini 2.5 and Claude 3.7 have native strategy adaptation ability without augmentations, we evaluated both models with summary and prior information disabled. In both cases, the models showed a collapse in performance following the chemistry change (Figure 11d,e). This shows that even models with strong native exploration abilities struggle with strategy adaptation when not provided with task-specific prompt augmentations.

4.3 Alchemy conclusions

Taken together, our results show that integrating information over long time horizons through adaptable strategies in context is a frontier challenge in LLMs. In particular, we show that even the strongest models are generally poor at meta-learning and strategy adaptation without task-specific augmentations. However, the emergence of these skills after prompting for summarization suggests that LLMs have a latent ability to improve and adapt through exploration. Likewise, the comparatively strong performance of Gemini 2.5 and Claude 3.7 relative to other models suggests that deficits in exploration ability can vanish with more general model improvements.

5 Discussion and limitations

This work provides critical insights into the active exploration capabilities of foundation models.

In Feature World, we find that exploration efficiency remains very close to optimal for most models with single-feature reward functions, and for at least one model with more complex reward functions.

While Gemini 1.5 Pro showed efficient information gathering in the 3D Feature World, as well as the ability to draw conclusions about the environment dynamics (when vision errors were excluded), the experiments also underscored that accurate multi-modal processing can be a significant challenge, potentially bottlenecking performance more than reasoning capabilities alone.

We show Alchemy is a challenging benchmark for meta-learning and strategy adaptation, where foundation models struggle without prompt augmentations. Critically, we find that inter-trial summarization unlocks these abilities, suggesting they are latent capabilities that are not fundamentally out of reach for future models.

One limitation of our work is that our 3D embodied Feature World experiments are a preliminary proof-of-concept with a limited scope, involving a single model in a low-complexity setting. Furthermore, by using a human-in-the-loop for motor control, we intentionally abstracted away the challenges of action generation and physical grounding. A full assessment of embodied foundation models would require integration with, or generation of, motor policies.

Overall, we demonstrate that the frontier of autonomous exploration lies in complex, multi-trial environments where models must continuously integrate information to meta-learn and adapt a world model. While challenging, these abilities can be elicited with prompt augmentations like summarization, suggesting no intrinsic barrier to their emergence. Benchmarks like Alchemy are crucial for testing these capabilities as models improve and we enter the “era of experience” [Silver and Sutton, 2025].

References

J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.

Anthropic. Claude 3.7 Sonnet and Claude Code, 2025. URL `https://www.anthropic.com/news/claude-3-7-sonnet`. Accessed: 2025-04-30.

A. P. Badia, P. Sprechmann, A. Vitvitskyi, D. Guo, B. Piot, S. Kapturowski, O. Tieleman, M. Arjovsky, A. Pritzel, A. Bolt, et al. Never Give Up: Learning Directed Exploration Strategies. In *International Conference on Learning Representations*, 2020.

G. Brockman, V. Cheung, T. Hester, J. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Y. Burda, H. Edwards, A. Storkey, and O. Klimov. Exploration by Random Network Distillation. *arXiv preprint arXiv:1810.12894*, 2018.

J. Coda-Forno, M. Binz, Z. Akata, M. Botvinick, J. Wang, and E. Schulz. Meta-in-context learning in large language models. In *Advances in Neural Information Processing Systems*, 2023.

W. Dai, N. Lee, B. Wang, Z. Yang, Z. Liu, J. Barker, T. Rintamaki, M. Shoeybi, B. Catanzaro, and W. Ping. NVLM: Open Frontier-Class Multimodal LLMs. *arXiv preprint arXiv:2409.11402*, 2024.

A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Embodied Question Answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini, et al. Molmo and PixMo: Open Weights and Open Data for State-of-the-Art Multimodal Models. *arXiv preprint arXiv:2409.17146*, 2024.

C. J. Donahue, M. F. Glasser, T. M. Preuss, J. K. Rilling, and D. C. Van Essen. Quantitative assessment of prefrontal cortex in humans relative to nonhuman primates. *Proc. Natl. Acad. Sci. U. S. A.*, 115 (22):E5183–E5192, May 2018.

A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.

V. Dwaracherla, S. M. Asghari, B. Hao, and B. Van Roy. Efficient Exploration for LLMs. *arXiv preprint arXiv:2402.00396*, 2024.

A. Ecoffet, J. Huizinga, J. Lehman, K. O. Stanley, and J. Clune. Go-Explore: a New Approach for Hard-Exploration Problems. *arXiv preprint arXiv:1901.10995*, 2019.

Y. Feng, Y. Wang, J. Liu, S. Zheng, and Z. Lu. LLaMA Rider: Spurring Large Language Models to Explore the Open World. *arXiv preprint arXiv:2310.08922*, 2023.

Gemini Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al. Gemini: A Family of Highly Capable Multimodal Models. *arXiv preprint arXiv:2312.11805*, 2023.

Google. Gemini 2.5: Our most intelligent AI model, 2025. URL `https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/#gemini-2-5-thinking`. Accessed: 2025-04-30.

Z. Guo, S. Thakoor, M. Pîslar, B. Avila Pires, F. Altché, C. Tallec, A. Saade, D. Calandriello, J.-B. Grill, Y. Tang, et al. BYOL-Explore: Exploration by Bootstrapped Prediction. In *Advances in Neural Information Processing Systems*, 2022.

S. Haq, N. Chhaya, P. Pandey, and P. Bhattacharya. Is your LLM trapped in a mental set? investigative study on how mental sets affect the reasoning capabilities of LLMs. *arXiv [cs.CL]*, Jan. 2025.

Z. Hu, C. Liu, X. Feng, Y. Zhao, S.-K. Ng, A. T. Luu, J. He, P. W. Koh, and B. Hooi. Uncertainty of Thoughts: Uncertainty-Aware Planning Enhances Information Seeking in Large Language Models. *arXiv preprint arXiv:2402.03271*, 2024.

X. Huang, W. Liu, X. Chen, X. Wang, D. Lian, Y. Wang, R. Tang, and E. Chen. WESE: Weak Exploration to Strong Exploitation for LLM Agents. *arXiv preprint arXiv:2404.07456*, 2024.

A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, et al. Mixtral of Experts. *arXiv preprint arXiv:2401.04088*, 2024.

A. Krishnamurthy, K. Harris, D. J. Foster, C. Zhang, and A. Slivkins. Can large language models explore in-context? *arXiv preprint arXiv:2403.15371*, 2024.

C. Lu, S. Hu, and J. Clune. Intelligent Go-Explore: Standing on the Shoulders of Giant Foundation Models. *arXiv preprint arXiv:2405.15143*, 2024a.

C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. *arXiv preprint arXiv:2408.06292*, 2024b.

A. Majumdar, A. Ajay, X. Zhang, P. Putta, S. Yenamandra, M. Henaff, S. Silwal, P. Mcvay, O. Maksymets, S. Arnaud, et al. OpenEQA: Embodied Question Answering in the Era of Foundation Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

F. A. Mansouri, M. J. Buckley, and K. Tanaka. Mnemonic function of the dorsolateral prefrontal cortex in conflict-induced behavioral adjustment. *Science*, 318(5852):987–990, Nov. 2007.

OpenAI. Hello GPT-4o, 2024. URL `https://openai.com/index/hello-gpt-4o`. Accessed: 2025-04-30.

OpenAI. OpenAI o3 and o4-mini system card. May 2025.

I. Osband, Y. Doron, M. Hessel, J. Aslanides, E. Sezener, A. Saraiva, K. McKinney, T. Lattimore, C. Szepesvari, S. Singh, B. Van Roy, R. Sutton, D. Silver, and H. Van Hasselt. Behaviour suite for reinforcement learning. *arXiv [cs.LG]*, Aug. 2019.

D. Paglieri, B. Cupiał, S. Coward, U. Piterbarg, M. Wolczyk, A. Khan, E. Pignatelli, Kuciński, L. Pinto, R. Fergus, J. N. Foerster, J. Parker-Holder, and T. Rocktäschel. BALROG: Benchmarking agentic LLM and VLM reasoning on games. *arXiv [cs.AI]*, Nov. 2024.

D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven Exploration by Self-supervised Prediction. In *International Conference on Machine Learning*, 2017.

M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

A. Z. Ren, J. Clark, A. Dixit, M. Itkina, A. Majumdar, and D. Sadigh. Explore until Confident: Efficient Exploration for Embodied Question Answering. *arXiv preprint arXiv:2403.15941*, 2024.

B. Romera-Paredes, M. Barekatain, A. Novikov, M. Balog, M. P. Kumar, E. Dupont, F. J. R. Ruiz, J. S. Ellenberg, P. Wang, O. Fawzi, P. Kohli, A. Fawzi, J. Grochow, A. Lodi, J.-B. Mouret, T. Ringer, and T. Yu. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2023.

A. Ruoss, F. Pardo, H. Chan, B. Li, V. Mnih, and T. Genewein. LMAct: A benchmark for in-context imitation learning with long multimodal demonstrations. *arXiv [cs.AI]*, Dec. 2024.

A. Saade, S. Kapturowski, D. Calandriello, C. Blundell, P. Sprechmann, L. Sarra, O. Groth, M. Valko, and B. Piot. Unlocking the Power of Representations in Long-term Novelty-based Exploration. *arXiv preprint arXiv:2305.01521*, 2023.

B. Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

D. Silver and R. S. Sutton. Welcome to the era of experience. *Google AI*, 1, 2025.

A. Tam, N. Rabinowitz, A. Lampinen, N. A. Roy, S. Chan, D. Strouse, J. Wang, A. Banino, and F. Hill. Semantic Exploration from Language Abstractions and Pretrained Representations. In *Advances in Neural Information Processing Systems*, 2022.

W. Tan, Z. Ding, W. Zhang, B. Li, B. Zhou, J. Yue, H. Xia, J. Jiang, L. Zheng, X. Xu, et al. Towards General Computer Control: A Multimodal Agent for Red Dead Redemption II as a Case Study. *arXiv preprint arXiv:2403.03186*, 2024.

Y. Tassa, Y. Doron, A. Mulhern, T. Erez, Y. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, G. Barth-Maron, M. Hessel, et al. DeepMind Control Suite. *arXiv preprint arXiv:1801.00662*, 2018.

S. Thrun and L. Pratt. Learning to learn: Introduction and overview. In *Learning to Learn*, pages 3–17. Springer US, Boston, MA, 1998.

E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

Y. Tong, D. Li, S. Wang, Y. Wang, F. Teng, and J. Shang. Can LLMs learn from previous mistakes? investigating LLMs' errors to boost for reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3065–3080, Stroudsburg, PA, USA, 2024. Association for Computational Linguistics.

T. Trinh, Y. T. Wu, Q. Le, H. He, and T. Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625:476–482, 2024.

G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv preprint arXiv:2305.16291*, 2023a.

J. X. Wang, M. King, N. Porcel, Z. Kurth-Nelson, T. Zhu, C. Deck, P. Choy, M. Cassin, M. Reynolds, F. Song, et al. Alchemy: A benchmark and analysis toolkit for meta-reinforcement learning agents. *arXiv preprint arXiv:2102.02926*, 2021.

Z. Wang, S. Cai, A. Liu, Y. Jin, J. Hou, B. Zhang, H. Lin, Z. He, Z. Zheng, Y. Yang, et al. JARVIS-1: Open-World Multi-task Agents with Memory-Augmented Multimodal Language Models. *arXiv preprint arXiv:2311.05997*, 2023b.

A. L. Zhang, T. L. Griffiths, K. R. Narasimhan, and Ofir Press. VideoGameBench: Can vision-language models complete popular video games? *arXiv [cs.AI]*, May 2025.

H. Zhu, R. Kapoor, S. Y. Min, W. Han, J. Li, K. Geng, G. Neubig, Y. Bisk, A. Kembhavi, and L. Weihs. EXCALIBUR: Encouraging and Evaluating Embodied Exploration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

A Technical Appendices and Supplementary Material

A.1 Detailed related work

Exploration in RL Information gathering is related to exploration in RL, which has been studied for tasks with sparse rewards such as Montezuma's Revenge and Pitfall in Atari, Deep-sea exploration and other tasks in the Behavior Suite for RL, and the DM-HARD-8 tasks [e.g., Burda et al., 2018, Ecoffet et al., 2019, Osband et al., 2019, Guo et al., 2022, Saade et al., 2023] as well as in unsupervised settings [e.g., Pathak et al., 2017, Guo et al., 2022]. These methods commonly derive an "intrinsic" reward from the error of a predictive model [e.g., Pathak et al., 2017, Burda et al., 2018, Guo et al., 2022] or by estimating the density of visited states [e.g., Saade et al., 2023]. Badia et al. [2020] use a combination of both of these types of intrinsic rewards and Tam et al. [2022] additionally use pre-trained representations. In contrast, this work relies on the prior knowledge of foundation models from internet-scale pre-training for exploration [e.g., Wang et al., 2023a, Feng et al., 2023, Lu et al., 2024a] rather than using random exploratory actions and intrinsic rewards. Also, existing RL environments [e.g., Todorov et al., 2012, Brockman et al., 2016, Tassa et al., 2018] often conflate exploration with other aspects of agent performance, making it difficult to isolate and assess a model's inherent exploratory capabilities. Such aspects include sparse or deceptive rewards and noisy, non-stationary, or multi-agent environments. We therefore chose and designed a suite of environments that allows us to systematically disentangle and control the factors influencing exploration.

Foundation models for games Foundation models have also been used to build agents that play games [e.g., Wang et al., 2023a,b, Feng et al., 2023, Tan et al., 2024], which often involves some form of exploration. Wang et al. [2023a] show that GPT-4 can reach impressive performance in Minecraft by incrementally building a skill library via an "automatic curriculum" stage where GPT-4 is prompted to propose novel tasks. Feng et al. [2023] prompt an LLM to explore an environment and subsequently use the collected experiences for fine-tuning the model. Unlike Wang et al. [2023a] and Feng et al. [2023], Wang et al. [2023b] and Tan et al. [2024] use image observations rather than relying on access to environment internal states. All of these works, however, focus more on improving agent performance rather than performing an explicit, systematic investigation of information gathering, meta-learning, and strategy adaptation with foundation models in controlled, zero-shot settings and in comparison to known optimal policies. LMAct benchmarks LLMs on simple games in the very long context regime and VideoGameBench tests VLMs on a collection of video games [Zhang et al., 2025, Ruoss et al., 2024]. However, neither of these works investigates exploration as a capability distinct from overall performance. The BALROG benchmark incorporates a range of existing games used as RL environments and investigates exploration among a number of key capabilities [Paglieri et al., 2024]. However, this is limited to a qualitative assessment and does not involve quantitative measurement of multiple clearly-defined facets of exploration as this work does.

Exploration with foundation models Several other works investigate exploration with foundation models, e.g., for text-based environments [Lu et al., 2024a, Huang et al., 2024], reinforcement learning from human feedback (RLHF) [Dwaracherla et al., 2024], and multi-armed bandit problems [Coda-Forno et al., 2023, Krishnamurthy et al., 2024]. Unlike Krishnamurthy et al. [2024] and Dwaracherla et al. [2024] and similar to Lu et al. [2024a], this work considers stateful environments. While Lu et al. [2024a] replace components of the exploration method introduced in Ecoffet et al. [2019] with an LLM, this work studies the ability of foundation models to gather information and test hypotheses in-context via zero-shot prompting rather than using LLMs in a more modular fashion. Also adopting more modular approaches, Hu et al. [2024] use foundation models as components in a larger exploration framework and Huang et al. [2024] propose to use a smaller agent to explore the environment and a larger agent to leverage the gathered information.

Active learning The field of active learning [Settles, 2009] has studied how to best acquire data to improve model predictions with methods that commonly focus on highly structured data (either i.i.d. or on a graph). In contrast, this work explores efficient knowledge acquisition in more general interactive environments.

Active embodied question answering This work studies a similar setup to embodied question answering (EQA) [e.g., Das et al., 2018, Zhu et al., 2023, Majumdar et al., 2024, Ren et al., 2024]. Similar to our work, agents in the EQA setting need to actively explore an environment to gather information. Unlike our tasks, EQA typically does not involve performing iterative experiments to infer unknown mechanisms in dynamic environments, and the optimal exploratory action typically does not depend on past observations.

AI for science Hypothesis generation and testing is central to the scientific method and recent works research the application of foundation models in this broader domain [e.g., Romera-Paredes et al., 2023, Trinh et al., 2024, Lu et al., 2024b]. Such works typically use foundation models in a highly structured protocol designed for the fixed domain in question. In contrast, this work explores the capabilities of base foundation models as the complexity of the domain varies.

Strategy adaptation The dorsolateral prefrontal cortex (dlPFC) is the brain region that has undergone the most reorganization in humans relative to other primates [Donahue et al., 2018]. It is involved in detecting when a previously-successful strategy no longer yields expected rewards and adjusting behavioral strategy accordingly [Mansouri et al., 2007]. Analogs of this capability have been studied in LLMs in the context of mental sets [Haq et al., 2025], and learning from mistakes [Tong et al., 2024]. However, these studies focus on single-turn static math and reasoning problems. To our knowledge, this is the first study to investigate strategy adaptation with LLMs in the context of either exploration or interactive environments.

A.2 Prompts for Feature World

Prompts used for the foundation models in Feature World. Prompts are provided verbatim, with the exception of newlines added to fit the text within the table boundaries.

Task	Prompt
Text Environment Single Factor Task	

```
You are playing a text-based game. Your goal is to find which object property leads to a non-zero reward in as few steps as
possible.

Game Rules:
- There are objects with different colors and shapes.
- Picking up an object gives you a reward of either 0 or 1.
- There is a single property, i.e., one particular color OR shape, that leads to a reward of 1.

{scene_description}

{action_reward_description}

Respond with this format, please be specific about the object:

* Action: pick up <colored> <object>
* Stop: <YES> or <NO>
*
* Which factor influence reward? <COLOR> or <SHAPE>  or <UNSURE>
* WINNING COMBINATION: <State the specific color or shape that leads to reward>

Explain your reasoning thoroughly.
```

Table 1: In-context prompt used for the text Feature World environments for the single-feature tasks.

Task	Prompt
Text Environment Multi Factor Task	

```
You are playing a text-based game. Your goal is to find which combination of object properties leads to a non-zero reward in
as few steps as possible.

Game Rules:
- There are objects with different colors, shapes, and textures.
- Picking up an object gives you a reward of either 0 or 1.
- There is a single combination of two properties, i.e., a color and shape, a shape and texture, or a color and texture,
that leads to a reward of 1.

{scene_description}

{action_reward_description}

Respond with this format, please be specific about the object:

* Action: pick up <colored> <textured> <object>
* Stop: <YES> or <NO>
*
* Which combination of factors influence reward? <COLOR, SHAPE> or <COLOR, TEXTURE> or <TEXTURE, SHAPE>  or <UNSURE>
* WINNING COMBINATION: <State the specific combination of properties (e.g., color and shape, shape and texture, or color and
texture.>

Explain your reasoning thoroughly.
```

Table 2: In-context prompt used for the text Feature World environments for the conjunction tasks.

Task	Prompt
Text Environment Single Factor Task Structured JSON	

```
You are playing a text-based game. Your goal is to find which object property leads to a non-zero reward in as few steps as
possible.

Game Rules:
- There are objects with different colors and shapes.
- Picking up an object gives you a reward of either 0 or 1.
- There is a single property, i.e., one particular color OR shape, that leads to a reward of 1.

{scene_description}

{action_reward_description}

Your response must conform to the following JSON format:
{{
    "next_object_picked_up": What object should be picked up next? This should be in format <COLOR> <SHAPE>.
    "stop": YES or NO, do you want to end the game after this step?
    "rewarding_factor": What factor do you think most is related to reward? <COLOR> or <SHAPE> or <UNSURE>
    "winning_combination": State the specific color or shape that leads to reward. If you're not sure, it's okay to say
    <UNSURE>
}}
```

Table 3: In-context prompt used for the text Feature World environments for the single-feature tasks, using a structured JSON output format[2].

Task	Prompt
Text Environment Multi Factor Task Structured JSON	

```
 You are playing a text-based game. Your goal is to find which combination of object properties leads to a non-zero reward in
 as few steps as possible.

Game Rules:
- There are objects with different colors, shapes, and textures.
- Picking up an object gives you a reward of either 0 or 1.
- There is a single combination of two properties, i.e., a color and shape, a shape and texture, or a color and texture,
that leads to a reward of 1.

{scene_description}

{action_reward_description}

Your response must conform to the following JSON format:
{{
    "next_object_picked_up": What object should be picked up next? This should be in format <COLOR> <TEXTURE> <SHAPE>.
    "stop": YES or NO, do you want to end the game after this step?
    "rewarding_factor": What factors do you think most are related to reward? <COLOR, SHAPE> or <COLOR, TEXTURE> or
    <TEXTURE, SHAPE>  or <UNSURE>
    "winning_combination": State the specific combination of properties (color and shape, shape and texture, or color and
    texture) that leads to
    reward. If you're not sure, it's okay to say <UNSURE>
}}
```

Table 4: In-context prompt used for the text Feature World environments for the conjunction tasks, using a structured JSON output format.

Task	Prompt
3D Environment Iterative Exploration: vision	

```
You are an expert video game player who is annotating videos of gameplay.

In this game, the player controls a robot in a factory room, which contains objects of various shapes and
colors, such as red planks, blue cubes, green cylinders, orange disks, yellow pyramids, etc.
The player can pick up and move objects using a blue laser beam.
The player is trying to place the correct type of object on the conveyor belt.
If the object is correct, the object disappears in the machine and the light on the machine turns green.
If the object is incorrect, the light on the machine turns red and the object is pushed off.

The possible colors are red, green, blue, yellow, purple, and orange.
The possible shapes are cylinder, cube, plank/board, pyramid, and disk.

Your goal is to accurately and comprehensively list every object that the player places on the input
conveyor belt, along with the timestamp of when the object was placed and whether the object is correct
or incorrect.

Your response should be in the following format:
0 [timestamp 0] <1st object placed on conveyor> : <correct / incorrect>
1 [timestamp 1] <2nd object placed on conveyor> : <correct / incorrect>
2 [timestamp 2] <3rd object placed on conveyor> : <correct / incorrect>
3 [timestamp 3] <4th object placed on conveyor> : <correct / incorrect>
...
```

Task	Prompt
3D Environment Iterative Exploration: reasoning	

```
Now we want to explain how this game works.
The goal of the game is to place all objects with the right property, such as a particular color or
shape, on the conveyor belt.
Let’s try to find the next action to take to figure out what factor (color or shape) determines the
correctness of the object.

If there is no history of objects yet, tell the player to pick up a random object you can see in the room
from the video.
If you have no video input yet, tell the player to explore the room.
Otherwise, follow the instructions below.

  Important: You have VERY FEW turns left. Choose your next action carefully to maximize information.

  Think step-by-step:

  1. What pattern do you see in the correct objects so far?
  2.  **Consider which colors and shapes have NEVER been correct. This eliminates BOTH the color AND shape
  from being correct.**
  3. What color or shape seems MOST promising to test next?
  4. Why will this choice give you the most useful information, even if it isn’t a correct object?

  Explain your reasoning thoroughly. Don’t just guess! Each turn is precious.

  After doing your reasoing, respond at the end with this format, please be specific about the object:

  * CORRECT PROPERTY: <COLOR> or <SHAPE> or <UNSURE>
  * NEXT COMMAND: place the <colored> <object> on the conveyor belt.
```

Table 5: In-context prompts used for the 3D Construction Lab environment in the exploration phase for the Gemini agent.

Task	**Prompt**

3D Environment Trajectory Review: vision	`You are an expert video game player who is annotating videos of gameplay.`
	
	`In this game, the player controls a robot in a factory room, which contains objects of`
	`various shapes and colors, such as red planks, blue cubes, green cylinders,`
	`orange disks, yellow pyramids, etc.`
	`The player can pick up and move objects using a blue laser beam. The player`
	`is trying to place the correct type of object on the conveyor belt. If the object`
	`is correct, the object goes through and the light on the machine turns green.`
	`If the object is incorrect, the light on the machine turns red and the object`
	`is pushed off.`
	
	`The possible colors are red, green, blue, yellow, purple, and orange.`
	`The possible shapes are cylinder, cube, plank/board, pyramid, and disk.`
	
	`Your goal is to accurately and comprehensively list every object that the`
	`player places on the input conveyor belt, along with the timestamp of when the object was placed`
	`and whether the object is correct or incorrect.`
	
	`Your response should be in the following format:`
	`0 [timestamp 0] <1st object placed on conveyor> : <correct / incorrect>`
	`1 [timestamp 1] <2nd object placed on conveyor> : <correct / incorrect>`
	`2 [timestamp 2] <3rd object placed on conveyor> : <correct / incorrect>`
	`3 [timestamp 3] <4th object placed on conveyor> : <correct / incorrect>`
	`...`

3D Environment Trajectory Review: reasoning	`Now we want to explain how this game works.`
	`The goal of the game is to place all objects with the right property, such as a particular color or shape,`
	`on the conveyor belt.`
	
	`Based on the observations above of which objects were placed on the conveyor belt`
	`and which ones were correct or incorrect,  explain your reasoning and state what the right object`
	`property is.`
	`The right property is either a specific shape or a specific color.`
	
	`Your response should be in the following format:`
	`REASONING: <Explain your reasoning for how you deduced the right object property.>`
	`TARGET PROPERTY: <State what the specific correct shape OR specific correct color is.>`

3D Environment Trajectory Review: generalization	`Based on what you determined the correct object property to be, state whether`
	`    each of the following objects would be correct if placed on the conveyor belt:`

Table 6: In-context prompts used for the 3D Construction Lab environment in the review phase for all agent conditions.

A.3 Additional results - Feature World

A.3.1 Description of optimal strategies

To illustrate the optimal strategy, consider a task where the hidden rewarding property is “red.” The strategy unfolds in two phases. The first phase is exploration, where the goal is to find a successful object by maximizing information gain. If an attempt on a “blue toy” fails, the agent learns that *if* color is the rule, “blue” is not the answer, and *if* shape is the rule, "toy" is not the answer. The optimal next action is to test an object with entirely new features, like a “yellow sphere”, to efficiently explore the remaining possibilities.

Once an action succeeds—for example, picking up a “red box”—the strategy shifts to the second phase: isolation. The agent must now disambiguate whether “red” or “box” is the true cause. The optimal way to do this is to test a new object that changes only one of these features, such as a “red sphere” or a “green box”, to definitively pinpoint the rewarding property.

A.4 Exploration in 3D embodied environments: Construction Lab

To further evaluate the foundation models in a 3D embodied environment, we implement an analogous task to the text-based environment in a factory-style simulation called Construction Lab. Construction Lab was introduced in **[reference-anonymized]** as a simulation environment that includes both game-like mechanics and simplified but non-trivial object manipulation and physical reasoning.

In this work, we focus on a task that requires the player to operate a simple machine called the Exchanger. The Exchanger requires objects with specific properties to be placed on an input conveyor belt (Figure 2). If an object matches the requirement, the input is consumed, a green light shows for a few seconds, and an output object is produced on an output belt. If the object is invalid, the machine rejects it by reversing the input belt and a red error light is activated. No cues are provided regarding the correct input object required, and thus the task entails determining what the correct object properties are through trial and error, observing how the machine responds to input objects, and drawing appropriate inferences.

Through the use of this 3D, visually rich environment that mirrors the challenges of the text-based environment, we are able to investigate the effects of visual complexity on active information gathering, reasoning, and hypothesis testing.

A.4.1 Task setup

A number of additional challenges must be addressed when performing this exploration task in a 3D embodied environment. First, the agent must assess both the current state of the environment and the consequences of any actions taken through vision. Second, the agent requires a motor control module to execute exploratory actions. We use Gemini 1.5 Pro’s multi-modal functionality to ingest video input from Construction Lab sub-sampled to 1.5 Hz and 320 x 240 resolution. To disentangle vision and reasoning performance from translation of natural language instructions into a complex keyboard-and-mouse action space, we adopt a setup in which instructions are provided to a human actor who performs the exploratory actions online.

We assess Gemini 1.5 Pro’s ability to generate these exploratory instructions by comparing against an optimal and random baseline, mirroring those in the text environment. The optimal strategy was performed by a single human performing the task according to an optimal policy that maximally reduced uncertainly about the correct property. The random strategy was performed according to a policy that selects a random object from the room at each step, with replacement.

As running the 3D environment and using human actors in the loop reduces experimental throughput, we limit ourselves to a single level of environment and reward function complexity. We choose the condition with 3 colors and 1 causal factor, as conditions with more colors had significant visual clutter. Each task is randomly generated as follows: at the beginning of the episode, 3 unique colors and 3 unique shapes are randomly selected from 6 colors and 5 shapes, and objects with each shape-color combination are placed in random locations in the environment, for a total of 9 objects. One property, either a shape or a color, is randomly selected as the correct property, for a total of 3 correct objects. The player and the Exchanger machine with input and output conveyor are likewise placed randomly in the room. A gameplay episode ends when either all 3 correct objects are placed on the input conveyor or 2 minutes have elapsed.

A.4.2 Gemini-based agent

The Gemini agent is implemented as follows: every 10 seconds, the model is fed the most recent 100 video frames (or 67 seconds) of gameplay and queried in two stages, during which gameplay is paused for the human actor. We implement a two-stage procedure with a vision stage and a reasoning stage, which we found improves accuracy for each stage compared with running both together. In the first stage, Gemini is asked to list, for every object placed on the input conveyor, the timestamp at which it was placed, its color and shape, and whether it was correct or not (as indicated by a red or green light on the machine). In the second stage, Gemini is provided the output of the first stage (subsequent video frames and list of objects placed with their reward values) and prompted to select a next exploratory action to maximize information gain, similar to the text environment. The human actor is provided only with the command generated by the second stage, such as "place the red cube on the conveyor."

All video trajectories are processed in the same way, regardless of how the exploration instructions were generated. Specifically, we truncate the video to include only the first 4 object attempts. Gemini is then called on the truncated video in three steps: vision, reasoning, and generalization. In the vision step, it is asked to list all the objects placed on the conveyor and whether they were correct, similar to the vision step in the exploration policy. In the reasoning step, it is asked to deduce the correct object property based on its observations. In the generalization step, it is asked to predict whether each object in a list of hypothetical objects would give a reward. See Appendix A.2 for specific prompts used.

A.4.3 Evaluation

To evaluate different aspects of performance for each agent type, we measure relevant property accuracy and number of objects until sufficient information is acquired to determine the correct property, assuming perfect reasoning. We also record the number of vision errors made by the VLM when listing objects in the full video, defined as misclassifying the shape, color, or correctness of an object placed on the conveyor, or omitting mention of an object placed on the conveyor. Because internal game states are not exposed in our experiments, we use manual human annotation of video trajectories to collect the above metrics and error counts. We collect a total of 15 trajectories for each agent type.

A.4.4 Results

In the exploration efficiency metric, we see the same trends in the results for the 3D embodied environment as for the text environment, with Gemini's exploration efficiency significantly outperforming the random baseline and approaching the optimal baseline (Figure 4b). These results suggest that the additional complexity of an imperfect vision system and partially observed environment state are not significant limitations in generalizing directed exploration capabilities to embodied 3D environments. In the accuracy metric (Figure 4c), the picture is more nuanced. For relevant property accuracy, the difference between performance with the Gemini agent and the random agent was not statistically significant ($p > 0.05$, paired sample t-test).

This result is interesting because VLM vision is also necessary for the exploration phase, where there was no discrepancy in performance. A likely reason for this is that the iterative nature of the exploration task makes it robust to occasional errors. Because the model must re-list all objects placed at each step, chance errors made during one step do not propagate to later steps.

To probe the reason for the gap in accuracy performance, we also computed results where we filtered out trajectories in which the vision step made an error (Figure 4c). In these results, accuracies for the Gemini and optimal agents are nearly identical and their differences with the random agent are statistically significant ($p < 0.05$, two sample t-test). These results suggest that errors in the vision step, rather than reasoning or exploration, are responsible for the relatively reduced accuracy in the Gemini agent condition.

Taken together, results in the Construction Lab show that the directed exploration capabilities of foundation models robustly generalize from text-based environments to embodied 3D environments, though overall accuracy of the system is somewhat reduced by imperfect performance of the VLM's object and action recognition in videos. This indicates that the challenges of multi-modal reasoning from realistic simulated video could be addressed by focusing on the vision and action recognition capabilities of foundation models separately from their reasoning capabilities.

A.5 Additional Alchemy details and results

A.5.1 Invariant principles in Alchemy

While the specific chemistry changes per episode, Alchemy includes invariant principles or abstract regularities that span all episodes. These invariants are crucial because discovering and exploiting them over many episodes is the essence of the meta-learning problem in Alchemy.Such invariant properties include, among others: 1. within an episode, stones with the same visual features have the same value and respond identically to potions, and potions of the same color likewise have the same effects; 2. potions come in fixed pairs (e.g., red/green, yellow/orange, pink/turquoise) which always have opposite effects; 3. the underlying causal graph topology is structured by a generative grammar, though some edges might be missing, creating "bottlenecks"; and 4. the maximum stone score is 15, and the minimum score is -3. It's important to note that all of our experiments were conducted without any post-training, and so all models are presumed to lack knowledge of these invariant task structures.[3] We therefore conduct experiments to test the effects of including various components of these invariant properties into the prompt. This allows us to disentangle the challenge of acquiring such meta-learned knowledge from that of already possessing it and being able to use it to inform smart exploration.

In the case with *no prior information*, the model is given the main prompt to introduce Alchemy, which describes the general gameplay mechanics (Figure 9). We hypothesized that, in order to learn which potions optimally improve stone reward value for a given combination of stone properties, the models require an understanding of the invariant properties of the Alchemy tasks, which provide a framework on which the model can integrate evidence from its observations. To test this hypothesis, we provided additional information about the reward, potion pairing, and causal mechanics of the game (*prior information* condition, Figure 9).

Main prompt	You are playing a text-based game called Alchemy where you place stones of different shapes (round or pointy), sizes (small or large), and colors (blue or purple) into potions that change the stones' properties. The goal is to maximize the reward value of the stones, and then place them into the cauldron to increase the total score. Placing a stone in the cauldron adds the current reward value of that stone to the total score. However, there might be more rules to the game than mentioned here.
Reward information	The maximum stone reward value is 15 and the minimum is -3.
Potion pairing information	There are six different potion colors, which come in three pairs: yellow/orange, red/green, and pink/turquoise. Potions in a pair have the opposite effect from each other on a given property.
Causal information	A stone's reward value is determined by its properties. The game is deterministic: a potion of a given color always has the same effect on a stone with given properties, and a stone with given properties always has the same reward value.

Figure 9: Prompt and different components comprising the *prior information* (reward information, potion pairing information, and causal information).

---

[3] Because the Alchemy environment was published in 2021, knowledge of it may be included in the pretraining data for the foundation models studied here. However, when we probed the models, we found they entirely lacked knowledge about potion pairs and min/max rewards and had limited knowledge of the environment in general.

A.5.2 Cross-trial summarization

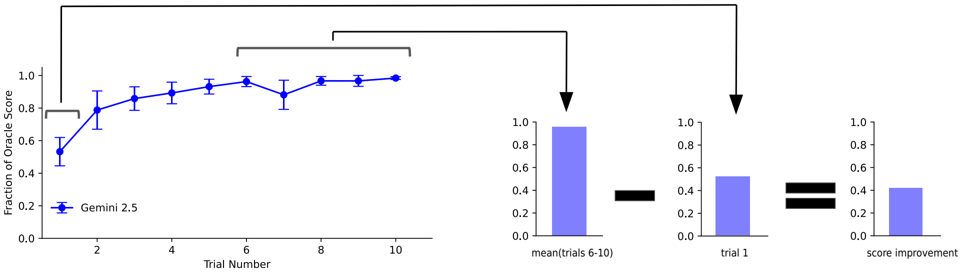
The multi-trial structure of Alchemy introduces a need to manage very long contexts and perform inference across multiple timescales and abstraction—a challenge that often arises in real-world, and particularly multi-modal, settings. Even in the case that it is possible to fit all observations, actions, and rewards from a multi-turn interaction within the model's context window, it is unclear whether current models are capable of natively managing ever-growing context while maintaining inference and reasoning capabilities. We therefore examined and along-context summarization method. We hypothesized that summarization would offer a benefit due to the multi-trial structure of the task that requires extended directed exploration and inference across multiple timescales.

Specifically, the summary is constructed at the end of each trial by asking the model to create a summary conditioned on the current trial's events as well as the summary from the previous trial. At each step, the model is asked to take actions based on the summarization as well as the current trial's events so far. We additionally have a *no summarization* condition, in which all episode events up until the current step are included in the context.

A.5.3 Alchemy experimental setup details

We implemented a custom version of symbolic Alchemy, modified so that observations and actions are given in natural language format. Episode chemistries are sampled every episode and held constant for the duration of 10 trials, although new stones and potions are resampled every trial. Trials last for a maximum of 20 steps, and the agent could choose to end the episode at any step. At each step, the agent could choose to select a stone and then either a potion or the cauldron (to cache the stone and obtain the reward), or to end the trial. See Figure 5 (d) for a visualization of the tasks.

We considered the condition of procedural generation in which the transition graph has random bottlenecks: This is the hardest version of the task in which certain transformations (e.g., the effect of a blue potion) are only possible if specific initial conditions of the stone are met (e.g., the stone being transformed is large and round).



(a) Computation of score improvement

Figure 10: Illustration of how score improvement is computed for an Alchemy episode.

A.5.4 Example model outputs in Alchemy

Below, we display examples of raw outputs of Gemini 2.5 from trail 1 and trial 5. For each trial, we show 1) the model's reasoning in step 1 of the trial, 2) the model's reasoning in step 5 of the trial, and 3) the model's summary of the trial. All outputs are drawn from the condition with summaries but no prior information.

For the reasoning steps, we also display the observation of the game state provided to the model by the environment, labeled GAME STATE UPDATE.

Condition | **Output**

Trial 1 Step 1

```
GAME STATE UPDATE:
New episode
New trial
New observation
You see the following stones:
0: purple small pointy with reward -1
1: blue small pointy with reward 1
2: purple large pointy with reward 1
You see the following potions:
0: pink
1: turquoise
2: turquoise
3: pink
4: yellow
5: red
6: yellow
7: red
8: red
9: turquoise
10: pink
11: orange

Current trial score: 0

MODEL RESPONSE:
OBSERVATION: This is the first action of the trial. I have not yet used any potions, so their effects are unknown.
REASONING: My initial goal is to understand the effects of the different potions. I will begin by testing the first available
potion, which is the pink potion at index 0. I am choosing to use stone 0 (purple small pointy, reward -1) for this
experiment. Since this stone has a negative reward, there is little risk in modifying it, and any positive change would be a
significant gain. This action will reveal what pink potions do to purple, small, and pointy stones.
ACTION: Place stone 0 in potion 0.
```

Trial 1 Step 5

```
GAME STATE UPDATE:
What stone do you use? 1
What potion do you use? 5
Outcome stone: purple small pointy with reward -1
New observation
You see the following stones:
0: purple large pointy with reward 1
1: purple small pointy with reward -1
2: purple large pointy with reward 1
You see the following potions:
0: None
1: None
2: None
3: pink
4: None
5: None
6: yellow
7: red
8: red
9: turquoise
10: pink
11: orange

Current trial score: 0

MODEL RESPONSE:
OBSERVATION: Placing stone 1 (blue small pointy, reward 1) into the red potion at index 5 changed its color from blue to
purple. This also inverted its reward value from 1 to -1. The stone is now a purple small pointy stone.
REASONING: The last action revealed that red potions change blue stones to purple and reduce their reward. My new stone 1 is
now a purple small pointy stone with a reward of -1, which is exactly what stone 0 was at the start of the trial. In step 2,
I learned that a turquoise potion changed stone 0 from a small purple stone to a large purple stone, increasing its reward
from -1 to 1. I will now attempt to replicate this effect on the new stone 1 using the turquoise potion at index 9. This will
both confirm my hypothesis about turquoise potions and improve the value of a stone.
```

Trial 1 Summary

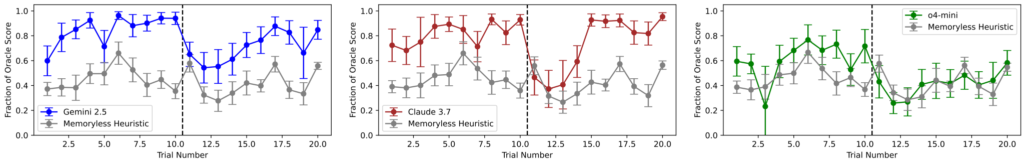
```
### Potion Effects
* **Pink Potion:** Appears to have no effect. It was tested on a ‘purple small pointy‘ stone with no change.
* **Red Potion:** Changes a stone’s color from ‘blue‘ to ‘purple‘. It does not affect ‘purple‘ stones. The effect on the
reward depends on the stone’s size: the reward is inverted for ‘small‘ stones but stays the same for ‘large‘ stones.
* **Turquoise Potion:** Changes the size of ‘purple small‘ stones to ‘purple large‘. It has no effect on ‘blue‘ stones or
already ‘large‘ stones. The reward is inverted.
* **Yellow Potion:** Changes a stone’s shape from ‘round‘ to ‘pointy‘. It has no effect on ‘pointy‘ stones. The reward is
inverted.
* **Orange Potion:** Changes a stone’s shape from ‘pointy‘ to ‘round‘. It has no effect on ‘round‘ stones. The reward is
inverted.
### Highest Reward Combination
The highest reward value seen in the game is **+1**. Based on the potion effects, the combinations of properties that result
in this highest reward
are:
* ‘blue small pointy‘
* ‘purple large pointy‘
* ‘blue large pointy‘
* ‘purple small round‘
```

Table 7: Outputs of Gemini 2.5 for Trial 1 of Alchemy in the condition with summarization and no prior information.

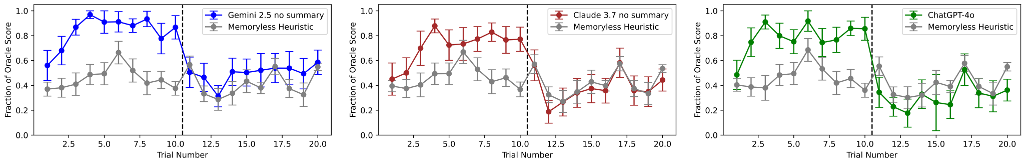
Condition	**Output**
Trial 5 Step 1	GAME STATE UPDATE: New observation You see the following stones: 0: blue small round with reward -1 1: purple large round with reward -1 2: blue large pointy with reward 15 You see the following potions: 0: orange 1: orange 2: red 3: pink 4: yellow 5: red 6: red 7: green 8: red 9: red 10: orange 11: pink MODEL RESPONSE: OBSERVATION: The game has started with one stone (Stone 2) already in the ‘blue large pointy‘ state, which I have identified as the highest possible reward state (+15). REASONING: Stone 2 currently has a reward of 15. Based on my previous findings, this is the maximum achievable reward. Using any potion on it would risk lowering its value. Therefore, the most logical first step is to place it in the cauldron immediately to lock in the +15 score. ACTION: Place stone 2 in the cauldron.
Trial 5 Step 5	GAME STATE UPDATE: What stone do you use? 0 What potion do you use? 0 Outcome stone: blue small round with reward -1 New observation You see the following stones: 0: blue small round with reward -1 1: None 2: None You see the following potions: 0: None 1: orange 2: red 3: pink 4: None 5: red 6: red 7: None 8: red 9: red 10: orange 11: pink Current trial score: 30 MODEL RESPONSE: OBSERVATION: The Orange potion had no effect on the ‘blue small round‘ stone. This confirms my hypothesis that Orange potions only work on ‘pointy‘ stones. REASONING: My only remaining stone is ‘blue small round‘ with a reward of -1. I cannot improve its properties towards the maximum reward state because I lack the necessary Yellow and Turquoise potions. Placing it in the cauldron would lower my score. I have many Red potions, and my current hypothesis is that they change ‘blue‘ stones to ‘purple‘. Since I cannot improve this stone, I will use this opportunity to test the effect of a Red potion on it to gather more data. ACTION: Place stone 0 in potion 2.
Trial 5 Summary	* **Green Potion:** Changes a stone’s color from ‘purple‘ to ‘blue‘. The reward change depends on the stone’s shape. * If the stone is ‘round‘, its reward increases by 2. * If the stone is ‘pointy‘, its reward increases by 14. * **Red Potion:** Changes a stone’s color from ‘blue‘ to ‘purple‘. This transformation consistently subtracts 2 from the stone’s reward. * **Yellow Potion:** Changes a stone’s shape from ‘round‘ to ‘pointy‘. This transformation consistently adds 2 to the stone’s reward. * **Orange Potion:** Changes a stone’s shape from ‘pointy‘ to ‘round‘. This transformation subtracts 2 from the stone’s reward. This potion has no effect on ‘round‘ stones. * **Turquoise Potion:** Changes a ‘purple small‘ stone into a ‘purple large‘ stone. This transformation adds 2 to the stone’s reward. It appears to have no effect on stones that are already ‘large‘ or are ‘blue‘. * **Pink Potion:** Has no effect on a stone’s properties or reward value. ### Highest Reward Combination The single combination of stone properties with the highest reward is: **blue large pointy** This specific combination results in a reward value of **+15**. This state is best achieved by transforming a ‘purple pointy‘ stone with a Green potion to gain the +14 reward bonus.

Table 8: Outputs of Gemini 2.5 for Trial 5 of Alchemy in the condition with summarization and no prior information.

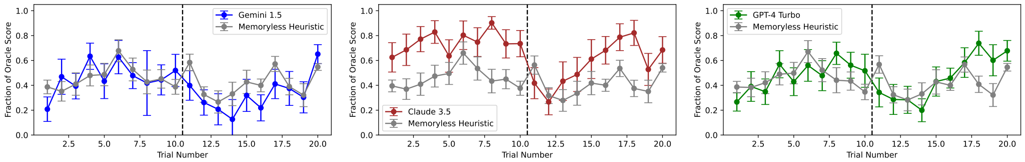
A.5.5 Strategy adaptation timeseries results



(a) Gemini 2.5 Pro, summary (b) Claude 3.7 Sonnet, summary (c) o4-mini, summary



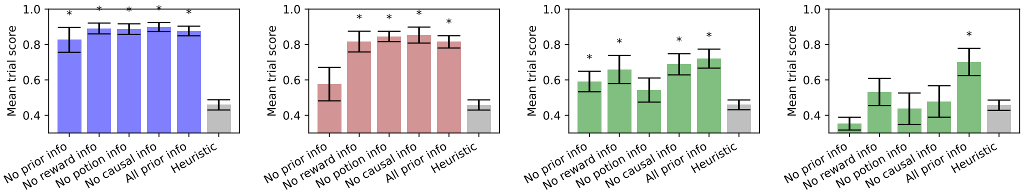
(d) Gemini 2.5 Pro, no summary (e) Claude 3.7 Sonnet, no summary (f) ChatGPT-4o, summary



(g) Gemini 1.5 Pro, summary (h) Claude 3.5 Sonnet, summary (i) GPT-4 Turbo, summary

Figure 11: Normalized model score by trial when an uncued change in chemistry occurs halfway through the episode. (a-i) Trace of score across 20 trials. The vertical dotted line denotes the point at which the change in chemistry occurs, following trial 10. Error bars represent standard error of the mean, across 10 replicates.

A.5.6 Effects of prior information ablation



(a) Gemini 2.5 (b) Claude 3.7 (c) o4-mini (d) ChatGPT-4o

Figure 12: Mean trial scores for each model with no summarization when various pieces of prior information have been removed from the prompt. (a) Prompt ablations for Gemini 2.5. (b) Prompt ablations for Claude 3.7. (c) Prompt ablations for o4-mini. N=10 replicates of 10-trial episodes. Error bars represent standard error of the mean. Asterisk indicates the mean is significantly different from that of the memoryless heuristic ($p < 0.05$, paired-sample t-test).

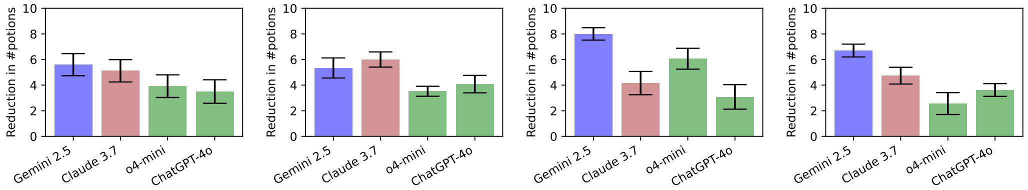
To further investigate the effect of prior information in the prompt, we performed ablations in which we removed either reward information, potion pairing information, or causal information from the prompt with prior information (see Figure 9). We ran the models with the ablated prompt and no summaries enabled and measured the performance.

We found that the effects of prior information ablation differed substantially depending on the model. Gemini 2.5 is the most robust to the ablations, showing only a slight decrease in performance when no prior information is provided (Figure 12a). Interestingly, Claude 3.7 showed no decrease in performance with any of the single ablations, but still showed a large decrease when all prior information was removed (Figure 12b). This suggests that Claude 3.7 is more robust at meta-learning without summaries, but still struggles without at least a small amount of initial prior information to build off of. Performance of o4-mini was reduced slightly without causal and reward information, and moderately without potion pair information or when no information is provided (Figure 12c). Performance of ChatGPT-4o was reduced substantially and became statistically insignificant compared to the heuristic policy with removal of any piece of prior information (Figure 12d). This shows that this model is less robust at learning principles on its own, potentially due to its lack of thinking ability.

A.5.7 Alchemy item usage

To investigate the mechanism behind the score improvement in the models, we analyzed reduction in the number of potions used by the models in later trials (Figure 13). The most apparent finding is that Gemini 2.5 shows a significantly larger reduction in potions used than the other models when summaries are enabled. This demonstrates that the summary has a substantial effect on the model's strategy, despite the fact that the summary only improves the already-high performance of Gemini 2.5 a small amount. We also found that Gemini and Claude reduced their potion use substantially more than ChatGPT-4o overall, suggesting that these models owe part of their superior performance to an ability learn to make more efficient use of potions.

Finally, to analyze the propensity of the models for clear reasoning errors in different conditions, we computed the fraction of trials in which a model places at least one negative-valued stone into the cauldron (Figure 14). Such an action is obviously counterproductive, and is never performed by the memoryless heuristic. We found that all models place negative stones into the cauldron at least a small number of times, but in most cases in less than 10% of trials. Interestingly, however, in the prior information conditions, Claude places negative stones in the cauldron in between 20% and 25% of trials, despite being a high performing model in these conditions. It is not clear why this occurs, but it suggests there is room for significant improvement in Claude on this task.



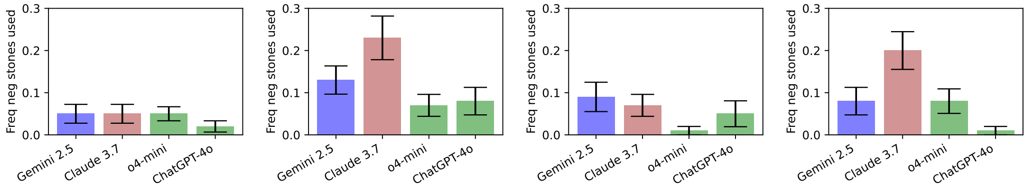
(a) No summary, no prior information

(b) No summary, prior information

(c) Summary, no prior information

(d) Summary, prior information

Figure 13: Reduction in the number of potions used over the course of the episode, computed as the potions used in the first episode minus the mean of the potions used in the last 5 episodes. (a-d) Same conditions as Figure 6. N=10 replicates of 10-trial episodes. Error bars represent standard error of the mean.



(a) No summary, no prior information

(b) No summary, prior information

(c) Summary, no prior information

(d) Summary, prior information

Figure 14: The fraction of trials in which at least one negative-valued stone was placed in the cauldron by the model. (a-d) Same conditions as Figure 6. N=10 replicates of 10-trial episodes. Error bars represent standard error of the mean.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See the introduction: Section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Experimental setups are described in sufficient detail in the main text and Appendix for reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
    (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
    (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
    (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
    (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Open access to the data and code for Alchemy are provided in the paper introducing that environment [Wang et al., 2021]. Open access to data and code for Feature World and for running running experiments with the LLMs are not provided due to organizational policies.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Sufficient setup information to understand the results is provided in the main text, with full details and exact prompts included in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars are included in all plots and represent standard error of the mean. *t*-tests are used to verify statistical significance for all key results. See Sections 3.2 and 4.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We use the public APIs for all models studied, and information on pricing for these are available on the corresponding public websites.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: Our research conforms to the NeurIPS Code of Ethics in all respects.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This work primarily benchmarks and analyzes the performance of existing models, rather than introducing new models or methods. While our finding that summarization enables meta-learning in models could be used to improve the autonomy of LLM agents, the focus is on characterization of different models and behaviors given our specific settings and benchmarks. For these reasons, we expect the primary impacts of our work to be an improved understanding of the existing capabilities of LLM agents and we do not anticipate any negative societal impacts for the research in this paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original paper that produced the Alchemy environment, which is licensed under Apache License 2.0.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.