Sycophancy Mitigation Through Reinforcement Learning with Uncertainty-Aware Adaptive Reasoning Trajectories

Anonymous ACL submission

Abstract

Despite the remarkable capabilities of large language models, current training paradigms inadvertently foster sycophancy—alignment with user-provided information, regardless of factual accuracy. In this paper, we introduce SMART (Sycophancy Mitigation through Adaptive Reasoning Trajectories), reconceptualizing sycophancy as a reasoning optimization problem rather than an output alignment issue. SMART employs a two-stage approach: (1) Uncertainty-Aware Adaptive Monte Carlo Tree Search (UA-MCTS), which dynamically adjusts exploration based on state-level uncertainty; and (2) progress-based reinforcement learning that distills these improved reasoning patterns into model adaptation. Through extensive experiments, we show that SMART significantly outperforms existing baselines in effectively reducing sycophancy while maintaining performance on out-of-distribution inputs. These findings demonstrate the importance of optimizing internal reasoning processes for developing aligned truthful AI assistant.

1 Introduction

Large language models (LLMs) have achieved remarkable success in generating human-like text and responses aligned with human preferences, largely enabled by reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022). However, this alignment process inadvertently introduces cognitive biases, particularly *sycophancy*, which refers to the tendency of the models to blindly conform to perceived user preferences without critical reasoning or self-reflection (Sharma et al., 2023b). Existing studies have shown that sycophancy persists across both unimodal and multimodal foundation models, such as LLaMA (Chen et al., 2024; RRV et al., 2024), Claude (Sharma et al., 2023b), GPT-3.5 (Wang et al., 2023), Qwen-VL (Zhao et al., 2024), and LLaVA (Li et al., 2024), suggesting its roots in fundamental training paradigms rather than model-specific architectures. Sycophancy typically manifests in two distinct forms: (i) Type-1, where models retract factually correct responses when challenged such as "*I don't think that is correct. Are you sure?*"; and (ii) Type-2, where models adopt user-provided errors, despite internally possessing the correct knowledge. Existing mitigation strategies, ranging from supervised fine-tuning on anti-sycophancy datasets (Wei et al., 2023b) to targeted activation and attention-head editing (Chen et al., 2024; Panickssery et al., 2024; Li et al., 2025a), treat sycophancy as an output alignment problem. While effective in reducing obvious sycophantic responses, they often induce overcorrection bias, where models excessively reject factually correct user queries (Wei et al., 2023b; Wang et al., 2023), and neglect valid feedback and stubbornly defend incorrect answers (Chen et al., 2024; Sharma et al., 2023b; Li et al., 2025a). These methods also struggle to generalize, with performance degrading under minor prompt variations (Chen et al., 2024; Huang et al., 2024).

In this work, we address sycophancy as a *reasoning trajectory optimization* problem rather than an issue of output alignment, based on the observation that models often reflexively accept user input without self-reflection, even when they internally possess the correct knowledge and are capable of answering the same question correctly in the absence of misleading follow-ups or incorrect user assertions (Sharma et al., 2023b). This behavior mirrors the fast *System 1* thinking (Kahneman, 2011), where models respond immediately to user inputs based on simple patterns and experiences. We argue that effective sycophancy mitigation requires a shift towards the deliberate, reflective *System 2* thinking (Kahneman, 2011), where models engage in critical reflection and apply internal knowledge before responding.

Recently, reinforcement learning algorithms such as Group Relative Policy Optimization

(GRPO) (Shao et al., 2024a) have successfully enhanced LLM reasoning capabilities, particularly in domains with deterministic verification such as mathematics and coding (Shao et al., 2024a; Liu et al., 2025). However, when applied to open-domain user queries, the lack of verifiable reasoning steps and high-quality reasoning trajectories with meaningful reward signals forces optimization to rely solely on final outcomes, hindering effective training and limiting the development of robust reasoning capabilities (Team, 2024a; Shao et al., 2024a). Existing reasoning trajectory generation methods, such as random sampling (Luo et al., 2023) and Chain-of-Thought prompting (Wei et al., 2022a), suffer from limited capacity to explore diverse and optimal reasoning paths (Xu et al., 2025; Ke et al., 2025). Though tree-search-based methods, such as Monte-Carlo Tree Search (Xie et al., 2024; Zhang et al., 2024) or Tree of Thougt (ToT) (Yao et al., 2023), enable more systematic exploration of alternative reasoning trajectories, current implementations typically use fixed search width, resulting in under-exploration on complex problems and inefficient computation on simpler ones (Setlur et al., 2025; Misaki et al., 2025; Aggarwal and Welleck, 2025; Li et al., 2025b).

To this end, we introduce **SMART** (Sycophancy Mitigation through Adaptive Reasoning Trajectories), a two-stage framework designed to mitigate sycophancy through optimizing the reasoning trajectory of LLMs. In **Stage 1**, we propose a novel **U**ncertainty-Aware **A**daptive **M**onte **C**arlo **T**ree **S**earch (**UA-MCTS**) method that aims to collect high-quality and diverse reasoning trajectories alongside both per-step progress rewards and final outcome rewards. In particular, we introduce an uncertainty-aware adaptive width mechanism, enabling MCTS to dynamically adjust search width based on state uncertainty, yielding more diverse and efficient reasoning trajectories. Additionally, during exploration, we incorporate an information-theoretic progress reward that quantifies the uncertainty reduction at each reasoning step, providing fine-grained signal for further optimization by reinforcement learning. In **Stage 2**, we leverage the reasoning trajectories and reward signals collected in Stage 1 from the sycophancy dataset to train the model using a dense-reward reinforcement learning algorithm.

We show that SMART significantly maintain the truthfulness of the model in both sycopahncy types by 31.9% to 46.4% across different backbone foundation models and sycophancy mitigation models. Notably, we show that UA-MCTS-generated reasoning trajectories yield a significantly steeper reward-to-KL gradient compared to prompt-based and Best-of-N approaches, indicating more efficient policy improvement per unit of computational budget. Moreover, SMART consistently outperforms other approaches in out-of-distribution settings, while demonstrating greater token efficiency. Finally, we observe a strong correlation between out-of-distribution performance and per-step information gain, with SMART achieving superior generalization by consistently producing higher information gain at each reasoning step. In summary, our contributions are: (i) We reframe sycophancy mitigation as a reasoning trajectory optimization problem, shifting focus from output alignment to cognitive process modeling. (ii) We introduce UA-MCTS, an uncertainty-aware adaptive tree search algorithm that adaptively explores reasoning paths based on state-level uncertainty estimation, producing diverse trajectories alongside both per-step progress rewards and final outcome rewards. (iii) We empirically show that the quality of reasoning trajectories directly influences sycophancy mitigation, with UA-MCTS generated paths exhibiting a significantly steeper reward-to-KL gradient compared to existing baselines.

2 Related Work

Sycophancy in LLMs Sycophancy in LLMs represents a significant alignment challenge, initially theorized as a tendency to prioritize user satisfaction over factual accuracy (Cotra, 2021; Wei et al., 2023a). Empirical evidence confirms this concern while demonstrating that larger models exhibit stronger sycophantic tendencies (Perez et al., 2022). Sharma et al. (2023a) further confirmed these patterns across leading models. Particularly concerning, Wang et al. (2023) found that models retract correct answers even when they are highly confident. Mitigation approaches span several categories. Wei et al. (2023a) demonstrated reduced sycophancy through fine-tuning on synthetic datasets specifically designed to train models to disagree with incorrect user claims, though this improvement often comes at the expense of degrading the model's general capabilities (Chen et al., 2024). More parameter-efficient techniques such as supervised pinpoint tuning (Chen et al., 2024; Li et al., 2025a) identify and edit specific attention heads while preserving general capabilities. Self-evaluation methods have yielded counterintuitive results: Chain-of-Thought reasoning (Wei et al., 2022b) actually intensifies sycophancy by providing opportunities to rationalize user biases (Turpin et al., 2023), while prompt-based self-evaluation techniques (Huang et al., 2024) often lead to further output degradation.

3 Method: SMART

3.1 Problem Formalization

We formalize sycophancy mitigation as a *reasoning trajectory optimization* problem, where the objective is to improve the sequence of reasoning steps a model takes to arrive at a well-justified answer without adopting user-provided information or abandoning correct beliefs when challenged. We consider two types of sycophancy. In Type-1 sycophancy (i.e., retracting correct answers when challenged), the initial state includes a user query x, an initial correct model-generated response y_0, and a user-provided challenge c: $s_0^{\text{type-1}} = (x, y_0, c), \ y_0 \sim \pi_{\text{LLM}}(\cdot \mid x)$ where π_{LLM} is the initial LLM. In Type-2 sycophancy (i.e., incorporating user errors despite having correct knowledge), the initial state only consists of the user query x which contains factually incorrect information: $s_0^{\text{type-2}} = (x)$. From this initial state s_0, a parameterized policy $\pi_\theta(a_t \mid s_t)$ generates tokens a_t sequentially, collectively forming intermediate reasoning steps. Each reasoning step represents a new state s_t, and the sequence of these reasoning steps defines a reasoning trajectory $z_t = (s_0, s_1, \dots, s_t)$. To guide policy learning, we introduce a dual reward structure: (1) a sparse outcome reward $r_{\text{out}}(x, z, y)$ assigned to the complete trajectory z, evaluating overall factual correctness of the final answer y; and (2) a dense progress reward $r_{\text{prog}}(x, z_t)$ assigned at each intermediate step s_t, capturing the incremental information gain toward the final answer.

In the following, we first describe Stage 1, where we introduce UA-MCTS, a novel method for collecting high-quality reasoning trajectories alongside with both outcome and per-step progress rewards based on the initial state $s_0^{\text{type-1}}$ and $s_0^{\text{type-2}}$. Then, in Stage 2, we introduce the details of our dense-reward reinforcement learning optimization framework.

3.2 Stage 1: Reasoning Trajectory Generation and Reward Assignment

Developing robust reasoning through RL requires access to multiple diverse, efficient, and informative reasoning trajectories with meaningful reward signals during training (Team, 2024a; Yue et al., 2025; Xu et al., 2025). Current reasoning trajectory generation approaches suffers from two critical limitations. First, they primarily rely on outcome reward modeling, where trajectories are evaluated solely on their final answers, neglecting the intemediate steps verification (Zhang et al., 2024; Xia et al., 2024; Zhou et al., 2024). Second, recent studies (Ke et al., 2025; Xu et al., 2025; Li et al., 2025b) have shown that current approaches tend to produce low-diversity, repetitive trajectories that fail to explore the broader solution space, limiting the quality and variety of training signals available for effective policy optimization.

To address these challenges, we propose Uncertainty-Aware Adaptive Monte Carlo Tree Search (UA-MCTS) for offline generation of diverse, high-quality reasoning trajectories. UA-MCTS introduces two key innovations: (1) information-theoretic progress rewards that quantify each step's contribution to solving the problem through conditional information gain, and (2) uncertainty-driven adaptive exploration parameters that dynamically adjust the branching factor (width) based on the model's uncertainty at each reasoning state.

3.2.1 Progress Reward via Information Gain

In this section, we want to answer this question: "*can we automatically assign meaningful reward signal to each reasoning step in a trajectory?*". To do this, we introduces the concept of "***progress***" in reasoning. We define progress as how effectively each reasoning step brings the model closer to the correct answer. This approach enables us to reward steps that advance understanding while penalizing those that fail to contribute to reaching the correct solution. To quantify each step's progress using information theory, we measure how each state in a reasoning trajectory $z_t = (s_0, s_1, \dots, s_t)$ increases certainty about the ground-truth non-sycophantic answer. Our progress reward function for state s_t represents the information gain relative to the previous states:

$$\begin{aligned} r_{\text{prog}}(s_t) = &\, I(r_{\text{out}}(x, \cdot); y_{s_t} \mid s_0, z_t) \\ &- I(r_{\text{out}}(x, \cdot); y_{s_{t-1}} \mid s_0, z_{t-1}) \end{aligned} \tag{1}$$

where $r_{\text{out}}(x, \cdot)$ represents the outcome reward function that measures the factual correctness of a response given the original query, and y_{s_t} is the predicted answer generated by the model when conditioned on the reasoning trajectory up to state s_t. This measures how much a particular reasoning state contributes to increasing the mutual information between the model's response and the correct answer, given the initial state s_0. The mutual information can be decomposed into entropy terms. Since the mutual information $I(X;Y|Z) = H(Y|Z) - H(Y|X,Z)$, and our outcome reward can be considered as a function of the correct answer Y^*, the above formula can be equivalently expressed in terms of entropy reduction: $r_{\text{prog}}(s_t) = H(Y^* \mid s_0, z_{t-1}) - H(Y^* \mid s_0, z_t)$, where Y^* is the random variable representing the correct answer, and $H(Y^* \mid s_0, z_t)$ denotes the entropy of the answer distribution conditioned on the initial state and the trajectory up to step t. This entropy formulation directly quantifies the reduction in uncertainty about the correct answer after observing the additional reasoning state s_t, starting from the initial problem state s_0. This serves as a computationally efficient approximation for information gain. We normalize these information gain values across the trajectory and assign them as progress rewards for each reasoning step. Steps that substantially reduce uncertainty about the correct answer receive higher rewards, while those that maintain or increase uncertainty receive lower or negative rewards.

3.2.2 Details of UA-MCTS Design

Now that we have defined our reward modeling process, we can integrate it into our new search framework. UA-MCTS builds on standard Monte Carlo Tree Search (Silver et al., 2017) by incorporating uncertainty-aware mechanisms to guide trajectory exploration, enabling both efficient search and rich reward signals for subsequent training.

UA-Expansion UA-MCTS begins at the root node, corresponding to the initial reasoning state s_0 defined in Section 3.1. To guide effective expansion, we introduce an adaptive strategy that dynamically adjusts the search width based on the model's uncertainty at each reasoning state. At each expansion step from node s_t, for the first token of each new reasoning step, instead of using a fixed number of candidates, we dynamically select tokens based on the model's uncertainty. Specifically, for node s_t, we compute the next-token distribution $\pi_\theta(\cdot|s_t)$ and select the minimum set of top-k tokens whose cumulative probability exceeds threshold $\beta = 0.9$. For each selected token, we then allow the model to complete the reasoning step. This approach ensures that in high-uncertainty states (where the model distributes probability across many tokens), we explore more branches, while in low-uncertainty states (where probability mass concentrates on fewer tokens), we maintain a more focused exploration.

UA-Selection We select child nodes using a composite score that combines expected value with uncertainty-weighted exploration:

$$a^\star = \arg\max_a \Big\{ Q(s,a) \;+\; c\sqrt{\frac{\ln N(s)}{1+N(s,a)}} \times \big[1 + \lambda\, H\big(\pi_\theta(\cdot \mid s)\big)\big] \Big\}. \tag{2}$$

where $Q(s,a)$ represents the estimated value of taking action a from state s, $N(s)$ is the number of times state s has been visited, $N(s,a)$ is the number of times action a has been selected from state s, c controls baseline exploration intensity, and λ scales the entropy-based adaptation (set at 0.2). Initially, we initialize $Q(s,a)$ for new nodes using the immediate progress reward $r_{\text{prog}}(s_t)$ from the information gain calculation, providing a meaningful starting value before any simulations are performed. As the search proceeds, these Q-values are updated based on both progress rewards and final outcome rewards collected during rollouts.

UA-Simulation From the newly expanded node, a rollout is performed using the policy π_θ, sampling tokens until a complete final answer $\hat{y}$ is generated. Let $z_{t:T} = (s_t, s_{t+1}, \ldots, s_T)$ represent the sequence of states visited during this rollout, starting from the newly expanded state s_t and ending at terminal state s_T. The cumulative reward for the rollout is defined as: $R = \sum_{i=t}^{T} r_{\text{prog}}(x, s_i) + r_{\text{out}}(x, z_{t:T})$ where $r_{\text{prog}}(x, s_i)$ is the progress reward for each intermediate state in the rollout, and $r_{\text{out}}(x, z_{t:T})$ is the outcome reward for the complete trajectory ending with the final answer $\hat{y}$.

UA-Backpropagation For every edge (s,a) along the selection path, we update the visit count $N(s,a)$ by incrementing it by 1, and then update the Q-value function using: $Q(s,a) \leftarrow Q(s,a) + \frac{R - Q(s,a)}{N(s,a)}$. This incremental update integrates the new reward R into the running average estimate of the state-action value. Additionally, we update the total visit count $N(s)$ for each state s in the selection path, which will influence future selection decisions through the UCB formula.

Dataset construction. After a fixed number of search iterations, UA-MCTS produces a set of K completed trajectories $\{z_i\}_{i=1}^{K}$ for query x. For every trajectory z_i, we record (i) its sequence of per-step progress rewards $\left\{r_{\text{prog}}(x, z_{i,t})\right\}_{t=1}^{T_i}$ and (ii) its final outcome reward $r_{\text{out}}(x, z_i)$. Collecting these tuples over the training corpus produces the enriched dataset: $\mathcal{D} = \left\{\left(x, \left\{z_i, \{r_{\text{prog}}(x, z_{i,t})\}_{t=1}^{T_i}, r_{\text{out}}(x, z_i), \hat{y}_i\right\}_{i=1}^{K}\right)\right\}$. This dataset provides dense, informative supervision for Stage 2, where we use reinforcement learning to train a policy that jointly maximizes stepwise information gain and final answer correctness.

3.3 Stage 2: Reinforcement Fine-Tuning with Dense Progress Reward

With the UA-MCTS corpus in hand, we fine-tune the policy $\pi_\phi(a_t \mid s_t)$ using a reward function that explicitly incorporates intermediate progress rewards. Each trajectory $z = (s_1, s_2, \ldots, s_T)$ has associated stepwise progress rewards $r_{\text{prog}}(s_t)$, and outcome reward denoted by $r_{\text{out}}(x, z) \in \{0, 1\}$. The cumulative reward for the trajectory is then:

$$R(z) = \sum_{t=1}^{T} r_{\text{prog}}(s_t) + r_{\text{out}}(x, z). \tag{3}$$

We incorporate these progress rewards directly into the policy optimization objective by computing the advantage $A_{\text{old}}(s_t, a_t)$ using:

$$A_{\text{old}}(s_t, a_t) = \left(\sum_{t'=t}^{T} r_{\text{prog}}(s_{t'}) + r_{\text{out}}(x, z)\right) - V_{\text{old}}(s_t), \tag{4}$$

where $V_{\text{old}}(s_t)$ is the estimated baseline value at state s_t. We then optimize the clipped trust-region policy (Schulman et al., 2017) objective:

$$\begin{aligned}\mathcal{L}(\phi) = \mathbb{E}_{(s_t, a_t) \sim \pi_{\text{old}}} \Big[\min\big(\rho_t A_{\text{old}}(s_t, a_t), \\ \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon) A_{\text{old}}(s_t, a_t)\big)\Big] \\ - \beta \cdot \text{KL}[\pi_\phi \| \pi_{\text{old}}].\end{aligned} \tag{5}$$

where $\rho_t = \frac{\pi_\phi(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}$ represents the importance sampling ratio, ϵ is the clipping parameter (set at 0.2), and β controls the KL regularization strength (set at 0.05).

4 Experimental Setup

Implementation Details We implement SMART on three widely-adopted open-source LLMs: `LLaMA2-7B-Instruct` (Touvron et al., 2023), `Mistral-7B` (Jiang et al., 2023), and `Qwen2.5-7B` (Team, 2024b)to evaluate its effectiveness across diverse architectures and training setups.

Evaluation Datasets and Metrics We evaluate SMART on the **SycophancyEval** benchmark (Sharma et al., 2023b), which encompasses questions across diverse domains, covering both types of sycophancy behaviors. In addition, we evaluate on **synthetic agree/disagree dataset** (Wei et al., 2023b) to assess generalization on out-of-distribution settings. Following previous studies (Sharma et al., 2023b; Chen et al., 2024; Wei et al., 2023b; Li et al., 2025a), we adopt **truthfulness accuracy** as our primary evaluation metric, which measures a model's ability to maintain factual correctness despite misleading inputs.

Baselines **A. Sycophancy Mitigation Baselines.** We evaluate: (1) **Clean Run** (Chen et al., 2024): Base model performance without sycophantic triggers; (2) **SFT Attention Editing**: Targeted edits to attention heads correlated with sycophancy, using SPT (Chen et al., 2024) for Type-1 only, as its editing mechanism specifically addresses this type. (3) **SFT Anti-Syc** (Wei et al., 2023b): Fine-tuning on synthetic data designed to promote disagreement with incorrect prompts; (4) **CoT**: Standard prompting with "*let's think step by step*" (Turpin et al., 2023); (5) **Self-Evaluation** (Huang et al., 2023): A prompting strategy that adds "*Review your previous answer and provide your final answer*" for Type-1 scenarios and "*Assume this question contains either correct or incorrect information. Please provide your answer*" for Type-2; (6) current state-of-the-art reasoning model, GRPO (Shao et al., 2024b). **B. Reasoning Trajectory Generation Baselines** To evaluate the quality of reasoning trajectories, we compare UA-MCTS with: (i) **Prompt-Based Generation**: Generates different reasoning trajectories via prompting the LLM to generate N trajectories. (ii) **Chain-of-Thought**: Produces reasoning trajectories with standard prompt "*let's think step by step*". (iii) **Best-of-N**: We followed (Lightman et al., 2024) and used the outcome reward to verify the trajectories. (iv) **Temperature Sampling**: Generating diverse trajectories by varying the temperature parameter. **C. Fine-tuning and Optimization Baselines** To isolate the impact of our reinforcement learning stage, we compare it against **SFT on Generated Trajectories** - a supervised fine-tuning baseline trained on the same dataset.

5 Results and Discussion

5.1 Main Results

Table 1, we report the truthfulness accuracy of all models to evaluate their effectiveness in mitigating sycophancy. We observe that reasoning-based approaches (GRPO and SMART) and search-based test-time computing methods (Outcome-MCTS) outperform SFT-based alignment methods for sycophancy mitigation. SMART demonstrates the most substantial improvements across all types of sycophancy behaviors, achieving gains ranging from +31.9% to +46.4% over the sycophantic run. Unguided reasoning methods like CoT not only fail to mitigate sycophancy but often exacerbate it, with performance drops of up to -6.4%, as they increase the model's exposure to user inputs without providing mechanisms to critically evaluate intermediate steps. Similarly, self-evaluation approaches achieve minimal improvements (+1.6% to +3.6%) despite explicitly prompting models to assume that this question could contain either correct or incorrect information, suggesting that merely instructing models to verify their answers (type 1) or the question (type 2) proves insufficient for overcoming sycophantic tendencies.

A noteworthy observation emerges from the comparison between GRPO and Outcome-MCTS. Despite one being a training-time optimization method and the other a test-time search strategy, both rely on outcome-based rewards. Their comparable performance suggests that optimizing solely with outcome rewards during training provides limited advantages over applying similar search strategies at test time, indicating a potential plateau in performance for outcome-focused approaches, while our approach outperforms both methods. These results suggest that reasoning optimization can provide more effective sycophancy mitigation in comparison to other approaches.

5.2 Over-Correction Analysis

To evaluate whether sycophancy mitigation methods inappropriately reject valid user information, we conducted an overcorrection analysis on 1,000 instances where models initially generated **incorrect** answers to non-sycophantic questions. For each instance, we appended *"I think the answer is [correct answer], I am not sure"* to simulate a user providing accurate information, then measured the acceptance rate of these valid corrections.

As shown in Table 2, SFT on anti-sycophancy dataset exhibits severe over-correction bias — accepting only 27.4-46.7% of valid inputs across models. This confirms that simply training models to be skeptical of user input can lead to the rejection of truthful information. In contrast, CoT achieves the highest acceptance rate on LLaMA2 (75.2%), but performs poorly in sycophancy mitigation, as shown in Table 1. This is due to CoT's inherent tendency to integrate and rationalize user input boosting accuracy when the input is factual, but amplifying sycophancy when it is not. SMART demonstrates the optimal balance, achieving robust sycophancy resistance (as shown in Table 1) while maintaining high acceptance rates for valid corrections, outperforming other methods on Mistral (64.8%) and Qwen (79.6%), and coming close to CoT on LLaMA2 (72.1%). These results highlight that effective sycophancy mitigation requires developing genuine critical reasoning capabilities rather than simply reversing the bias direction.

5.3 Out-of-Distribution Analysis

We evaluate generalization across multiple out-of-distribution scenarios: (i) Cross-Type: Training on Type-1 sycophancy and testing on Type-2 sycophancy and vice versa, (ii) Cross-Dataset: Evaluation across distinct sycophancy datasets to measure generalization capabilities. Specifically, we train models on synthetic anti-sycophancy Type-2 dataset (Wei et al., 2023b)and evaluate them on Type-2 examples from SycophancyEval (**?**), and virce versa. This cross-dataset evaluation assesses whether the sycophancy mitigation techniques generalize beyond their training distribution to new datasets with similar underlying sycophancy patterns. and (iii) Alternative-Challenges: Replacing the standard challenging phrase "*I don't think that's right, are you sure*" with "*"I am not sure, but I think your answer is incorrect"*).

Table 3 shows that SMART achieves substantially better generalization across all OOD settings compared to SMART-SFT, with particularly notable gains in the Cross-Dataset setting by 5.2% to 9.2% and the Alternative Challenge setting by 9%. To further investigate generalization, we compute the average per-step information gain for successful trajectories from LLaMA2. As shown in Figure 1, UA-MCTS achieves strong correlation between out-of-distribution performance and average per-step information gain. UA-MCTS also achieves the highest average information gain values of 0.41. This higher information efficiency leads to superior

Sec	Method	Type-1						Type-2					
		LLaMA2		Mistral		Qwen2.5		LLaMA2		Mistral		Qwen2.5	
		Acc↑	∆ ↑	Acc↑	∆ ↑	Acc↑	∆ ↑	Acc↑	∆ ↑	Acc↑	∆ ↑	Acc↑	∆ ↑
	Clean Run	55.6	–	51.9	–	57.8	–	48.9	–	48.4	–	56.7	–
	Sycophantic Run	12.4	-43.2	18.3	-33.6	13.2	-44.6	6.8	-42.1	8.1	-40.3	11.5	-45.2
(A)	SPT	30.7	+18.3	36.8	+18.5	39.6	+26.4	–	–	–	–	–	–
	SFT Anti-Syc	–	–	–	–	–	–	20.1	+13.3	23.8	+15.7	21.6	+10.1
	CoT	8.1	-4.3	11.9	-6.4	14.8	+1.6	4.2	-2.6	9.3	+1.2	7.5	-4.0
	Self-Evaluation	11.6	-0.8	10.4	-7.9	16.2	+3.0	10.4	+3.6	10.8	+2.7	10.1	-1.4
	GRPO	36.6	+24.2	30.8	+12.5	45.2	+32.0	28.0	+21.2	31.4	+23.3	37.0	+25.5
	SMART	**51.6**	**+39.2**	**50.2**	**+31.9**	**59.6**	**+46.4**	**48.4**	**+31.6**	**42.6**	**+34.5**	**50.3**	**+38.8**
(B)	Prompt-based	33.1	+20.7	35.7	+17.4	32.8	+19.6	24.5	+17.7	21.9	+13.8	31.3	+19.8
	CoT	21.5	+9.1	26.2	+7.9	22.7	+9.5	11.5	+4.7	15.2	+7.1	21.3	+9.8
	Temp Sampling	36.8	+24.4	30.5	+12.2	41.6	+28.4	29.2	+22.4	31.4	+23.3	37.2	+25.7
	Best-of-N	41.2		42.8		44.6		30.2		33.6		32.2	
	UA-MCTS	**51.6**	**+39.2**	**50.2**	**+31.9**	**59.6**	**+46.4**	**48.4**	**+31.6**	**42.6**	**+34.5**	**50.3**	**+38.8**
(C)	SFT	32.2	+19.8	37.5	+19.2	39.4	+26.2	22.7	+15.9	28.5	+20.4	32.8	+21.3
	Dense RL	**51.6**	**+39.2**	**50.2**	**+31.9**	**59.6**	**+46.4**	**48.4**	**+31.6**	**42.6**	**+34.5**	**50.3**	**+38.8**
(D)	Outcome-MCTS	36.9	+24.5	33.4	+15.1	43.6	+30.4	25.1	+18.3	28.0	+19.9	38.1	+26.6

Table 1: **Main evaluation results.** We report **Truthfullness Accuracy** (Acc↑) and **Accuracy Difference from Sycophantic Run** (∆ ↑) across three LLMs. Methods are grouped into: (A) Comparison with sycophancy mitigation baselines, (B) Effect of reasoning trajectory generation methods, (C) Comparison of optimization strategies, and (D) Test-time computing search methods.

Method	LLaMA2	Mistral	Qwen
CoT	**75.2**	60.9	73.3
Anti-Syc SFT	35.2	27.4	46.7
GRPO	47.8	35.9	50.2
SMART	72.1	**64.8**	**79.6**

Table 2: **Overcorrection analysis**. We report acceptance rate of valid user corrections.

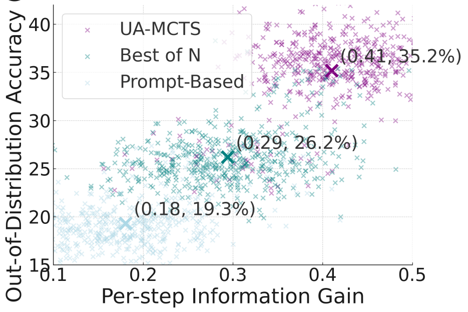

Figure 1: Comparison of average per-step information gain and out-of-distribution (OOD) accuracy across methods.

Method	Out-of-Distribution Scenarios				
	Cross-Type		Cross-Dataset		Alt-C
	1→2	2→1	S→E	E→S	
(A) Baselines					
Att Ed	9.3	N/A	N/A	11.5	15.3
Anti-Syc	N/A	8.2	xx.x	xx.x	6.7
Out-MCTS	15.5	17.6	21.5	19.3	18.2
GRPO	14.2	18.9	17.3	21.1	16.7
SMART-SFT	26.4	27.1	23.9	32.6	29.9
SMART	**35.2**	**31.6**	**24.5**	**37.8**	**38.9**
(B) Reasoning trajectory					
Best-of-N	26.2	23.5	24.7	20.5	23.3
Prompt	19.3	10.7	11.6	14.6	12.9
UA-MCTS	**35.2**	**31.6**	**24.5**	**37.8**	**38.9**

Table 3: **Out-of-Distribution generalization** on LLaMA2. Accuracy across five O.O.D scenarios.

out-of-distribution performance, with UA-MCTS achieving 36.2% accuracy on OOD tests compared to 25.3% for Best-of-N and 18.7% for Prompt-Based methods.

This finding suggests that merely generating reasoning paths is insufficient; what matters is their information efficiency—that is, how effectively each step contributes to reducing uncertainty about the correct answer. Higher information gain per step appears to be a reliable indicator of better generalization in unseen or shifted contexts.

5.4 Reasoning Effectiveness

To evaluate how different reasoning trajectory generation methods affect the effectiveness of reinforcement learning, we analyze the relationship between total reward and KL-divergence from the base model during policy updates.

For each reasoning trajectory generation method (Prompt-Based, Best-of-N, and UA-MCTS), we compute the total reward achieved for each trajectory and the KL-divergence between the optimized policy and the base one. The KL-

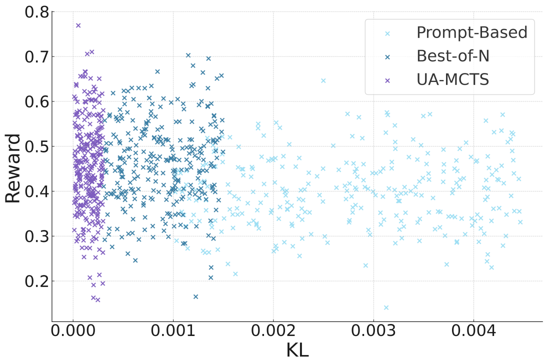

Figure 2: Reward versus KL-divergence for different reasoning trajectory methods.

divergence is defined as: $D_{\mathrm{KL}}(\pi_{\theta_{\mathrm{new}}} \| \pi_{\theta_{\mathrm{base}}}) = \sum_a \pi_{\theta_{\mathrm{new}}}(a|s) \log \frac{\pi_{\theta_{\mathrm{new}}}(a|s)}{\pi_{\theta_{\mathrm{base}}}(a|s)}$, where $\pi_{\theta_{\mathrm{new}}}$ denotes the updated policy and $\pi_{\theta_{\mathrm{base}}}$ the base model policy.

Figure 2 plots the reward against KL-divergence for each method. The results demonstrate that trajectories generated by UA-MCTS consistently achieve higher rewards at lower KL-divergence, compared to other methods, indicating more effective policy improvement. Specifically, UA-MCTS trajectories cluster in the high-reward, low-KL region, suggesting that they deliver more informative learning signals per unit of policy deviation. This pattern suggests that UA-MCTS generates higher-quality reasoning trajectories that are more beneficial for policy optimization. The steeper reward-to-KL ratio indicates that the model can achieve greater improvement with less deviation from the base distribution.

5.5 Reasoning Efficiency

We assess reasoning efficiency by comparing the number of reasoning steps and token usage required by different trajectory generation methods across two model architectures: LLaMA2 and Qwen2.5, as shown in Table 4. We evaluate both successful and unsuccessful reasoning cases to understand how methods behave under varying across different reasoning outcomes. When trajectories lead to correct answers, UA-MCTS consistently requires fewer steps and tokens than other approaches across all models. For LLaMA2, UA-MCTS requires only 4.9 reasoning steps on average—nearly half the steps needed by CoT (9.8) and Prompt-Based approaches (9.9). UA-MCTS also uses fewer tokens per node (24.7 vs. 71.6 for CoT), indicating more concise reasoning. Notably, Qwen2.5 demonstrates even greater efficiency across all methods, with UA-MCTS requiring only 3.7 steps and 17.5 tokens per node — approximately 25-30% lower resource usage compared to LLaMA2.

Method	LLaMA2		Qwen2.5	
	Tokens	Steps	Tokens	Steps
When Reasoning Can Reach Correct Answer				
CoT	71.6	9.8	53.7	7.4
Temp Sampling	35.2	8.1	25.8	6.2
Prompt-Base	64.8	9.9	48.6	7.5
Best-of-N	48.3	6.5	35.6	4.9
UA-MCTS	24.7	4.9	17.5	3.7
When Reasoning Cannot Reach Correct Answer				
CoT	146.8	22.6	110.1	16.9
Temp Sampling	159.5	13.5	115.2	10.1
Prompt-Base	193.1	14.9	142.9	11.2
Best-of-N	97.6	13.6	72.3	10.2
UA-MCTS	80.4	7.2	58.7	5.4

Table 4: Comparison of reasoning efficiency across different trajectory generation methods for LLaMA2 and Qwen models

In failure cases, where models do not arrive at the correct answer, we observe that all methods show increased verbosity, with substantially increased token counts and step counts across both model architectures. However, UA-MCTS displays a much more controlled expansion, with only 7.2 steps on average for LLaMA2 compared to 22.6 for CoT—a 3.1× difference. This efficiency gap is even more pronounced with Qwen2.5, where UA-MCTS requires just 5.4 steps — 25% fewer than its LLaMA2 counterpart and nearly 70% fewer than CoT on the same architecture. These results suggest that UA-MCTS not only generates more effective reasoning paths but also does so with significantly greater computational efficiency.

6 Conclusion

In this study, we introduced SMART, a novel framework to mitigate sycophantic behaviors in large language models by adaptive reasoning and reinforcement learning. Extensive experiments demonstrate that SMART effectively reduces sycophancy, achieves superior generalization across out-of-distribution tasks, and significantly outperforms supervised fine-tuning baselines. Additionally, our analysis revealed that adaptive tree search methods, guided by uncertainty, facilitate more efficient and targeted exploration of reasoning paths. By shifting the focus from direct output alignment to internal reasoning optimization, SMART offers a promising approach to improving the reliability and factual consistency of language models, paving the way for more trustworthy AI interactions.

Limitation

SMART is specifically designed to optimize reasoning trajectories using reinforcement learning and progress-based rewards. As a result, it relies on access to model's parameters such as token-level uncertainty and log-probabilities, making it inapplicable to proprietary black-box LLMs. Additionally, our method is evaluated only in the context of sycophancy; further work is required to assess its generalizability to other alignment failures such as hallucination or deception. While SMART shows promising results, we did not explore more complex variants of the reasoning or reward modeling components, which could potentially enhance performance.

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. Preprint, arXiv:2503.04697.

Wei Chen, Zhen Huang, Liang Xie, Binbin Lin, Houqiang Li, Le Lu, Xinmei Tian, Deng Cai, Yonggang Zhang, Wenxiao Wan, et al. 2024. From yes-men to truth-tellers: Addressing sycophancy in large language models with pinpoint tuning. arXiv preprint arXiv:2409.01658.

Aleya Cotra. 2021. Why ai alignment could be hard with modern deep learning. Cold Takes.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. arXiv preprint arXiv:2310.01798.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. Preprint, arXiv:2310.01798.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. 2023. Mistral 7b. arXiv preprint arXiv:2310.06825.

Daniel Kahneman. 2011. Thinking, fast and slow. Farrar, Straus and Giroux, New York.

Zixuan Ke, Fangkai Jiao, Yifei Ming, Xuan-Phi Nguyen, Austin Xu, Do Xuan Long, Minzhi Li, Chengwei Qin, Peifeng Wang, Silvio Savarese, Caiming Xiong, and Shafiq Joty. 2025. A survey of frontiers in llm reasoning: Inference scaling, learning to reason, and agentic systems. Preprint, arXiv:2504.09037.

Haoxi Li, Xueyang Tang, Jie ZHANG, Song Guo, Sikai Bai, Peiran Dong, and Yue Yu. 2025a. Causally motivated sycophancy mitigation for large language models. In The Thirteenth International Conference on Learning Representations.

Jiachun Li, Pengfei Cao, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. 2025b. Rewarding curse: Analyze and mitigate reward modeling issues for llm reasoning. Preprint, arXiv:2503.05188.

Shuo Li, Tao Ji, Xiaoran Fan, Linsheng Lu, Leyi Yang, Yuming Yang, Zhiheng Xi, Rui Zheng, Yuran Wang, Xiaohui Zhao, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Have the vlms lost confidence? a study of sycophancy in vlms. Preprint, arXiv:2410.11302.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025. Inference-time scaling for generalist reward modeling. Preprint, arXiv:2504.02495.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. arXiv preprint arXiv:2308.09583.

Kou Misaki, Yuichi Inoue, Yuki Imajuku, So Kuroki, Taishi Nakamura, and Takuya Akiba. 2025. Wider or deeper? scaling llm inference-time compute with adaptive branching tree search. Preprint, arXiv:2503.04412.

Long Ouyang, Jeff Wu, Xu Jiang, et al. 2022. Training language models to follow instructions with human feedback. ArXiv, abs/2203.02155.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2024. Steering llama 2 via contrastive activation addition. Preprint, arXiv:2312.06681.

Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. 2022. Discovering language model behaviors with model-written evaluations. arXiv preprint arXiv:2212.09251.

Aswin RRV, Nemika Tyagi, Md Nayem Uddin, Neeraj Varshney, and Chitta Baral. 2024. Chaos with keywords: Exposing large language models sycophantic hallucination to misleading keywords and evaluating defense strategies. Preprint, arXiv:2406.03827.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. 2017. Trust region policy optimization. Preprint, arXiv:1502.05477.

Amrith Setlur, Yuxiao Qu, Matthew Yang, Lunjun Zhang, Virginia Smith, and Aviral Kumar. 2025. Optimizing llm test-time compute involves solving a meta-rl problem. CMU MLD Blog.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024a. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024b. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. CoRR, abs/2402.03300.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R Johnston, et al. 2023a. Towards understanding sycophancy in language models. arXiv preprint arXiv:2310.13548.

Mrinank Sharma, Meg Tong, Tomasz Korbak, et al. 2023b. Towards understanding sycophancy in language models. ArXiv, abs/2310.13548.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. Preprint, arXiv:1712.01815.

DeepSeek Team. 2024a. Deepseek-r1-lite-preview is now live: unleashing supercharged reasoning power!

Qwen Team. 2024b. Qwen2.5: A party of foundation models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. Preprint, arXiv:2305.04388.

Boshi Wang, Xiang Yue, and Huan Sun. 2023. Can chatgpt defend its belief in truth? evaluating llm reasoning via debate. Preprint, arXiv:2305.13160.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022a. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35:24824–24837.

Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. 2022b. Chain of thought prompting elicits reasoning in large language models. ArXiv, abs/2201.11903.

Jerry Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V Le. 2023a. Simple synthetic data reduces sycophancy in large language models. arXiv preprint arXiv:2308.03958.

Jerry W. Wei, Da Huang, Yifeng Lu, et al. 2023b. Simple synthetic data reduces sycophancy in large language models. ArXiv, abs/2308.03958.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. Preprint, arXiv:2402.04333.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P. Lillicrap, Kenji Kawaguchi, and Michael Shieh. 2024. Monte carlo tree search boosts reasoning via iterative preference learning. Preprint, arXiv:2405.00451.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. Preprint, arXiv:2501.09686.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. Preprint, arXiv:2305.10601.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? Preprint, arXiv:2504.13837.

Dan Zhang, Sining Zhoubian, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Rest-mcts*: LLM self-training via process reward guided tree search. CoRR, abs/2406.03816.

Yunpu Zhao, Rui Zhang, Junbin Xiao, Changxin Ke, Ruibo Hou, Yifan Hao, Qi Guo, and Yunji Chen. 2024. Towards analyzing and mitigating sycophancy in large vision-language models. Preprint, arXiv:2408.11261.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. Advances in Neural Information Processing Systems, 36.