ABBEL: LLM Agents Acting through Belief Bottlenecks Expressed in Language

Aly Lidayan[†,*] **Jakob Bjorner**[*‡]**, Satvik Golechha**[*]**, Alane Suhr**[†]
dayan@berkeley.edu, jbjorner3@gatech.edu, zsatvik@gmail.com, suhr@berkeley.edu

Abstract

As the length of multi-step interactive language tasks increases, it becomes computationally impractical to keep full interaction histories in context. We propose a general and interpretable approach: Acting through Belief Bottlenecks Expressed in Language (ABBEL), which replaces long multi-step interaction history by a *belief state*, i.e., a natural language summary of what has been discovered about task-relevant unknowns. Under ABBEL, at each step the agent first updates the prior belief with the most recent observation from the environment, then uses only the updated posterior belief to select an action. We systematically evaluate frontier models under ABBEL across six diverse multi-step environments, finding that (1) ABBEL significantly reduces context lengths, enabling near-constant memory use over interaction steps, (2) the generated beliefs are interpretable, and (3) bottlenecks can reduce unnecessary reasoning. However, it is challenging to generate beliefs that are both concise and sufficient, and in some environments we observed inferior performance due to discarding valuable information or belief update errors. Motivated by this, we show that Reinforcement Learning is effective for improving the ability of LLM agents to generate and reason through belief bottlenecks. Training Qwen2.5-7B-Instruct under both ABBEL and full history settings, ABBEL quickly catches up with a 40% increase in performance while maintaining near-constant belief lengths over interaction steps.

1 Introduction

Tasks such as software development and scientific research can span hundreds or thousands of interaction turns, exceeding the practical context limits of even frontier models. These limitations necessitate the development of methods that can compress interaction histories while preserving the most relevant information for effective decision-making. While work on maintaining minimal sufficient statistics for sequential decision-making stretches back to Åström [1965], LLMs provide a unique opportunity for expressing such information in *language*, a medium that is both flexible and interpretable. The information in the interaction history required to solve a task can generally be captured by a posterior belief over the values of task-relevant variables. Rolling the history into such a belief state could, in principle, bottleneck the growing context length without harming performance. Furthermore, recent work suggests that LLMs can accurately update natural language descriptions of beliefs [Arumugam and Griffiths, 2025], and prompting language agents to explicitly generate a belief before acting can even enhance performance [Kim et al., 2025].

In light of this, we propose **ABBEL** (**Acting through Belief Bottlenecks Expressed in Language**), a framework for maintaining compact and interpretable contexts where an agent generates and acts on natural language belief states instead of full interaction histories (Figure 1). E.g., in word guessing

[*]These authors contributed equally to this work. [†]UC Berkeley EECS. [‡]Georgia Institute of Technology

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Bridging Language, Agent and World Models.

game *Wordle*[2], the full history of guesses and feedback is replaced by the current belief over the letters of the secret word. ABBEL alternates between updating a belief state given new observations, and selecting an action based solely on the current belief. Thus, ABBEL relies on the ability of the language agents to propagate the correct information at each step: they must maintain *sufficient* information for selecting good actions, while discarding superfluous information, e.g., repeated feedback that a letter is not in the secret word, to generate belief states that are *compact* enough to keep the context length manageable in long horizon settings.

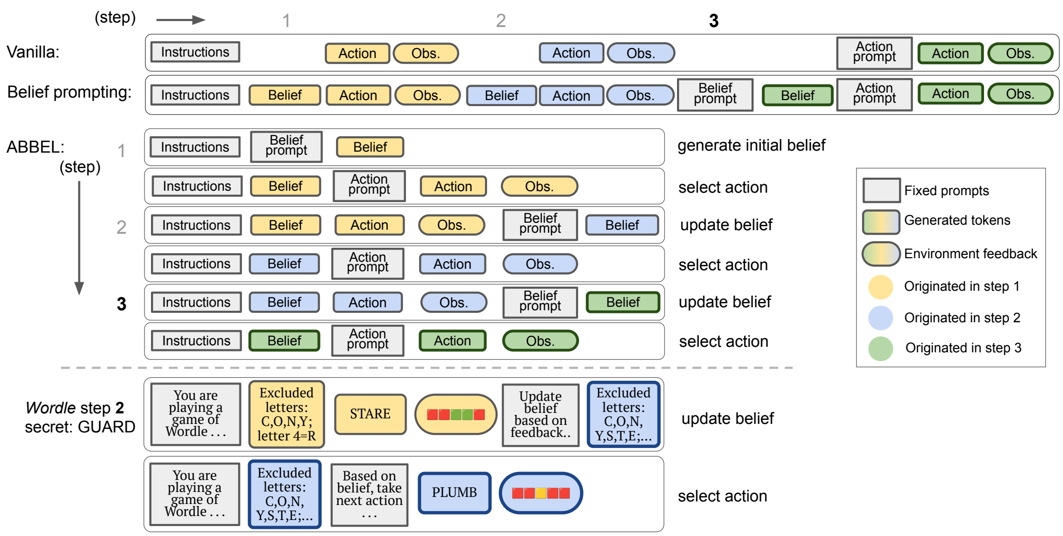


Figure 1: Top: Overview of the belief update and action selection contexts over three timesteps under ABBEL, in contrast to the typical multi-step paradigm (Vanilla) or simply prompting for belief generation (Belief prompting) which keep all past timesteps in context. Bottom: an example step of ABBEL in *Wordle*; actions are word guesses, and observations provide feedback on each letter of the guess; see Appendix A for the full trace.

We systematically evaluate current frontier models under ABBEL across six multi-step environments with varying levels of reasoning complexity and structure, comparing to ablations on both the belief generation and bottleneck components. We find that in many environments, the generated belief states are intelligible and significantly shorter than the full interaction history without significantly impacting performance, and conditioning on self-generated beliefs can also reduce unnecessary reasoning. While the history grows linearly with interaction steps, the belief lengths grow much more slowly, even decreasing in some environments as possibilities are ruled out. However, for each model we find environments where the reduced context decreases task performance, and identify key causes: propagating erroneous beliefs across steps, hallucinating false memories of previous steps, and repeating uninformative actions because the belief doesn't change without new information.

Considering the significant divergence between ABBEL and typical LLM pre-training settings, and the observed weaknesses from the frontier models, we propose to use RL to post-train LLM agents under ABBEL to better generate and reason through belief state bottlenecks. Training Qwen2.5-7B-Instruct in a simplified version of *Wordle*, we find ABBEL's performance quickly increases to match the success rate of the same model trained in the full context setting, demonstrating the efficacy of this approach.

2 Related Work

Multi-step interaction through beliefs. Various approaches have been proposed for maintaining compact representations of interaction history in multi-step exploration tasks. Hard-coded summary

[2]In *Wordle* the player has six tries to guess a five-letter secret word, receiving feedback about each letter (not in the secret, in the secret in a different position, or in the correct position) after every guess.

statistics have proven effective for bandit problems [Krishnamurthy et al., 2024b, Nie et al., 2025], but lack the flexibility needed for more complex environments. Arumugam and Griffiths [2025] show that frontier models can be effective at belief updating, but they hand-craft the initial prior beliefs for each environment, whereas in realistic settings such priors are often unavailable, and they select actions following a fixed heuristic.

Practical Approaches to Long-Context Management. Several recent systems have developed practical solutions for managing long contexts in interactive tasks. Context compression methods [Chevalier et al., 2023] generate dense vector representations that, while computationally efficient, sacrifice interpretability. IterativeAgent from PaperBench [Starace et al., 2025] simply prunes the first 30% of context when reaching limits while preserving the system prompt and initial user message, incentivizing high-level problem-focused planning at each step through ReACT-style prompting. Wang et al. [2025] address long-horizon tasks in SWE-bench and webshop environments using an LLM summarizer to condense context while maintaining task-relevant information, but this relies on a detailed hand-crafted prompt specifying what information should be maintained. Örwall [2024] focus on specialized tools for LLMs in SWE-bench, combining ReACT-style prompting with context pruning strategies. These approaches demonstrate the practical necessity of belief-like representations but have not been systematically evaluated for their ability to maintain sufficient information for optimal decision-making across diverse task structures.

RL for LLM Context Summarization Training LLMs for multi-step interaction with custom contexts instead of full histories is a developing area, and recent works like VeRL-Agent [Feng et al., 2025] and rLLM [Tan et al., 2025] take the first pass in implementing frameworks with such functionality. Recently MEM1 [Zhou et al., 2025] built on VeRL to post-train language models with multi-step RL to generate and act on context summaries. However, MEM1 combines the belief state with all reasoning for both belief updating and action selection in a single "internal state", but reasoning is generally extraneous information which does not allow for the interpretability and controllability of a true belief bottleneck.

See Appendix D for more related work.

3 Formulation

We model each environment as a Partially Observable Markov Decision Process, using *Wordle* as an example environment for grounding our formulation. In *Wordle* the objective is to guess a secret 5-letter word in under 7 turns by guessing a 5-letter word at each step. Each *task* corresponds to a randomly sampled hidden initial state s_0, e.g., `(secret:GUARD, step:0)`. At each step the agent selects an action a_t from the action space, e.g., 5-letter English words. The hidden state s_{t+1} is updated based on s_t and a_t, which in *Wordle* simply increments the step counter. The agent receives reward r_t and observation o_t both conditioned on a_t and s_t, e.g., $r_t = 1$ if $a_t = \texttt{GUARD}$ and $\texttt{step} < 7$ otherwise $r_t = 0$, and o_t is feedback on each letter in a_t (not present in the secret word, present at a different position, or present at the guessed position) and the new step count (see Fig. 1).

We model LLM agents as sampling actions from context-conditioned policies $a_t \sim \pi(\cdot|c_t)$. In the typical multi-step paradigm, the context includes the full interaction history of observations and actions $h_t = a_1 o_2 a_2 o_3 ... a_{t-1} o_{t-1}$, as shown in Fig. 1 (*Vanilla*), while in ABBEL it contains no steps before the current belief. The agent is called twice at each step t: first conditioned on the environment instructions p_i (e.g., how to play *Wordle*) and the last belief, action, and observation, and belief prompt p_b to generate new belief $b_t \sim \pi(\cdot|p_i b_{t-1} a_{t-1} o_{t-1} p_b)$ (*Update belief* in Fig. 1). Next, all steps before t are removed from the context, and π is called with action prompt p_a to select the next action $a_t \sim \pi(\cdot|p_i b_t p_a)$ (*Select action* in Fig. 1), resulting in a new observation o_t from the environment. See Appendix A for the full details. We measure the performance of π in each environment by its expected performance across the task distribution, e.g., the uniform distribution over all possible 5-letter secret words.

4 Evaluating Frontier Models With Belief Bottlenecks

We investigate to what extent current frontier models can already generate and reason through natural language belief states as bottlenecks in reasoning.

4.1 Environments

We evaluate across six multi-step environments from Tajwar et al. [2025] spanning various levels of reasoning complexity and structure. *Wordle* and *Mastermind* demand complex reasoning using highly structured feedback on each position of a secret word or code. *Twenty Questions* and *Guess My City* involve iteratively narrowing down a search space of topics or cities by asking a sequence of questions. In contrast, both actions and observations in *Murder Mystery* and *Customer Service* are free-form descriptive sentences: actions correspond to clue-gathering or troubleshooting instructions, and observations, generated by GPT-4o-mini, describe what the detective discovers or how the customer responds. Table 1 summarizes key characteristics of each environment; for more details see Tajwar et al. [2025].

Table 1: Characteristics of evaluation environments.

Environment	**Horizon**	**Complex Reasoning**	**Information Structure**	**Answer Space Size**
Murder Mystery	20	No	Low	3 (suspects)
Customer Service	20	No	Low	$\sim 10^2$ (faulty parts)
Twenty Questions	20	No	Medium	$\sim 10^3$ (e.g., animals)
Guess My City	20	No	Medium	$\sim 10^3$ (all cities)
Wordle	6	Yes	High	2315 (5-letter words)
Mastermind	12	Yes	High	10^4 (4-digit numbers)

4.2 Models and Frameworks

We evaluate Gemini-2.5-Pro, DeepSeek-R1, and DeepSeek-V3 with chain-of-thought prompting. For each model, we compare ABBEL with two frameworks. The first is a standard multi-step interaction framework (Fig. 1, *Vanilla*) where at each step the agent is prompted with the initial instructions followed by the full interaction history of actions and observations, and finally a prompt to generate the next action. The second framework (Fig. 1, *Belief prompting*) follows ABBEL in first prompting to update beliefs and then prompting to select an action given the beliefs at each step, but the full interaction history remains in context, ablating the information bottleneck aspect of ABBEL. We sample 40 tasks from each environment and report the mean and standard error.

4.3 Results

Belief State Compactness and Interpretability. We first investigate if ABBEL can reduce the context length for frontier models, by examining the compactness of belief states generated through ABBEL across different models and tasks, shown in Fig. 2. In most cases, beyond the first few steps the belief states were significantly shorter than the length of the interaction history (the gray lines).While the history always grows linearly with interaction steps, the belief lengths grew more slowly, plateauing or even decreasing in some environments as possibilities were ruled out, with the exception of Gemini 2.5 Pro in *Twenty Questions* and *Guess My City*. By inspection we found that all models generated intelligible natural language beliefs, which allowed us to better understand their behavior. For instance, in *Twenty Questions* we find that Gemini 2.5 Pro concatenates all information from the observations, explaining why the length grows linearly with time on par with the history, whereas DeepSeek R1 maintains a compact description of the posterior beliefs (see Appendix B for examples).

Task Performance. We next analyze how well frontier models perform under each framework. Fig. 3 presents the average success rates for each setting. We find that Gemini 2.5 Pro with ABBEL mostly maintains or even exceeds the performance of the full-context settings despite significant reductions in context length. However, the Deepseek models generally perform worse under all frameworks and show greater drops in performance under ABBEL, though Deepseek R1 achieves similar performance in Twenty Questions with significantly shorter beliefs compared to Gemini 2.5 Pro.

We then examine the performance of BELIEF PROMPTING to separately study the effects of prompting for belief generation and the belief state bottleneck. Firstly, we find that BELIEF PROMPTING rarely

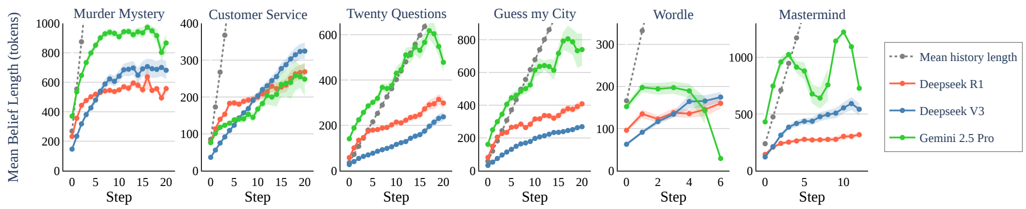


Figure 2: Average length of beliefs generated under ABBEL compared to full interaction histories. While history grows linearly over interaction steps, the belief lengths generally grow more slowly and are significantly shorter after the first few steps.

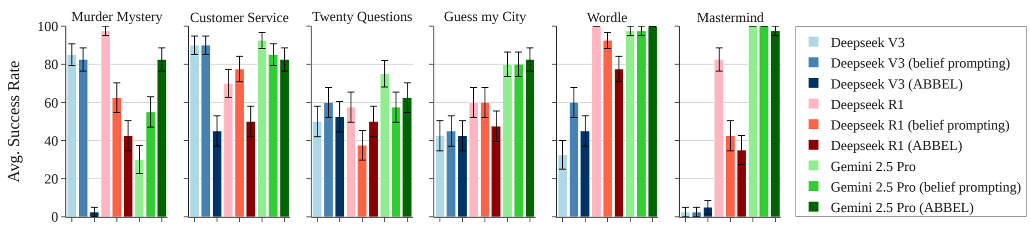


Figure 3: Performance of frontier models across six environments with no intervention, prompting for beliefs before acting (BELIEF PROMPTING), and ABBEL. Error bars indicate SEM. Gemini 2.5 Pro mostly maintained performance under ABBEL despite significantly reduced contexts.

outperforms the baselines and sometimes substantially decreases performance, suggesting limitations to prior findings that this is a helpful intervention in isolation [Kim et al., 2025]. Secondly, we investigate belief *sufficiency*, comparing ABBEL and BELIEF PROMPTING to control for the effect of belief generation. We observe that the weaker Deepseek models generally struggle more with generating sufficient beliefs in environments with low information structure (*Customer Service* and *Murder Mystery*), where it is more ambiguous what information should be maintained in the beliefs. Even Gemini-2.5-Pro fails to generate fully sufficient beliefs across all environments, as evidenced by the small performance drop in *Mastermind*.[3]

Impact on Reasoning. Finally, we investigate how ABBEL affects reasoning. Figure 4 shows the average length of reasoning used for action selection for DeepSeek-R1 and Gemini-2.5-Pro.[4] We find that conditioning on belief states generated by ABBEL and BELIEF PROMPTING rather than full histories significantly reduces reasoning length for comparable performance in several environments. This suggests that in multi-step environments, reasoning models may naturally integrate information from the interaction history as the first step of reasoning, and access to beliefs allows them to skip this part of the reasoning process. We also find ABBEL often uses even less reasoning than BELIEF PROMPTING while achieving similar success rates (e.g., Deepseek R1 in *Twenty Questions*, *Guess my City* and *Mastermind*). Inspecting the reasoning traces (see Appendix C.2 for examples), we find that R1 has a strong prior to ignore the belief state and reconstruct a posterior from the interaction history when available, so belief bottlenecks provide an additional benefit of preventing unnecessary extra reasoning over histories when beliefs are sufficient. For some environments, even the total length of both action and belief reasoning for R1 with ABBEL was less than the baseline, with no drop in success rate (see Fig. 6). Accounting for all tokens involved at each interaction step, including both input contexts and output reasoning, beliefs and actions, we find ABBEL uses fewer tokens and requires less memory in most environments once the interaction exceeds 5 steps (see figs 7 and 8).

We additionally inspect the traces to get further insight into the challenges of reasoning through a belief bottleneck. We find that performance of ABBEL is impacted when the agent does not

[3] Surprisingly, Gemini-2.5-Pro performs much better under ABBEL than the baseline for Murder Mystery. We find that in this game the model often continues gathering clues and runs out of steps before making an accusation, while an explicit belief state over the murderer's identity biases it to make accusations earlier.

[4] Only reasoning summaries were available for Gemini-2.5-Pro, which likely correlates with reasoning length.

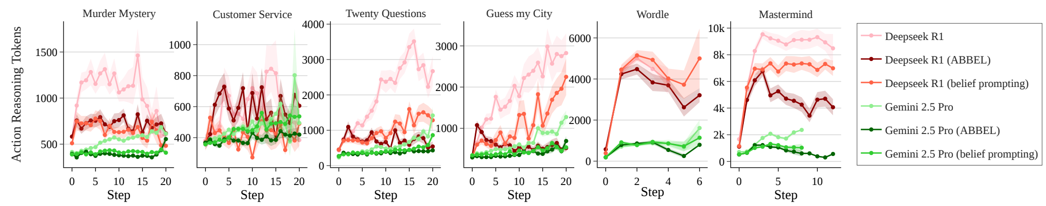


Figure 4: Reasoning trace lengths across steps (some models have no data at higher steps because all episodes ended early). Access to prior beliefs reduces reasoning in most environments, while ABBEL reduces reasoning even more than belief prompting alone.

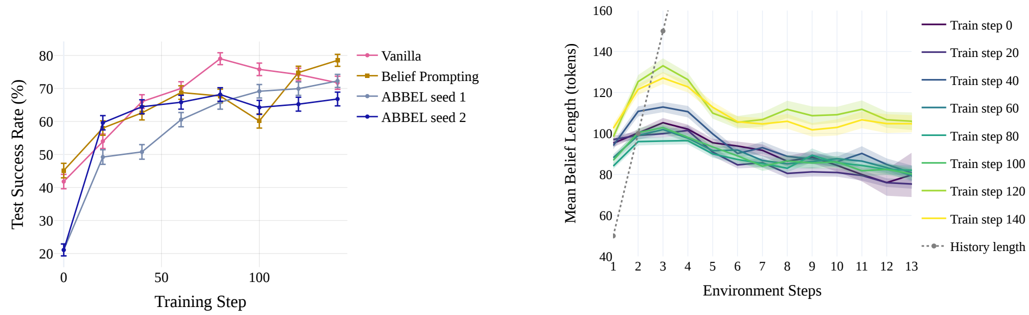
update the belief state after uninformative observations (e.g., in *Customer Service* when the customer responds “I’m not sure"), causing it to take the same action again, whereas if it can see previous actions it is much less likely to repeat an uninformative action. We also find many cases in the environments requiring more complex reasoning (*Wordle* and *Mastermind*) where belief state errors are introduced and propagated from one step to the next. In the latter case, models may self-correct their beliefs if they receive contradictory observations, but the true posterior and the wasted turns may be irrecoverable; whereas access to the full history enables earlier error detection and perfect posterior reconstruction. We find two main causes of belief state errors: incorrectly updating on the new observation due to mistakes in reasoning (e.g., falsely assuming that a character cannot be repeated in the secret), and hallucinating false memories of past interactions (see C.1 for an example).

The findings from this section collectively suggest that ABBEL can already lead to significantly shorter yet interpretable contexts for frontier models, belief generation is a natural first step of reasoning for multi-step interaction, and belief bottlenecks also have potential for improving reasoning efficiency. However, for each model we found environments where there was still significant room for improvement in either the task performance or the compactness of the generated beliefs.

5 Reinforcement Learning to Act through Belief Bottlenecks

Though we found in Section 4 that ABBEL’s significant context reduction can sometimes impact performance, there is a significant divergence between ABBEL and typical LLM pre-training settings. Thus, we propose the use of Reinforcement Learning (RL) to improve LLMs’ abilities to generate and reason through belief bottlenecks. RL does not require any task-specific knowledge about the content or format of the belief states: outcome-based rewards naturally incentivize maintaining and correctly updating information relevant for completing the task, while penalties on the length of the belief state can be used to encourage compactness. Further motivating our experimentation, RL has been shown to improve general abilities across task structures and input distribution shifts compared to SFT alone for general multi-step exploration [Nie et al., 2025, Kirk et al., 2024, Tajwar et al., 2025] in addition to calibrating models to their parametric knowledge [Eisenstein et al., 2025].

We train Qwen2.5-7B-Instruct with COT prompting on *Combination Lock*, a simple 3-digit version of *Wordle* proposed by Arumugam and Griffiths [2025]. We use outcome-based rewards only, training with GRPO in VeRL-agent [Feng et al., 2025], a multi-context synchronous rollout framework (for full details see Appendix E). In line with our findings from section 4.3, the initial performance of ABBEL is significantly lower than either the baseline (VANILLA) or BELIEF PROMPTING. We find that RL training is effective, leading to ABBEL’s success rate quickly increases to bridge this gap (see Fig. 5a). ABBEL learns to generate longer belief states over training, although they are still significantly shorter than the full history of actions and environment feedback past the first two steps (Fig. 5b). However, the cumulative regret of the learnt policies (Fig. 9) show that ABBEL still explores less efficiently than the other frameworks after RL. This may be due to phenomenon observed in Section 4.3 where mistakes in belief updating get propagated across steps, leading to inferior action choices. We leave it to future work to investigate sample-efficient training methods which both improve belief update accuracy and encourage compactness of generated belief states.



(a) Test success rate over training.

(b) ABBEL belief lengths over training.

Figure 5: Test performance of Qwen2.5-7B-Instruct trained in *Combination Lock* under each framework. (a) Success rates over training steps show ABBEL quickly reduces its performance gap with other frameworks. (b) ABBEL learns to generate longer beliefs over training, but they remain significantly shorter than the interaction history beyond the second environment step.

6 Discussion

We introduce ABBEL, a general framework for LLM agents to maintain manageable and interpretable contexts for long horizon interactive tasks via generating natural language beliefs. Evaluating frontier models in ABBEL across diverse multi-step environments, we find that they can maintain interpretable beliefs that are significantly shorter than full interaction histories without impacting performance, and the bottleneck can reduce unnecessary reasoning over full interaction histories. However, we find some environments where frontier models generate beliefs that are either as long as the history or cause inferior task performance under ABBEL. We identify key causes of performance drops, including belief update errors generating false beliefs that propagate across steps. This suggests that when used purely as an inference framework, ABBEL is most effective in settings that do not involve complex reasoning for belief updating. We then propose reinforcement learning in ABBEL as a general method for post-training LLM agents to generate and reason through beliefs more effectively. In *Combination Lock*, a task requiring significant reasoning, we show that RL quickly reduces the performance gap between ABBEL and models trained with full history access, with a 40% increase in success rate while belief lengths remain near-constant over interaction steps.

Future Work. We observe ABBEL still explores less efficiently after RL; investigating methods for further improving performance while maintaining compact beliefs is an important next step. Another interesting future direction is studying methods for training models to generate even more compact beliefs (such as length penalties) while preserving interpretability. Evaluating ABBEL on more realistic tasks with much longer horizons, robustness to injection of redundant or distracting information, and the potential of ABBEL's interpretable belief states for better diagnosis of failure modes in LLM reasoning are also left for future work.

7 Acknowledgments

We would like to thank Stuart Russell, Kartik Goyal, Dilip Arumugam, Syrielle Montariol, and Cameron Allen for valuable discussions. This material is supported in part by the Center for Human-Compatible AI (CHAI) and an Ai2 Young Investigator Award.

References

Dilip Arumugam and Thomas L. Griffiths. Toward efficient exploration by large language model agents, 2025. URL `https://arxiv.org/abs/2504.20997`.

Karl Johan Åström. Optimal control of markov processes with incomplete state information i. *Journal of mathematical analysis and applications*, 10:174–205, 1965.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts, 2023. URL https://arxiv.org/abs/2305.14788.

Jacob Eisenstein, Reza Aghajani, Adam Fisch, Dheeru Dua, Fantine Huot, Mirella Lapata, Vicky Zayats, and Jonathan Berant. Don't lie to your friends: Learning what you know from collaborative self-play, 2025. URL https://arxiv.org/abs/2503.14481.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025.

Jeonghye Kim, Sojeong Rhee, Minbeom Kim, Dohyung Kim, Sangmook Lee, Youngchul Sung, and Kyomin Jung. Reflact: World-grounded decision making in llm agents via goal-state reflection. *arXiv preprint arXiv:2505.15182*, 2025.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity, 2024. URL https://arxiv.org/abs/2310.06452.

Akshay Krishnamurthy, Keegan Harris, Dylan J Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context? *Advances in Neural Information Processing Systems*, 37:120124–120158, 2024a.

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

Allen Nie, Yi Su, Bo Chang, Jonathan N. Lee, Ed H. Chi, Quoc V. Le, and Minmin Chen. Evolve: Evaluating and optimizing llms for in-context exploration, 2025. URL https://arxiv.org/abs/2410.06238.

Thomas Schmied, Jörg Bornschein, Jordi Grau-Moya, Markus Wulfmeier, and Razvan Pascanu. Llms are greedy agents: Effects of rl fine-tuning on decision-making abilities, 2025. URL https://arxiv.org/abs/2504.16078.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. Paperbench: Evaluating ai's ability to replicate ai research, 2025. URL https://arxiv.org/abs/2504.01848.

Fahim Tajwar, Yiding Jiang, Abitha Thankaraj, Sumaita Sadia Rahman, J Zico Kolter, Jeff Schneider, and Ruslan Salakhutdinov. Training a generally curious agent, 2025. URL https://arxiv.org/abs/2502.17543.

Sijun Tan, Michael Luo, Colin Cai, Tarun Venkat, Kyle Montgomery, Aaron Hao, Tianhao Wu, Arnav Balyan, Manan Roongta, Chenguang Wang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. rllm: A framework for post-training language agents. https://pretty-radio-b75.notion.site/rLLM-A-Framework-for-Post-Training-Language-Agents-21b81902c146819db63cd98a54ba5f31, 2025. Notion Blog.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. Openhands: An open platform for AI software developers as generalist agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=OJd3ayDDoF.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents, 2025. URL https://arxiv.org/abs/2506.15841.

Albert Örwall. Moatless Tools, June 2024. URL https://github.com/aorwall/moatless-tools.

A Belief Bottleneck Rollout

A.1 Rollout Algorithm

Algorithm 1 Belief Bottleneck Rollout

Require: Instructions p_i, horizon $H \in \mathbb{N}$, step function $T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S} \times \mathcal{O}$, initial state s_0
Require: Belief generation prompt p_b, action selection prompt p_a, policy π.
  $t \leftarrow 0$
  $s \leftarrow s_0$
  $b \leftarrow$ "This is the start of the game. No beliefs right now."
  while $t \leq H$ **do**
    $a \leftarrow \pi(p_i b p_a)$ ▷ Action selection
    $s, o \leftarrow T(s, a)$ ▷ Environment step: get new state and observation.
    $t \leftarrow t + 1$
    $b \leftarrow \pi(p_i b a o p_b)$ ▷ Belief update
  end while

A.2 Sample ABBEL Step in Wordle

Here, we show one sample and associated prompts for the game *Wordle* with Deepseek R1 (illustrated in Fig. 1).

Game State

Secret: GUARD
Step: 2 out of 6
Goal Reached: No

Instructions

You are playing a game of Wordle.
Format your response in the following way: <Think> Any step-by-step, short and concise thinking to strategically determine the next guess for the secret word </Think> <Answer> your guess of what the word should be </Answer>
The game begins now, please make your first guess about the secret five-letter word!

Current Belief State

Excluded letters: C, O, N, Y
Included letters: R (must be in position 1, 3, 4, or 5)
Target word contains R and excludes C, O, N, Y.

Action Prompt

This is the game information:
`<game_instructions>`
You are currently taking your attempt `<step>` out of `<horizon>` attempts. Look at the current belief state and history and give an answer based on it.
Give an answer that leads to optimal exploration and do not be greedy unless it is the last attempt. Try to maximize the amount of information you have so that you can solve the task correctly.
Belief: `<curr_belief>`
Please format your response as: <Answer>your answer in the correct format mentioned </Answer>

Action and Observation

Action: STARE
Observation:
First letter, s, is not in the target word
Second letter, t, is not in the target word
Third letter, a, is correct and in the correct position in the target word
Fourth letter, r, is correct and in the correct position in the target word
Fifth letter, e, is not in the target word

Belief Update Prompt

This is the game information:
`<game_instructions>`
Look at the current belief and the agent's action and environment response on that belief. You have to update the current belief based on the action and response, while maintaining important information about the game state needed to take optimal future actions.
Current belief: `<curr_belief>`
Agent's action: `<action>`
Environment's response: `<observation>`
Output the updated belief state inside <BELIEF> and </BELIEF> tags.
Understand that only the generated belief is fed to the agent to pick the next action, not the history, so be sure to include all necessary information.

Updated Belief State

Excluded letters: C, O, N, Y, S, T, E
Included letters: A (position 3), R (position 4)
Target word contains A and R, excludes the listed letters, and has A in position 3 and R in position 4.

B Generated Belief State Examples

We compare examples of the belief states generated by DeepSeek R1 and Gemini 2.5 Pro operating under ABBEL. Each are at step 15 of the *Twenty Questions* game, where the answer is "waistcocat".

DeepSeek R1

The garment is a standalone structured, strapless, sleeveless formal bridal piece, not layered over other garments (confirmed). Excludes outerwear (boleros, jackets), full gowns, corsets, bustiers, and undergarments. Focus narrows to standalone bridal separates with stiffened fabrics (taffeta, satin) or layered construction for structural elegance, lacking boning. Key possibilities: embellished strapless bridal tops (beaded/embroidered), corsage-inspired designs with tailored support, or reinforced bridal camisoles/decorative tulle overlays. Must function independently (paired with skirts) without waist-flared elements or basque terminology. Core attributes: bridal formality, structural integrity via tailoring or stiff fabrics, no corsetry associations.

Gemini 2.5 Pro

I am playing a game of 20 Questions to identify a piece of clothing. I have now used 15 out of 20 questions.

Here's what I know so far:

- **Fact 1:** The item is worn on the upper body.
- **Fact 2:** It is NOT typically worn as outerwear.

- **Fact 3:** It typically has a collar.
- **Fact 4:** It typically has buttons all the way down the front.
- **Fact 5:** It is typically worn by men.
- **Fact 6:** It is typically worn for formal or business occasions.
- **Fact 7:** It has short sleeves.
- **Fact 8:** It features decorative vertical pleats or embroidery on the front.
- **Fact 9:** It is NOT a Guayabera shirt.
- **Fact 10:** It is NOT considered a form of national dress for a specific country.
- **Fact 11:** It is specifically worn with a tuxedo.
- **Fact 12:** It is NOT called a "short-sleeved tuxedo shirt".
- **Fact 13:** Its decorative front does NOT feature ruffles.
- **Fact 14:** Its decorative front does NOT consist of a distinct, separate panel of fabric (a "bib").
- **Fact 15:** The decorative vertical pleats on the front are NOT known as "pin-tucks".

Based on the cumulative information, my confidence has been updated.

- **Analysis:** The "No" answer to my last question, "Are the decorative vertical pleats on the front known as 'pin-tucks'?" is a major setback. My leading hypothesis of a "pleated-front shirt" or "pin-tuck shirt" has been invalidated. This means that if the decorative elements are pleats, they are a very obscure type.
- **Re-evaluation:** I must now reconsider the other half of Fact 8: "…pleats **or** embroidery". Having exhausted the most common types of pleated fronts (ruffles, bib, pin-tucks), it is highly likely that the decorative elements are not pleats at all, but rather a form of embroidery. The object remains a very unusual item: a short-sleeved formal shirt worn with a tuxedo, featuring vertical decorations that are not ruffles, not a bib, and not pin-tucks.
- **Current Prime Candidate:** There is no longer a single prime candidate for the item's name. The identity is very mysterious. The immediate goal is to confirm the nature of the decoration. My new leading hypothesis is that the item is a shirt with vertical **embroidery** on the front.

My confidence in any specific name is now low. However, I am confident that the next step must be to pivot away from pleats and investigate the "embroidery" possibility directly.

C Sample Frontier Model Reasoning Traces

C.1 Hallucinating Past Observations During Belief Update Reasoning

We provide an example ABBEL belief update from Deepseek R1 in Mastermind, demonstrating how LLMs may hallucinate observations from previous steps when reasoning through a belief bottleneck in multi-step environments.

Game State

Secret: 4 5 1 8
Step: 5 out of 12
Goal Reached: No

Prior Belief

Exclusions: 0, 6, and 9 are not in the code.
Possible Codes:
- 4 5 [1/2/3] [7/8]
- 4 5 [7/8] [1/2/3]

Action and Observation

Action: 4 5 1 7
Observation: 3 exact matches, 0 partial matches

Deepseek R1 (ABBEL) Belief Update Reasoning

...but previous feedback said that the third guess (4 5 7 8) had a partial match...

`4568` was never guessed; this hallucinated previous feedback leads R1 to conclude that 7 was the partial match and must be in position 4, ruling out the true code `4518` from the posterior:

Updated Belief State

Exclusions: 0, 1, 6, 8, 9 are not in the code.
Possible Codes:
- 4 5 2 7
- 4 5 3 7

C.2 Belief State Reconstruction under Vanilla and BELIEF PROMPTING

We highlight how frontier models reconstruct previously stated information in the Vanilla and BELIEF PROMPTING settings.

We find that Deepseek R1 reconstructs a posterior belief from the interaction history as the first step of reasoning about the next action:

Deepseek R1 (vanilla) Action Selection Reasoning

Okay, let's see. So, the history so far has three guesses: `CLOUD`, `PULSE`, and `FAULT`. Let me recap the feedback from each.

- **From `CLOUD`:**
  `C, O, D` are not in the target. `L` (second letter) is present but wrong position. `U` (fourth letter) is present but wrong position.
- **From `PULSE`:**
  `P, S, E` are not in the target. `U` (second letter) and `L` (third letter) are present but wrong positions.
- **From `FAULT`:**
  `F, A` are not in the target. `U` (third), `L` (fourth), and `T` (fifth) are present but wrong positions.

Putting this together, the letters that are definitely in the target word are `L`, `U`, and `T`. Because in `FAULT`, even though `F` and `A` are excluded, `U, L, T` are confirmed. Additionally, from `CLOUD`, `L` and `U` are confirmed but not in positions 2 and 4 respectively. From `PULSE`, `U` and `L` are again confirmed but positions 2 and 3 are wrong.
...

We find that Deepseek R1 with belief prompting sometimes fails to leverage the explicit belief state in its context, and wastefully reconstructs a posterior belief from the interaction history:

Deepseek R1 (BELIEF PROMPTING) Action Selection Reasoning

Belief State
First letter is `S`. Excluded letters: `P, O, I, L, E`. Correct letters: `A` in position 3, `R` in position 4, `T` in position 5. Target word format: `S_ART`. Possible letters for position 2: `B, C, D, F, G, H, J, K, M, N, Q, U, V, W, X, Y, Z`.

Reasoning
Okay, so I'm trying to figure out the next guess for Wordle. Let's recap what we know so far. The first guess was `SPOIL`. The response told us that `S` is correct in the first position. The letters `P, O, I, L`, and `E` (from the fifth letter in `SPOIL` and the fifth in `STARE`) are excluded. Then the second guess was `STARE`, which confirmed that `S` is still correct in position 1. The letter `T` is present but not in position 2. Also, `A` is correct in position 3, `R` in position 4, and `E` is excluded. Now the target word structure is `S _ A R T`. Wait, no, wait. Wait, let me check again.
Wait, in `STARE`, the fifth letter was `E`, which is excluded. But the response for `STARE` said that the fifth letter `E` is not in the target word. So the target word structure is `S _ A R _`, since the fifth letter isn't `E`. Wait, hold on. No, the correct letters from `STARE`: third letter `A` (position 3), fourth letter `R` (position 4). The fifth letter `E` is excluded, so the fifth position can't be `E`. But the first letter is `S`. The second letter is `T` in `STARE`, but it's in the target word in a different position. So `T` exists but isn't in position 2. So the target word has `T` somewhere else. But the structure we have is `S _ A R _`, with the fifth letter not `E`, but from the first guess `SPOIL`, the fifth letter `L` is excluded. So fifth position can't be `L` or `E`. Wait, but what about the third guess? Wait, the user only provided two guesses: `SPOIL` and `STARE`. Then the belief state says the target word is `S _ A R T`. Wait, how did `T` get into the fifth position? Because in `STARE`, the fifth letter `E` was wrong, but the correct letters in `STARE` are `S` (position 1), `A` (3), `R` (4). Then `T` in position 2 is present but in a different spot. So `T` must be in position 5? Because the only other positions are 2 (which was `T` in `STARE` but it's not there), or 5. Because positions 1, 3, 4 are already `S, A, R`. So `T` must be in position 5. That makes the target word `S _ A R T`. So fifth letter is `T`. So structure is `S _ A R T`.
...

Finally, we find that Deepseek R1 under ABBEL reasons directly with the belief state:

Deepseek R1 (ABBEL) Action Selection Reasoning

Belief State
Excluded letters: S, L, T, R, C, D, H, V
Confirmed present letters: A (position 2), E (position 4), N (position 5)
Possible letter positions:
1: [B, F, G, I, J, K, M, O, P, Q, U, W, X, Y, Z]
2: [A]
3: [B, F, G, I, J, K, M, O, P, Q, U, W, X, Y, Z]
4: [E]
5: [N]

Reasoning Okay, let's see. I need to figure out the best word to guess for the third attempt in Wordle. The excluded letters are S, L, T, R, C, D. So I can't use any of those. The confirmed letters are A in position 2 and E in position 4. So the word must be `_ A _ E _`.
My first thought is to maximize the information gain. Since this is the third attempt and there are three more left, I shouldn't be too greedy yet. I need to test possible letters in the unknown positions (1, 3, 5) while using letters that aren't excluded.
Looking at the possible letters for positions 1, 3, and 5: they include B, F, G, H, I, J, K, M, N, O, P, Q, U, V, W, X, Y, Z. I should choose a word that uses as many of these as possible in the unknown slots to check which ones are present.
...

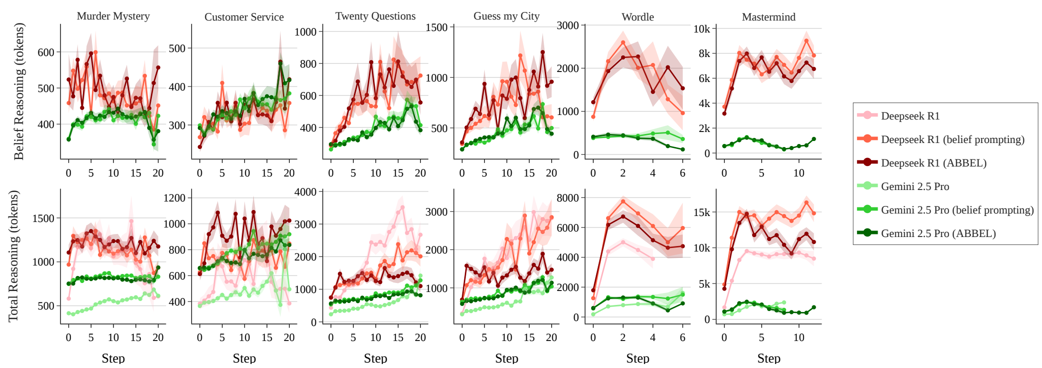


Figure 6: Reasoning trace length for belief generation (top) and the total reasoning length at each step, summing the belief and action selection reasoning lengths (bottom).

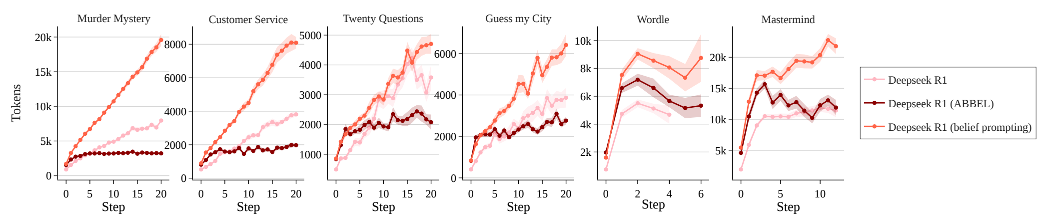


Figure 7: The total number of tokens involved at each step, including both input (i.e., the context) and output (i.e., reasoning, actions and belief states).

D More Related Work

Zhou et al. [2025] perform RL on LLM agents that combine belief update and action selection reasoning in an “internal state” representation, and demonstrate horizon generalization after training. However, the combined internal state does not allow the user to specify limits for belief length or encourage shorter beliefs without impacting the ability for sophisticated action reasoning, nor is it possible to fully isolate the information used for action selection to diagnose suboptimal behavior.

Recent work has identified specific failure modes when deploying language models for exploration tasks. Prior work on diagnosing exploration suboptimality in LLMs [Schmied et al., 2025, Krishnamurthy et al., 2024a] does not examine the crucial ability of LLMs to integrate knowledge from environment feedback- apparent suboptimality in action selection may be optimal given incorrect or incomplete beliefs. Schmied et al. [2025] diagnose three primary failure modes for non-reasoning pre-trained LLMs in bandit problems: greediness, frequency bias (copying the most frequent action in context regardless of reward), and the knowing-doing gap (describing correct behavior but failing to execute it). However, these diagnostics assume the LLMs are operating from perfect beliefs, missing errors in integrating observations into posterior beliefs.

E RL details

E.1 Environment Details

Combination Lock has the same feedback dynamics as *Wordle* with 3-character codes and guesses, while additionally enforcing that all three characters of the secret code and of every guess must be unique. Unique secret codes of 3 vocabulary characters were sampled, with a larger disjoint vocabulary and increased horizon at test time (see Table 2).

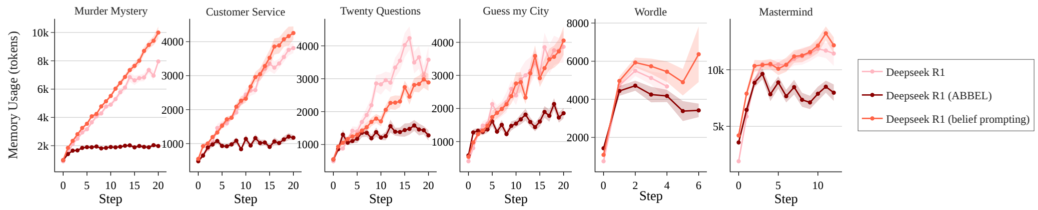


Figure 8: The memory usage at each step, defined as max(input + output tokens for belief updating, input + output tokens for action selection), representing the inference-time memory requirement. After the first few steps, ABBEL uses significantly less memory than the other frameworks.

Table 2: Characteristics of the *Combination Lock* environments.

Setting	Horizon (H)	Vocabulary	Answer Space Size
Train	12	012345689	720 (3 unique digits)
Test	16	qawsedrftgyhujik	3360 (3 unique letters)

We prompted Qwen2.5-7B-Instruct to first think step by step between <think>...</think> tags, and then generate actions or beliefs between <action></action> or <belief>...</belief> tags. Invalid generations did not count as an environment step, i.e. did not impact regret, but we limited the number of generation calls per game to H (VANILLA) or $2H$ (ABBEL and BELIEF PROMPTING); see Table 3 for details. Each trajectory ends in success once the secret code is guessed, or failure if either the generation limit or environment horizon is exceeded, with reward defined as follows to encourage succeeding with as few guesses as possible:

$$\mathcal{R} = \begin{cases} (H + 1 - \text{environment steps taken})/H & \text{if trajectory successful} \\ -1 & \text{otherwise.} \end{cases} \tag{1}$$

We leave belief state length penalties to encourage compactness for future work.

E.2 Training details

See Table 4 for the training settings and hyper parameters used.

Table 3: Handling of invalid generations in *Combination Lock*.

Case	Description	Outcome

Valid action	The action generation is correctly formatted as `<action>[c1, c2, c3]</action>` with three unique characters.	Both generation and environment steps are incremented, and feedback is presented in a newline separated list. e.g,:
		`8 is in Position 1!`
		`6 is not in Position 2, but is in the lock`
		`9 is not in the lock`

Invalid action	Most often errors take the form of `[action>...</action>` or repeated characters.	Generation step is incremented, and the model receives a message stating the action is invalid, reiterating the required format and prompting regeneration.
Invalid belief	Not using `<belief></belief>` tags. Errors tend to result from forgotten beginning/ending angle brackets or misspellings of `belief`.	Generation step is incremented, and the model receives a message stating the belief is invalid, reiterating the required format and prompting regeneration.

Table 4: Settings used in experiments. The mini batch at every gradient update step was set to the number of tensors present in the step to prevent off-policy updates, which have been shown to result in unstable training behavior with Qwen models.

Name	value
Optimization Algorithm	GRPO
AdamW learning rate	1e-7
batch_size	16
GRPO n rollouts	2
mini_batch	N/A
training_steps	140
num_epochs (calculated equivalent)	3.2
Learning rate decay	0.0
Gradient clipping	1.0

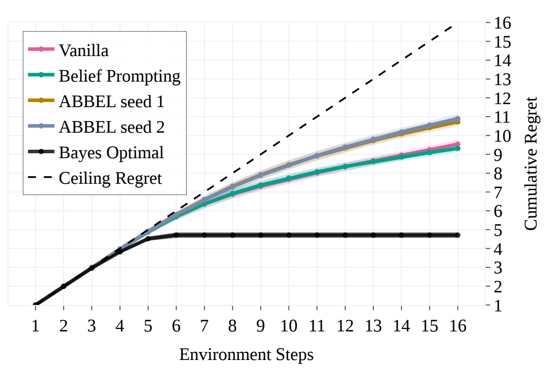


Figure 9: Cumulative regret curves show that ABBEL still takes more attempts to find the secret code after RL fine-tuning than models trained with the full history in context (VANILLA and BELIEF PROMPTING).