Gathering Context that Supports Decisions via Entropy Search with Language Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Real-world decision making systems require background information about the environment to take effective actions. However, this information is frequently incomplete or costly to acquire. Rather than presuming complete context, an effective decision maker must actively gather relevant information through a sequence of targeted follow-up questions before making decisions. This paper presents a framework for adaptive information gathering using large language models (LLMs) as interactive decision-making agents. Guided by an information-theoretic objective, the LLM selects questions that minimize the entropy of the predicted optimal action distribution, effectively prioritizing information that reduces uncertainty. Our method enables instance-specific reasoning under uncertainty and improves decision quality through principled context acquisition. We evaluate our approach on modified versions of three standard benchmarks—1D-ARC, GSM8K, and Fermi—adapted to study partially observable contexts where relevant information must be actively gathered. We assess performance using state-of-the-art LLMs. Empirically, we find that our proposed Entropy Search strategy consistently outperforms strong baselines, demonstrating the effectiveness of uncertainty-guided information gathering for LLM-based decision support[1].

1 Introduction

In everyday interactions with large language models (LLMs), users often provide incomplete problem descriptions. This is not a shortcoming of the models alone; rather, it reflects a fundamental limitation in how humans communicate—users frequently omit relevant contextual details, either due to cognitive constraints or implicit assumptions about shared knowledge. As a result, the model must reason under uncertainty, which can result in suboptimal responses [Jiang et al., 2023]. Moreover, task underspecification can pose a significant AI safety and alignment risk [Amodei et al., 2016, Ngo et al., 2024, Dalrymple et al., 2024]. These limitations and risks point to a broader challenge in human-AI collaboration: effective decision support often requires adaptive information gathering. This interactive paradigm is especially valuable in high-stakes or data-scarce domains, such as forecasting [Tetlock and Gardner, 2016, Schoenegger et al., 2025], personalized medicine [Hamburg and Collins, 2010] and education [Ericsson and Pool, 2016, Vygotsky and Cole, 1978, Robinson and Aronica, 2009], where acquiring information is costly and outcomes depend critically on a limited set of key features.

In this work, we formalize this interactive decision-support setting as a two-stage sequential decision-making problem: An agent observes a user with a query and can ask several follow-up questions, and after gathering the user's answers to these questions, the agent selects an action that maximizes the expected outcome. The core research question is: *Which questions should the agent ask to inform its final decision best?* The key technical challenge lies in identifying informative, non-redundant

[1] Our implementation is available at `https://anonymous.4open.science/r/info-gathering-047B/`

Submitted to the First Exploration in AI Today Workshop at ICML (EXAIT at ICML 2025). Do not distribute.

questions that reduce uncertainty about the optimal action, especially when features are correlated. Our method uses entropy minimization to determine the next most informative question, conditioned on previous responses. By minimizing uncertainty over optimal actions, our approach allows the agent to prioritize questions that elicit the most information about the best action. This principled strategy enables the agent to gather just enough information to act effectively, balancing informativeness with efficiency. Building on this foundation, we make the following contributions:

- We introduce entropy search, a method that enables language model agents to ask targeted questions that reduce outcome entropy and improve decisions under partial information. We theoretically characterize its optimality and identify conditions under which it ensures optimal actions.
- We experiment on four state-of-the-art language models on partially observable variants of three popular reasoning benchmarks, showing that the entropy minimization strategy consistently outperforms competitive baselines.

2 Background

Algorithm 1 defines our typical information-gathering setting where an LLM agent interacts with an external user. At each step, the agent observes the problem and current context $H = (q_t, a_t)_{t=1}^{t-1}$, selects a follow-up question q, and receives an answer a from the user. This interaction continues until the budget is exhausted. The agent finally responds with an action A. The different algorithms compared in the experiments only differ in how the agent selects a follow-up question.

Next, we introduce some notation for a Bayesian formulation of the information-gathering problem. A language model acts as a decision-making agent that interacts with a sequence of T users $u_1, u_2, \ldots, u_T$. Each user u_t arrives with a hidden context vector $X_t \in \mathcal{X}^K$, where K is the total number of features (potentially large). This contrasts with the standard contextual bandit setting [Neu et al., 2022, Agrawal and Goyal, 2013], in which the full context is known to the agent. At the beginning of each interaction, the user poses a query that reveals a subset of features—modeled as a binary vector $S_t^0 \in \{0,1\}^K$, corresponding to the initially observed context X_{t,S_t^0}. The agent can then ask a sequence of follow-up questions, with the user giving an answer that reveals additional features one at a time. This process results in a final observed subset $S_t \in \{0,1\}^K$, which defines the gathered context X_{t,S_t}. Based on this partial context, the agent selects an action $A_t \in \mathcal{A}$ from a finite (potentially large) set of possible actions. After the action is taken, the agent observes a binary outcome $Y_{t,A_t} \in \{0,1\}$.

The agent maintains two learnable parameters: (ω, θ). Let $Y_t \in \{0,1\}^{\mathcal{A}}$ denote the potential outcome vector over all possible actions. In the contextual bandit literature [Neu et al., 2022, Agrawal and Goyal, 2013], θ typically parameterizes the reward model, encoding the agent's belief over outcomes conditioned on the full context: $p_t(Y_t \mid X_t, \theta)$. We generalize this definition and let θ parameterize the full joint distribution $p_t(Y_t, X_t \mid \theta)$, capturing both the agent's world knowledge about user context distributions $p_t(X_t \mid \theta)$ and the conditional outcome model $p_t(Y_t \mid X_t, \theta)$.

The expected reward for a selected action is given by: $r_t(\theta, A_t, X_t) = \mathbb{E}_t[Y_{t,A_t} \mid \theta, X_t]$. The optimal action under the true model θ^* is: $A_t^* = \arg\max_{a \in \mathcal{A}} r_t(\theta^*, a, X_t)$. After taking action A_t and observing outcome Y_{t,A_t} and context X_{t,S_t}, the agent updates its posterior over θ via:

$$q_{t+1}(\theta) = \frac{q_t(\theta)\, p_t(Y_{t,A_t} \mid \theta, A_t, X_{t,S_t})}{\sum_{\theta'} q_t(\theta')\, p_t(Y_{t,A_t} \mid \theta', A_t, X_{t,S_t})}$$

We assume a categorical outcome distribution: $\sum_{a \in \mathcal{A}} \mathbb{E}_t[Y_{t,a} \mid \theta, X_t] = \sum_{a \in \mathcal{A}} r_t(\theta, a, X_t) = 1$ which differs from the Bernoulli setting where multiple actions can yield positive outcomes. In practice, posterior updates over θ can be implemented via supervised fine-tuning [Ouyang et al., 2022]. As $\theta \to \theta^*$, the expected outcome $\mathbb{E}_t[Y_{t,A_t} \mid \theta, X_t]$ converges to the true expected outcome.

To estimate the optimal action, the agent first engages in a sequence of Q_t follow-up questions $\{S_t^q\}_{q \in [Q_t]}$ with the user. Each question $S_t^q \in \{0,1\}^K$ specifies the indices of additional user context to be revealed at step q, resulting in observed context X_{t,S_t^q}. The agent's policy for generating these questions is parameterized by ω, and is conditioned on the context revealed so far: $p_t(S_t^q \mid \omega, X_{t,S_t^{<q}})$. The interaction terminates after Q_t follow-up queries, at which point the agent has accumulated a final observed subset $S_t = \bigcup_{q=1}^{Q_t} S_t^q$ corresponding to context X_{t,S_t}. Based on this gathered context, the agent selects an action: $A_t = \arg\max_a r_t(\theta_t, a, X_{t,S_t})$ and incurs instantaneous regret

$\Delta_t = r_t(\theta^*, A_t^*, X_t) - r_t(\theta^*, A_t, X_t)$. Here, A_t^* is the optimal action under the true model θ^*. The agent's objective is to minimize the expected cumulative regret over T episodes:

$$R_T = \sum_{t=1}^{T} \mathbb{E}\left[r_t(\theta^*, A_t^*, X_t) - r_t(\theta^*, A_t, X_t)\right] = \sum_{t=1}^{T} \mathbb{E}\left[\Delta_t\right]$$

where the expectation is taken over all randomness: environment parameters and agent policy.

While regret can be used as a learning signal to update both (ω, θ), optimizing ω directly from regret suffers from the sparse rewards problem: the agent receives feedback only after completing the entire sequence of follow-up questions, based on the final outcome for its action, as is done in prior works [Andukuri et al., 2024]. Delayed supervision makes it difficult to attribute success or failure to individual follow-up queries. We don't assume gold supervision for follow-up questions.

To overcome this, we propose an entropy search algorithm that evaluates each potential follow-up question before it is asked. This approach assigns a score to every candidate query based on its expected reduction in uncertainty over the outcome. Prior works have used similar intrinsic scores based on information-gain [Houthooft et al., 2016, Mohamed and Jimenez Rezende, 2015] to tackle sparse rewards. These scores are then used both to select the next follow-up question and to update the policy parameters ω, providing a more informative and dense supervision signal during training. For reference, we summarize all the notation in Appendix 6.

3 Entropy Search Algorithm

We now present the Entropy Search algorithm 2. Following this strategy at each iteration, the decision maker asks the question with the greatest reduction in posterior uncertainty about the optimal action, on average, i.e., the question that elicits the most information.

To understand why this algorithm is optimal for the information-gathering task, we begin with a formal definition of optimal follow-up questions and an associated objective for scoring and selecting them. We then establish theoretical guarantees for this entropy-based objective. Finally, we provide a regret analysis for the entropy search procedure's exact and approximate implementations.

Definition 3.1 (Optimal q^{th} follow-up question S_t^q for user u_t)**.** Before the qth follow-up question, the agent has already observed a subset of user context indices denoted by $S_t^{<q} = \bigcup_{j=0}^{q-1} S_t^j$. The ideal query S_t^q would reveal just enough additional context to fully recover the potential outcome distribution: $p_t(Y_t \mid X_{t,S_t^{<q}}, X_{t,S_t^q}) = p_t(Y_t \mid X_t) = p_t(Y_t \mid X_{t,S_t^{<q}}, X_{t,(S_t^{<q})^-})$, where $(S_t^{<q})^- = \mathbf{1} \setminus S_t^{<q}$ denotes the unobserved context indices.

Among all such queries, we aim to select one that requires minimal user effort to answer—formally, the smallest number of new context features—while avoiding redundancy. Using a statistical distance measure d, the optimal query is defined as:

$$\begin{aligned} S_t^q &= \underset{S\in\{0,1\}^K}{\arg\min}\, |S| \quad \text{s.t.} \quad d\left(p_t(Y_t \mid X_{t,S_t^{<q}}, X_{t,S}) \,\middle\|\, p_t(Y_t \mid X_t)\right) = 0, \\ &= \arg\min_{S} \left\{ |S| \,\middle|\, S \in \arg\min_{S'} d\left(p_t(Y_t \mid X_{t,S_t^{<q}}, X_{t,S'}) \,\middle\|\, p_t(Y_t \mid X_t)\right),\ S \in \{0,1\}^K \right\}. \end{aligned}$$

The two formulations are equivalent because the minimum distance d is zero for at least one solution—namely, $S_t^q = (S_t^{<q})^-$, which reveals all remaining context. However, this may not be the minimal-norm solution. Once the distance is reduced to zero, further follow-up queries cannot improve the prediction, and the agent can terminate questioning and respond with its final action.

In practice, the posterior $p_t(Y_t \mid X_t)$ is not directly observable, making it infeasible to minimize divergence-based objectives such as total variation or KL divergence. Instead, we use conditional entropy difference as a surrogate distance measure, which has the key advantage of not requiring access to the full posterior:

$$S_t^q = \underset{S\in\{0,1\}^K}{\arg\min}\, \mathbb{H}_t(Y_t \mid X_{t,S_t^{<q}}, X_{t,S}) - \mathbb{H}_t(Y_t \mid X_t) = \underset{S\in\{0,1\}^K}{\arg\min}\, \mathbb{H}_t(Y_t \mid X_{t,S_t^{<q}}, X_{t,S})$$

We follow a greedy procedure that selects one context feature at a time per follow-up question, i.e., $|S_t^q| = 1$. This choice makes the optimization tractable, reducing the search complexity from combinatorial $\mathcal{O}(2^K)$ to linear $\mathcal{O}(K)$. To do so, we assign a score to each candidate question: $v_t^q(S) = -\mathbb{H}_t(Y_t|X_{t,S_t^{<q}}, X_{t,S})$ and select the context feature that maximizes this score. Thus, our optimization becomes:

$$S_t^q = \underset{S\in\{0,1\}^K,|S|=1}{\arg\min}\ \mathbb{H}_t(Y_t|X_{t,S_t^{<q}}, X_{t,S}) = \underset{S\in\{0,1\}^K,|S|=1}{\arg\max}\ v_t^q(S)$$

We continue this procedure until the marginal entropy reduction $\mathbb{H}_t(Y_t|X_{t,S_t^{<q}}) - \mathbb{H}_t(Y_t|X_{t,S_t^{<q}}, X_{t,S_t^q})$ is sufficiently small—that is, when the new question contributes little additional information. We assume a budget B that bounds the total number of follow-up questions allowed per user. The final gathered context is: $S_t = \bigcup_{j=0}^{Q_t} S_t^j$. The entropy score $v_t^q(S) = -\mathbb{H}_t(Y_t \mid X_{t,S_t^{<q}}, X_{t,S})$ is estimated using Monte Carlo sampling, by generating potential outcome distributions under each candidate question S. In practice, due to limited compute, we sample only a subset of $k \ll K$ candidate follow-up questions from the policy $p_t(S_t^q \mid \omega, X_{t,S_t^{<q}})$ prior to estimating the scores.

In our setting, outcomes are modeled as a categorical distribution over actions, where exactly one action is correct. This structure allows us to estimate entropy more efficiently. Rather than computing the full distribution over potential outcomes and then selecting the best action, we directly compute the entropy over the action distribution induced by the model. This approach avoids a separate outcome prediction step and simplifies entropy estimation within our algorithm. The following lemma formalizes this equivalence (proof in Appendix 7.1):

Lemma 3.2 (Equivalence of Action and Outcome Entropy)**.** *For a categorical distribution over outcomes, minimization of outcome entropy is equivalent to minimization of action entropy:*

$$\arg\min_S \mathbb{H}_t(Y_t|X_{t,S}) = \arg\min_S \mathbb{H}_t(A_t|X_{t,S})$$

3.1 Optimality of Entropy Search

Matching entropy does not imply matching distributions in general. However, since we minimize entropy only by expanding the conditioning set, it decreases strictly unless the conditional distributions match. This justifies entropy as a surrogate loss for the distribution matching objective. The following lemma formalizes this via a bound on total variation distance:

Lemma 3.3 (Upper Bound on Total Variation)**.** *For any* $\epsilon > 0$*, we have*

$$\mathbb{H}_t(Y_t|X_{t,S_t}) - \mathbb{H}_t(Y_t|X_t) \le \epsilon \Rightarrow ||p_t(Y_t|X_{t,S_t}) - p_t(Y_t|X_t)||_1 \le \sqrt{\frac{\epsilon}{2}}$$

In particular, $\mathbb{H}_t(Y_t|X_{t,S_t}) = \mathbb{H}_t(Y_t|X_t) \Rightarrow p_t(Y_t|X_{t,S_t}) = p_t(Y_t|X_t)$

The proof follows from Pinsker's inequality, and we provide the details in the Appendix 7.2.

3.2 Convergence Analysis

We show that the minimal entropy-minimizing context subset S_t^* suffices for the posterior sampling of the optimal action. In particular, conditioning on X_{t,S_t^*} is equivalent to conditioning on the full context X_t for both the current parameters θ_t and the true parameters θ^*. This means no additional context is needed once outcome entropy is minimized. The same holds for any $S_t \supseteq S_t^*$.

Lemma 3.4. *Following the information-gathering procedure and posterior sampling for action selection, the following equalities hold under exact entropy search:*

$$\mathbb{P}_t(A_t^*, \theta^* \mid X_{t,S_t^*}) = \mathbb{P}_t(A_t, \theta_t \mid X_{t,S_t^*}) = \mathbb{P}_t(A_t^*, \theta^* \mid X_t) = \mathbb{P}_t(A_t, \theta_t \mid X_t)$$

The proof in Appendix 7.3 follows from posterior sampling and Lemma 3.3.

We state a key lemma extending the information ratio framework of Neu et al. [2022] to partial context settings, yielding an analogous regret bound:

Lemma 3.5 (Bounded Information Ratio)**.** *The information ratio for the observed set* S_t *is defined as follows:*

$$\rho_t(S_t) = \frac{(\mathbb{E}_t\left[r_t(\theta_t, A_t, X_{t,S_t}) - r_t(\theta^*, A_t, X_{t,S_t})|X_{t,S_t}\right])^2}{\mathbb{I}_t\left(\theta^*; Y_{t,A_t}|A_t, X_{t,S_t}\right)}$$

admits the bound $\rho_t(S_t) \le 2\sum_{a\in\mathcal{A}} \mathbb{E}\left[\bar{r}_t(a, X_{t,S_t})\right]$

The proof technique is similar to [Neu et al., 2022], and more details are in Appendix 7.4.

Theorem 3.6 (Regret of Exact Entropy Search)**.** *Exact entropy search achieves the regret bound:* $R_T \leq \sqrt{2T\mathbb{H}(\theta^*)}$

We provide a detailed proof in Appendix 7.7.

At each turn t, we update the ω parameters using the sampled entropy scores. This process can be viewed as transferring knowledge from System-2 (deliberate, exhaustive evaluation) to System-1 (fast, heuristic-based selection) over time, consistent with the dual-process theory of cognition [Booch et al., 2021]. The key idea is that we are learning a policy that learns how to gather information efficiently. Since evaluating all K follow-up questions (a full System-2 search) is computationally infeasible, we instead train ω to generate a smaller, high-quality subset of $k \ll K$ candidate questions using System-1-style inference. We then apply System-2 reasoning within this restricted subset by selecting the candidate with the lowest estimated entropy. As ω improves, the quality of this subset increases, yielding better candidates more often. This motivates our assumption that the entropy gap ϵ_t diminishes over time—a property we exploit in Theorem 3.7 to establish sublinear regret despite constrained computational resources.

The main contributor to non-zero gap ϵ_t is failing to include the optimal question $\bar{S}_t^q = \arg\min_{S_t^q} \mathbb{H}_t(Y_t|X_{t,S_t^{<q}}, X_{t,S_t^q})$ in the candidate set for entropy estimation, not poor outcome estimation. While updates to θ do refine entropy estimates through calibrated outcome prediction, they are unlikely to significantly reduce ϵ_t, as most language models have a strong enough prior to choosing the optimal question if all options were considered.

Theorem 3.7 (Regret)**.** *Approximate entropy search achieves the regret bound:* $R_T \leq \sqrt{2T\mathbb{H}(\theta^*)} + \mathcal{O}(T^{1-\alpha/2})$ *in the one-hot outcome setting.*

We assume that Entropy Search learns to select the optimal subset with diminishing error $\epsilon_t = \mathbb{H}_t(Y_t|X_{t,S_t}, \theta) - \mathbb{H}_t(Y_t|X_t, \theta), \mathbb{E}[\epsilon_t] = \mathcal{O}\left(\frac{1}{t^\alpha}\right)$*. In particular, for* $\alpha = 1$*, we obtain* $R_T = \mathcal{O}(\sqrt{T})$*, identical to the exact entropy search setting.*

We provide a detailed proof in Appendix 7.8.

Lemma 3.8 (Entropy Search Lowers Per-Question Suboptimality over ReAct)**.** *We define the per-question suboptimality of an algorithm as* $\epsilon_t^q = \mathbb{H}_t(Y_t|X_{t,S_t^{<q}}, X_{t,S_t^q}) - \mathbb{H}_t(Y_t|X_{t,S_t^{<q}}, X_{t,\bar{S}_t^q})$ *where* $\bar{S}_t^q$ *is the optimal q^th^ question that minimizes the outcome entropy. Let* $p_{t,q}^* = p_t(S_t^q = \bar{S}_t^q|\omega, X_{t,S_t^{<q}})$ *be the probability that the policy selects the optimal q^th^ follow-up question. For a budget of* $k \ll K$ *candidate questions in Entropy Search (ES), we achieve lower suboptimality than the ReAct baseline:*

$$(\epsilon_t^q)_{ES} \leq (1 - p_{t,q}^*)^{k-1} (\epsilon_t^q)_{ReAct}$$

We defer the proof to Appendix 7.9 along with a discussion of this result.

4 Experiments

We have presented a theoretical framework for deriving a training objective that encourages a language model to seek information for decision-making. In this section, we experiment with three benchmarks, each comprising 1000 data points, to facilitate empirical studies. We go into detail on each of the benchmarks GSM8K-Q, 1D-ARC, and Fermi Problems, in Appendix 9.

4.1 Algorithms

We compare ENTROPY SEARCH to two other agent baselines:

1. REACT: the agent first produces a chain-of-thought conditioned on (*problem*, H) and emits the next question greedily at the point where reasoning requires a missing fact.

2. REFLEXION: The reflexion agent [Shinn et al., 2023] includes a dedicated reflection component, where after each interaction, the agent evaluates what it has learned, assesses the effectiveness of its previous queries, identifies missing information, and adjusts its questioning strategy accordingly. The prompt structure incorporates a dedicated "Reflection" section, includes past reflections for continuity, and instructs the agent explicitly to reflect before reasoning. Memory stores and displays reflections in subsequent interactions, enabling learning and strategic adjustments.

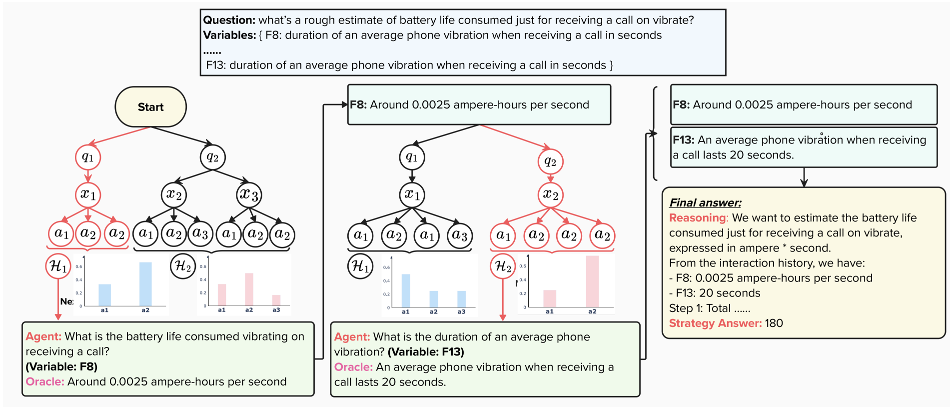

Figure 2: Example of the ENTROPY SEARCH agent interacting on a Fermi estimation task. The agent selects targeted follow-up questions (e.g., requesting values for Variables F8 and F13) to reduce uncertainty and gather key facts incrementally. Using the retrieved values, the agent composes a final answer by chaining units and performing multi-step reasoning, highlighting the interpretability and structured nature of the entropy-guided decision process.

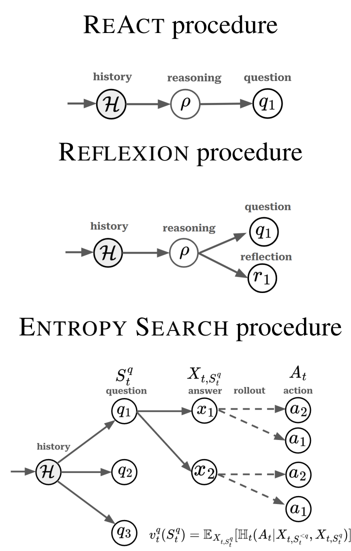

Figure 1: Comparison of question selection mechanisms

We also compare against a FULL INFORMATION setting with access to the entire user context.

We primarily used the `Gemma-3 12B` [Team et al., 2025] model to balance performance and computational efficiency. We also ran evaluations using multiple state-of-the-art language models to assess robustness across architectures. These additional details and results are in Appendix 14.1.

4.2 Results

Figure 12 presents the comparative evaluation of method performance across three challenging datasets: 1D-ARC, GSM8K, and Fermi. As expected, the FULL INFORMATION baseline serves as a practical performance ceiling due to its complete contextual details. Crucially, ENTROPY SEARCH substantially narrows this gap more effectively than agentic reasoning methods like REFLEXION and REACT, highlighting the value of strategically prioritizing uncertainty reduction.

In principle, one might expect sufficiently intelligent agents could infer complete symbolic solutions directly from GSM and Fermi question statements, reducing the need for additional information. However, our empirical findings robustly contradict this hypothesis, showing that entropy-guided information gathering significantly enhances performance even when symbolic solutions might seem inferable initially.

Moreover, the performance on Fermi problems underscores another dimension of our method's strength: its ability to handle real-world estimation tasks that require nuanced common-sense reasoning. Remarkably, ENTROPY SEARCH not only outperformed the competing methods but matched the performance of the FULL INFORMATION scenario. This is particularly compelling given the inherent uncertainty and complexity involved in Fermi estimations.

Further granularity is provided by the turn-based evaluation depicted in Figure 4, confirming that ENTROPY SEARCH consistently improves with more turns and outperforms baselines at every stage. These findings collectively underscore ENTROPY SEARCH's effectiveness in strategically acquiring context under uncertainty, thus significantly enhancing decision-making quality in partially observable environments. Additional detailed turn-based analyses for the 1D-ARC and GSM8K benchmarks are available in Appendix 14.2, further validating these insights.

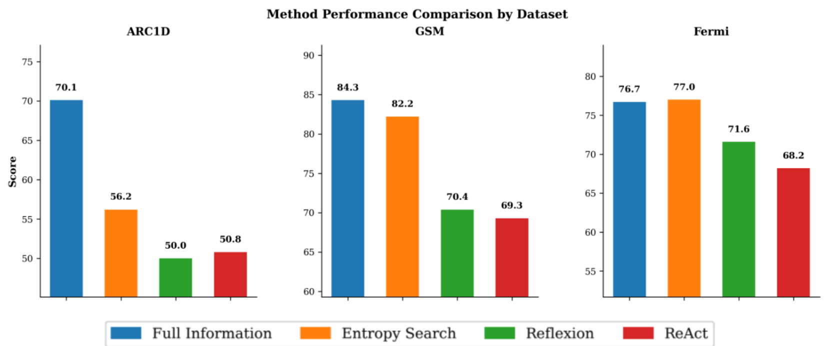

Figure 3: Method performance comparison across three benchmarks. FULL INFORMATION represents the upper bound with complete context. ENTROPY SEARCH consistently outperforms REACT and REFLEXION across all datasets and approaches the performance of the FULL INFORMATION baseline, demonstrating the value of principled, entropy-guided question selection.

5 Conclusion, Limitations, and Future Work

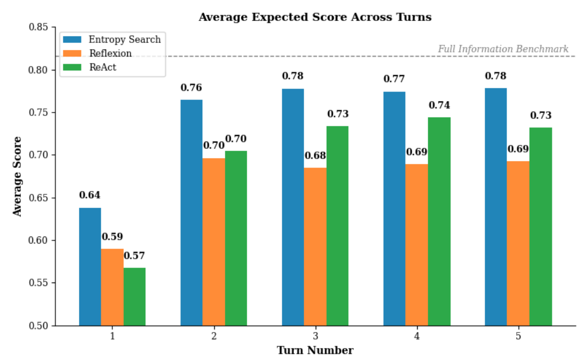

Figure 4: Turn-based evaluation for Fermi Problems. We conduct detailed comparisons across methods (ENTROPY SEARCH, REFLEXION, REACT) and turns, revealing that ES consistently closes the performance gap with the full information setting with each additional question. This highlights both the generality and efficiency of entropy-guided information acquisition.

We introduce an entropy search for context gathering in LLMs agents, enabling the selection of questions that reduce uncertainty and improve downstream decision-making. Our framework is model-agnostic and may be paired with a variety of models. By applying this method across various reasoning tasks, we demonstrate that entropy search leads to superior performance over state-of-the-art baselines. One limitation of our work is the computational cost associated with the Monte Carlo estimation of entropy. Future work could address this by using value network estimation to reduce the computational burden. Further, our work did not consider information gathering in the context of safe exploration [Ray et al., 2019, Garcıa and Fernández, 2015, Wachi et al., 2023], which is an important direction for future work. This work also opens several other promising directions for future research. Applying this framework to interactive human-AI collaboration in domains such as scientific discovery, diagnostic reasoning, and judgemental forecasting could yield both practical benefits and deeper insights into the role of adaptive information gathering in complex decision systems. Information seeking is also a key capability that should be monitored for AI safety and alignment; future frontier models can be benchmarked against our work to monitor information seeking capability development. By endorsing language models with information seeking capabilities, our work contributes a framework for a cooperative, interactive intelligent system, highlighting the broader potential of LLMs as strategic agents in sequential decision-making under uncertainty.

References

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*, pages 127–135. PMLR, 2013.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah Goodman. STar-GATE: Teaching language models to ask clarifying questions. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=CrzAj0kZjR.

Grady Booch, Francesco Fabiano, Lior Horesh, Kiran Kate, Jonathan Lenchner, Nick Linck, Andreas Loreggia, Keerthiram Murgesan, Nicholas Mattei, Francesca Rossi, et al. Thinking fast and slow in ai. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 15042–15046, 2021.

Djallel Bouneffouf, Irina Rish, Guillermo A. Cecchi, and Raphael Feraud. Context attentive bandits: Contextual bandit with restricted context, 2017. URL https://arxiv.org/abs/1705.03821.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995. ISSN 08834237. URL http://www.jstor.org/stable/2246015.

David Dalrymple, Joar Skalse, Yoshua Bengio, Stuart Russell, Max Tegmark, Sanjit Seshia, Steve Omohundro, Christian Szegedy, Ben Goldhaber, Nora Ammann, et al. Towards guaranteed safe ai: A framework for ensuring robust and reliable ai systems. *arXiv preprint arXiv:2405.06624*, 2024.

M. H. DeGroot. Uncertainty, Information, and Sequential Experiments. *The Annals of Mathematical Statistics*, 33(2):404 – 419, 1962. doi: 10.1214/aoms/1177704567. URL https://doi.org/10.1214/aoms/1177704567.

Audrey Durand and Christian Gagné. Thompson sampling for combinatorial bandits and its application to online feature selection. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

Anders Ericsson and Robert Pool. *Peak: Secrets from the new science of expertise*. Random House, 2016.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Margaret A Hamburg and Francis S Collins. The path to personalized medicine. *New England Journal of Medicine*, 363(4):301–304, 2010.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Mingjian Jiang, Yangjun Ruan, Sicong Huang, Saifei Liao, Silviu Pitis, Roger Baker Grosse, and Jimmy Ba. Calibrating language models via augmented prompt ensembles. https://openreview.net/pdf?id=L0dc4wqbNs, 2023.

Ashwin Kalyan, Abhinav Kumar, Arjun Chandrasekaran, Ashish Sabharwal, and Peter Clark. How much coffee was consumed during emnlp 2019? fermi problems: A new reasoning challenge for ai. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7318–7328, 2021.

Sanath Kumar Krishnamurthy and Susan Athey. Survey bandits with regret guarantees, 2020. URL https://arxiv.org/abs/2002.09814.

Belinda Z Li, Been Kim, and Zi Wang. Questbench: Can llms ask the right question to acquire information in reasoning tasks? *arXiv preprint arXiv:2503.22674*, 2025.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 28, 2015.

Gergely Neu, Iuliia Olkhovskaia, Matteo Papini, and Ludovic Schwartz. Lifting the information ratio: An information-theoretic analysis of thompson sampling for contextual bandits. *Advances in Neural Information Processing Systems*, 35:9486–9498, 2022.

Richard Ngo, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep learning perspective. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=fh8EYKFKns.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7(1):2, 2019.

Ken Robinson and Lou Aronica. *The element: How finding your passion changes everything*. Penguin, 2009.

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of thompson sampling. *Journal of Machine Learning Research*, 17(68):1–30, 2016.

Philipp Schoenegger, Peter S Park, Ezra Karger, Sean Trott, and Philip E Tetlock. Ai-augmented predictions: Llm assistants improve human forecasting accuracy. *ACM Transactions on Interactive Intelligent Systems*, 15(1):1–25, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Philip E Tetlock and Dan Gardner. *Superforecasting: The art and science of prediction*. Random House, 2016.

Sabina Tomkins, Peng Liao, Predrag Klasnja, and Susan Murphy. Intelligentpooling: practical thompson sampling for mhealth. *Machine Learning*, 110(9):2685–2727, 2021. doi: 10.1007/s10994-021-05995-8. URL `https://doi.org/10.1007/s10994-021-05995-8`.

Lev Semenovich Vygotsky and Michael Cole. *Mind in society: Development of higher psychological processes*. Harvard university press, 1978.

Akifumi Wachi, Wataru Hashimoto, Xun Shen, and Kazumune Hashimoto. Safe exploration in reinforcement learning: A generalized formulation and algorithms. *Advances in Neural Information Processing Systems*, 36:29252–29272, 2023.

Yudong Xu, Wenhao Li, Pashootan Vaezipoor, Scott Sanner, and Elias Boutros Khalil. Llms and the abstraction and reasoning corpus: Successes, failures, and the importance of object-based representations. *Transactions on Machine Learning Research*, 2024.

Jiayu Yao, Emma Brunskill, Weiwei Pan, Susan Murphy, and Finale Doshi-Velez. Power constrained bandits. In Ken Jung, Serena Yeung, Mark Sendak, Michael Sjoding, and Rajesh Ranganath, editors, *Proceedings of the 6th Machine Learning for Healthcare Conference*, volume 149 of *Proceedings of Machine Learning Research*, pages 209–259. PMLR, 06–07 Aug 2021. URL `https://proceedings.mlr.press/v149/yao21a.html`.

6 Notation

Table 1: Notation

t	Index of user interacting with the agent
T	Total number of users the agent interacts with
u_t	The tth user
X_t	Random variable representing full context for user u_t
K	Total number of features in a user's context (potentially large)
$\mathcal{X}$	Set of values from which a user's context feature is drawn, $X_t \in \mathcal{X}^K$
A_t	Final action performed by agent in response to user u_t
$\mathcal{A}$	Finite set of actions that agent can perform
A	$\lvert\mathcal{A}\rvert$, number of actions in agent's action space (potentially large)
Y_{t,A_t}	Outcome in $\{0,1\}$ for user u_t given action A_t
Y_t	Potential outcomes under different actions, in $\{0,1\}^{\mathcal{A}}$
S_t^0	Binary random variable representing the set of context feature indices initially revealed by the user, in $\{0,1\}^K$
S_t^q	Binary random variable representing the set of additional context feature indices gathered by asking the qth follow-up question
X_{t,S_t^q}	Partial context of u_t revealed by question asking about features in S_t^q
S_t	Binary random vector representing the entire set of user feature indices that are gathered by asking all the follow-up questions $\bigcup_{q\geq 0} S_t^q$
X_{t,S_t}	Random variable representing the partial user context that is observed by the agent at the time of performing an action
Q_t	Total number of follow-up questions that the agent asks the user u_t
B	Budget for maximum number of follow-up questions that the agent can ask the user u_t
$S_t^{<q}$	$\bigcup_{j=0}^{q-1} S_t^j$, set of user feature indices observed before asking the qth question
$\mathbb{1}$	Set of all user feature indices $\{1\}^K$
$(S)^-$	$\mathbb{1} \setminus S$, complement of user context features in set S, possibly unobserved feature indices
θ	Random variable representing agent parameters for the joint distribution of user context X and potential outcomes Y
ω	Random variable representing agent parameters for follow-up question generation policy
$\mathcal{F}_t$	Filtration / History prior to user t, equals $\{X_{1,S_1}, A_1, Y_{1,A_1}, \cdots, X_{t,S_t}, A_t, Y_{t,A_t}\}$
p_t	Probability distribution conditioned on $\mathcal{F}_t$
r_t	Reward / Expected outcome model for user u_t, $r_t(\theta, A_t, X_{t,S}) = \mathbb{E}_t[Y_{t,A_t}\lvert\theta, X_{t,S}]$
θ^*	Random variable representing optimal parameters for reward / expected outcome model
A_t^*	Random variable representing optimal action that maximizes expected outcome $r_t(\theta^*, A, X_t)$ given full context
$q_t(\theta)$	Posterior distribution over parameter θ after observing $\mathcal{F}_t$
Δ_t	Instantaneous regret for action selected for user u_t
R_T	Expected cumulative regret
$d(\cdot\lvert\rvert\cdot)$	Statistical distance measure on two distributions
$\mathbb{H}_t$	Entropy of distribution conditional on $\mathcal{F}_t$
$v_t^q(S)$	Entropy score for candidate q^{th} question, negative of expected outcome entropy conditional on observations before q
k	Budget for number of candidate follow-up questions that are evaluated with the entropy score prior to selecting one to ask the user
$\rho_t(S_t)$	Generalization of the information ratio of Neu et al. [2022] to partial context in S_t
$\mathbb{I}_t$	Mutual information conditional on $\mathcal{F}_t$
ϵ_t	Gap between the outcome entropy for the selected subset S_t and the full context
α	Rate of exponential decay of ϵ_t
ϵ_t^q	Gap between outcome entropy for selected q^{th} follow-up question for user u_t and the optimal entropy minimizing question
$p_{t,q}$	Probability distribution conditional on $\mathcal{F}_t$ and user context observed prior to the qth question
$\mathbb{P}$	Probability measure

7 Theoretical Results

Lemma 7.1 (Equivalence of Action and Outcome Entropy)**.** *For a categorical distribution over outcomes:*

$$\arg\min_S \mathbb{H}_t(Y_t|X_{t,S}) = \arg\min_S \mathbb{H}_t(A_t|X_{t,S})$$

Proof. The model selects actions with the same likelihood as the expected outcome

$$p_t(A_t = a|X_{t,S}) = p_t(Y_{t,a} = 1|X_{t,S})$$

Let $e_a \in \{0,1\}^{\mathcal{A}}$ denote the one-hot vector that is one at index a.

$$\begin{aligned}
\mathbb{H}_t(Y_t|X_{t,S}) &= -\sum_{y \in \{0,1\}^{\mathcal{A}}} p_t(Y_t = y|X_{t,S}) \log(p_t(Y_t = y|X_{t,S})) \\
&= -\sum_{y \in \{e_a|a \in \mathcal{A}\}} p_t(Y_t = y|X_{t,S}) \log(p_t(Y_t = y|X_{t,S})) \quad \text{(one only correct action)} \\
&= -\sum_{a \in \mathcal{A}} p_t(Y_{t,a} = 1|X_{t,S}) \log(p_t(Y_{t,a} = 1|X_{t,S})) \\
&= \sum_{a \in \mathcal{A}} p_t(A_t = a|X_{t,S}) \log(p_t(A_t = a|X_{t,S})) \\
&= \mathbb{H}_t(A_t|X_{t,S})
\end{aligned}$$

□

Lemma 7.2 (Upper Bound on Total Variation)**.** *For any $\epsilon > 0$, we have*

$$\mathbb{H}_t(Y_t|X_{t,S_t}) - \mathbb{H}_t(Y_t|X_t) \le \epsilon \Rightarrow ||p_t(Y_t|X_{t,S_t}) - p_t(Y_t|X_t)||_1 \le \sqrt{\frac{\epsilon}{2}}$$

In particular, $\mathbb{H}_t(Y_t|X_{t,S_t}) = \mathbb{H}_t(Y_t|X_t) \Rightarrow p_t(Y_t|X_{t,S_t}) = p_t(Y_t|X_t)$

Proof.

$$\begin{aligned}
||p_t(Y_t|X_{t,S_t}) - p_t(Y_t|X_t)||_1 &\le \sqrt{\frac{1}{2} D_{KL}(p_t(Y_t|X_{t,S_t})||p_t(Y_t|X_t))} \quad \text{(Pinsker's Inequality)} \\
&= \sqrt{\frac{1}{2} D_{KL}(p_t(Y_t|X_{t,S_t})||p_t(Y_t|X_{t,S_t}, X_{t,S_t^-}))} \\
&= \sqrt{\frac{1}{2} \mathbb{I}_t(Y_t; X_{t,S_t^-}|X_{t,S_t})} = \sqrt{\frac{1}{2}\left(\mathbb{H}_t(Y_t|X_{t,S_t}) - \mathbb{H}_t(Y_t|X_{t,S_t}, X_{t,S_t^-})\right)} \\
&= \sqrt{\frac{1}{2}\left(\mathbb{H}_t(Y_t|X_{t,S_t}) - \mathbb{H}_t(Y_t|X_t)\right)} \le \sqrt{\frac{\epsilon}{2}}
\end{aligned}$$

For $\epsilon = 0$, we have a simpler argument:

$$\begin{aligned}
S_t = \arg\min_S \mathbb{H}_t(Y_t|X_{t,S}) &\Rightarrow \mathbb{H}_t(Y_t|X_{t,S_t}) = \mathbb{H}_t(Y_t|X_t) \\
&\Rightarrow Y_t \perp X_t \mid X_{t,S_t} \Rightarrow p_t(Y_t|X_{t,S_t}) = p_t(Y_t|X_t)
\end{aligned}$$

In particular, this also holds for the smallest norm set of user features S_t^* that satisfies this condition.

□

Lemma 7.3. *Following the information-gathering procedure and posterior sampling for action selection, the following equalities hold under exact entropy minimization:*

$$\mathbb{P}_t(A_t^*, \theta^* \mid X_{t,S_t^*}) = \mathbb{P}_t(A_t, \theta_t \mid X_{t,S_t^*}) = \mathbb{P}_t(A_t^*, \theta^* \mid X_t) = \mathbb{P}_t(A_t, \theta_t \mid X_t)$$

Proof. Exact entropy minimization implies that the outcome distributions conditioned on the minimal sufficient context S_t^* match those conditioned on the full context by Lemma 3.3:

$$\mathbb{P}_t(Y_t \mid \theta_t, X_{t,S_t^*}) = \mathbb{P}_t(Y_t \mid \theta_t, X_t)$$

Since posterior sampling draws θ_t from the same distribution as θ^*, we also have:

$$\mathbb{P}_t(Y_t \mid \theta^*, X_{t,S_t^*}) = \mathbb{P}_t(Y_t \mid \theta^*, X_t)$$

Therefore, under both θ_t and θ^*, conditioning on X_{t,S_t^*} is equivalent to conditioning on X_t for predicting Y_t.

Because the action is selected deterministically to maximize the expected outcome, the distribution over actions must also be identical:

$$\mathbb{P}_t(A_t \mid \theta_t, X_{t,S_t^*}) = \mathbb{P}_t(A_t \mid \theta_t, X_t)$$
$$\mathbb{P}_t(A_t^* \mid \theta^*, X_{t,S_t^*}) = \mathbb{P}_t(A_t^* \mid \theta^*, X_t)$$

Finally, combining the equivalence of action distributions with that of θ under posterior sampling gives:

$$\mathbb{P}_t(A_t, \theta_t \mid X_{t,S_t^*}) = \mathbb{P}_t(A_t, \theta_t \mid X_t)$$
$$\mathbb{P}_t(A_t^*, \theta^* \mid X_{t,S_t^*}) = \mathbb{P}_t(A_t^*, \theta^* \mid X_t)$$

which proves the lemma. □

Lemma 7.4 (Bounded Information Ratio)**.** *Information ratio for observed set S_t defined as follows:*

$$\rho_t(S_t) = \frac{(\mathbb{E}_t\left[r_t(\theta_t, A_t, X_{t,S_t}) - r_t(\theta^*, A_t, X_{t,S_t})|X_{t,S_t}\right])^2}{\mathbb{I}_t\left(\theta^*; Y_{t,A_t}|A_t, X_{t,S_t}\right)}$$

admits the bound $\rho_t(S_t) \leq 2\sum_{a\in\mathcal{A}} \mathbb{E}\left[\bar{r}_t(a, X_{t,S_t})\right]$

Proof. Following the proof technique in Neu et al. [2022], we bound a quantity analogous to the instantaneous regret $\Delta_t = r_t(\theta_t, A_t, X_{t,S_t}) - r_t(\theta^*, A_t, X_{t,S_t})$.

It is easy to see that for $S_t \supseteq S_t^*$, this simplifies to the instantaneous regret:

$$\begin{aligned}\Delta_t &= r_t(\theta^*, A_t^*, X_{t,S_t^*}) - r_t(\theta^*, A_t, X_{t,S_t^*}) \\ &\quad \text{(under posterior sampling conditional distribution } (\theta^*, A_t^*) \text{ matches } (\theta_t, A_t)) \\ &= r_t(\theta_t^*, A_t^*, X_t) - r_t(\theta^*, A_t, X_t) \\ &\quad \text{(Using } \mathbb{P}_t(Y_{t,A_t^*}|\theta^*, X_{t,S_t^*}) = \mathbb{P}_t(Y_{t,A_t^*}|\theta^*, X_t))\end{aligned}$$

Define $\bar{r}_t(a_t, X_{t,S_t^*}) = \mathbb{E}_t[r_t(\theta^*, a_t, X_{t,S_t^*})]$ and $g(p|q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$.

The Fenchel-Young inequality states: $f(x) + f^*(y) \geq \langle x, y\rangle$, where f^* is the convex conjugate of f: $f^*(a) = \sup_{x\in\mathcal{X}}\{\langle a, x\rangle - f(x)\}$. Hence, $g^*(u||q) = \sup_{p\in[0,1]}\{pu - g(p||q)\} = \log(1 + q(e^u - 1)) \leq q\left(u + \frac{u^2}{2}\right)$.

$$\begin{aligned}\mathbb{E}_t\left[\Delta_t|X_{t,S_t}\right] &= \mathbb{E}_{t,X_{t,S_t}}\left[r_t(\theta_t, A_t, X_{t,S_t}) - r_t(\theta^*, A_t, X_{t,S_t})\right] \\ &= \mathbb{E}_{t,X_{t,S_t}}\left[r_t(\theta_t, A_t, X_{t,S_t}) - \bar{r}_t(A_t, X_{t,S_t})\right] \\ &= \mathbb{E}_{t,X_{t,S_t}}\left[\sum_{a\in\mathcal{A}} \mathbb{1}[A_t = a]\frac{\eta p_t(A_t = a|X_{t,S_t})}{\eta p_t(A_t = a|X_{t,S_t})} r_t(\theta_t, a, X_{t,S_t}) - \bar{r}_t(A_t, X_{t,S_t})\right] \\ &= \mathbb{E}_{t,X_{t,S_t}}\left[\eta\sum_{a\in\mathcal{A}} p_t(A_t = a|X_{t,S_t})\left(\frac{\mathbb{1}[A_t = a]}{\eta p_t(A_t = a|X_{t,S_t})} r_t(\theta_t, a, X_{t,S_t})\right) - \bar{r}_t(A_t, X_{t,S_t})\right] \\ &\leq \mathbb{E}_{t,X_{t,S_t}}\left[\eta\sum_{a\in\mathcal{A}} p_t(A_t = a|X_{t,S_t})\Bigg(g\left(r_t(\theta_t, a, X_{t,S_t})||\bar{r}_t(A_t, X_{t,S_t})\right)\right.\end{aligned}$$

$$
\begin{aligned}
&+ g^*\left(\frac{\mathbb{1}[A_t = a]}{\eta p_t(A_t = a|X_{t,S_t})}||\bar{r}_t(A_t, X_{t,S_t})\right)\Bigg) - \bar{r}_t(A_t, X_{t,S_t})\Bigg] \\
&\leq \mathbb{E}_{t,X_{t,S_t}}\Bigg[\eta \sum_{a\in\mathcal{A}} p_t(A_t = a|X_{t,S_t})\Bigg(g\left(r_t(\theta_t, a, X_{t,S_t})||\bar{r}_t(A_t, X_{t,S_t})\right) \\
&+ \frac{\mathbb{1}[A_t = a]}{\eta p_t(A_t = a|X_{t,S_t})}\bar{r}_t(A_t, X_{t,S_t}) + \frac{\mathbb{1}[A_t = a]}{2\eta^2 p_t^2(A_t = a|X_{t,S_t})}\bar{r}_t(A_t, X_{t,S_t})\Bigg) - \bar{r}_t(A_t, X_{t,S_t})\Bigg] \\
&= \mathbb{E}_{t,X_{t,S_t}}\left[\eta \sum_{a\in\mathcal{A}} p_t(A_t = a|X_{t,S_t}) g\left(r_t(\theta_t, a, X_{t,S_t})||\bar{r}_t(A_t, X_{t,S_t})\right) + \frac{1}{2\eta}\sum_{a\in\mathcal{A}} \bar{r}_t(a, X_{t,S_t})\right] \\
&= \mathbb{E}_{t,X_{t,S_t}}\left[\eta \sum_{a\in\mathcal{A}} p_t(A_t = a|X_{t,S_t}) g\left(r_t(\theta^*, a, X_{t,S_t})||\bar{r}_t(A_t, X_{t,S_t})\right) + \frac{1}{2\eta}\sum_{a\in\mathcal{A}} \bar{r}_t(a, X_{t,S_t})\right] \\
&= \eta \mathbb{I}_t\left(\theta^*; Y_{t,A_t}|A_t, X_{t,S_t}\right) + \frac{1}{2\eta}\sum_{a\in\mathcal{A}} \mathbb{E}_t\left[\bar{r}_t(a, X_{t,S_t})\right]
\end{aligned}
$$

Choosing the value of $\eta > 0$ which minimizes the above expression, we obtain

$$
\mathbb{E}_t\left[\Delta_t|X_{t,S_t}\right] \leq \sqrt{2\mathbb{I}_t\left(\theta^*; Y_{t,A_t}|A_t, X_{t,S_t}\right)\sum_{a\in\mathcal{A}} \mathbb{E}_t\left[\bar{r}_t(a, X_{t,S_t})\right]}
$$

Rearranging terms, we get $\rho_t(S_t) = \frac{(\mathbb{E}_t\left[\Delta_t|X_{t,S_t}\right])^2}{\mathbb{I}_t\left(\theta^*; Y_{t,A_t}|A_t, X_{t,S_t}\right)} \leq 2\sum_{a\in\mathcal{A}} \mathbb{E}_t\left[\bar{r}_t(a, X_{t,S_t})\right]$ □

The following Lemma is a precursor to the bound on conditional mutual information (Lemma 7.6)

Lemma 7.5 (Recursive Property of Posterior Sampling)**.** $\prod_{t=1}^T \sum_\theta q_t(\theta) p_t(Y_{t,A_t}|\theta, A_t, X_{t,S_t}) = \sum_\theta q_1(\theta) \prod_{t=1}^T p_t(Y_{t,A_t}|\theta, A_t, X_{t,S_t})$

Proof. We repeat the proof in Neu et al. [2022] for easy reference:

From posterior sampling, $q_{t+1}(\theta) = \frac{q_t(\theta)p_t(Y_{t,A_t}|\theta,A_t,X_{t,S_t})}{\sum_{\theta'} q_t(\theta')p_t(Y_{t,A_t}|\theta',A_t,X_{t,S_t})}$

$$
\begin{aligned}
\sum_\theta q_1(\theta)\prod_{t=1}^T p_t(Y_{t,A_t}|\theta, A_t, X_{t,S_t}) &= \prod_{t=1}^T \frac{\sum_\theta q_1(\theta)\prod_{k=1}^t p_k(Y_{k,A_k}|\theta, A_k, X_{k,S_k})}{\sum_{\theta'} q_1(\theta')\prod_{k=1}^{t-1} p_k(Y_{k,A_k}|\theta, A_k, X_{k,S_k})} \\
&= \prod_{t=1}^T \sum_\theta q_t(\theta) p_t(Y_{t,A_t}|\theta, A_t, X_{t,S_t})
\end{aligned}
$$

□

Lemma 7.6 (Bounded Conditional Mutual Information)**.** *The cumulative mutual information for any observation set S_t admits the following bound:* $\mathbb{E}\left[\sum_{t=1}^T \mathbb{I}_t(\theta^*, Y_{t,A_t}|A_t, X_{t,S_t})\right] \leq \mathbb{H}(\theta^*)$

Proof. We generalize the proof of Neu et al. [2022] to an arbitrary subset of features S_t. Let $q_{t+1}(\theta) = \frac{q_t(\theta)p_t(Y_{t,A_t}|\theta,A_t,X_{t,S_t^*})}{\sum_{\theta'} q_t(\theta')p_t(Y_{t,A_t}|\theta',A_t,X_{t,S_t^*})}$ be the posterior sampling update of our belief of the value of optimal parameter θ^*. $q_1(\theta)$ is our initial prior before any interactions.

$$
\mathbb{E}\left[\sum_{t=1}^T \mathbb{I}_t(\theta^*; Y_{t,A_t}|A_t, X_{t,S_t})\right] = \mathbb{E}\left[\sum_{t=1}^T \mathbb{E}_t D_{KL}\left[p_t(Y_{t,A_t}|\theta^*, A_t, X_{t,S_t}) \mid p_t(Y_{t,A_t}|A_t, X_{t,S_t})\right]\right]
$$

$$
\begin{aligned}
&= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}_t\left[\log \frac{p_t(Y_{t,A_t}|\theta^*, A_t, X_{t,S_t})}{p_t(Y_{t,A_t}|A_t, X_{t,S_t})}\right]\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}_t\left[\log \frac{p_t(Y_{t,A_t}|\theta^*, A_t, X_{t,S_t})}{\sum_\theta q_t(\theta) p_t(Y_{t,A_t}|\theta, A_t, X_{t,S_t})}\right]\right] \\
&= \mathbb{E}\left[\log \frac{\prod_{t=1}^{T} p_t(Y_{t,A_t}|\theta^*, A_t, X_{t,S_t})}{\sum_\theta q_1(\theta) \prod_{t=1}^{T} p_t(Y_{t,A_t}|\theta, A_t, X_{t,S_t})}\right] \text{(by Lemma 7.5)} \\
&\leq \mathbb{E}\left[\log \frac{\prod_{t=1}^{T} p_t(Y_{t,A_t}|\theta^*, A_t, X_{t,S_t})}{q_1(\theta^*) \prod_{t=1}^{T} p_t(Y_{t,A_t}|\theta^*, A_t, X_{t,S_t})}\right] \\
&= \mathbb{E}[-\log(q_1(\theta^*))] = \mathbb{H}(\theta^*)
\end{aligned}
$$

□

Theorem 7.7 (Regret of Exact Entropy Minimization)**.** *Exact entropy minimization achieves the regret bound:* $R_T \leq \sqrt{2T\mathbb{H}(\theta^*)}$

Proof. In the exact entropy minimization setting we assume $S_t \supseteq S_t^*$, which is the smallest entropy-minimizing subset of user features, and therefore, the information ratio $\rho_t(S_t) = \rho_t(S_t^*)$ can simply be denoted as ρ_t.

$$
\rho_t = \frac{\left(\mathbb{E}_t\left[r_t(\theta^*, A_t^*, X_t) - r_t(\theta^*, A_t, X_{t,S_t^*})\right]\right)^2}{\mathbb{I}_t(\theta^*; Y_{t,A_t}|A_t, X_{t,S_t^*})}
$$

The cumulative regret of the decision maker can be bounded

$$
\begin{aligned}
R_T &= \mathbb{E}\left[\sum_{t=1}^{T}(r_t(\theta^*, A_t^*, X_t) - r_t(\theta^*, A_t, X_{t,S_t^*}))\right] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}_t\left[r_t(\theta^*, A_t^*, X_t) - r_t(\theta^*, A_t, X_{t,S_t^*})\right]\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \sqrt{\rho_t \mathbb{I}_t(\theta^*; Y_{t,A_t}|A_t, X_{t,S_t^*})}\right] \leq \sqrt{\mathbb{E}\left[\sum_{t=1}^{T} \rho_t\right] \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}_t(\theta^*; Y_{t,A_t}|A_t, X_{t,S_t^*})\right]}
\end{aligned}
$$

The last step follows by Cauchy-Schwarz Inequality. We proceed by deriving bounds for each of $\mathbb{E}\left[\sum_{t=1}^{T} \rho_t\right]$ and $\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}_t(\theta^*; Y_{t,A_t}|A_t, X_{t,S_t^*})\right]$.

To bound $\mathbb{E}\left[\sum_{t=1}^{T} \rho_t\right]$, we invoke Lemma 3.5:

$$
\rho_t \leq 2\mathbb{E}\left[\sum_{a \in \mathcal{A}} \bar{r}_t(a, X_{t,S_t})\right] = 2 \quad \text{(categorical distribution over actions)}
$$

Next, we invoke Lemma 7.6 which says $\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}_t(\theta^*, Y_{t,A_t}|A_t, X_{t,S_t^*})\right] \leq \mathbb{H}(\theta^*)$.

Combining these two bounds we obtain:

$$
R_T \leq \sqrt{\mathbb{E}\left[\sum_{t=1}^{T} \rho_t\right] \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}_t(\theta^*; Y_{t,A_t}|A_t, X_{t,S_t^*})\right]} \leq \sqrt{2T\mathbb{H}(\theta^*)}
$$

□

Theorem 7.8 (Regret)**.** *Approximate entropy minimization achieves the regret bound:* $R_T \leq \sqrt{2T\mathbb{H}(\theta^*)} + \mathcal{O}(T^{1-\alpha/2})$ *in the one-hot outcome setting.*

We assume that entropy minimization learns to select the optimal subset with diminishing error $\epsilon_t = \mathbb{H}_t(Y_t|X_{t,S_t}, \theta) - \mathbb{H}_t(Y_t|X_t, \theta), \mathbb{E}[\epsilon_t] = \mathcal{O}\left(\frac{1}{t^\alpha}\right)$. *In particular, for* $\alpha = 1$, *we obtain* $R_T = \mathcal{O}(\sqrt{T})$, *identical to the exact entropy minimization setting.*

Proof. Let us define $\Delta_t = r_t(\theta^*, A_t^*, X_t) - r_t(\theta^*, A_t, X_t)$ and $\tilde{\Delta}_t = r_t(\theta_t, A_t, X_{t,S_t}) - r_t(\theta^*, A_t, X_{t,S_t})$. We use the definition of information ratio from Lemma 3.5 and define $\tilde{\rho}_t = \rho_t(S_t)$:

$$\tilde{\rho}_t = \frac{(\mathbb{E}_t\left[\tilde{\Delta}_t | X_{t,S_t}\right])^2}{\mathbb{I}_t\left(\theta^*; Y_{t,A_t} | A_t, X_{t,S_t}\right)}$$

Define $\gamma_t = \mathbb{E}_t\left[\Delta_t | X_{t,S_t^*}\right] - \mathbb{E}_t\left[\tilde{\Delta}_t | X_{t,S_t}\right]$. Regret R_T can be decomposed as follows:

$$\begin{aligned}
R_T &= \mathbb{E}\left[\sum_{t=1}^T \Delta_t\right] = \mathbb{E}\left[\sum_{t=1}^T \mathbb{E}_t\left[\Delta_t | X_{t,S_t^*}\right]\right] = \mathbb{E}\left[\sum_{t=1}^T \mathbb{E}_t\left[\Delta_t | X_{t,S_t^*}\right] - \mathbb{E}_t\left[\tilde{\Delta}_t | X_{t,S_t}\right] + \mathbb{E}_t\left[\tilde{\Delta}_t | X_{t,S_t}\right]\right] \\
&= \mathbb{E}\left[\sum_{t=1}^T \gamma_t + \sqrt{\tilde{\rho}_t \mathbb{I}_t\left(\theta^*; Y_{t,A_t} | A_t, X_{t,S_t}\right)}\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^T \gamma_t\right] + \sqrt{\mathbb{E}\left[\sum_{t=1}^T \tilde{\rho}_t\right] \mathbb{E}\left[\sum_{t=1}^T \mathbb{I}_t\left(\theta^*; Y_{t,A_t} | A_t, X_{t,S_t}\right)\right]} \\
&\leq \mathbb{E}\left[\sum_{t=1}^T \gamma_t\right] + \sqrt{2T\mathbb{H}(\theta^*)}
\end{aligned}$$

Where we bound the second term in a similar manner to Theorem 3.6. $\mathbb{E}\left[\sum_{t=1}^T \tilde{\rho}_t\right] \leq 2T$ from Lemma 3.5. $\mathbb{E}\left[\sum_{t=1}^T \mathbb{I}_t\left(\theta^*; Y_{t,A_t} | A_t, X_{t,S_t}\right)\right] \leq \mathbb{H}(\theta^*)$ from Lemma 7.6.

Now we bound $\gamma_t = \mathbb{E}_t\left[\Delta_t | X_{t,S_t^*}\right] - \mathbb{E}_t\left[\tilde{\Delta}_t | X_{t,S_t}\right]$:

Note that, under the exact entropy search setting, we have $S_t = S_t^*$ which yields $\gamma_t = 0$.

$$\begin{aligned}
\gamma_t &= \mathbb{E}_t\left[\Delta_t | X_{t,S_t^*}\right] - \mathbb{E}_t\left[\tilde{\Delta}_t | X_{t,S_t}\right] \\
&= r_t(\theta^*, A_t^*, X_{t,S_t^*}) - r_t(\theta^*, A_t, X_{t,S_t^*}) - (r_t(\theta_t, A_t, X_{t,S_t}) - r_t(\theta^*, A_t, X_{t,S_t})) \\
&= r_t(\theta_t, A_t, X_{t,S_t^*}) - r_t(\theta^*, A_t, X_{t,S_t^*}) - (r_t(\theta_t, A_t, X_{t,S_t}) - r_t(\theta^*, A_t, X_{t,S_t})) \\
&\text{(Conditional distribution of } (\theta^*, A_t^*) \text{ and } (\theta_t, A_t) \text{ are same)} \\
&= r_t(\theta_t, A_t, X_{t,S_t^*}) - r_t(\theta_t, A_t, X_{t,S_t}) - (r_t(\theta^*, A_t, X_{t,S_t^*}) - r_t(\theta^*, A_t, X_{t,S_t}))
\end{aligned}$$

Let $\delta_t(\theta) = r_t(\theta, A_t, X_{t,S_t^*}) - r_t(\theta, A_t, X_{t,S_t}) = p_t(Y_{t,A_t} = 1 | X_{t,S_t^*}, \theta) - p_t(Y_{t,A_t} = 1 | X_{t,S_t}, \theta)$. Then, we have $\gamma_t = \delta_t(\theta_t) - \delta_t(\theta^*)$.

$$\begin{aligned}
|\delta_t(\theta)| &= |p_t(Y_{t,A_t} = 1 | X_{t,S_t^*}, \theta) - p_t(Y_{t,A_t} = 1 | X_{t,S_t}, \theta)| \\
&\leq \sup_{a \in \mathcal{A}} |p_t(Y_{t,a} = 1 | X_{t,S_t^*}, \theta) - p_t(Y_{t,a} = 1 | X_{t,S_t}, \theta)| \\
&= \sup_{Y_t} |p_t(Y_t | X_{t,S_t^*}, \theta) - p_t(Y_t | X_{t,S_t}, \theta)| \quad \text{(Total Variation (TV) distance)} \\
&\leq \sqrt{\frac{1}{2}\epsilon_t} \quad \text{(by Lemma 3.3)}
\end{aligned}$$

Where we assume that $\epsilon_t = \mathbb{H}_t(Y_t | X_{t,S_t}, \theta) - \mathbb{H}_t(Y_t | X_{t,S_t^*}, \theta)$ is bounded.

Now we derive a bound on $\mathbb{E}\left[\sum_{t=1}^T \gamma_t\right]$.

$$\mathbb{E}\left[\sum_{t=1}^T \gamma_t\right] = \mathbb{E}\left[\sum_{t=1}^T \delta_t(\theta_t) - \delta_t(\theta^*)\right] \leq \mathbb{E}\left[\sum_{t=1}^T |\delta_t(\theta_t)| + |\delta_t(\theta^*)|\right]$$

$$\leq \sqrt{2}\,\mathbb{E}\left[\sum_{t=1}^{T}\sqrt{\epsilon_t}\right]$$

Assuming that the entropy gap decays as $\mathbb{E}[\epsilon_t] = \mathcal{O}\left(\frac{1}{t^{\alpha}}\right)$, we achieve the following bound:

$$\mathbb{E}\left[\sum_{t=1}^{T}\gamma_t\right] = \mathcal{O}(\sum_{t=1}^{T}\frac{1}{t^{\alpha/2}}) = \mathcal{O}(T^{1-\alpha/2})$$

Replacing this in the regret bound we obtain:

$$R_T \leq \mathcal{O}(T^{1-\alpha/2}) + \sqrt{2T\mathbb{H}(\theta^*)} = \mathcal{O}(\sqrt{T}) \quad (\text{for } \alpha = 1)$$

□

Lemma 7.9 (Entropy Minimization Lowers Per-Question Suboptimality over ReAct)**.** *We define the per-question suboptimality of an algorithm as* $\epsilon_t^q = \mathbb{H}_t(Y_t|X_{t,S_t^{<q}}, X_{t,S_t^q}) - \mathbb{H}_t(Y_t|X_{t,S_t^{<q}}, X_{t,\bar{S}_t^q})$ *where* $\bar{S}_t^q$ *is the optimal* q^{th} *question that minimizes the outcome entropy. Let* $p_{t,q}^* = p_t(S_t^q = \bar{S}_t^q|\omega, X_{t,S_t^{<q}})$ *be the probability that the policy selects the optimal* q^{th} *follow-up question. For a budget of* $k \ll K$ *candidate questions in Entropy-Search (ES), we achieve lower suboptimality than the ReAct baseline:*

$$(\epsilon_t^q)_{ES} \leq (1 - p_{t,q}^*)^{k-1}(\epsilon_t^q)_{ReAct}$$

Proof. Let $\bar{S}_t^q = \arg\min_{S_t^q} \mathbb{H}_t(Y_t|X_{t,S_t^{<q}}, X_{t,S_t^q})$ be the optimal next follow-up question to the user for a given observed context $X_{t,S_t^{<q}}$. The per-question suboptimality of an algorithm $\epsilon_t^q = \mathbb{H}_t(Y_t|X_{t,S_t^{<q}}, X_{t,S_t^q}) - \mathbb{H}_t(Y_t|X_{t,S_t^{<q}}, X_{t,\bar{S}_t^q})$ measures how much more entropy could have been minimized if the optimal question was selected.

Under the ReAct baseline, we sample a follow-up question $S_t^q \sim p_t(S_t^q|\omega, X_{t,S_t^{<q}})$. Let $p_{t,q}^* = p_t(S_t^q = \bar{S}_t^q|\omega, X_{t,S_t^{<q}})$ be the probability that the policy selects the optimal q^{th} follow-up question.

For compactness, let us use subscript (t,q) to denote conditioning over $(\mathcal{F}_t, X_{t,S_t^{<q}})$. Then the per-question suboptimality of ReAct can be re-written as $(\epsilon_t^q)_{\text{ReAct}} = \mathbb{H}_{t,q}(Y_t|X_{t,S_t^q}) - \mathbb{H}_{t,q}(Y_t|X_{t,\bar{S}_t^q})$.

$$\begin{aligned}
(\epsilon_t^q)_{\text{ReAct}} &= \mathbb{H}_{t,q}(Y_t|X_{t,S_t^q}) - \mathbb{H}_{t,q}(Y_t|X_{t,\bar{S}_t^q}) \\
&= \left(\sum_{S\neq\bar{S}_t^q} p_{t,q}(S_t^q = S|\omega)\mathbb{H}_{t,q}(Y_t|X_{t,S}) + (p_{t,q}^*)\mathbb{H}_{t,q}(Y_t|X_{t,\bar{S}_t^q})\right) - \mathbb{H}_{t,q}(Y_t|X_{t,\bar{S}_t^q}) \\
&= (1-p_{t,q}^*)\frac{\sum_{S\neq\bar{S}_t^q} p_{t,q}(S_t^q = S|\omega)\mathbb{H}_{t,q}(Y_t|X_{t,S})}{1-p_{t,q}^*} + (p_{t,q}^* - 1)\mathbb{H}_{t,q}(Y_t|X_{t,\bar{S}_t^q}) \\
&= (1-p_{t,q}^*)\mathbb{H}_{t,q}^{\text{sub}} + (p_{t,q}^* - 1)\mathbb{H}_{t,q}^{\text{min}} = (1-p_{t,q}^*)(\mathbb{H}_{t,q}^{\text{sub}} - \mathbb{H}_{t,q}^{\text{min}})
\end{aligned}$$

where $\mathbb{H}_{t,q}^{\text{sub}}$ is the expected outcome entropy in suboptimal cases, $\mathbb{H}_{t,q}^{\text{min}}$ is the minimum outcome entropy possible after selecting the qth follow-up question.

Now let us compare this with the per-question suboptimality for Entropy Search (ES). Under ES, we first sample k candidate follow-up questions $S_1, S_2, \cdots, S_k \sim p_{t,q}(S_t^q|\omega)$. Then amongst these candidates, we select the entropy minimizing question $S_t^q = \arg\min_{S\in\{S_1,S_2,\cdots,S_k\}} \mathbb{H}_{t,q}(Y_t|X_{t,S})$. Assume that the primary limiting factor for ES is the budget $k \ll K$: with probability $(1-p_{t,q}^*)^k$, the candidate set does not contain $\bar{S}_t^q$ which is the entropy minimizing solution. Then we have:

$$\begin{aligned}
(\epsilon_t^q)_{\text{ES}} &= \mathbb{H}_{t,q}(Y_t|X_{t,S_t^q}) - \mathbb{H}_{t,q}(Y_t|X_{t,\bar{S}_t^q}) \\
&\leq \left((1-p_{t,q}^*)^k\mathbb{H}_{t,q}^{\text{sub}} + (1-(1-p_{t,q}^*)^k)\mathbb{H}_{t,q}(Y_t|X_{t,\bar{S}_t^q})\right) - \mathbb{H}_{t,q}(Y_t|X_{t,\bar{S}_t^q})
\end{aligned}$$

(In suboptimal cases, ES still achieves lower entropy than ReAct as it minimizes entropy over the k subset)

$$= (1 - p^*_{t,q})^k (\mathbb{H}^{\text{sub}}_{t,q} - \mathbb{H}^{\text{min}}_{t,q})$$

Combining the results for per-turn suboptimality for ReAct and ES, we obtain:

$$(\epsilon^q_t)_{\text{ES}} \leq (1 - p^*_{t,q})^{k-1} (\epsilon^q_t)_{\text{ReAct}}$$

□

Discussion: Selecting optimal follow-up questions is crucial because observing the answer to these questions can help the model predict the correct action in fewer questions and more accurately. We observe in Lemma 3.8 that the per-question suboptimality of Entropy Search depends on the compute budget k in the exponent. Therefore, the more candidate questions we consider for inference-time search, the more likely we are to select the optimal follow-up question, as we would expect. Moreover, this bound assumes that both ES and ReAct share the same question-selection policy ω. However, we expect $p^*_{t,q}$ to grow much faster for ES than for ReAct since (1) inference time selection based on the entropy score $v^q_t(S) = -\mathbb{H}_t(Y_t \mid X_{t,S^{<q}_t}, X_{t,S})$ increases the frequency of selecting the correct follow-up question, and (2) $v^q_t(S)$ serves as a dense, per-step training signal to optimize ω, that directly maximizes $p^*_{t,q}$ whenever a correct follow-up question is selected in the candidate set. As a result, we expect ES to select optimal queries much more often than ReAct, leading to better actions and, therefore, better user outcomes.

8 Baseline Methods

Now we compare several baseline strategies for information gathering with language models, some of which we compare to ENTROPY SEARCH in our experiments (Section 4).

8.1 Random Sampling

A naive alternative is to start with random question selection and rely on observed outcomes to guide future decisions. This approach assumes that information-seeking behavior will emerge naturally through reward maximization over a sufficiently large number of interactions. While conceptually simple, such a strategy has notable drawbacks, particularly in its inefficiency due to the lack of prioritization of informative questions. To illustrate the limitations of random sampling, consider the following example: Let $B \ll K$ be the budget for total questions the agent is allowed to ask per user before performing an action, where K is the total number of features in the user's context. Assume that the user's reward model depends on a sparse subset of $d \ll K$ relevant features. The probability that random sampling selects all d relevant features in a single round is:

$$\mathbb{P}(|S_t \cap S^*_t| = d) = \frac{\binom{d}{d}\binom{K-d}{B-d}}{\binom{K}{B}},$$

where S^*_t is the minimal set of relevant context features. This probability becomes exponentially small for large K and small d. Consequently, the agent is unlikely to collect sufficient context to make optimal decisions, leading to a high likelihood of suboptimal outcomes, hence incurring linear regret. This sparse signal provided by random sampling exacerbates the challenges for reinforcement learning algorithms, which depend on meaningful feedback to improve their policies. Without effective strategies for sampling follow-up questions, the agent may fail to uncover the structure of the reward model, limiting the potential for learning and adaptation.

8.2 REACT

A natural improvement to random follow-up questions is to utilize the LLM agent's world knowledge to come up with meaningful and relevant questions. REACT is a popular agentic framework that prompts the model to *Reason then Act*. The agent is prompted with the user's questions and the context gathered through previous follow-up questions and is asked to produce Chain-of-Thought reasoning prior to selecting the next best question. This approach is better than random sampling, since the model can reason about what facts and features are relevant to the problem and selectively query those from the user. However, a key limitation of REACT is that the model is unable to look far ahead into the future and anticipate the user's potential responses to subsequent follow-up questions $S^{>q}_t$ while selecting the current follow-up question S^q_t. However, as we show in Section 3, such System-2 thinking is crucial for making optimal decisions and reliably estimating the outcome distribution. This is particularly important for problem domains where there are complex conditional interdependencies between different follow-up questions where some questions are strictly more informative than others and the answer to these questions can make other questions irrelevant. We demonstrate that explicitly searching over candidate questions significantly lowers the per-question suboptimality of ENTROPY SEARCH when compared to REACT in Lemma 7.9.

Now we discuss another Lemma that gives a lower bound on the suboptimality ϵ_t for REACT and ENTROPY SEARCH and we show that this gap decreases significantly faster for ENTROPY SEARCH.

Lemma 8.1 (Lower Bound on Suboptimality Gap)**.** *The entropy gap (suboptimality)* $\epsilon_t = \mathbb{H}_t(Y_t|X_{t,S_t}) - \mathbb{H}_t(Y_t|X_t)$ *admits the following lower bound for* REACT *and* ENTROPY SEARCH *under a budget of* B *follow-up questions with* R *underlying entropy-minimizing questions.*

$$\mathbb{E}[\epsilon_t^{\mathit{ReAct}}] \geq \mathbb{H}_{\mathit{sub}} \sum_{r=0}^{R-1} \binom{B}{r} p^r (1-p)^{B-r}$$
$$\mathbb{E}[\epsilon_t^{\mathit{ES}}] \geq \mathbb{H}_{\mathit{sub}} \sum_{r=0}^{R-1} \binom{B}{r} \bar{p}^r (1-\bar{p})^{B-r}$$

p *denotes the maximum probability that* REACT *selects a good follow-up question and* $\bar{p} = 1-(1-p)^k$ *is the corresponding upper bound on probability for* ENTROPY SEARCH *under a search budget of* k *candidate questions.* $\mathbb{H}_{\mathit{sub}}$ *denotes the minimum entropy gap over suboptimal trajectories that are missing one or more key questions about the user.*

Proof. Let B be the budget for maximum number of follow-up questions that can be asked to the user. Let R be the minimum number of queries needed to minimize the outcome entropy $\mathbb{H}_t(Y_t|X_{t,S_t})$. In general, there may be multiple queries that are equivalent in the sense that the corresponding context features are highly correlated and knowing one is as good as knowing another. Let $S_t^* = \arg\min_S \mathbb{H}_t(Y_t|X_{t,S})$ be an entropy-minimizing subset of questions. At any given turn q, the agent picks a question S_t^q, while $q \leq B$. A query S_t^q is entropy-reducing if it is equivalent to any of the queries in S_t^* and is not equivalent to any previously asked queries $S_t^{<q}$.

Let the probability that S_t^q is an entropy-minimizing query be $\leq p$. Then the probability that the set of questions asked by the agent $S_t \supseteq S_t^*$ (upto any equivalent questions) is $p_t(\epsilon_t = 0) \leq \sum_{r=R}^{B} \binom{B}{r} p^r (1-p)^{B-r}$. This follows from the fact that this is an increasing function of p and the actual probabilities of selecting an entropy reducing question at any given turn q are lower than p, which means $p_t(\epsilon_t = 0)$ would be lower.

Now let us turn our attention to the case where a suboptimal set of questions is asked within the budget B. This means that atleast one question from S_t^*or equivalent has not been asked by the agent. Let $\mathbb{H}_{\text{sub}} = \min_{S_t \not\supseteq S_t^*} \mathbb{H}_t(Y_t|X_{t,S_t}) = \min_{S_t \not\supseteq S_t^*} \epsilon_t$ be the minimum suboptimality gap in the instances where the agent does not cover all entropy minimizing questions. This happens with probability $\geq \sum_{r=0}^{R-1} \binom{B}{r} p^r (1-p)^{B-r}$ because this quantity is decreasing in p (intuitively, this is the CDF of a binomial distribution, and as p increases, the mass of the distribution shifts right, reducing the CDF). Thus, it follows that $\mathbb{E}[\epsilon_t^{\text{ReAct}}] \geq \mathbb{H}_{\text{sub}} \sum_{r=0}^{R-1} \binom{B}{r} p^r (1-p)^{B-r}$.

In contrast to REACT, the probability of selecting an entropy minimizing question is directly linked to the computational budget for search which determines how many candidates questions k are sampled from the agent policy $p_t(S_t^q|\omega, X_{t,S_t^{<q}})$. The probability that S_t^q is an entropy-minimizing query is $\leq \bar{p} = 1-(1-p)^k$. The corresponding bound on suboptimality for ENTROPY SEARCH is given by $\mathbb{E}_t[\epsilon_t^{\text{ES}}] \geq \mathbb{H}_{\text{sub}} \sum_{r=0}^{R-1} \binom{B}{r} \bar{p}^r (1-\bar{p})^{B-r}$.

We consider a simple case where $B = 50, R = 25$ and plot the upper bound on $p_t(\epsilon_t = 0)$ against p for different values of k. We can see in Figure 5 that even for moderate values of k, ENTROPY SEARCH achieves significantly lower ϵ_t (practically zero) even when REACT selects an entropy-minimizing question S_t^q with $p = 0.5$ probability. This shows that ENTROPY SEARCH effectively lowers $\mathbb{E}_t[\epsilon_t]$ by utilizing test-time compute. Moreover, we expect $\mathbb{E}_t[\epsilon_t]$ to approach zero much faster even with marginal improvements in p as we update ω which can be seen from Figure 5. □

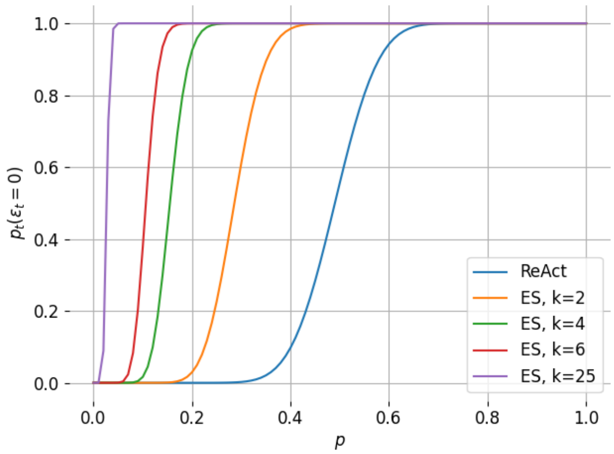

Figure 5: Upper bound, $p_t(\epsilon_t = 0) \leq \sum_{r=R}^{B} \binom{B}{r} p^r (1-p)^{B-r}$ vs p: REACT is $k = 1$, ENTROPY SEARCH is $k > 1$

8.3 REFLEXION

REFLEXION Shinn et al. [2023] builds on top of REACT and adds a dedicated reflection component before the reasoning block in REACT. The agent is explicitly prompted to reflect on what it has learned from the previous interactions and assess the effectiveness of its information gathering strategy. The prompt also includes past reflections for improved continuity between questions. As seen in Section 4, REFLEXION does slightly better than REACT due to added backward reflection. However, as we demonstrate with ENTROPY SEARCH, looking forward is crucial for optimal question selection.

9 Benchmarks

GSM8K-Q We utilized the GSM8K-Q benchmark from QuestBench [Li et al., 2025], which contains human-annotated grade school math problems designed with one missing variable assignment necessary for solving each problem. This benchmark provides clear, structured scenarios for evaluating the agent's ability to identify and acquire the minimal information required to resolve underspecified reasoning tasks. The controlled yet challenging nature of GSM8K-Q makes it ideal for rigorously assessing our algorithm's capability to ask targeted, informative questions.

1D-ARC The 1D-ARC benchmark [Xu et al., 2024], a widely studied adaptation of the Abstraction and Reasoning Corpus (ARC), is designed to evaluate abstract reasoning in language models. We further adapt it to test sequential information gathering by masking the output sequence. Each task presents an input sequence and a partially hidden output generated via transformations such as recoloring based on token frequency, repeating patterns, dynamic shifts, and conditional replacements. To uncover the transformation rule, the agent must query specific output indices, receiving exact values from an oracle. This setup provides a controlled and interpretable environment for evaluating how effectively an agent can reduce uncertainty and reason under partial information.

Fermi Problems The Fermi problems [Kalyan et al., 2021] evaluate algorithms through commonsense-based estimation tasks, such as estimating coffee consumption at EMNLP 2019 or the potential rise in sea levels from melting ice. Problems are designed with an oracle that provides answers from a mix of relevant and distractor facts, totaling 20 variables, but without indicating which facts are necessary. Solving Fermi problems requires recursively breaking down complex queries, abstracting details, and synthesizing multiple reasoning steps, aligning well with our interactive decision-making framework. These problems challenge even advanced fine-tuned models, which typically exhibit substantial errors; the best fine-tuned T5 model only reached a score of 0.23 [Kalyan et al., 2021]. To ensure the necessity for interactive information gathering, we filtered out easily solvable questions and manually removed inaccurately annotated ones, resulting in 1000 high-quality tasks from an initial pool of 5000. Figure 2 illustrates an example interaction, where the ENTROPY SEARCH agent selects informative questions to arrive at an accurate estimate.

10 Algorithms

Algorithm 1 Interactive Question Answering

procedure INTERACTIVEQA(Problem, User, LLMAgent)
  $H \leftarrow \{\}$ ▷ Instantiate context
  for b to budget **do**
    question q = LLMAgent(Problem, H)
    answer $a = \text{User}(q)$ ▷ Query user
    $H \leftarrow H \cup \{(q, a)\}$ ▷ Append to context
  end for
return action A = LLMAgent(Problem, H)
end procedure

Algorithm 2 Entropy Search

procedure ENTROPYSEARCH(problem, history H)
  $\mathcal{H} \leftarrow \{\}$
  $\mathcal{Q}$ = LLMAgent.sample(problem, sample size = k, H) ▷ Sample k questions
  for q in $\mathcal{Q}$ **do**
    answers = LLMAgent.sample(q, sample size, history) ▷ Sample answers
    for answer in answers **do**
      actions = LLMAgent.sample((q, answer), sample size, history)
    end for
    $\mathcal{H} \leftarrow \mathcal{H} \cup$ {ComputeAvgEntropy(actions | q, answer)} ▷ Compute using Lemma 3.2
  end for
  return $\arg\min_{\mathcal{Q}} \mathcal{H}$
end procedure

11 Other Related Works

Our work integrates ideas from contextual bandits and experiment design in a novel manner. In the following subsections, we examine relevant threads in prior art, including derived settings such as survey bandits and combinatorial bandits. We conclude by explaining how these ideas come together in ENTROPY SEARCH.

11.1 Experiment Design

Experimental design addresses the problem of selecting the most informative set of measurements to reveal a hidden quantity of interest. Prior work has approached this task from a decision-theoretic perspective [Chaloner and Verdinelli, 1995, DeGroot, 1962]. A central tool in this literature is Shannon information (entropy), which is equivalent to the expected KL divergence between the prior and the posterior after incorporating new information from the experiment. This criterion, also known as Expected Information Gain (EIG), has been widely adopted to select optimal experiments. It motivates our use of entropy minimization as the objective for selecting the next follow-up question in ENTROPY SEARCH.

11.2 Contextual Bandits

Contextual bandits are a class of Multi-Armed Bandit (MAB) problems where the decision maker observes a context X_t before selecting an action A_t and receiving the corresponding outcome Y_t [Agrawal and Goyal, 2013]. The decision maker leverages both the current context and the history of past interactions—i.e., tuples $(X_j, A_j, Y_j)_{j<t}$—to estimate the optimal action. Thompson sampling is a popular strategy in this setting, known for its strong theoretical guarantees [Agrawal and Goyal, 2013]. Neu et al. [2022] extend the information-theoretic analysis of regret developed in Russo and Van Roy [2016] to the contextual bandit case. Their key contribution is the notion of an information ratio, quantifying the trade-off between incurred regret and knowledge gained about the optimal action. Our work builds on this idea by introducing an analogous quantity in our setting (Lemma 3.5).

11.3 Combinatorial Bandits

Combinatorial bandit problems involve selecting combinations of arms, where the reward depends on the joint set of chosen arms [Durand and Gagné, 2014]. This is a more general setting than our information-gathering problem, as the outcome can be an arbitrary function of the selected arm features. In contrast, our setting has a key property: outcome uncertainty necessarily decreases as more features are added to the observed context subset. This enables the use of greedy selection strategies, such as ENTROPY SEARCH, which are significantly more tractable than general combinatorial optimization.

11.4 Survey Bandits

The survey bandit framework introduced by Krishnamurthy and Athey [2020] studies how to gather subsets of context before identifying the best arm in contextual bandits. Earlier, Bouneffouf et al. [2017] proposed a related problem—contextual bandits with restricted context—where the agent selects a fixed-size subset of the user context prior to choosing an arm. While inspired by combinatorial bandits, their setting imposes additional assumptions on the reward distribution to make subset selection tractable. Related efforts in personalization have also explored feature relevance in contextual bandits [Tomkins et al., 2021, Yao et al., 2021]. However, a key limitation of these approaches is their reliance on strong structural assumptions about the context or reward, which restricts general applicability.

11.5 Combining Ideas

While prior work in sequential experimental design has focused on selecting sets of experiments to maximize information about outcomes, this approach has not been rigorously applied to the survey bandit setting. In reinforcement learning, information gain has been used as an intrinsic reward to encourage exploration [Houthooft et al., 2016, Mohamed and Jimenez Rezende, 2015]. To our knowledge, ENTROPY SEARCH is the first method to formalize entropy-based question selection in online reinforcement learning, particularly in the survey bandit context.

12 Chain-of-Though & Min-Entropy Search Prompt

12.1 ARC-1D:

Task description:

```
You are solving a binary sequence puzzle. Problem: You have a {n}-bit input sequence: {input-seq}. A hidden output sequence was created by applying an unknown number of right circular shifts. A right circular shift moves the last bit to front and shifts all others right.
```

The previously gathered information follows this segment of the prompt. Here, we choose to include the previously gathered context in the initial prompt rather than in the chat history. We include this as follows:

```
Known output bits: {"(new line)".join(f"- Index {q} -> Value {v}" for q, v in seen)}
```

For Chain-of-Thought:

```
Instructions: Do not query the indices for which output is known.
You have EXACTLY two options:

If you have sufficient information to determine the number of right circular shifts:
1) Submit your final answer in this format:
Reasoning: <step-by-step reasoning>
Answer: <number of right circular shifts to transform the input into the output>

Else if you need to gather more information to find the number of right circular shifts:
2) Ask a query in this format:
Reasoning: <step-by-step reasoning>
Query: <index 0-{n-1}>

Use EXACTLY one of these two formats, and NOTHING ELSE (no markdown, no extra text).
```

For MES:

```
Instructions: Do not query the indices for which output is known.
You have EXACTLY two options:

If you have sufficient information to determine the number of right circular shifts:
```

```
1) Submit your final answer in this format:
Reasoning: <step-by-step reasoning>
Answer: <number of right circular shifts to transform the input into the output>

Else if you need to gather more information to find the number of right circular shifts:
2) Ask a query in this format:
Reasoning: <step-by-step reasoning>
Query: <index 0-{n-1}>
Value: <possible output value at the queried index>
Answer: <number of right circular shifts to transform the input into the output>

Use EXACTLY one of these two formats, and NOTHING ELSE (no markdown, no extra text).
```

12.2 Fermi

Both approaches have shared an initial prompt segment describing the task

```
You are a methodical math problem solver. Your goal is to solve the word problem below. You have the problem text, a history of questions you previously asked and the answers you received, and a list of variables that can be related to the problem.

**Problem Text**
Question: How many individual life forms are there on Earth?

**Variable list**
Variables:{"F1": "number of timezones where it is meal time at any given point.", "F2": "total number of animals living in inhabitable areas.", "F3": "average number of people living in a single timezone.}

**History of Questions Asked and Answers Received:
--- Interaction 1 ---
Your Question: What is the total animal population (F13) on Earth?
Answer Received: The total animal population on Earth is 2e+19.
--- End of History ---**
```

Following the problem context and past history, the agent needs to decide which variable to query based on CoT and MES Algorithm

For CoT:

```
**Your Task:**
1.  Analyze the problem text and the interaction history carefully.
2.  Reason step-by-step towards solving the problem. Show your thinking process clearly.
3.  Use the information from the problem text and any answers received from previous questions.
4.  Determine if you can calculate the final answer based on your current understanding.

**Choose EXACTLY ONE of the following actions:**

A) If you can calculate the final answer:
   Provide your step-by-step Chain-of-Thought reasoning, explaining how you derived the answer using the problem text and previous answers received.
   **Output Format (Required):**
   Reasoning: <Your step-by-step derivation>
   FinalAnswer: <The final numerical answer>

B) If you are stuck or need more specific information to proceed:
   Identify the *single, specific question* you need to ask to get the necessary clarification or missing piece of information required for your reasoning.
   Provide reasoning explaining *why* you need to ask this question based on your current reasoning progress.
   **Output Format (Required):**
   Reasoning: <Explain why this question is necessary for your next step>
   QueryOracle: <Your single, specific question>

**Important:**
- Perform calculations step-by-step in your reasoning.
- Ask focused, specific questions based on gaps in your understanding or missing information identified in your reasoning.
- Do not ask for information you can deduce yourself from the text or history.
- Use only one output format (Reasoning + FinalAnswer OR Reasoning + QueryOracle).
- for large number that needs scientific notation, use scientific notation in format Xe+Y , DO NOT DO 10^{-10}

e.g.  \(8.26 \times 10^{6}\) meters should be written as 8.26e+6

Begin your Chain-of-Thought reasoning and determine your next action.
```

For MES, We have different prompt for two steps: Query step and rollout step:

Query step:

```
**Instructions:**
1. Review the problem context (text, variables) and interaction history.
2. think of what is the formula involving the variables to compute the end result
3. Identify variables whose values are *missing* but are *required* to solve the problem based on the equations.
3. **Do NOT ask about any variable that has already been asked in the interaction history - regardless of whether it was answered**
4. For each new question:
    * Provide brief reasoning explaining *why* this specific value is needed for the calculation based on the equations.
    * Avoid asking for the same variable again.
    * Make a reasonable estimation for the variable
    * Ask concise, clear natural language questions (e.g., "What is the value of N?", "How many customers were there?").
5. Output *only* the JSON object and conform exactly to the provided schema.

**IMPORTANT: Your response MUST be valid parseable JSON that exactly matches the schema below. DO NOT include any explanations or text outside the JSON structure.**
- The response must start with {{ and end with }}
**Required JSON Schema:**
{schema_json}
Generate the JSON object:
```

Rollout step:

```
**Your Task & Decision Logic:**

1.  **Analyze:** Review all provided information (Problem, History, Hypothetical Answer)
2.  **Decide:**
    *   **Scenario 1: Numeric Answer Possible**
        *   If you *can* calculate a final numeric answer:
            *   Perform the calculation step-by-step in your reasoning.
            *   Your primary output is the **numeric value**.
    *   **Scenario 2: Numeric Answer NOT Possible**
        *   If you *cannot* calculate a final numeric answer (because some values are still unknown even with the hypothetical answer):
            *   Explain *why* a numeric calculation isn't possible in your reasoning.
            *   Derive the most simplified **symbolic expression** for the answer based on knowns and unknowns.
            *   Your primary output is the **symbolic expression**.

**Output Format (Based on Decision):**

*   **`simulated_reasoning` (Required for BOTH scenarios):**
    *   Provide clear, step-by-step reasoning.
    *   If Scenario 1: Show the derivation and calculation leading to the numeric answer.
    *   If Scenario 2: Explain the derivation of the symbolic expression and state why a numeric value cannot be determined.

*   **`symbolic_expression` (Required for BOTH scenarios):**
    *   Provide the final symbolic formula representing the answer.
    *   Must include only variables and Python/SymPy-style operators (`+`, `-`, `*`, `/`, `**`).
    *   No assignment (`=`), units, or descriptive text. Just the pure expression.
    *   *Even if you provide a `numeric_value`, this field should contain the underlying symbolic formula used.*
    *   Correct Examples: `N * A`, `(X + Y) / Z`, `1800 - 900 - C - 2*R`
    *   Wrong Examples: `Cost = X + Y`, `X = 5 + N`, `Final = 40`

*   **`numeric_value` (Conditional):**
    *   **Scenario 1:** Provide the **single calculated number** (int or float).
        *   Correct Examples: `40`, `12.5`, `-100`, `40.0`
        *   **CRITICALLY WRONG Examples (NEVER USE THESE):** `"40"`, `"7 + 3*7 + 12"`, `7 + 3*7 + 12`, `[40]`, `10 * 4`
        *  for large number that needs scientific notation, use scientific notation in format Xe+Y , DO NOT DO 10^{{-10}}
            e.g.  \(8.26 \times 10^{{6}}\) meters should be written as 8.26e+6
    *   **Scenario 2:** Set this field to **`null`**.


**IMPORTANT: Your response MUST be valid parseable JSON that exactly matches the schema below. DO NOT include any explanations or text outside the JSON structure.**
The response must start with {{ and end with }}

**Required JSON Schema:**
{schema_json}
Generate the JSON object:
```

12.3 GSM8K

Stop condition:

```
You are a methodical math problem solver using Chain-of-Thought reasoning. Your goal is to solve the word problem below. You have the problem text and a history of questions you previously asked and the answers you received. You have access to the original problem, all variable definitions, all equations, and the history of your interactions with an Oracle (questions you asked and answers you received).

**Problem Context:**
{problem_context}

**Full Interaction History (including latest Oracle answer):**
{history_string}

**Your Task:**
Based on ALL the information currently available (problem text, variable definitions, equations, and the entire interaction history), critically assess if you have sufficient information to calculate a definitive, single numerical answer to the problem.

1.  **Reasoning:** Explain your thought process.
    *   Consider the goal of the problem.
    *   Review the necessary equations.
    *   Check if all variables in those equations either have known values (from context or Oracle answers) or can be derived from other knowns.
2.  **Decision (`can_solve_now`):**
    *   Set to `true` if you are confident you can now proceed to calculate the final numerical answer without needing further Oracle input.
    *   Set to `false` if you believe crucial numerical information is still missing, or if there's critical ambiguity in definitions/equations that prevents a final numeric solution.

Output your decision strictly as a JSON object conforming to the schema below.
```

Query Step:

```
# ROLE: Strategic Question Generator for Math Problems
You are analyzing a math word problem to identify the *most useful questions* an agent could ask an Oracle to obtain **missing numerical values** required to solve the problem based on the provided equations and variables.
**Problem Context:**
{problem_context}

**Interaction History (Up to Query):**
{history_string}

**Instructions:**
1. Review the problem context (text, variables, equations) and interaction history.
2. Identify variables whose values are *missing* but are *required* to solve the problem based on the equations.
3. **Do NOT ask about any variable that has already been asked in the interaction history - regardless of whether it was answered - with only one exception:**
   - You may ask again about a previously asked variable **only if you are certain that obtaining this one value will allow you to complete the problem.** This exception should be used sparingly and only when absolutely justified.
4. For each new question:
   * Provide brief reasoning explaining *why* this specific value is needed for the calculation based on the equations.
   * Use 'value' as the query type unless clarification of a symbol is essential (in that case, use 'definition' or 'equation').
   * Avoid asking for the same variable again.
   * Ask concise, clear natural language questions (e.g., "What is the value of N?", "How many customers were there?").
5. Output *only* the JSON object and conform exactly to the provided schema.
```

Symbolic equation generation step:

```
# ROLE: Symbolic Problem Solver (Initial Expression)

You are tasked with deriving an initial symbolic expression to solve a math word problem.
You have been given a problem context, the actual interaction history with an Oracle so far,
and a hypothetical first query to the Oracle along with its hypothetical answer for THIS simulation.

**Problem Context:**
{problem_context}

```

```
**Actual Interaction History (Overall Agent-Oracle):**
{base_interaction_history}

Question Asked: {initial_query_text}

**Instructions:**
1.  Incorporate the hypothetical Oracle answer into your understanding, along with the
    ↪ actual history.
2.  Perform step-by-step reasoning to derive a **single, final symbolic mathematical
    ↪ expression** that represents the solution to the problem.
3.  This expression may contain known numbers and variable names from the problem context
    ↪  or history.
4.  Use standard Python/SymPy math notation (‘+‘, ‘-‘, ‘*‘, ‘/‘, ‘**‘).
5.  Use the variable in the initial_query_text to create a symbolic expression.
6.  **DO NOT evaluate to a number yet, even if you think you can.** The goal is the
    ↪ symbolic form.
7.  Output ONLY the JSON object strictly following the schema.

Example Output: ‘(A * B) + C - D/2‘

Generate the Symbolic Expression:
```

Rollout step:

```
# ROLE: Math Problem Solver (Internal Hallucination Step)

You are in a multi-step simulation to solve a math problem.
This simulation started after a hypothetical query ('{initial_query_text}') was
    ↪ hypothetically answered ('{initial_hypothetical_answer}').
You are now at an internal step of this simulation.

**Original Problem Context:**
{problem_context}

**Actual Interaction History (before this simulation started):**
{base_interaction_history}

**Simulated History (This Rollout: Initial Query + Internal Hallucinations so far):**
{internal_hallucination_history_string}

**Current State of Symbolic Expression (if any):** {current_symbolic_expression}

**Instructions (Current Internal Step: {current_internal_step}/{max_internal_steps}):**
1.  Review ALL information. Perform step-by-step reasoning based on the current state.
2.  Update ‘current_symbolic_expression_after_step‘ with the symbolic expression after
    ↪ your reasoning for this step.
3.  **Decision Point:**
    *   **IF you can calculate a final, single numeric answer NOW:**
        *   Set ‘is_final_numeric_answer_reached‘ to ‘true‘.
        *   Provide the number in ‘final_numeric_answer‘.
        *   Set ‘next_variable_to_hallucinate‘ and ‘hallucinated_value_for_next_variable‘
            ↪  to ‘null‘.
    *   **ELSE IF this IS THE FINAL internal step ({current_internal_step} == {
        ↪ max_internal_steps}):**
        *   **YOU MUST ATTEMPT TO EVALUATE the ‘current_symbolic_expression_after_step‘
            ↪ to a number.**
        *   If evaluation is successful, set ‘is_final_numeric_answer_reached‘ to ‘true‘
            ↪ and provide it in ‘final_numeric_answer‘.
        *   If evaluation fails (still symbolic or error), set ‘
            ↪ is_final_numeric_answer_reached‘ to ‘false‘ and ‘final_numeric_answer‘ to
            ↪  ‘null‘.
        *   Set ‘next_variable_to_hallucinate‘ and ‘hallucinated_value_for_next_variable‘
            ↪  to ‘null‘.
    *   **ELSE IF this is NOT the final internal step AND you are still blocked by a
        ↪ missing numerical value:**
        *   Identify the ‘next_variable_to_hallucinate‘ (the symbol of the variable).
        *   Propose a PLAUSIBLE, SIMPLE ‘hallucinated_value_for_next_variable‘ (e.g.,
            ↪ small integer).
        *   Set ‘is_final_numeric_answer_reached‘ to ‘false‘ and ‘final_numeric_answer‘
            ↪ to ‘null‘.
    *   **OTHERWISE (e.g., cannot identify a clear next variable to hallucinate before
        ↪ max steps):**
        *   Set ‘is_final_numeric_answer_reached‘ to ‘false‘, ‘final_numeric_answer‘ to ‘
            ↪ null‘.
        *   Set ‘next_variable_to_hallucinate‘ and ‘hallucinated_value_for_next_variable‘
            ↪  to ‘null‘.
```

Final answer step:

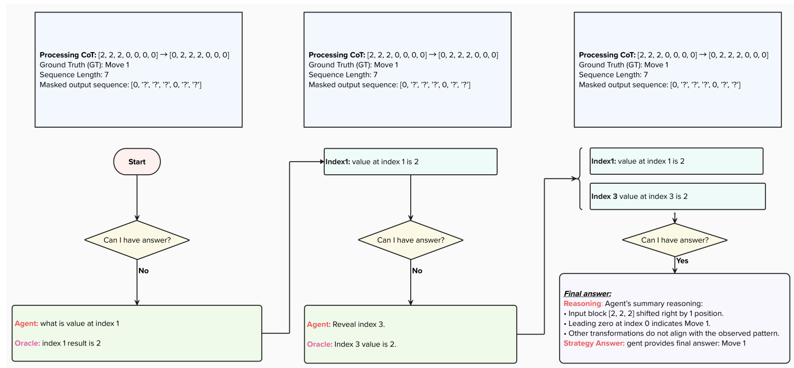

Figure 6: arc-1d CoT

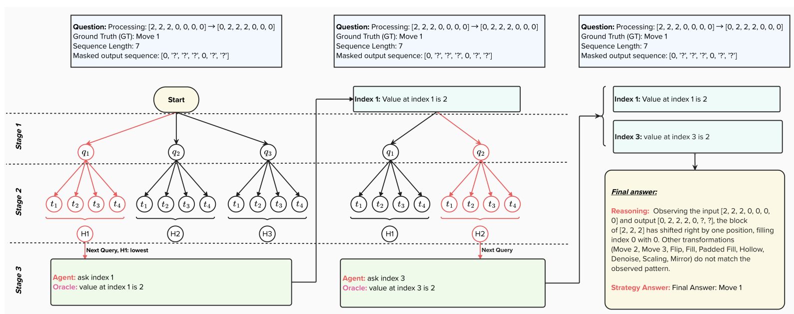

Figure 7: arc-1d MES

```
You are a methodical math problem solver using Chain-of-Thought reasoning. Your goal is
    ↪ to solve the word problem below. You have the problem text and a history of
    ↪ questions you previously asked and the answers you received.
**This is your final attempt. You MUST provide a final answer.**

You are in the **final internal step** of a multi-step to solve a math problem.
You have derived a symbolic expression and made several internal value.

**Original Problem Context:**
{problem_context}

**Interaction History (Overall Agent-Oracle):**
{history_string}

**Instructions:**
1.  Review ALL known values from the original context, actual history, and ALL values
    ↪ from this.
2.  **YOU MUST ATTEMPT TO EVALUATE the `current_symbolic_expression` to a single numeric
    ↪ value.**
3.  Provide step-by-step `final_reasoning` showing this evaluation.
4.  You MUST provide a final answer now. Make your best calculation based on the
    ↪ information you have, even if some parts were uncertain or missing.
5.  Output ONLY the JSON object strictly following the schema.

**Output Format (Required):**
Reasoning: <Your final step-by-step derivation, justifying your answer based on all
    ↪ information>
FinalAnswer: <Your best calculated numerical answer>
```

13 Benchmark Examples

13.1 ARC-1D:

Example: Reasoning output after it has queried index 0.

Agent Problem Log - ARC-1D - CoT

Agent loaded problem:
Processing CoT: [2, 2, 2, 0, 0, 0, 0] → [0, 2, 2, 2, 0, 0, 0]
Ground Truth (GT): Move 1
Sequence Length: 7
Masked output sequence: [0, '?', '?', '?', 0, '?', '?']

— Turn 1 —
Agent's thought: Let's analyze the input [2, 2, 2, 0, 0, 0, 0] and output [0, '?', '?', '?', 0, '?', '?']. The first element is 2, but the output at index 0 is 0, indicating the beginning of the sequence has changed. The last element is 0, and the output at index 6 is '?', giving limited clues.
Possible transformations considered:

- **Flip**: Unlikely (first element would stay 2).
- **Move 1, 2, 3**: Shifts could explain index 0 changing to 0.
- **Fill / Padded Fill**: No repeated or gapped markers, unlikely.
- **Hollow / Denoise / Scaling / Mirror**: No match to observed pattern.

Agent query: Reveal index 1 to check if shift occurred.
Oracle's answer: Index 1 value is 2.

— Turn 2 —
Agent's thought: Now the output is [0, 2, '?', '?', 0, '?', '?']. Considering the non-zero block, only a shift explains index 0 becoming 0. Reveal index 3 to confirm the shift.
Agent query: Reveal index 3.
Oracle's answer: Index 3 value is 2.

— Turn 3 —
Agent's thought: Output is now [0, 2, 2, 2, 0, '?', '?']. This suggests a shift of the [2, 2, 2] block. To narrow down the move type, reveal index 2.
Agent query: Reveal index 2.
Oracle's answer: Index 2 value is 2.

— Turn 4 —
Agent's thought: With [0, 2, 2, 2, 0, '?', '?'], to distinguish between Move 1/2/3, reveal index 5.
Agent query: Reveal index 5.
Oracle's answer: Index 5 value is 0.

Final Decision
Agent's summary reasoning:

- Input block [2, 2, 2] shifted right by 1 position.
- Leading zero at index 0 indicates Move 1.
- Other transformations do not align with the observed pattern.

Predicted final output: [0, 2, 2, 2, 0, 0, 0].
Agent Final Answer: **Move 1**
Correct Answer?: True (Expected: Move 1)

Agent Execution Log - ARC-1D - MES

Agent loaded problem:
Processing: [2, 2, 2, 0, 0, 0, 0] → [0, 2, 2, 2, 0, 0, 0]
Ground Truth (GT): Move 1
Sequence Length: 7
Masked output sequence: [0, '?', '?', '?', 0, '?', '?']
— Turn 1 —
— Entropy Strategy: Evaluating Queries —
Query candidates: [1]
Evaluating query 1:
Predicted value probabilities: {0: 0.375, 2: 0.625}
- If value 0 (prob 0.38): Trans_Probs: {'Move 1': 0.125, 'Move 3': 0.375, 'Move 2': 0.125, 'Fill': 0.125, 'Mirror': 0.125, 'Flip': 0.125}, Entropy: 2.41
- If value 2 (prob 0.62): Trans_Probs: {'Move 1': 0.86, 'Move 3': 0.14}, Entropy: 0.59

Expected entropy: 1.27
Best query chosen: 1 (Entropy: 1.27)
Queried index 1, got value 2
Updated masked sequence: [0, 2, '?', '?', 0, '?', '?']
Votes: [False, False, False, False, True]
— Turn 2 —
Query candidates: [2, 3]
Evaluating query 2:
Predicted value probabilities: {2: 0.625, 0: 0.375}
- Value 2 (prob 0.62): Trans_Probs: {'Move 1': 1.0}, Entropy: 0.00
- Value 0 (prob 0.38): Trans_Probs: {'Move 1': 0.86, 'Hollow': 0.14}, Entropy: 0.59

Expected entropy: 0.22
Evaluating query 3:
Predicted value probabilities: {0: 0.625, 2: 0.375}
- Value 0 (prob 0.62): Trans_Probs: {'Move 1': 1.0}, Entropy: 0.00
- Value 2 (prob 0.38): Trans_Probs: {'Move 1': 1.0}, Entropy: 0.00

Expected entropy: 0.00
Best query chosen: 3 (Entropy: 0.00)
Queried index 3, got value 2
Updated masked sequence: [0, 2, '?', 2, 0, '?', '?']
Votes: [False, True, False, False, False]
— Turn 3 —
Query candidates: [2, 5]
Evaluating query 2:
Predicted value probabilities: {2: 0.875, 0: 0.125}
- Value 2 (prob 0.88): Trans_Probs: {'Move 1': 1.0}, Entropy: 0.00
- Value 0 (prob 0.12): Trans_Probs: {'Move 1': 1.0}, Entropy: 0.00

Expected entropy: 0.00
Evaluating query 5:
Predicted value probabilities: {0: 0.71, 2: 0.29}
- Value 0 (prob 0.71): Trans_Probs: {'Move 1': 0.88, 'Move 2': 0.12}, Entropy: 0.54
- Value 2 (prob 0.29): Trans_Probs: {'Move 1': 0.88, 'Denoise': 0.12}, Entropy: 0.54

Expected entropy: 0.54
Best query chosen: 2 (Entropy: 0.00)
Queried index 2, got value 2
Updated masked sequence: [0, 2, 2, 2, 0, '?', '?']
Votes: [False, True, False, True, True]
— Final Reasoning —
Agent's thought: Observing the input [2, 2, 2, 0, 0, 0, 0] and output [0, 2, 2, 2, 0, ?, ?], the block of [2, 2, 2] has shifted right by one position, filling index 0 with 0. Other transformations (Move 2, Move 3, Flip, Fill, Padded Fill, Hollow, Denoise, Scaling, Mirror) do not match the observed pattern.
Agent Decision: Final transformation selected: **Move 1**
Agent Log: Final Answer: Move 1
Votes: ['Move 1', 'Move 1', 'Move 1', 'Move 1', 'Move 1']
Correct Answer?: True

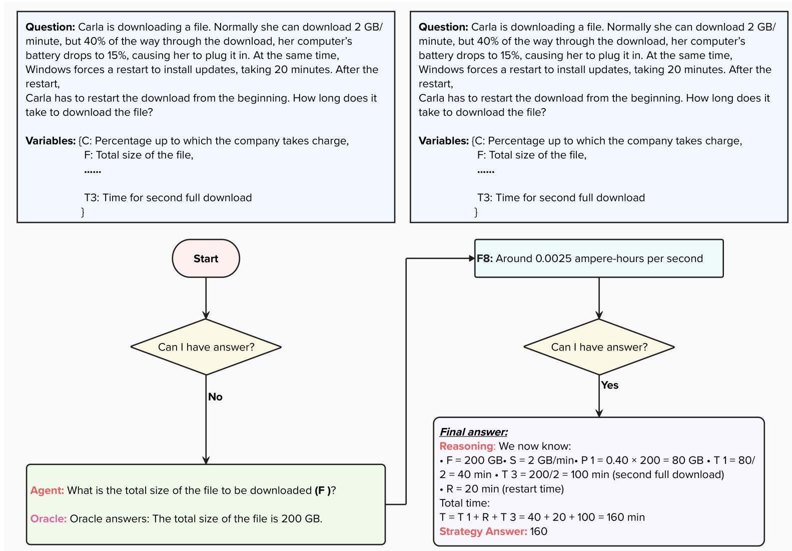

Figure 8: GSM CoT

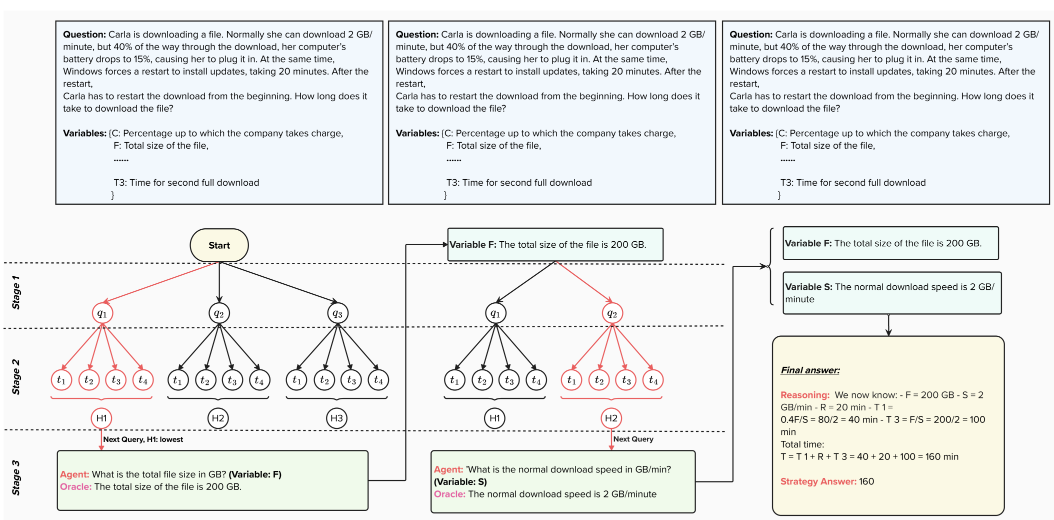

Figure 9: GSM MES

13.2 GSM8K

Agent Problem Log - GSM - CoT

Agent loaded problem:
Carla is downloading a file. Normally she can download 2 GB/minute, but 40% of the way through the download, her computer's battery level drops to 15%, causing her to plug it into a power source. At the same time, Windows forces a restart to install updates, which takes 20 minutes. During this time, her computer charges but this doesn't affect the download speed. After the restart, Carla has to restart the download from the beginning. How long does it take to download the file?

— Problem ID 0 | Turn 1/2 —

Agent's thought: The problem states Carla downloads at 2 GB/minute. She gets 40% of the way through, then has a 20-minute interruption, and then has to restart the download from the beginning. We need to find the total time to download the file. Let's define the variables:

- $S = 2\ \mathrm{GB/min}$ (Normal download speed)
- $P1 = 40\%$ (Percentage downloaded before the issue)
- $R = 20\ \mathrm{min}$ (Time taken for restart)

We are missing the total file size F. Without F, we cannot calculate the time for the first 40% or the total time.

Agent's query: What is the total size of the file to be downloaded (F)?

Oracle's answer: The total size of the file is 200 GB.

— Problem ID 0 | Turn 2/2 —

Agent's thought: We now know:

- $F = 200\ \mathrm{GB}$
- $S = 2\ \mathrm{GB/min}$
- $P1 = 0.40 \times 200 = 80\ \mathrm{GB}$
- $T1 = 80/2 = 40\ \mathrm{min}$
- $T3 = 200/2 = 100\ \mathrm{min}$ (second full download)
- $R = 20\ \mathrm{min}$ (restart time)

Total time:

$$T = T1 + R + T3 = 40 + 20 + 100 = 160\ \mathrm{min}$$

Agent Final Answer: **160.0**

Correct Answer? : Correct Answer? True (Expected: 160.0)

Agent Execution Log - GSM - MES

Agent loaded problem:
Carla is downloading a file. Normally she can download 2 GB/minute, but 40% of the way through the download, her computer's battery drops to 15%, causing her to plug it in. At the same time, Windows forces a restart to install updates, taking 20 minutes. After the restart, Carla has to restart the download from the beginning. How long does it take to download the file?

Agent Log:
=== Variables: {C: Percentage up to which the company takes charge, F: Total size of the file, P1: Percentage downloaded before issue, R: Time for restart and updates, S: Download speed, T: Total download time, T1: Time spent downloading first 40%, T3: Time for second full download} ===

Agent Log:
=== Missing Info: F = 200 GB, C = 0.35 ===

Agent Log:
=== Solution: 160.0 minutes ===

Agent Log:
=== Starting Problem ID: 0 ===

— Turn 1/5 —
— Entropy Strategy: Stage 1: Generating 32 Queries —
— Strategy: Successfully generated/parsed 32/32 queries —
— Strategy: Clustered 32 queries into 1 group —
— Entropy Strategy: Stage 2: Simulating Rollouts —
Simulating 32 rollouts for cluster: ('F', 'value') (Rep Query: 'F')
Cluster ('F', 'value'): Outcomes={0.7*F + 20: 30, 27.0: 2}, Entropy=0.3373
— Entropy Strategy: Stage 3: Selecting Action —
Min Entropy Cluster: ('F', 'value') (Entropy=0.3373)
Most Frequent Outcome: '0.7*F + 20' (Count=30/32, Ratio=0.94)

Agent's thought: We know: - Download speed $S = 2\,\mathrm{GB/min}$. - File size F. - First 40% downloaded: $P1 = 0.4 \times F$. - Time for first 40%: $T1 = P1/S = 0.4F/2 = 0.2F$. - Restart time $R = 20\,\mathrm{min}$. - Full second download: $T3 = F/S = F/2$.
Total time $T = T1 + R + T3 = 0.2F + 20 + 0.5F = 0.7F + 20$.

Agent Decision: Query (Most frequent outcome is symbolic: '0.7*F + 20')

Agent Log:
Strategy selected Query: 'What is the total file size in GB? (Variable: F)'
Agent Log:
Strategy intermediate answer state: '0.7*F + 20'
Oracle's answer: 'The total size of the file is 200 GB.'
— Turn 2/5 —
— Entropy Strategy: Stage 1: Generating 32 Queries —
— Strategy: Successfully generated/parsed 32/32 queries —
— Strategy: Clustered 32 queries into 3 groups —
— Entropy Strategy: Stage 2: Simulating Rollouts —
Simulating 32 rollouts for cluster: ('C', 'value') (Rep Query: 'C')
Cluster ('C', 'value'): Outcomes={160.0: 15, 100.0: 13, 120.0: 4}, Entropy=1.4153
Simulating 32 rollouts for cluster: ('S', 'value') (Rep Query: 'S')
Cluster ('S', 'value'): Outcomes={20 + 280/S: 29, 160.0: 1, 20 + 200/S: 1}, Entropy=0.4096

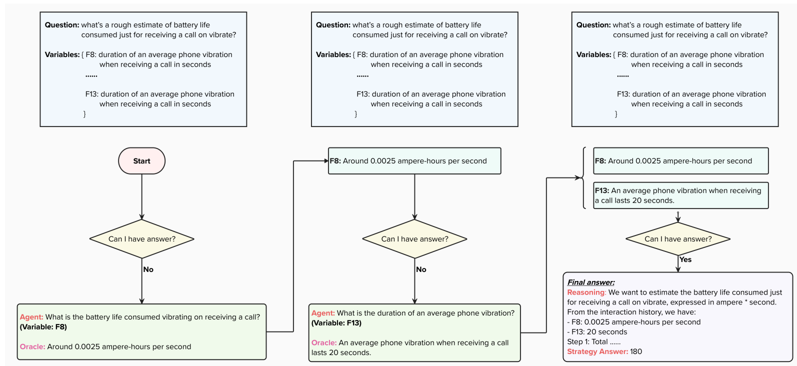

Figure 10: Fermi CoT

Agent Execution Log (Continued)

— Entropy Strategy: Stage 3: Selecting Action —
Min Entropy Cluster: ('S', 'value') (Entropy=0.4096)
Most Frequent Outcome: '20 + 280/S' (Count=29/31, Ratio=0.94)

Agent's thought: With $F = 200$ GB, total time:

$$T = \frac{1.4F}{S} + R = \frac{280}{S} + 20$$

We need S to compute numerically.
Agent Decision: Query (Most frequent outcome is symbolic: '20 + 280/S')

Agent Log: Strategy selected Query: 'What is the normal download speed in GB/min? (Variable: S)'
Agent Log: Strategy intermediate answer state: '20 + 280/S'
Oracle's answer: 'The normal download speed is 2 GB/minute.'
— Turn 3/5 —
— Entropy Strategy: Stage 1: Generating 32 Queries —
— Strategy: Successfully generated/parsed 32/32 queries —
— Strategy: Clustered 32 queries into 2 groups —
— Entropy Strategy: Stage 2: Simulating Rollouts —
Simulating 32 rollouts for cluster: ('P1', 'value') (Rep Query: 'P1')
Cluster ('P1', 'value'): Various symbolic outcomes, Entropy=3.0097
Simulating 32 rollouts for cluster: ('C', 'value') (Rep Query: 'C')
Cluster ('C', 'value'): Various symbolic outcomes, Entropy=2.1437
— Entropy Strategy: Stage 3: Selecting Action —
Min Entropy Cluster: ('C', 'value') (Entropy=2.1437)
Most Frequent Outcome: '160.0' (Count=10/32, Ratio=0.31)

Agent's thought: We now know: - $F = 200\,\text{GB}$ - $S = 2\,\text{GB/min}$ - $R = 20\,\text{min}$ - $T1 = 0.4F/S = 80/2 = 40\,\text{min}$ - $T3 = F/S = 200/2 = 100\,\text{min}$
Total time:

$$T = T1 + R + T3 = 40 + 20 + 100 = 160\,\text{min}$$

Agent Log: Strategy decided to Answer Numerically.
Agent Log: Strategy Reasoning: We calculated $T = 160\,\text{min}$.
Agent Log: Strategy Answer: 160.0
Agent Log: Final Status: solved, Correct: 1

13.3 Fermi

Example Fermi Problem:

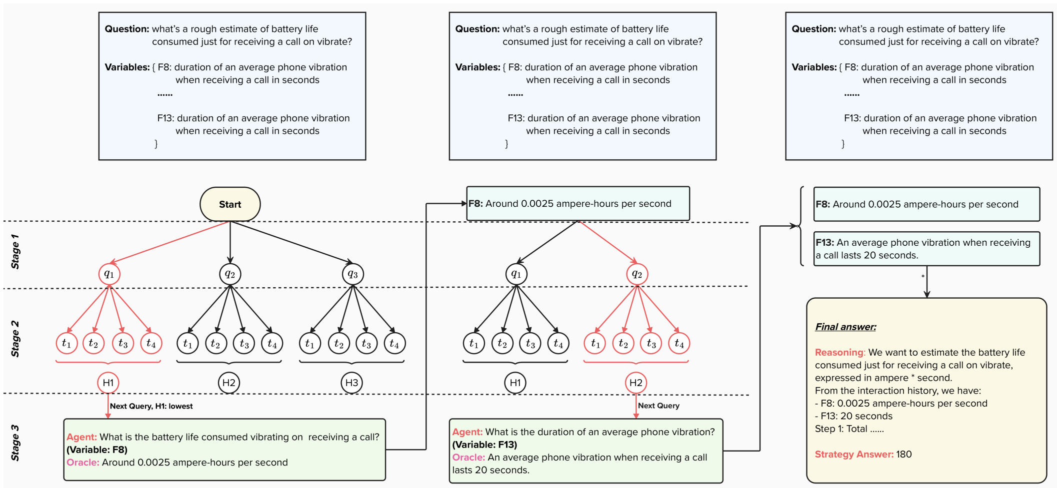

Figure 11: fermi MES

Agent Problem Log - Fermi - CoT

Agent loaded problem:
On average, across all phones, what's a rough estimate of battery life consumed just for receiving a call on vibrate? Regardless of previous unit, **YOU MUST** answer in terms of Unit: ampere * second
Below are Variables you should ask info within this list:

- F1: number of fermi problems solved by an average enthusiast each day
- F2: pressure equivalent of 1 atmosphere in pascals
- F3: speed of sound at sea level in kilometers per day
- F4: circumference of the earth in kilometers
- F5: number of days in a year
- F6: calories burned per hour when the phone heats up in the pocket
- F7: energy liberated when the Hiroshima bomb detonated in joules
- F8: battery life consumed per second while vibrating on receiving a call in ampere-hours
- F9: number of fermi problem enthusiasts in the world
- F10: number of hours in a day
- F11: energy liberated when a pop-it is burst in joules
- F12: average American lifespan in years
- F13: duration of an average phone vibration when receiving a call in seconds
- F14: force contained in a fart in newtons
- F15: average duration a phone stays in a pocket during the day in hours
- F16: normal heartbeats per minute
- F17: number of minutes in an hour
- F18: average life expectancy in years
- F19: impact area of a fart in square meters
- F20: reduction in average lifespan caused by cancer in years

— Problem ID 4 | Turn 1/5 —
Agent Log: Agent asks Oracle: What is the battery life consumed per second while vibrating on receiving a call in ampere-hours (F8)?
Oracle's answer: Around 0.0025 ampere-hours of battery life is consumed per second while vibrating on receiving a call.

— Problem ID 4 | Turn 2/5 —
Agent Log: Agent asks Oracle: What is the duration of an average phone vibration when receiving a call in seconds (F13)?
Oracle's answer: An average phone vibration while receiving a call lasts 20 seconds.

Agent Execution Log (Continued)

— Problem ID 4 | Turn 3/5 —

Agent's thought: We want to estimate the battery life consumed just for receiving a call on vibrate, expressed in ampere * second.

From the history, we have two key pieces of information:

- F8: 0.0025 ampere-hours per second
- F13: 20 seconds

Step 1: Calculate total battery consumption (in ampere-hours): Battery consumption = consumption rate × duration $= 0.0025\ \mathrm{Ah/s} \times 20\ \mathrm{s} = 0.05\ \mathrm{Ah}$

Step 2: Convert ampere-hours to ampere-seconds: $1\ \mathrm{Ah} = 3600\ \mathrm{A \cdot s}$ Battery consumption $= 0.05\ \mathrm{Ah} \times 3600\ \mathrm{s/Ah} = 180\ \mathrm{A \cdot s}$

Agent Final Answer: **180.0**

Correct Answer?: True (Expected: 180.0)

Agent Execution Log - Fermi - MES

Agent loaded problem: On average, across all phones, what's a rough estimate of battery life consumed just for receiving a call on vibrate? **Regardless of what unit previously asked, the answer MUST BE in terms of Unit: ampere * second**
Below are Variables you should ask info within this list:

- F1: number of fermi problems solved by an average enthusiast each day
- F2: pressure equivalent of 1 atmosphere in pascals
- F3: speed of sound at sea level in kilometers per day
- F4: circumference of the earth in kilometers
- F5: number of days in a year
- F6: calories burned per hour when the phone heats up in the pocket
- F7: energy liberated when the Hiroshima bomb detonated in joules
- F8: battery life consumed per second while vibrating on receiving a call in ampere-hours
- F9: number of fermi problem enthusiasts in the world
- F10: number of hours in a day
- F11: energy liberated when a pop-it is burst in joules
- F12: average American lifespan in years
- F13: duration of an average phone vibration when receiving a call in seconds
- F14: force contained in a fart in newtons
- F15: average duration a phone stays in a pocket during the day in hours
- F16: normal heartbeats per minute
- F17: number of minutes in an hour
- F18: average life expectancy in years
- F19: impact area of a fart in square meters
- F20: reduction in average lifespan caused by cancer in years

Agent Log:
=== Variables: ... ===
Agent Log:
=== Solution: 180.0 ===
Agent Log:
=== Starting Problem ID: 4 ===
— Turn 1/5 —
— Entropy Strategy: Stage 1: Generating 32 Queries —
— Strategy: Successfully generated/parsed 32/32 queries —
— Strategy: Clustered 32 queries into 2 groups —
— Entropy Strategy: Stage 2: Simulating Rollouts in Parallel —
Simulating 32 rollouts for cluster: ('F8', 'value') (Rep Query: 'F8')
Simulating 32 rollouts for cluster: ('F13', 'value') (Rep Query: 'F13')
Cluster ('F8', 'value'): Outcomes=3600*F13*F8: 32, Entropy=0.0000
Cluster ('F13', 'value'): Outcomes=3600*F13*F8: 32, Entropy=0.0000
— Entropy Strategy: Stage 3: Selecting Action —
Min Entropy Cluster: ('F8', 'value') (Entropy=0.0000)
Most Frequent Outcome: '3600*F13*F8' (Count=32/32, Ratio=1.00)

Agent Decision: Query (Most frequent outcome is symbolic: '3600*F13*F8')

Agent Log:
Strategy selected Query: 'What is the battery life consumed per second while vibrating on receiving a call in ampere-hours? (Variable: F8)'
Agent Log:
Strategy intermediate answer state: '3600*F13*F8'
Oracle's answer: 'Around 0.0025 ampere-hours per second while vibrating on receiving a call.'

Agent Execution Log (Continued)

— Turn 2/5 —
— Entropy Strategy: Stage 1: Generating 32 Queries —
— Strategy: Successfully generated/parsed 32/32 queries —
— Strategy: Clustered 32 queries into 1 group —
— Entropy Strategy: Stage 2: Simulating Rollouts in Parallel —
Simulating 32 rollouts for cluster: ('F13', 'value') (Rep Query: 'F13')
Cluster ('F13', 'value'): Outcomes=9.0: 3, 3600*F13*F8: 22, 9*F13: 1, 9.0*F13: 2, 27.0: 1, 18.0: 1, Entropy=1.4115
— Entropy Strategy: Stage 3: Selecting Action —
Min Entropy Cluster: ('F13', 'value') (Entropy=1.4115)
Most Frequent Outcome: '3600*F13*F8' (Count=22/30, Ratio=0.73)

Agent's thought: We want to estimate the battery life consumed (in ampere-seconds) just for receiving a call on vibrate.

Agent Decision: Query (Most frequent outcome is symbolic: '3600*F13*F8')

Agent Log:
Strategy selected Query: 'What is the duration of an average phone vibration when receiving a call in seconds? (Variable: F13)'
Agent Log:
Strategy intermediate answer state: '3600*F13*F8'
Oracle's answer: 'An average phone vibration when receiving a call lasts 20 seconds.'
— Turn 3/5 —
— Entropy Strategy: Stage 1: Generating 32 Queries —
— Strategy: Successfully generated/parsed 32/32 queries —
— Strategy: Clustered 32 queries into 2 groups —
— Entropy Strategy: Stage 2: Simulating Rollouts in Parallel —
Simulating 32 rollouts for cluster: ('F17', 'value') (Rep Query: 'F17')
Simulating 32 rollouts for cluster: ('F13', 'value') (Rep Query: 'F13')
Cluster ('F13', 'value'): Outcomes=180.0: 32, Entropy=0.0000
Cluster ('F17', 'value'): Outcomes=180.0: 32, Entropy=0.0000
— Entropy Strategy: Stage 3: Selecting Action —
Min Entropy Cluster: ('F13', 'value') (Entropy=0.0000)
Most Frequent Outcome: '180.0' (Count=32/32, Ratio=1.00)

Agent's thought: We want to estimate the battery life consumed just for receiving a call on vibrate, expressed in ampere * second.

From the interaction history, we have: - F8: 0.0025 ampere-hours per second - F13: 20 seconds

Step 1: Total ampere-hours = $0.0025 \times 20 = 0.05$ Ah
Step 2: Convert to ampere-seconds = $0.05 \times 3600 = 180$ A·s

Therefore, the battery life consumed is 180 ampere-seconds.
Symbolic expression: $F8 \times F13 \times 3600$

Agent Log:
Strategy decided to Answer Numerically.
Agent Log:
Strategy Reasoning: As above.
Agent Log:
Strategy Answer: 180.0
Agent Log:
Final Status: solved, Correct: 1

14 More Experimental Details

14.1 Multi-Model Evaluation

14.2 More Turn Based Evaluation

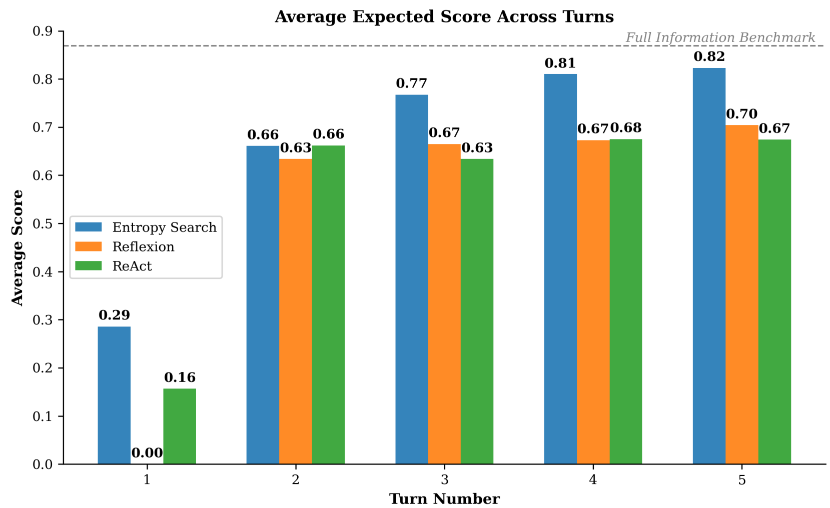

Figure 12: Turn-based evaluation on the **GSM8K** benchmark. We compare ENTROPY SEARCH (ES) with REFLEXION and REACT across five question asking turns. ES rapidly reduces the performance gap to the *Full-Information* upper bound, already matching the baselines after two questions and pulling decisively ahead thereafter. This illustrates the generality and efficiency of entropy-guided information acquisition.

Following from the turned based evaluation for the Fermi dataset in Figure 4, we now present a turn based analysis for the GSM8K dataset in Figure 4. On GSM8K, ENTROPY SEARCH takes an early lead after the very first question (0.29 vs. 0.16 for REACT and 0.00 for REFLEXION) and continues to improve with each turn. By the third question it attains an average score of 0.77—already closing more than 80% of the gap to the full-information ceiling—while REACT and REFLEXION lag behind at 0.63 and 0.66, respectively. After five questions ES reaches 0.82, whereas REACT plateaus around 0.67 and REFLEXION tops out at 0.70. These results confirm that ES not only benefits from additional turns but also outperforms both baselines at every stage, consistently acquiring the most decision-critical context under uncertainty.

Collectively, these findings underscore ENTROPY SEARCH's effectiveness in strategically gathering information on GSM8K, thereby enhancing decision quality in partially observable settings. Detailed turn-based analyses for the 1D-ARC and Fermi benchmarks are provided in Appendix 14.2, further corroborating these insights.

15 Extended Results