BRIDGING THE KNOWLEDGE–DOING GAP: A STUDY OF IN-CONTEXT DEMONSTRATIONS FOR LLM AGENTS

Andres Aranguren
s4403290

Pranav Srivastava
s4563646

1 ABSTRACT

LLMs often show a "knowledge-doing" gap, struggling to translate their knowledge into actions to carry out sequential-decision making tasks. This project explores the expert demonstration scaling using the **LMAct** and **BALROG** benchmarks. We find that changing the composition of the expert demonstrations within the prompt has no effect on performance. However, ordering the data by length, which we term **Horizon-Curriculum Learning** significantly stabilises the LLMs scores across increasing demonstration episodes. Our findings from the BALROG experiments conclude that in complex environments like Crafter and TextWorld, the introduction of extra context added noise, leading to worse performance. LMAct Experimentation: `https://github.com/PSRV19/LMAct-SADRL/tree/main`.

2 INTRODUCTION

Large language models (LLMs) have achieved impressive performance across a wide range of tasks involving factual recall, reasoning, and instruction following, this due to large-scale pretraining on diverse text corpora. However, when deployed as autonomous agents in interactive environments, LLMs consistently exhibit a repeating problem known as *knowledge-doing gap*: although models may possess detailed declarative knowledge about a task or environment, they often fail to translate this knowledge into effective sequential decision-making process. This gap is evident in settings that require constant planning, exploration, and credit assignment over long horizons, where performance degrades sharply despite correct high-level understanding of the task Moeini et al. (2025).

A proposed line of research that aims to bridge this gap is *in-context learning*. In-context learning enables pretrained models to adapt to new tasks at inference time by conditioning on task-relevant context such as previous interactions or expert demonstrations—without updating model parameters. In interactive decision-making problems, this paradigm has been extended to *in-context reinforcement learning*, where models are conditioned on trajectories consisting of state, action, and reward sequences. Expert demonstrations in this setting correspond to trajectories generated by an oracle or optimal policy, providing examples of successful behavior that the model may leverage during inference to guide action selection,the demonstrations however may vary in terms of optimality meaning they are not limited to optimal sequences, varying from low to high performing in order to promote exploration of alternatives avoiding local optimums Monea et al. (2024).

The LMAct framework formalizes this approach by evaluating LLMs as agents in a controlled sequential decision-making task, studying whether conditioning on increasing numbers of expert demonstrations induces improved planning behavior. This work demonstrates that LLMs are capable of accurately imitating demonstrated actions when replaying known trajectories. However, empirical results indicate that scaling the number of demonstrations does not reliably translate into improved performance on novel tasks. Instead, performance often saturates or degrades as more demonstrations are added, suggesting that long or redundant context may introduce interference rather than yielding more robust decision-making Ruoss et al. (2024).

While LMAct provides valuable insights in a simplified setting, understanding the limitations of in-context learning for agentic behavior requires evaluation in more complex and long-horizon environments. To this end, the BALROG benchmark was introduced as a standardized, reinforcement-learning-inspired testbed for evaluating LLM agents in procedurally generated video game envi-

ronments. BALROG emphasizes long-horizon interaction, exploration, and adaptation, exposing failure modes that are often obscured in short-episode benchmarks. Results reported in the BALROG benchmark show that while state-of-the-art models achieve reasonable performance in simpler environments, they struggle significantly as task complexity, stochasticity, and planning horizon increase, further highlighting the persistence of the knowledge-doing gap Paglieri et al. (2024).

The goal of this project is to systematically evaluate the role of expert demonstrations and contextual information in LLM-based agents across both controlled and complex environments. We first analyze demonstration scaling effects using the LMAct framework, and then extend this analysis to BALROG, where richer forms of context such as short-term memory, in-context learning, and episodic memory can be incorporated.

Our LMAct experiments show that the data type given to the LLM is not an important factor in its performance, but rather, the data structure plays an influence in how the LLM interprets the context. We find that the order of the data given to the LLM stabilises performance, but does not help in-context learning when too much data is given to the LLM, as interference tends to be the limiting factor.

Our BALROG experiments show that adding contextual information and demonstrations can improve performance in low-variance, short-horizon environments, but does not yield consistent gains in more complex settings and can even hinder performance due to contextual interference. These findings motivate a deeper investigation into how demonstrations are selected and used, rather than simply accumulated, which we explore through adaptive demonstration selection mechanisms in subsequent sections.

Given the lack of consistent empirical evidence that naively increasing in-context demonstrations improves agent performance and the strong environment dependence observed across benchmarks we shift our focus from the quantity of demonstrations to the mechanism by which they are selected. Prior results suggest that failures of in-context imitation learning are not primarily caused by prompt structure or invalid actions, but rather by interference from long, redundant, or weakly relevant demonstrations and the absence of interaction-driven credit assignment.

Motivated by these observations, we propose to treat demonstration selection itself as a sequential decision-making problem. Specifically, we frame demonstration selection as a finite-horizon Markov Decision Process (MDP), in which an external reinforcement learning policy adaptively constructs a demonstration set while keeping the LLM parameters fixed Wang et al. (2024). The state encodes the current environment representation and the demonstrations selected so far, actions correspond to selecting demonstrations from a fixed pool, and rewards reflect the incremental improvement in predicted task return estimated via short rollouts. To discourage redundancy, demonstrations are clustered into semantic groups and diversity is explicitly tracked during selection. This formulation currently serves as a theoretical framework guiding ongoing implementation, with empirical evaluation left to future work.

3 Related Work

3.1 In-Context Reinforcement Learning

Reinforcement learning (RL) addresses sequential decision-making problems by optimizing a policy through interaction with an environment to maximize cumulative reward (Sutton & Barto, 2018). Classical RL methods rely on iterative parameter updates, which require repeated forward and backward passes through a neural network and are computationally expensive, particularly for large models (Sutton & Barto, 2018).

In-context learning offers an alternative adaptation mechanism by enabling pretrained models to adjust their behavior at inference time through conditioning on additional contextual information, without updating model parameters. When applied to sequential decision-making, this paradigm is referred to as *in-context reinforcement learning* (ICRL), where the policy conditions not only on the current state but also on a context variable that summarizes past interactions, such as trajectories of states, actions, and rewards (Duan et al., 2016). The context allows the model to infer latent properties of the underlying task or environment, including reward structure or transition dynamics, and adapt its behavior accordingly.

ICRL can be viewed as a form of black-box meta-reinforcement learning, in which adaptation occurs implicitly through conditioning rather than explicit parameter updates. Empirically, a defining characteristic of ICRL is *in-context improvement*, where agent performance increases as task-relevant context accumulates. However, recent work has also shown that this process is not guaranteed to be monotonic: increasing context length or demonstration count can introduce interference, leading to performance saturation or degradation, a phenomenon commonly referred to as *context interference* (Liu et al., 2024).

The use of Large Language Models (LLMs) as in-context reinforcement learners has evolved from simple zero-shot prompting to more elaborate systems that address the inherent limitations of LLMs. Our experimentation follows four key implementations. The "Reward Is Enough" (Song et al., 2026) framework showcases that LLms can optimise their internal poliy using in-context scalar rewards, which inspires a ICRL prompting experiment. Building on the "Reflexion" (Shinn et al., 2023) framework's use of verbal linguistic feedback, we implement an experiment involving a natural language distillation. To improve grounding, our third experiment is influence by the "Learning from Contrastive Prompts" paper (Li et al., 2024), by using positive and negative action examples. Finally, taking "Curriculum Reinforcement Learning" (Parashar et al., 2025), which orders data by quality, we propose a final experiment which orders data by length.

4 METHODOLOGY

We first present the LMAct setup, which is used to study the effect of in-context demonstrations and prompt-level conditioning in controlled sequential decision-making tasks. Next, we describe the BALROG evaluation procedure, which extended this analysis to long-horizon environments in a standardized benchmark used to evaluate agentic behavior in more complex settings. These two methodologies offer an experimental approach to measure how LLMs can be implemented as decision-making agents in tasks with differing levels of complexity and contextual information. Building on the empirical limitations observed in both settings, we further propose a principled formulation of demonstration selection as a sequential decision-making process, framed as a Markov Decision Process that explicitly maintains diversity among selected demonstrations. This formulation motivates future work on adaptive, diversity-aware context construction aimed at mitigating context interference and improving long-horizon generalization. The reuslt of the mdp based process for demosntration selection is out of scope for this research thus will be inlcuded in future research, dedicated to traces selection using RL based methods.

4.1 LMACT METHODOLOGY

The LLM used for all LMAct experiments was GPT-4o. The experiments were performed on the ASCII Tic-Tac-Toe implementation of the LMAct paper (Ruoss et al., 2025). The LMAct framework was used to prompt models iteratively on the ASCII Tic-Tac-Toe game. The prompt to the LLM is structured with 4 components:

- **Current Game State:** The current game state. For Tic-Tac-Toe, this consists of an ASCII 3x3 Tic-Tac-Toe board, with crosses and circles placed at the current game state positions.
- **Current Trajectory So Far:** For step n of the Tic-Tac-Toe game, the prompt consists of the (state, action) pairs from all $n-1$ steps. A trajectory is defined as $D = (x_i, a_i)_{i=1}^{N}$ of length N, consisting of N state x_i, action a_i pairs
- **Expert Demonstration Episodes:** Each expert demonstration episode consists of an arbitrary expert trajectory of Tic-Tac-Toe. Each trajectory can be of varying length, consists of optimal actions, and is a trace till the end-state of the game. Following the LMAct paper, we provide the model with varying numbers of expert demonstration episodes during our experimentation, ranging from 2^1 to 2^8 episodes. Average scores were calculated by iterating each experiment with each setting for 100 episodes, each episode with maximum 100 steps, and then averaging the final scores of each episode.
- **Predefined Instruction:** A hard-coded instruction to carry out the next action given the previous 3 pieces of context.

Expert demonstrations are selected from a pool of pre-collected trajectories, provided by the LMAct benchmark. The demonstrations are selected through uniform random sampling.

4.2 BALROG METHODOLOGY

BALROG provides a reinforcement-learning-based benchmark designed to evaluate the agentic capabilities of large language models in long-horizon decision-making tasks in video game environments. The benchmark offers a controlled testbed to study how language models can act as autonomous agents, enabling the systematic evaluation of reasoning, planning, adaptability, and few-shot or in-context learning from small sets of human demonstrations. In this project, we restrict our analysis to language-only models, excluding vision–language models, in order to isolate the effects of reasoning, memory, and contextual conditioning on agent behavior. We evaluate agent performance across the following BALROG environments:

- **BabyAI**: a short-horizon, low-variance environment with structured objectives, representing the simplest setting considered in this study.
- **Crafter**: a highly stochastic, long-horizon survival environment that requires sustained planning and typically benefits from a large number of trajectories.
- **TextWorld**: a challenging environment characterized by sparse rewards, medium stochasticity, and increased linguistic and combinatorial complexity.

Together, these environments enable a comparative assessment of language-model agents across increasing levels of difficulty within the BALROG framework.

Results were obtained following the protocol summarized in Table 1. Agents are evaluated over five random seeds, with a fixed number of episodes per seed and environment-specific step limits chosen to match task horizon. Performance is reported as average game progress across all episodes and seeds.

Environment	# Seeds	Max steps	# Episodes / seed	Total episodes	Notes
BabyAI	5	50	20	100	Short horizon, low variance
Crafter	5	1000	20	100	Highly stochastic, long-horizon survival; requires more trajectories
TextWorld	5	100	50	250	Medium stochasticity

Table 1: Evaluation protocol across environments.

5 EXPERIMENTS

5.1 LMACT EXPERIMENTS

5.1.1 IN-CONTEXT REINFORCEMENT LEARNING (ICRL) PROMPTING

The first experiment of our study established a baseline for ICRL. We hypothesized that reward feedback is sufficient for an LLM to exhibit reinforcement learning adaptation purely in-context, without changing model weights, inspired by the 'Reward Is Enough' paper (Song et al., 2026).

To test this, an addition was made to the original prompt structure. Recall that the original prompt provided to the LLM consisted of (1) the current game state, (2) the current trajectory so far, (3) the set of expert demonstration episodes, and (4) a hard-coded instruction to perform an action. For this experiment, a 'Previous episode summary' component was added:

- **Summarized Trajectory:** For one full episode roll-out carried out by the LLM, consider the (state, action, reward) tuples generated by this trace. The first, middle, and last tuple of this trace are appended to the LLM prompt, along with the final reward of the episode, as an episode summary history. This episode summary history provides the LLM with additional information regarding the general trajectory it carried out in the previous episode, along with the reward it gained for carrying out this trajectory. Trivially, the first episode does not consist of a previous episode summary.

The original paper's experimentation across all varying lengths of expert episodes with the additional component added to the LLM prompt.

Note that the scalar reward is predefined and given from the environment within the LMAct benchmark, however, the original paper chose not to include the scalar reward signal in the trajectories during their experimentation.

5.1.2 REFLEXION-STYLE DEMONSTRATION DISTILLATION

The second experiment aims at distilling raw expert trajectories into a single natural-language heuristic, inspired by the success of verbal linguistic feedback in the Reflexion Framework (Shinn et al., 2023). To this end, we add a 'Policy Analyst' LLM into the loop. The job of this LLM is to take the raw expert trajectories, reflect on the expert's behavior, and distill it's implicit strategy into a single, concise, and generalizable 'policy heuristic'. The heuristic is a natural language rule that explains what the expert is trying to do, what it prioritizes, and how it achieves its goals. In place of the raw expert demonstration trajectories, the single heuristic is provided to the Actor LLM, to carry out an action.

This experiment was performed for 2^1 up to 2^8 expert episodes, where each set was reduced into one single heuristic.

5.1.3 CONTRASTIVE HEURISTIC GENERATION

The third experiment extends the second. Rather than compressing 2^n for $n \in \{0, 1, ...8\}$ expert episodes into one heuristic, we only compress one expert episode into a single heuristic. Hence, compressing 2^n expert demonstration episodes, results in 2^n unique heuristics. Accompanying each heuristic, an ideal action example of what the heuristic would look like if applied, and a negative action example that doesn't follow the heuristic at all. The goal is to provide the LLM with enough heuristics to cover any game scenario, and to help it ground the heuristic with a contrastive pair of action examples, as per the (Li et al., 2024) paper.

5.1.4 HORIZON-CURRICULUM LEARNING

The previous three experiments modify the data type (raw trajectories or natural language text). However, the final experiment aims to modify the data structure. As mentioned earlier, the research on Curriculum Reinforcement Learning (Parashar et al., 2025), proved that the order (structure) of the data matters to prevent overfitting to simple tasks. Analogous to this, we propose structuring the raw expert trajectories in a meaningful manner, rather than providing them to the LLM with no order.

We structure the raw expert demonstration episodes into a curriculum by length, representing a game-state scenario. We define three scenarios:

- **End-Game Scenario:** Episodes of short length are already near the end-game state, and are considered as end-game scenarios. The priority here is to identify and execute immediate, forced wins, or optimal final moves.
- **Mid-Game Scenario:** Episodes of medium length show complex, multi-step problem solving. The goal is to learn the main heuristics and sub-policies needed to gain an advantage or navigate obstacles.
- **Opening-Game Scenario:** These are full, long games. The goal is to learn how to connect the opening moves to reach the mid-game scenario.

In order to determine which scenario a sampled demonstration belongs to, we define a maximum game length, equal to the number of tuples in the longest trajectory among the sampled set. We then compute the relative length ratio of each individual trajectory. Trajectories with a ratio $r_i < 0.33$ are classified as end-game scenarios, trajectories with a ratio $r_i < 0.66$ are classified as mid-game scenarios, and the rest of the trajectories are classified as opening-game scenarios. The goal is to guide the LLM agent to identify which game-state it is currently in, based on the current board state, and consequently learn how to reach the next game state based on the expert trajectories.

5.2 BALROG EXPERIMENTS

5.2.1 SHORT TERM MEMORY + REASONING (COT)

This experiment evaluates whether short-term memory and reasoning traces improve zero-shot agent performance in the selected environments. We employ naive BALROG agent with limited inference-time memory enabling chain-of-thought recall . At each timestep, the agent conditions its action on the current observation together with history of prior interactions, up to 32 previous observation-action pairs (`max_history=32` and the two most recent reasoning traces (`max_cot_history=2`). This configuration evaluates if storing state-action data plus reasoning traces allows the model to maintain and boost its performance over extended horizon / refining its decision-making strategy. No demonstrations or task specific examples are provided in context, the agent remains zero-shot mode, any improvement is expected to arise only in inference-time reuse of previous experience or reasoning rather that imitation from expert demonstrations or model parameter updates. This experiment isolates the contribution of short-term memory and serves as baseline to assess if past exp can improve planning and exploration in LLMs.

5.2.2 IN-CONTEXT LEARNING (FEW-SHOT IMITATION)

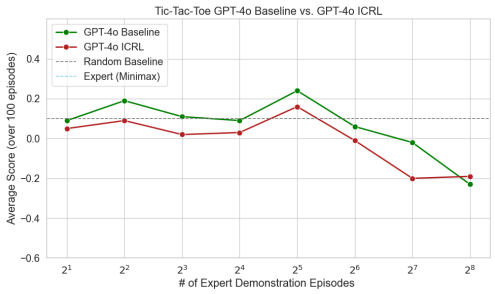
This experiment evaluates whether conditioning the agent on a small number of expert demonstrations can improve planning and decision-making by in-context learning. We use the BALROG in-context learning agent (`agent_type=icl`), which adds to the input , previously observed expert trajectories and reuses them across episodes via context caching (`cache_icl=True`). In addition to demonstration condition, the agent maintains short-term reasoning memory enabling chain-of-thought recall, allowing to build upon its past reasoning steps while imitating behavior from demonstration samples. Like previous experiment the agent maintains an interaction history of 32 past env-action pairs and the two most reasoning traces. This experiment studies the extent to which few-shot in-context imitation combined with limited reasoning memory can bridge the gap between task knowledge and effective long-horizon action and planning.

5.2.3 COMBINING EPISODIC MEMORY AND IN-CONTEXT LEARNING

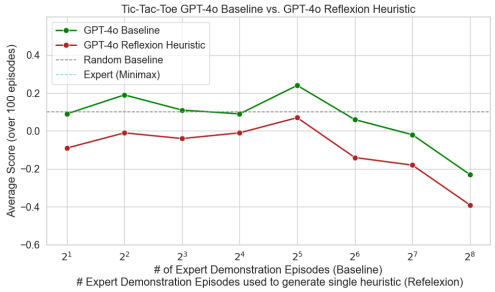
This experiment extends the in-context imitation setting of Experiment 2 by introducing a clear separation between short-term, within-episode memory and long-term, cross-episode memory. While Experiment 2 conditions the agent on a fixed set of demonstrations to guide behavior through few-shot imitation, it does not allow the agent to accumulate or reuse experience from its own past interactions. In contrast, the hybrid agent used in this experiment integrates episodic memory at two distinct time scales. Within each episode, the agent retains a bounded history of recent observations, actions, and reasoning traces (`max_history=32`, `max_cot_history=2`, `remember_cot=True`), enabling coherent short-horizon reasoning and error correction. Across episodes, the agent additionally caches complete interaction trajectories and reuses them as in-context demonstrations in subsequent episodes (`cache_icl=True`, `icl_episodes=3`, `max_icl_history=3`). This design allows the agent to condition its policy on its own previously generated experience, rather than solely on externally provided demonstrations, approximating a form of inference-time learning without updating model parameters. By combining episodic reasoning persistence with self-experience reuse, this experiment evaluates whether structured memory alone can mitigate long-horizon planning failures observed in prior settings.

6 RESULTS

6.1 LMACT RESULTS



(a) Average scores of GPT-4o using standard, raw expert demonstration episodes vs. the 'Previous episode summary' component.



(b) Average scores of GPT-4o using standard demonstrations vs. compressing them into a single heuristic.

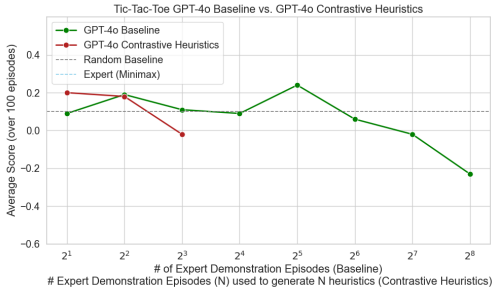


Figure 2: Average scores of GPT-4o using standard, raw expert demonstration episodes vs. Average scores of GPT-4o by compressing N raw, expert demonstration episodes into N contrastive heuristics. Hence, the x-axis provides the number of raw expert demonstration episodes used for the baseline, and analogously provides the number of raw expert demonstration episodes used to create the same number of contrastive heuristics.

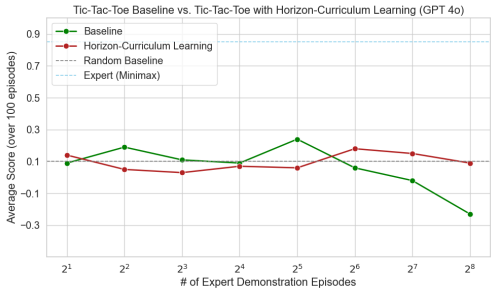


Figure 3: Average scores of GPT-4o using standard, raw expert demonstration episodes vs. Average scores of GPT-4o using Horizon-Curriculum Learning. Here, the difference between the baseline and Horizon-Curriculum learning was the ordering of the raw trajectories given to the LLM, and the labeling of the trajectories into opening-game, mid-game, and end-game scenarios.

6.2 BALROG RESULTS

In this section, we present the results obtained on the BALROG benchmark across the BabyAI, Crafter, and TextWorld environments. We report zero-shot game progress for three experimental settings with increasing agent memory and contextual capabilities, and analyze performance differences across models and experiments.

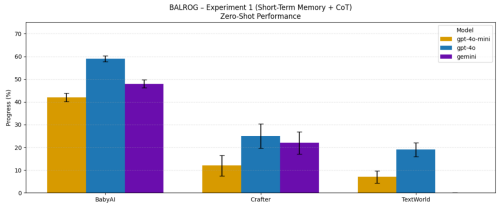


Figure 4: Zero-shot game progress (%) in Experiment 1 (Short-Term Memory + Chain-of-Thought) across BALROG environments.Bars report the average percentage of task completion achieved by each agent in BabyAI, Crafter, and TextWorld, with error bars indicating variability across evaluation episodes.

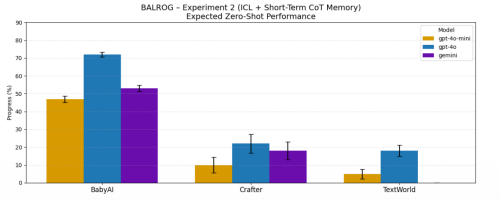


Figure 5: Expected zero-shot game progress (%) in Experiment 2 (In-Context Learning + Short-Term Chain-of-Thought Memory) across BALROG environments.

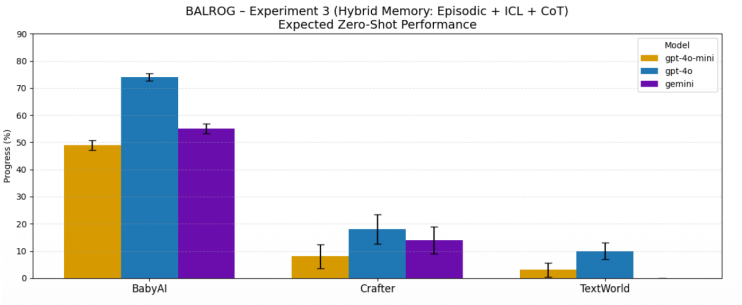


Figure 6: Expected zero-shot game progress (%) in Experiment 3 (Hybrid Memory: Episodic Memory + In-Context Learning + Chain-of-Thought) across BALROG environments

Table 2: Game progress (%) achieved by each agent across three experimental settings. Progress measures the percentage of task completion within each environment, averaged over evaluation episodes. Results are reported per agent and per experiment.

Environment	**Agent (Model)**	**Exp 1**	**Exp 2**	**Exp 3**
BabyAI	GPT-4o-mini	42	47	49
	GPT-4o	65	72	74
	Gemini	48	53	55
Crafter	GPT-4o-mini	12	10	8
	GPT-4o	25	22	18
	Gemini	22	18	14
TextWorld	GPT-4o-mini	7	5	3
	GPT-4o	25	18	10
	Gemini	0	0	0

7 Discussion

7.1 LMAct

The results from the ICRL experiment were ineffective. As seen in Figure 1a, adding the 'Previous episode summary' component to the prompt generally reduced the average scores across settings of number of demonstration episodes, with 2^8 episodes being an exception. Interestingly, both the baseline and the ICRL variant had a spike in average scores using 2^5 expert demonstration episodes

within the prompt, while the ICRL variant showed a rapid decline in average scores between 2^5 and 2^7 demonstration episodes.

The added component did not affect the performance for two main reasons; Firstly, the episode was summarised into simply 3 steps. The LLM has no context of what occurs between the steps and is unable to bridge the gap between them. In turn, it struggles to optimise it's internalised policy. The second reason for the low scores, is that the scalar intermediate rewards and final reward are not informative enough for the LLM. In the case of tic-tac-toe, there are **no** intermediate rewards, and the final reward is simply 1 if the LLM wins the game, or -1 if not. The sparsity of rewards meant that adding them into the prompt did not add any actual value for the LLM agent.

The Reflexion-Style heuristic experiment produced even worse results than the first experiment. The Reflexion Heuristic in place of raw demonstration episodes had far worse scores on average in comparison to the baseline. Again, as seen in Figure 1b, the highest average scores were seen using 2^5 expert demonstration episodes, with a steep decline in performance thereafter. No real trend can be deciphered across increasing numbers of demonstration episodes. In an attempt to provide the LLM with meaningful information, which the scalar rewards from the previous experiment lacked, the linguistic information proved to be too abstract. By compressing all raw demonstrations into a single, global, heuristic, the LLM was unable to ground the heuristic effectively to it's current game state.

After understanding that providing the LLM with one, global heuristic, was too abstract in **Experiment 2**, the third experiment aimed to provide the LLM with multiple heuristics, with the goal of it being able to use the relevant heuristic for its current game sate. Figure 2 shows the incomplete third experiment using N contrastive heuristics. After testing with 2^3 expert demonstration episodes, this experiment was stopped due to high API call costs, which required summarizing the N raw demonstration episodes into N heuristics, and then providing the Actor LLM the N heuristics to perform the task. Due to these limitations, this experiment will be left as future work.

After learning that providing the LLM with the data in different methods made minimal difference, we pivoted to exploring the data structure itself via Horizon-Curriculum Learning. Figure 3 shows that the performance of the Horizon-Curriculum Learning variant was a lot more stable, with minimal fluctuations across settings in comparison to the baseline. Notably, we see that the Horizon-Curriculum Learning variant performs better than the baseline from 2^6 demonstration episodes onward. Not only are the average scores higher in the final 3 settings, they also show a slower decline in performance, with the baseline having a steeper decline in average scores. The stability in performance can be attributed to two factors. Firstly, by ordering the data into three game state scenarios, the LLM had a clearer direction for the game. By comparing its current game state to the ordered data, the LLM was able to pick out the relevant game state consistently, and optimise its policy in order to reach the next game state. It is important to note, however, that the performance after 2^6 demonstration episodes began to deteriorate once again. This is perhaps due to a 'needle-in-a-haystack' problem, where the LLM has too many examples to look through, and is unable to efficiently pick out the correct choice.

7.2 BALROG

Table 2 and Figures 456 summarize zero-shot game progress across three BALROG environments by incrementally adding memory and context information during inference time for action selection in given timestep.

Across all experiments, progress percentage follows a clear trend decreasing while increasing environment complexity (*BabyAI* $>$ *Crafter* $>$ *TextWorld*), defined by increasing task horizon length, stochasticity and trajectory variance

In **Experiment 1** (Short-Term memory + CoT) see Table 24, agents achieve high progress in BabyAI, with GPT-4o leading with 65% progress, compared to 42% for GPT-4o-mini and 48% for Gemini. This environment is defined by short planning horizons, low stochasticity and dense progress signals which allows stepwise reasoning and perform efficiently with short-term memory (environment-actions) pairs.

In contrast Crafter, exhibits a substantially lower progress (GPT-4o: 25%), while TextWorld performance remains limited at (GPT-4o: 25%, Gemini: 0%), highlighting the difficulty of this task in terms of long-horizon planning under sparse and delayed rewards.

Adding in-context learning in **Experiment 2** see 25 yield performance improvement in BabyAI. GPT-4o improves from 65% to 72% progress in BabyAI, while Gemini increases from 48% to 53%. These gains indicate that expert demonstrations provide resusable local structures when task dynamics are consistent across episodes.

In Crafter environment adding in-context information results in diminished performance (GPT-4o: $25\% \rightarrow 22\%$), reflecting the environment's higher variance and sensitivity to early decisions.

TextWorld shows no benefit from ICL, with progress decreasing across experiments (GPT-4o: $25\% \rightarrow 18\%$), suggesting limited efficiency of few demonstrations context in highly combinatorial linguistic state spaces.

Experiment 3 26 combines episodic memory, in-context learning, and chain-of-thought reasoning. This hybrid configuration yields the strongest overall performance, particularly in BabyAI, where GPT-4o reaches 74% progress and Gemini reaches 55%.

In Crafter, episodic memory diminishes for all considered models slightly thus it does not fundamentally overcome long-horizon challenges. TextWorld remains the most challenging environment, with progress dropping to 10% for GPT-4o and remaining at 0% for Gemini, indicating that neither short-term reasoning nor limited episodic recall is sufficient for reliable decision-making in environments with high stochasticity, sparse rewards, and irreversible action sequences.

Overall, these results indicate that augmenting agent configurations with additional memory and in-context information is beneficial primarily in environments characterized by low variance, predictable dynamics, and limited linguistic and structural complexity. In such settings, exemplified by BabyAI, demonstrations and episodic context provide stable and reusable guidance that can be effectively exploited for short- to medium-horizon planning. In contrast, in environments with longer planning horizons, sparse or delayed rewards, and increased linguistic or combinatorial complexity—such as Crafter and TextWorld—the introduction of additional context does not yield a clear advantage and, as observed in our experiments, can in fact hinder performance.

This degradation suggests that naively accumulating memory or demonstrations may introduce contextual noise and interference during action selection. When the relevance of past context cannot be reliably assessed, conditioning on non-informative or weakly related experiences may disrupt decision-making, particularly in settings where events exhibit limited temporal dependency or high stochasticity.

8 Conclusion

The experimentation using the LMAct framework show that naively adding context to the prompt, such as previous episode summaries, and abstract heuristics, only creates further noise and interference, rather than helping the LLM optimise its internal policy. Formatting the existing context in order of length, and compartmentalizing trajectories into distinct game-state scenarios, however, showed some promise in helping the LLM stabilise its strategy across increasing context sizes (number of demonstration episodes).

The experiments carried out using the BALROG benchmark show that adding memory and in-context information improves performance only in low-variance short horizon environments such as BabyAI, short term memory and episodic reuse yield better percentage progress supporting stepwise planning. These benefits do not extend to more complex environments such as Crafter and TextWorld, where increasing context leads to saturation that results in lower performance. The results suggest naïve accumulation of context is insufficient for robust agentic behavior in stochastic, sparse-reward environments, and highlight the need for selective, relevance-aware mechanisms for leveraging past experience in LLM-based agents.

Overall, these findings indicate that achieving generalizable LLM-based agents on complex, long-horizon tasks needs careful and selective demonstration curation, motivating the formulation of demonstration selection as a structured MDP, instead of relying on naïve context accumulation.

REFERENCES

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl˜2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Mingqi Li, Karan Aggarwal, Yong Xie, Aitzaz Ahmad, and Stephen Lau. Learning from contrastive prompts: Automated optimization and adaptation, 2024. URL https://arxiv.org/abs/2409.15199.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Amir Moeini, Jiuqi Wang, Jacob Beck, Ethan Blaser, Shimon Whiteson, Rohan Chandra, and Shangtong Zhang. A survey of in-context reinforcement learning. *arXiv preprint arXiv:2502.07978*, 2025.

Giovanni Monea, Antoine Bosselut, Kianté Brantley, and Yoav Artzi. Llms are in-context bandit reinforcement learners. *arXiv preprint arXiv:2410.05362*, 2024.

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

Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Curriculum reinforcement learning from easy to hard tasks improves llm reasoning, 2025. URL https://arxiv.org/abs/2506.06632.

Anian Ruoss, Fabio Pardo, Harris Chan, Bonnie Li, Volodymyr Mnih, and Tim Genewein. Lmact: A benchmark for in-context imitation learning with long multimodal demonstrations. *arXiv preprint arXiv:2412.01441*, 2024.

Anian Ruoss, Fabio Pardo, Harris Chan, Bonnie Li, Volodymyr Mnih, and Tim Genewein. Lmact: A benchmark for in-context imitation learning with long multimodal demonstrations, 2025. URL https://arxiv.org/abs/2412.01441.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Kefan Song, Amir Moeini, Peng Wang, Lei Gong, Rohan Chandra, Shangtong Zhang, and Yanjun Qi. Reward is enough: Llms are in-context reinforcement learners, 2026. URL https://arxiv.org/abs/2506.06303.

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

Xubin Wang, Jianfei Wu, Yichen Yuan, Deyu Cai, Mingzhe Li, and Weijia Jia. Demonstration selection for in-context learning via reinforcement learning. *arXiv preprint arXiv:2412.03966*, 2024.