Measuring Belief Updates in Curious Agents

Joschka Strüber [*1 2] **Ilze Amanda Auzina** [*1 2] **Shashwat Goel** [3 4] **Susanne Keller** [1 2] **Jonas Geiping** [1 3 4]
Ameya Prabhu[† 1 2] **Matthias Bethge**[† 1 2]

Abstract

To effectively perform open-ended tasks, language models must identify gaps in their knowledge, take actions to acquire new information, and update their internal world models accordingly. This raises a key question, how can we assess whether their reasoning chains and multi-turn actions contribute to improving *beliefs* in their internal world model? In this paper, we demonstrate a simple, scalable method of *measuring belief updates* by sequentially assessing the log-probabilities that a language model assigns to the true belief across multi-turn actions. We assess model belief updates on a multi-turn RL reasoning benchmark, 'Twenty Questions'. Our findings show that recent Qwen3 models struggle to update their beliefs, even when the quality of generated questions is controlled for. Through counterfactual experiments, we validate that fine-tuning teaches the student models to perform coherent belief updates, which they could not do before. Intriguingly, we find that measuring model beliefs also allows detecting reward-hacking in RL-trained models. Overall, we offer a novel perspective on measuring and understanding intermediate beliefs of language models.

1. Introduction

Language Model (LM) agents are increasingly capable of solving complex reasoning tasks in real-world contexts incorporating tool-use, interactions and search (Song et al., 2025; Motwani et al., 2024). However, to effectively reason in open-ended scenarios, LM agents must be able to spot gaps in their knowledge, ask targeted questions and integrate the answers into their beliefs. These *curiosity-driven learning* capabilities are essential for step-by-step tasks with incomplete information. For example, when assisting with medical diagnosis, LM agents must ask maximally informative questions for which responses would progressively update their belief about the diagnosis. Beliefs at each step should then guide the next query and the final decisions.

We examine the sequential information seeking capabilities of LM agents using the classic deductive reasoning game—*Twenty Questions*. In this game, a secret word is uncovered through a trajectory of informative questions and, hence, progressive belief updates are crucial for integrating previously obtained information. Without accurate belief updates, even well-phrased questions will fail to lead the agent towards the final solution.

To assess whether models are able to integrate new information, we propose a simple, scalable metric for measuring *belief updates*. We sequentially compute the log-probability that the agent assigns to the true belief y at every step s_t, denoted by $\pi_\theta(y|s_t)$. Via this metric, we are able to answer the following questions:

(1) Are agents able to update their beliefs coherently?

(2) Does learning to ask better questions via training lead to improvement in belief updates?

(3) Does measuring beliefs help detect reward hacking despite task success?

Our findings show that recent Qwen3 models struggle to update their beliefs across different model sizes (Section 2.1), even when the quality of generated questions is controlled for (Section 3.3). This may explain why smaller Qwen3 models struggle in multi-turn conversations—their inability to update beliefs hinders effective knowledge acquisition. Finetuning on question trajectories from capable API models enable even small models at performing coherent belief updates (Section 3.1). Measuring model beliefs also allows us to detect and quantify reward-hacking in RL-trained models (Section 3.2). We verify that our proposed belief-update measure is not confounded by simply increasing context length leading to smaller log probabilities (Section 3.4). Overall, our work shows a novel way to understand LM agent capabilities in the context of information pursuit and updating world models.

[*]Equal contribution [†] Equal supervision [1]Tübingen AI Center [2]University of Tübingen [3]ELLIS Institute Tübingen [4]Max Planck Institute for Intelligent Systems. Correspondence to: Joschka Strüber <joschka.strueber@bethgelab.org>, Ilze Amanda Auzina <ilze.auzina@bethgelab.org>.

ICML 2025 Workshop on Assessing World Models.

Table 1. Pass@1 (% won) results with mean and standard deviation across 16 iterations. All finished games of the GRPO model were the result of reward hacking the judge model (*).

Model	Pass@1
Qwen3-1.7B	0.36% ± 0.31%
Qwen3-1.7B + SFT	8.63% ± 1.61%
Qwen3-1.7B + GRPO	*62.33% ± 1.65%
Qwen3-4B	3.11% ± 0.87%
Qwen3-4B + SFT	11.00% ± 1.12%
Qwen3-8B	3.87% ± 0.71%
Qwen3-14B	5.89% ± 0.77%
Gemini 2.0 Flash	15.80% ± 1.93%

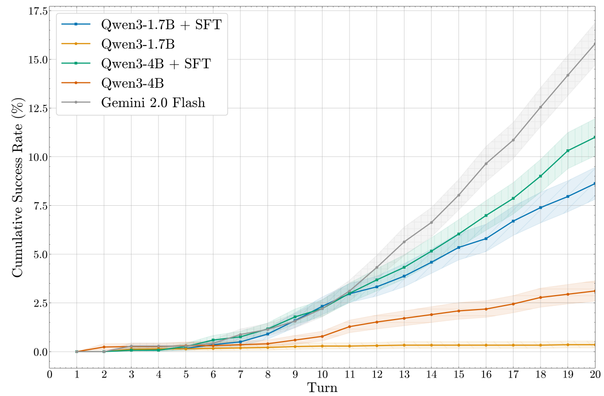

Figure 1. **Cumulative win rate across timesteps.** The number of questions needed for success, and overall success rate, improve after SFT on high-quality trajectories.

2. Methodology

Twenty Questions is a two-player deductive reasoning game. The game consists of 20 turns, where one player, the questioner, asks a closed question, and the other player, the responder, answers 'yes', 'no', or 'finished' given the secret word. In the context of multi-turn RL, the responder can be seen as the environment with which the agent interacts, and it also acts as a judge by deciding whether the final guess was successful (Abdulhai et al., 2023a).

Secret Words The secret words for our environment were taken from the Corpus of Contemporary American English (COCA) (Davies, 2008). We filtered a subset of the most frequent singular nouns (2,635). The final dataset consists of N=2,367 train and N=263 test secrets.

Models As questioners we compare `Qwen3` model family without thinking (Table 1)(Qwen Team, 2025). As judge we specify `Qwen2.5-14B-Instruct` with CoT based on preliminary experiments. Additionally, we evaluate also `Gemini 2.0 Flash`, a medium-sized API model (Gemini Team, 2024).

Supervised Fine-Tuning We further fine-tune `Qwen3` 1.7B and 4B models to investigate the influence of sft training on question generation and belief updates. The data was collected by rejection sampling from multiple larger capable models. Overall, the training set contains 18,033 game turns from 1,071 different games selected from the 718 most difficult words.

GRPO Finally, we apply GRPO training to the `Qwen3` 1.7B SFT model as work by (Swamy et al., 2025) has alluded that long horizon problems should benefit from RL training more than from SFT (Shao et al., 2024). The train dataset size (N=1000 secrets) and test (N=263 secrets), GRPO rollouts set to n=5.

2.1. Ensuring the Benchmark is Informative

First, we verify that the 20 Questions benchmark is a difficult task even for modern LM agents (Table 1). Our findings confirm that even for `Gemini 2.0 Flash`, a leading model in the AI landscape, the present benchmark poses a difficult challenge as only 15.8% trajectories are successful. `Qwen` models pass@1 rates are even lower, around 6% or less, and the performance further drops with decreasing model sizes. On the contrary, by simply SFT-ing the `Qwen` instruct models we are able to minimize the gap from the best performing model `Gemini` roughly by half for `Qwen3-1.7B` and three times for `Qwen3-4B`. Furthermore, SFT reduces the number of questions asked for a successful guess (Figure 8). Lastly, even though, the quantitative result of GRPO training lead to the highest success rate of 62.33%, qualitative examination the model's behavior revealed that the model had learned to trick the judge rather than to ask better questions (Table 1).

The presented success rates clearly indicate that the current state-of-the-art models struggle with multi-step reasoning, however, simply looking at success rates overlooks the complexity of the task: a model can ask perfectly reasonable questions, yet fail to finish the game in twenty questions. This behavior would not be captured by the success rate, therefore, in the following section we explicitly explore whether model's curiosity is guiding it towards the right answer, even when it fails.

3. Measuring Belief Updates

At each stage of a multi-turn trajectory or long CoT, we can consider a LM agent to be acting under uncertainty. The goal of each action is to bring the agent closer to the correct solution. Importantly, which action helps an agent the most towards achieving its goal depends on the agent's

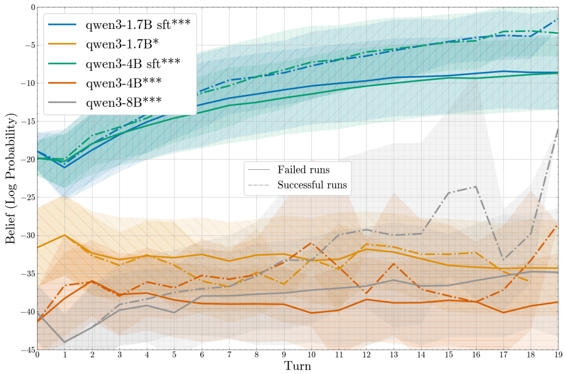

Figure 2. **Belief Updates by Outcome** Irrespective of outcome, SFT models progressively improve beliefs. Asterisks indicate the p-value of the mean of per-secret differences of log-probabilities between successful vs unsuccessful runs (permutation test). *: $p < 0.01$. ***: $p < 0.0001$.

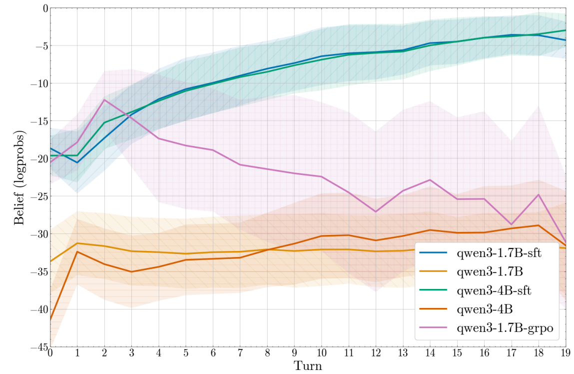

Figure 3. **Controlling for Question Generation** The models' ability to update beliefs when controlling for question quality (on Gemini's successful trajectories).

own understanding (Åström, 1965). At any given moment, different actions might be optimal for different agents. It is thus important to measure the agent's own beliefs to evaluate an action. How can we do this for Language Models?

We leverage a simple insight. Language Models are trained to provide probabilities over subsequent tokens. At any intermediate state, we can obtain their belief that a particular string is the final solution, simply by prompting them in a way that elicits their current guess at the answer. Specifically, to measure a model's belief for an answer string y given the prefix rollout s_t until turn t, we measure the log-probability assigned by the model to y with an elicitation prompt E.

$$\pi_\theta(y|s_t) = \log p_\theta\big(y \mid s_t + E\big), \tag{1}$$

For the setting of Twenty Questions, y is the secret word for that sample, s_t are all previously asked questions and responses, and E is "Is the secret word ", the prefix for making a final guess.

3.1. Long-Horizon Belief Updates

We first tested the hypothesis whether the improved results of the SFT models align with their own belief updates as measured by Equation 1. The results obtained (Figure 2) confirm the success rate scores (Table 1): for both SFT models the belief updates follow a clear upward trend - with each additional turn the model's belief in the correct answer is strengthened. On the contrary, Qwen3-1.7B and -4B models do not show any trend towards increasing belief updates neither for successful nor unsuccessful runs. For Qwen3-8B we can observe a small positive trend in the belief updates for the successful run, but with a large variation. In addition, both SFT models and the GRPO model with further RL training show much higher beliefs at the start of the game before the first question is asked than the base models. This indicates that SFT training enabled them to learn the distribution of possible solution candidates despite the split into train and test secrets.

Despite the visibly small differences between the log-probabilities of successful and failed trajectories, permutation test (Appendix A.1.4) indicates significant differences across all models, confirming that belief updates differ between true successful updates and failed runs. Nonetheless, despite the strong evidence observed, the SFT-to-base model difference may be confounded by the possibility that only the SFT models were capable of generating informative questions. We disentangle these factors through a counterfactual analysis described in the next section.

3.2. Counterfactual Analysis: Question Generation

To separate the model's ability to acquire information from updating beliefs, we select pre-generated successful trajectories, thereby removing the question generation as a possible confounder. In particular, we select the optimal paths (shortest successful trajectories) from `Gemini` generated paths for each secret word. By comparing the trends of Figure 2 and Figure 3 we can conclude that the observed belief-update capability difference between the SFT and base Qwen models also holds when both models are presented with exactly the same successful trajectories. This finding verifies the previously made observation that SFT training results in models that are able to integrate evidence over many turns.

On the other hand, smaller `Qwen` base models lack this capability even when the task is simplified by removing the question generation part. Lastly, by looking at the model belief-update trends, the 'hacking' model can be easily detected without qualitatively examining the model

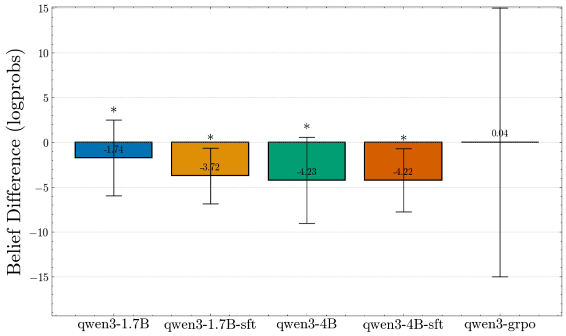

Figure 4. **Contradictory Evidence** Model capability to update beliefs when presented with contradictory evidence (”yes” and ”no” swapped) on previously successful trajectories.

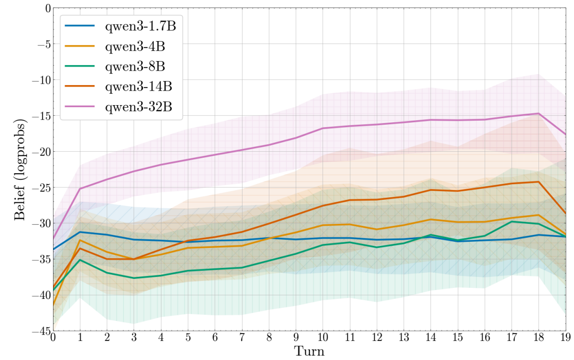

Figure 5. **Controlling for Model Size** Model ability to update beliefs across different model sizes when controlling for question generation (successful trajectories only).

generations. With each turn, the model’s beliefs become less certain, indicating that it is pushed away from the true answer.

This finding is further highlighted in Figure 4 by examining belief difference when presented with contradictory evidence on previously successful trajectories. In this experiment, we take the same successful game rollouts from `Gemini` as before and flip the judge’s response to the penultimate question. The figure shows the difference between the original belief and the belief after receiving a contradictory response that excludes the correct secret from the solution space.

For all other models, we see a negative shift in beliefs given the change, yet for the model that exploited reward hacking, the mean remains zero, indicating no difference. Surprisingly, the belief difference of the Qwen3-4B base model is very similar to the more capable SFT models. Although further tests are needed to fully draw this conclusion, this provides evidence that the base model is able to recognize whether the response to a given question should be “yes” or “no” in a binary fashion for the secret, it cannot accumulate the information it received when given a chain of questions and responses to update its belief.

3.3. Effect of Model Size on Belief Updates

Thus far the models discussed are of relatively small size, therefore, one might argue that the model’s are simply too small to have belief-updating capabilities unless specifically fine-tuned for this. To validate this claim, we extend our analysis by repeating the above experiment with Qwen3-14B and Qwen3-32B. As reported in Figure 5, model size correlates with model belief update trends. However, the absolute increase is still greater for the small SFT models, suggesting that belief-update capability is an unexplored model feature, at least in the context of `Qwen` models.

3.4. Beliefs On Failed Trajectories

Finally, for the belief-update measure to be a meaningful metric, it should be able to distinguish between successful and failed trajectories. To test this, we evaluate the belief updates of the 1.7B model after SFT on questions from the untuned 4B model. The data is further split into games that were won, games that were lost, but belong to a secret that was solved at least once, and games of secrets that `Qwen3-4B` was never able to guess. Although initially both won and lost only trajectories are similar, they exhibit different trends, confirming that using log-probabilities to assess model beliefs is a valid metric to understand model belief change (see Figure 6).

Recently, there have been multiple publications in the domain of LLM reasoning with findings that were specific to the `Qwen` model family. Any attempts to generalize them to other model families were unsuccessful. To show that this is not the case for belief updates, we repeat most experiments with `Llama-3.2` models in Appendix A.3, including inference results of the instruction tuned models with and without further training and counterfactual belief updates on successful Gemini games.

4. Related Work

There is a long history of evaluating machine learning models and their problem-solving capabilities on games such as checkers, chess or go (Samuel, 1959; Campbell et al., 2002; Silver et al., 2016). Although these early studies used agents specifically trained for the games they were evaluated on, recently the focus has shifted to games that evaluate the capabilities of general-purpose foundation models, such as LMs. For a recent benchmark for testing the agentic capabilities of LLMs and VLMs in games see Paglieri et al. (2025). The Twenty Questions game as a benchmark was introduced by De Bruyn et al. (2022a) and De Bruyn et al.

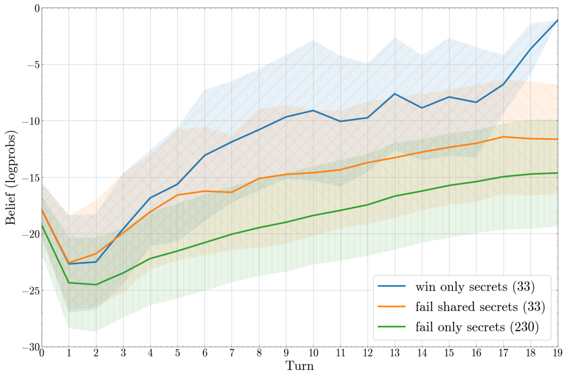

Figure 6. **Belief Updates on Failed Trajectories** Model beliefs of Qwen3-1.7-SFT on the correct answer update more slowly in failed trajectories from the untrained Qwen3-4B.

(2022b) where the goal was to evaluate the capability of off-the-shelf LMs to answer yes-no questions with different prompting techniques and SFT. This has been extended by Zhang et al. (2024b) to more capable LMs and PPO training. Abdulhai et al. (2023b) introduced the LMRL Gym to train LLMs with online and offline RL algorithms. Richardeau et al. (2024) learn to ask questions to fingerprint black-box LMs. Mazzaccara et al. (2024) show that the expected information gain can be used to choose better candidates for chosen-rejected pairs for DPO. Chen et al. (2024) proposed BrainKing, a modified version that introduces a limited amount of errors by the Answerer as deception. Mazzaccara et al. (2024) measure the Expected Information Gain of each question during training, but this depends on the set of secrets, quickly becomes computationally infeasible, and is challenging to generalize beyond Twenty Questions.

Multi-turn reasoning and its evaluation has been studied in many different contexts ((Patel et al., 2024), (Xie et al., 2024), (Lee & Hockenmaier, 2025)). Our method of beliefs as intermediate rewards can be applied to other benchmark settings that include a ground truth final outcome (e.g., (Patel et al., 2024)), without the need for human annotation of reasoning steps (Golovneva et al., 2022), setting-specific process reward models (Zhang et al., 2024a) or self-evaluation (Xie et al., 2024).

5. Conclusion

Our work offers a novel perspective on measuring and understanding how model internal beliefs play a role in information acquisition. As agents become more integrated in our daily life, the capability of updating internal beliefs is crucial in decision making processes. Our proposed *belief-update* measure showcases that recent Qwen3 models struggle to update their beliefs and this in turn affects how effective the model is in seeking information. Moreover, we show that finetuning LMs with rejection sampling is a viable solution, as the capability to update beliefs of finetuned models significantly improves. Furthermore, measuring model belief updates also allows to detect models with odd behaviors, such as, reward hacking.

Finally, the belief-update measure could be further explored in the context of RL training. In many long-horizon games sparse rewards pose a difficult optimization challenge. Utilizing agent's own beliefs is a simple, yet effective way how to solve this problem by allowing the agent to guide itself towards the correct option or conversely penalizing the agent if its actions do not lead to increase in beliefs.

References

Abdulhai, M., White, I., Snell, C., Sun, C., Hong, J., Zhai, Y., Xu, K., and Levine, S. Lmrl gym: Benchmarks for multi-turn reinforcement learning with language models. *arXiv preprint arXiv:2311.18232*, 2023a.

Abdulhai, M., White, I., Snell, C., Sun, C., Hong, J., Zhai, Y., Xu, K., and Levine, S. Lmrl gym: Benchmarks for multi-turn reinforcement learning with language models, 2023b. URL https://arxiv.org/abs/2311.18232.

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

Campbell, M., Hoane, A. J., and Hsu, F.-h. Deep blue. *Artif. Intell.*, 134(1–2):57–83, January 2002. ISSN 0004-3702. doi: 10.1016/S0004-3702(01)00129-1. URL https://doi.org/10.1016/S0004-3702(01)00129-1.

Chen, Y., Li, Y., Yan, S., Liu, S., Liang, J., and Xiao, Y. Do large language models have problem-solving capability under incomplete information scenarios? In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2225–2238, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.131. URL https://aclanthology.org/2024.findings-acl.131/.

Davies, M. The corpus of contemporary american english (coca), 2008. URL https://www.english-corpora.org/coca/.

De Bruyn, M., Lotfi, E., Buhmann, J., and Daelemans, W. Is it smaller than a tennis ball? language models play the game of twenty questions. In Bastings, J., Belinkov, Y., Elazar, Y., Hupkes, D., Saphra, N., and Wiegreffe, S. (eds.), *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 80–90, Abu Dhabi, United Arab Emirates (Hybrid), December 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.blackboxnlp-1.7. URL https://aclanthology.org/2022.blackboxnlp-1.7/.

De Bruyn, M., Lotfi, E., Buhmann, J., and Daelemans, W. 20Q: Overlap-free world knowledge benchmark for language models. In Bosselut, A., Chandu, K., Dhole, K., Gangal, V., Gehrmann, S., Jernite, Y., Novikova, J., and Perez-Beltrachini, L. (eds.), *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pp. 494–508, Abu Dhabi, United Arab Emirates (Hybrid), December 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.gem-1.46. URL https://aclanthology.org/2022.gem-1.46/.

Gemini Team. Introducing gemini 2.0: our new ai model for the agentic era, December 2024. URL https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/. Accessed: 2025-05-22.

Golovneva, O., Chen, M., Poff, S., Corredor, M., Zettlemoyer, L., Fazel-Zarandi, M., and Celikyilmaz, A. Roscoe: A suite of metrics for scoring step-by-step reasoning. *arXiv preprint arXiv:2212.07919*, 2022.

Grover, A., Gummadi, R., Lazaro-Gredilla, M., Schuurmans, D., and Ermon, S. Variational rejection sampling. In Storkey, A. and Perez-Cruz, F. (eds.), *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 823–832. PMLR, 09–11 Apr 2018. URL https://proceedings.mlr.press/v84/grover18a.html.

Lee, J. and Hockenmaier, J. Evaluating step-by-step reasoning traces: A survey. *arXiv preprint arXiv:2502.12289*, 2025.

Llama Team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Mazzaccara, D., Testoni, A., and Bernardi, R. Learning to ask informative questions: Enhancing LLMs with preference optimization and expected information gain. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 5064–5074, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.291. URL https://aclanthology.org/2024.findings-emnlp.291/.

Motwani, S. R., Smith, C., Das, R. J., Rafailov, R., Laptev, I., Torr, P. H., Pizzati, F., Clark, R., and de Witt, C. S. Malt: Improving reasoning with multi-agent llm training. *arXiv preprint arXiv:2412.01928*, 2024.

Paglieri, D., Cupiał, B., Coward, S., Piterbarg, U., Wolczyk, M., Khan, A., Pignatelli, E., Kuciński, Ł., Pinto, L., Fergus, R., Foerster, J. N., Parker-Holder, J., and Rocktäschel, T. BALROG: Benchmarking agentic LLM and VLM reasoning on games. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fp6t3F669F.

Patel, N., Kulkarni, M., Parmar, M., Budhiraja, A., Nakamura, M., Varshney, N., and Baral, C. Multi-LogiEval: Towards evaluating multi-step logical reasoning ability of large language models. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20856–20879, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1160. URL https://aclanthology.org/2024.emnlp-main.1160/.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Richardeau, G., Merrer, E., Penzo, C., and Tredan, G. The 20 questions game to distinguish large language models, 09 2024.

Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):211–229, 1959.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*, 2024. URL https://github.com/volcengine/verl.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Song, H., Jiang, J., Min, Y., Chen, J., Chen, Z., Zhao, W. X., Fang, L., and Wen, J.-R. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.

Swamy, G., Choudhury, S., Sun, W., Wu, Z. S., and Bagnell, J. A. All roads lead to likelihood: The value of reinforcement learning in fine-tuning. *arXiv preprint arXiv:2503.01067*, 2025.

Xie, Y., Goyal, A., Zheng, W., Kan, M.-Y., Lillicrap, T. P., Kawaguchi, K., and Shieh, M. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*, 2024.

Zhang, D., Zhoubian, S., Hu, Z., Yue, Y., Dong, Y., and Tang, J. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772, 2024a.

Zhang, Y., Lu, J., and Jaitly, N. Probing the multi-turn planning capabilities of LLMs via 20 question games. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1495–1516, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.82. URL https://aclanthology.org/2024.acl-long.82/.

A. Appendix

A.1. Methodology

A.1.1. Experimental Setup

We used models from the `Qwen3` model family as Questioner and `Qwen2.5` as Answerer model. The Answerer models that we trained have 1.7B and 4B parameters, respectively. To ensure low error rates and high answer quality, the Answerer is the larger 14B model.

To be able to use a moderate context length during training, we disabled reasoning for the Questioner models that we train. This model uses the default sampling parameters during RL training. Although increasing the temperature during training is known to improve exploration during single-step RL, we observed that it can significantly increase the chance of divergence in our multi-step environment. The Answerer LM generates a chain-of-thought before making a final decision, but we utilized greedy sampling to ensure consistent behavior from the environment.

We used verl's `FSDPSFTTrainer` to perform SFT training (Sheng et al., 2024). For multi-turn SFT we used the library's `Multiturn-SFT-Dataset`, whereas for single-turn SFT we implemented a modified version of the `SFTDataset` that accepts multi-turn conversations as masked prompts and only performs SFT on the next assistant response.

A.1.2. Rejection Sampling and Supervised Fine-Tuning

Although modern LMs are certainly aware of the rules of Twenty Questions, they were most likely not trained on actual gameplay data. To ensure successful RL training and as a baseline to compare against, we employ Supervised Fine-Tuning (SFT) with Rejection Sampling (Grover et al., 2018). We use a range of cheap API models, such as Gemini-2.0-Flash, and capable open-weight models that can be run locally, such as Gemma-3-27b-it and Qwen2.5-72B-Instruct, to collect high-quality SFT data. For each secret word, we used the games that required the fewest or second-lowest number of questions as SFT data. These games likely contain questions with a higher information gain than longer or unsuccessful games. In addition, we collect the overall success rate of the models for each secret word. The 1,000 easiest nouns with the highest are reserved for RL training and only the SFT data of the difficult ones are used for cold start SFT. The remaining easier secrets are reserved for reinforcement learning with GRPO.

We fine-tune our policy on high-quality game transcripts by minimizing the cross-entropy to the best-performing actions. This directly maximizes the likelihood of the shortest successful question sequences.

$$\mathcal{L}_{\mathrm{SFT}}(\theta) \;=\; -\,\mathbb{E}_{(s,q)\sim\mathcal{D}_{\mathrm{SFT}}}\big[\log \pi_\theta(q \mid s)\big]$$

where $\mathcal{D}_{\mathrm{SFT}}$ is the set of (state, question) pairs from the top-2 shortest successful games per secret word (**?**). The state s consists of a system prompt and instructions, followed by a sequence of turns with previous questions q and oracle answers a.

A.1.3. Group Relative Policy Optimization

Group Relative Policy Optimization (GRPO) extends and simplifies PPO to optimize relative improvements across groups of game-play rollouts for the same secret (Shao et al., 2024). The GRPO objective is

$$\begin{aligned}
\mathcal{L}_{\mathrm{GRPO}}(\theta) &= -\sum_{g\in G}\mathbb{E}_t\Big[\min\Big(r_t^g(\theta)A_t^g, \\
&\qquad\qquad \mathrm{clip}\big(r_t^g(\theta), 1-\epsilon, 1+\epsilon\big)A_t^g\Big)\Big], \\
r_t^g(\theta) &= \frac{\pi_\theta^g(q_t \mid s_t)}{\pi_{\theta_{\mathrm{old}}}^g(q_t \mid s_t)}, \\
A_t^g &= \text{advantage for group } g.
\end{aligned}$$

In the base setting, we use a success-based reward, returning 1.0 if the game is won and 0 if it is lost.

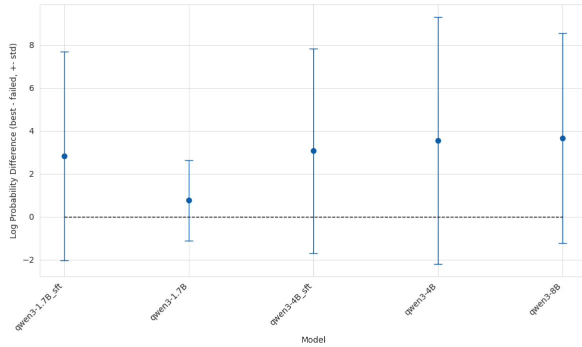

Figure 7. Mean per-secret log-probability difference between successful and failed runs (on a models' own trajectories). The differences are small compared to the noise at a per-secret level, especially for Qwen3-1.7B.

A.1.4. Statistical Analysis

To compute p-values of the log probabilities of successful vs. failed runs, we take the two randomly sampled trajectories per secret, skipping secrets with only failures or only successes. We take the mean log probability difference per step (up to the point where one trajectory finishes), and then compute the mean of per-secret mean differences. We then use a permutation test (randomly assigning the "success" and "failure" labels for each trajectory pair) to obtain a p-value that the observed mean of differences is larger than zero. Although this shows significant effects for all models, on a per-sample basis there is considerable amount of noise (Figure 7).

A.1.5. Prompts

A.1.6. Example Gameplay for the Questioner without Chain-of-Thought

```
<|im_start|>system
You are the Questioner in a game of 20 Questions, and your goal is to determine the secret
     word.
The secret is randomly drawn from the 2500 most frequent nouns of the English language.

Ask clear, concise, and strategic yes/no questions that will help you narrow down the
    possibilities.
Consider previous answers to inform your subsequent questions, and keep track of the
    information you gather.
You have a maximum of 20 questions to guess the secret word correctly. Focus on deductive
    reasoning,
and avoid open-ended questions. Start with a broad question and refine your queries as you
     progress.
<|im_end|>
<|im_start|>user
Ask a question to gain additional information about the secret or guess what the secret is
    .

Instructions:
1. Ask a question that can be answered with "Yes" or "No" to help you deduce the secret
    word.
2. Your question must be a single, brief question. Do not provide any additional
    commentary or reasoning.

Ask your question:
<|im_end|>
<|im_start|>assistant
Is the secret word a living thing?
<|im_end|>
<|im_start|>user
No
<|im_end|>
```

```
<|im_start|>assistant
Is the secret word a part of the human body?
<|im_end|>
<|im_start|>user
No
<|im_end|>
<|im_start|>assistant
Is the secret word a common object found in a house?
<|im_end|>
<|im_start|>user
No
<|im_end|>
<|im_start|>assistant
Is the secret word a concept or idea?
<|im_end|>
<|im_start|>user
Yes
<|im_end|>
<|im_start|>assistant
Is the secret word related to emotions?
<|im_end|>
<|im_start|>user
No
<|im_end|>
<|im_start|>assistant
Is the secret word related to time?
<|im_end|>
```

A.1.7. Example Gameplay for the Judge without Chain-of-Thought

```
<|im_start|>system
You are the Answerer in a game of 20 Questions.
The secret is randomly drawn from the 2500 most frequent nouns of the English language.
The secret word you have in mind is '[ANSWER]'.
Your role is to answer the questioner's questions with short and precise responses: 'Yes'
    or 'No'.
Do not provide extra hints or commentary beyond your answer.

Remain consistent and ensure that your responses accurately reflect the secret word.
Remember, your goal is to let the questioner deduce the word within 20 questions.
<|im_end|>
<|im_start|>user
Given the following question, is the correct response "Yes" or "No"?
If the question exactly matches the secret, the answer should be "Finished".

Question: '[QUESTION]'

Instructions:
1. Answer the question to the best of your abilities.
2. Provide your chain-of-thought reasoning step by step within XML <think>...</think> tags
    .
3. Don't think longer than three brief sentences.
4. Then, provide your final answer within XML tags as <answer>[Yes/No/Finished]</answer>
5. Answer with "No" if the question is empty.

Give your answer in the format <think>...</think><answer>[Yes/No]</answer>:
<|im_end|>
<|im_start|>assistant
[Yes/No/Finished]
<|im_end|>
```

A.2. Sampling Parameters

For the questioner LM, we use the default parameters recommended by the Qwen team for Qwen3 with thinking disabled (Qwen Team, 2025). These are shown in the table below. To ensure a consistently behaving environment, we use greedy sampling for the judge model.

Table 2. Sampling parameters for `Qwen3 (+ SFT)` as questioner

Parameter	Value
Temperature	0.7
Top-p	0.8
Top-k	20
Max Tokens	1024

A.3. Additional Results with Llama

To ensure that these findings are not specific to the `Qwen3` model family, we replicate most of our experiments using a different model family, `Llama-3.1/3.2` (Llama Team, 2024). We used the default sampling parameters as shown below:

Table 3. Sampling parameters for `Llama-3.1/3.2 (+ SFT)` as questioner

Parameter	Value
Temperature	0.6
Top-p	0.9
Top-k	-1
Max Tokens	1024

A.3.1. Ensuring the Benchmark is Informative

Table 4. Pass@1 (% won) results with mean and standard deviation across 16 iterations with additional experiments using Llama LMs. All finished games of the GRPO model were the result of reward hacking the judge model (*).

Model	Pass@1
Llama-3.2-1B-Instruct	5.25% $\pm$ 1.76 %
Llama-3.2-1B-Instruct + SFT	9.36% $\pm$ 1.79 %
Qwen3-1.7B	0.36% $\pm$ 0.31%
Qwen3-1.7B + SFT	8.63% $\pm$ 1.61%
Qwen3-1.7B + GRPO	*62.33% $\pm$ 1.65%
Llama-3.2-3B-Instruct	5.80% $\pm$ 1.05 %
Llama-3.2-3B-Instruct + SFT	12.93% $\pm$ 1.70 %
Qwen3-4B	3.11% $\pm$ 0.87%
Qwen3-4B + SFT	11.00% $\pm$ 1.12%
Qwen3-8B	3.87% $\pm$ 0.71%
Qwen3-14B	5.89% $\pm$ 0.77%
Gemini 2.0 Flash	15.80% $\pm$ 1.93%

A.3.2. Counterfactual Analysis: Question Generation

A.4. Maximum Belief Updates

Our figures show the development of aggregated beliefs for each turn. This can be misleading, because different models have varying degrees of initial beliefs before the first turn and because finished game (with usually higher beliefs) are not part of all later turns once they are finished. To ensure that our findings are consistent, we also report the distribution of differences between the minimum and maximum belief of each trajectory on the successful Gemini rollouts.

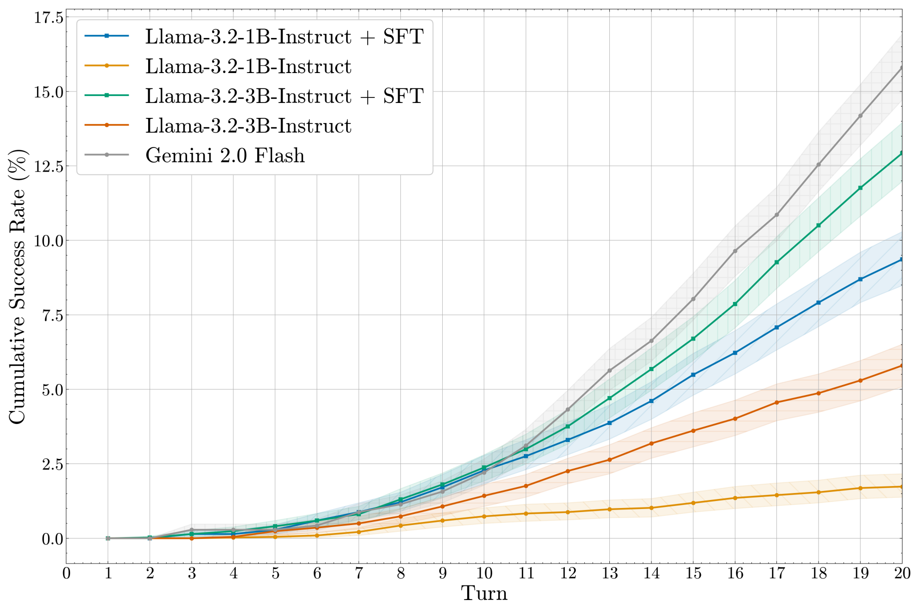

Figure 8. **Cumulative win rate across timesteps.** The number of questions needed for success, and overall success rate, improve after SFT on high-quality trajectories.

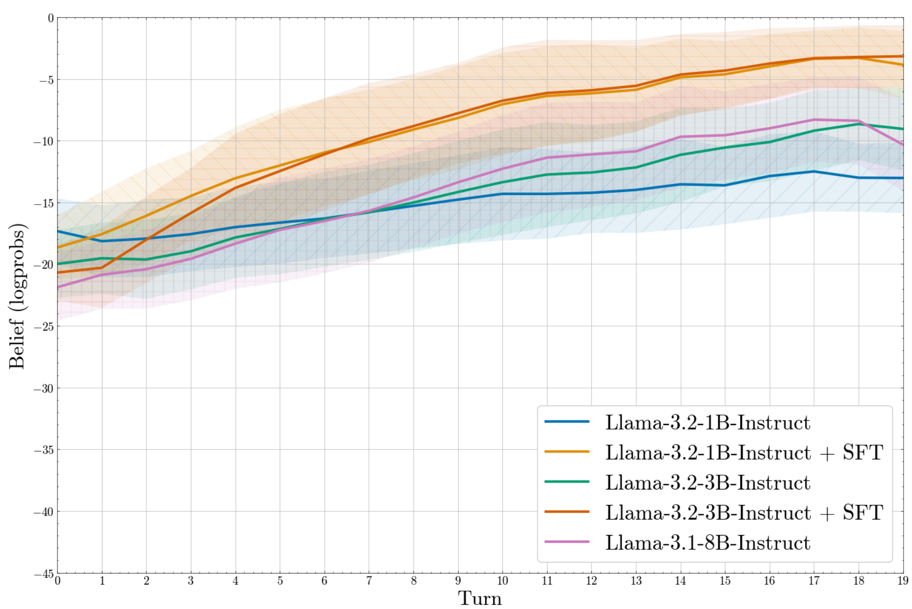

Figure 9. **Controlling for Question Generation** The models' ability to update beliefs when controlling for question quality (on Gemini's successful trajectories).

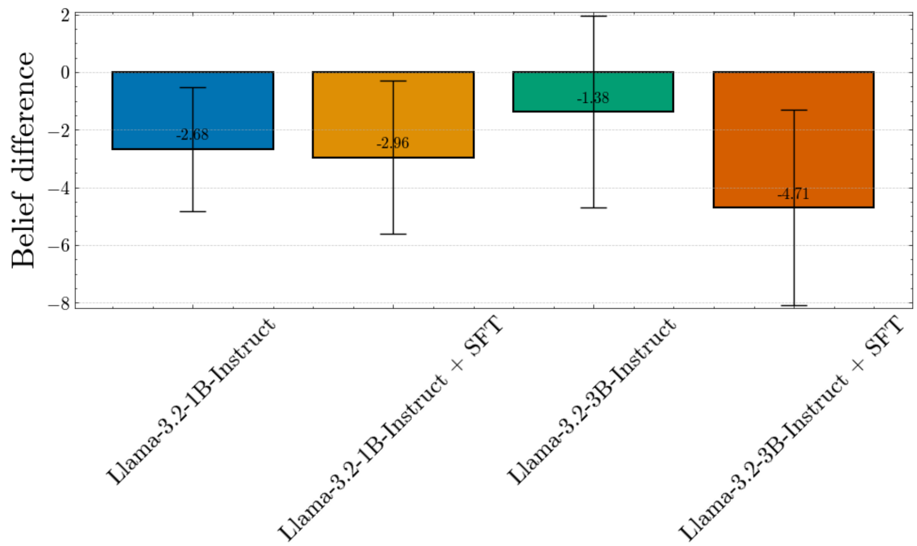

Figure 10. **Contradictory Evidence** Model capability to update beliefs when presented with contradictory evidence (”yes” and ”no” swapped) on previously successful trajectories.

Table 5. Mean of the difference of maximum and minimum beliefs of each trajectory for various models (mean ± standard deviation) across the successful Gemini evaluations. If the minimum occurs after the maximum belief, we flip the sign to show that model became less certain of the ground truth solution over the course of the game.

Model	**Mean of deltas**
Llama-3.2-1B-Instruct	7.34 ± 2.79
Llama-3.2-1B-Instruct + SFT	16.20 ± 3.31
Qwen3-1.7B	5.98 ± 2.91
Qwen3-1.7B + SFT	18.21 ± 3.88
Qwen3-1.7B + GRPO	-26.24 ± 6.69
Llama-3.2-3B-Instruct	11.91 ± 3.62
Llama-3.2-3B-Instruct + SFT	19.03 ± 3.11
Qwen3-4B	15.74 ± 5.29
Qwen3-4B + SFT	18.07 ± 3.50
Llama-3.1-8B-Instruct	14.72 ± 4.17
Qwen3-8B	15.79 ± 5.87
Qwen3-14B	18.80 ± 6.85