WHAT AM I MISSING? QUESTION-ANSWERING AS HIDDEN STATE PROBING

Anonymous authors
Paper under double-blind review

ABSTRACT

The study of test-time reasoning in large language models (LLMs) has attracted increasing interests due to its profound impact on many tasks, with early work tracing back to approaches such as chain-of-thought reasoning. The basic mechanism underlying the test-time reasoning process, however, is still under-studied: from the same input prompt, and even the same partial solution, LLMs can produce varied answers if sampled multiple times. In this paper, we leverage question-asking as an inference-time intervention that articulates information about the model's hidden state. Specifically, we present a student-teacher setting where a student asks questions to a teacher. We train a probe on the student's hidden state before and after asking a question and find it is predictive of the trajectory's final correctness, even before generating the teacher's answer. This suggests there is a meaningful signal from the *self-diagnosis* that occurs during question generation rather than information transfer from the teacher. We then frame question-asking as a sequential decision problem, using this probe as a quality score, and define a gating policy to ask questions that maximize likelihood of correctness. We find that the success of question-asking as an intervention is largely dependent on the model's self-consistency. Our empirical results show a gap between detection and recovery; while our gating policy captures model correctness and uncertainty, interventions are equally likely to harm correct trajectories as they are to recover incorrect ones. This gap between diagnosis and correction has broader implications on language models' capacity for self-refinement under uncertainty. [1]

1 INTRODUCTION

Since the introduction of chain-of-thought reasoning, test-time technologies have become a significant field of study for large language models (LLMs). This has also led to the introduction of Large Reasoning Models (LRMs) that undergo post-training to encourage the model to produce intermediate tokens before formulating the final answer Kambhampati et al. (2025). There have been various interventions to improve LLMs' performance at inference time by manipulating intermediate reasoning tokens Wei et al. (2022); Snell et al. (2024); Zhang et al. (2025a). For example, Muennighoff et al. (2025) demonstrate that simply replacing the stop token with a continuation such as "wait," often leads to significant improvements in accuracy. In addition, other works treat language model generation as an exploration in the generation space and employ branching to search for optimal reasoning trajectories Li et al. (2026); Huang et al. (2025); Chang et al. (2025). The third class of intervention is through question-asking (QA) — questions are an effective mechanism for clarification Aliannejadi et al. (2019); Rao and Daumé III (2018), gathering new information Hu et al. (2024), or encouraging the decomposition of complex tasks Press et al. (2023). These interventions aimed to compensate for the same fundamental issue: LLMs struggle to consistently self-correct reasoning without external feedback Huang et al. (2023); Tyen et al. (2024).

In this paper, we leverage question-asking as an inference-time intervention, investigating whether and how it can be used to guide and correct test-time reasoning, where question-asking is theorized to contain more semantic information, which explicitly encourages the language model to guide itself toward the correct answer Ambati et al. (2025). We hypothesize that the process of formulating a question can inherently change the model's reasoning trajectory at a semantic level — introducing new

[1] Our code is available at `https://github.com/chufeiluo/belief-state-qa`

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

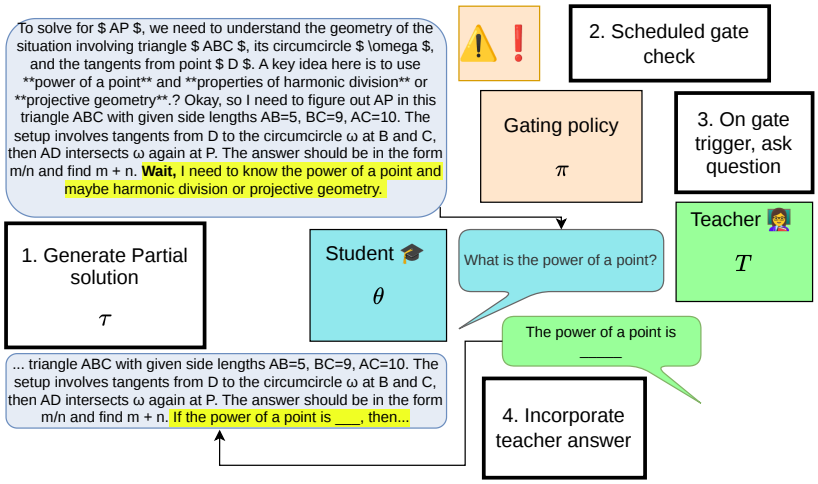


Figure 1: An illustration of our experimental setting. A student θ generates a partial solution τ for a target task. First, it chooses a step in the reasoning trajectory to pause the generation. At scheduled intervals determined by rule-based or external signal, we check a gating policy π. We sample n questions from the student and choose the best to be sent to the teacher T. After answering, the student incorporates the answer into a rewrite of the uncertain step.

perspectives or angles on a problem — and examine how this translates to self-recovery. We propose two specific hypotheses: First, if the reasoning tokens are meaningful, then introducing new semantic information would help steer the trajectory. Second, if a model is truly "proficient" at reasoning, it would be able to ask questions that can correct its own logic. The objective of this study is to leverage question-asking to predict the correctness of the reasoning trajectory, and to introduce interventions to question and recover from uncertain steps. We frame our study as a decision-making problem in a student-teacher setup, where a student is asking questions to a teacher and re-incorporating the teacher's answer into its own reasoning trajectory. We introduce a gating policy that uses hidden-state probing to assess two axes of good questions: (i) *which to ask*: what student questions would lead to the most fruitful outcomes? and (ii) *when to ask*: when is an intervention likely beneficial to the student?

Our contributions are summarized as follows:

- We demonstrate that prompting a reasoning model to generate questions at test-time produces a measurable shift in the hidden state *before* incorporating the answer. We train a hidden state probe and empirically demonstrate the probe's scoring is reasonably well-calibrated to the final correctness rate. Additionally, we observe the *range* of predicted correctness from a sampled set is well-correlated to correctness rate, more than per-set semantic diversity.
- We frame question-asking as a sequential decision problem in a student-teacher setting and propose a pipeline with a gating mechanism based on our probe-estimated correctness, to intervene on a reasoning trajectory when it is likely to be wrong. This pipeline frames question-asking as trajectory recovery, where we need to shift incorrect answer distributions towards a desired direction.
- We present thorough ablations to analyze the mechanisms of question-asking and find, despite identifying incorrect trajectories, there exists a gap between detection and recovery. We show that the success of an intervention is strongly correlated to baseline self-consistency. Even though we can predict correctness from the hidden state, the student does not necessarily utilize this signal to assess its own trajectory.

2 Related Work

Self-correction and self-refinement. There is a large body of work that studies whether language models can improve their own reasoning through iterative refinement Madaan et al. (2023) or, more

broadly, self-evolution Tao et al. (2024). Wang et al. (2022) propose *self-consistency*, which samples multiple chains-of-thought (CoTs) and aggregates via majority vote. This provides a critical baseline for test-time compute: if repeated sampling achieves the same gains as interaction, external feedback may be unnecessary. Zelikman et al. (2022) introduce STaR, where models iteratively refine reasoning using hints derived from correct answers. Similarly, Shinn et al. (2023) propose Reflexion, where models critique failed attempts and retry using feedback signals. These approaches demonstrate that retrying with additional information can improve performance. However, they rely on oracle signals (e.g., correctness labels or ground-truth answers) to guide refinement. Huang et al. (2023) and Tyen et al. (2024) demonstrate that LLMs struggle with self-correction in the absence of external feedback. We investigate whether there are internal signals of correctness, and whether or not they translate into action.

Asking clarifying questions. Several works study whether models can improve reasoning by asking follow-up questions. Self-Ask (Press et al., 2023) prompts a model to decompose problems into sub-questions that it asks itself, or feeds into an external search engine. Prior work in dialogue and information retrieval (Rao and Daumé III, 2018; Aliannejadi et al., 2019) studies when clarification questions are worth asking, framing the problem as a trade-off between cost and expected utility. Our work differs in two key ways. First, we explicitly separate the *student* (reasoner) and *teacher* (answering model), allowing us to analyze the effect of interaction. Second, we evaluate whether asking questions provides value beyond simply triggering regeneration.

Internal representations and probing. There are many works that study what information is encoded in neural representations via probing classifiers (Belinkov, 2022; Wang and Xu, 2025; Press et al., 2023). Zhu et al. (2025) finds that it is possible to predict the final correctness from the input question with a linear probe, which they use to inform their budget for test-time scaling. In contrast, we use the hidden state shift over a question and provide detailed analysis of the actual utility of our probe at inference time.

3 METHODOLOGY

3.1 SETUP

Our setup is illustrated in Figure 1. We frame our study as a sequential decision problem, where a student is gated by an external policy to ask questions to a teacher. To focus on the model's capacity for self-correction, most of our study is self-play — i.e. the same model plays the role of both student and teacher with the same available information. Let x denote a problem instance and $y \in \mathcal{Y}$ a discrete answer. A student model θ induces a distribution $p_\theta(y \mid x)$ over answers by sampling a full rollout. A question q posed to a teacher p_T produces a response a, after which the student updates its answer distribution to $p_\theta(y \mid x, q, a)$.

The intuitive idea is simple; we do not need to intervene on problems the student already answers well on its own. We wish to find a score s_t that measures the expected quality of the current trajectory. At a time step t, we decide whether to trigger the intervention. Let $\tau_t = (x, z_{1:t})$ denote the partial reasoning trajectory. We intervene on the partial reasoning trajectory with a gating policy $\pi(\tau_t)$, formally defined as the following threshold decision at step t:

$$\pi(\tau_t) = \begin{cases} \pi(\cdot \mid \tau_t), & s_t \leq \gamma \\ \varnothing, & \text{otherwise.} \end{cases} \tag{1}$$

3.2 PROBING TEST-TIME QUESTION-ASKING

Using this definition of question-asking as a decision made over a partial trajectory τ_t, we seek to predict the effectiveness of question-asking as a test-time intervention strategy. We study the value of the question itself, and also the timing of the question — that is, does there exist a critical point where a trajectory can be rescued and guided towards correctness. Empirically, we find that we can predict $p_\theta(y \mid x, q, a)$ using a probe trained on the hidden state shift induced by the question q. Using this probe, we can approximate the correctness of the current reasoning trajectory. We utilize this

correctness score to decide whether or not to ask a question, formulated as a sequential decision problem.

Belief State Probe. We measure the belief state as the answer distribution obtained by sampling full rollouts from a reasoning trajectory. For a partial reasoning trajectory τ_t, the student maintains a distribution Y_t over the answer space obtained by sampling continuations, i.e. $Y_t := p_\theta(y \mid \tau_t)$. At time t, the student samples a question q_t from a feasible set $\mathcal{Q}_\theta(\tau_t)$, and updates the context to the triplet (τ_t, q_t, a_t). The answer distribution after a QA intervention Y_q is defined as:

$$Y_q := p_\theta(y \mid \tau_t, q_t, a_t).$$

We train a hidden state probe to predict the correctness rate at time step t *before generating the teacher answer*. That is, for a hidden state h_θ, we train a probe to predict the correctness rate $P(y_q = y\star)$ where $y_q \sim Y_q$. We take the hidden state as the embedding of the last token at the final layer of the model, i.e. the same embedding used for token prediction. We concatenate the hidden states $h_\theta(\tau_t)$ and $h_\theta(\tau_t, q_t)$, before and after asking the question.

Question-Asking as a Test-time Intervention We formulate what to ask as the question that maximizes the expected intervention outcome, or maximally increases Δ_{acc}. Next, we discuss the timing — when to ask — as a decision made over the belief state τ. Unlike Equation (4), we assume no access to the baseline answer distribution $\mathbb{P}_{y\sim p_\theta(\cdot|x)}(y = y^\star)$. We sample N discrete questions $Q(\tau_t)$, and take the score that triggers the gating policy $\pi(\tau_t)$ as the following:

$$s_t = \min_{q\in Q(\tau_t)} \mathbb{E}y \sim p_\theta(\cdot \mid \tau_t, q)[\mathbf{1}[y = y^\star]] \tag{2}$$

Note the minimum term — this again derives from the idea of treating a hidden state as a belief state, and asking questions is an articulation of possible trajectories that could occur from a specific reasoning trajectory. If there is at least one question sample that is predicted to lead to a weak belief state, then the gate triggers. This is intuitively equating model uncertainty to trajectory variance; that is, if a bad final state is reachable from the current reasoning step, then that can be treated as information on the trajectory as a whole. We then choose the **maximum**-scoring question from the sampled set to ask the teacher, the teacher answers, and we *rewrite* the step triggered with the teacher's answer in context.

Gate Scheduling Another benefit of studying math problems is that there are many well-trained process reward models (PRMs) that provide an external source of correctness. We test several timing strategies to decide *when* to check the gating policy:

- **Fixed interval:** Always trigger the gate check every n reasoning steps. Additionally, we run m iterations of the QA loop and keep the final answer at iteration m.
- **Adaptive** (PRM-weak): Only trigger the gating decision on steps where the PRM score falls below a threshold α. We additionally continue to employ the idea of answer mass as an early stopping criteria — that is, if all sampled trajectories exceed the threshold and there is low variance between the samples, we assume sufficiency and stop.

3.3 Measuring Intervention Quality

We study the effect of test-time interaction through the change in the student's answer distribution. This can be conceptualized as a Jensen-Shannon divergence between the student's distribution in the answer space pre- and post- question, i.e. $\Delta_q(x) = D_{\mathrm{JS}}\big(p_\theta(y \mid x) \,\|\, p_\theta(y \mid x, q, a)\big)$. We refer to $\Delta_q(x)$ as the *distributional shift* induced by question q.

The quantity $\Delta_q(x)$ captures the magnitude of change, but not whether the change improves correctness. One primary goal of this study is to assess the direction of the change in trajectory induced by a question q, such that we can distinguish and characterize helpful from harmful questions. Let $y^\star$ denote the correct answer.

We define the expected accuracy change induced by a question and a sampled answer as:

$$\Delta_{\mathrm{acc}}(q; x) = \mathbb{P}_{y\sim p_\theta(\cdot|x,q,a)}(y = y^\star) - \mathbb{P}_{y\sim p_\theta(\cdot|x)}(y = y^\star) \tag{3}$$

In practice, we estimate this quantity via K discrete rollouts sampled over the reasoning trajectory with and without the question:

$$\hat{\Delta}_{\text{acc}}(q; x) = \frac{1}{K}\sum_{i=1}^{K} \mathbf{1}[y_{i,q} = y^\star] - \frac{1}{K}\sum_{i=1}^{K} \mathbf{1}[y_i = y^\star] \tag{4}$$

where $y_{i,q} \sim p_\theta(\cdot \mid x, q, a)$ and $y_i \sim p_\theta(\cdot \mid x)$.

An intervention by our pipeline is *helpful* if $\Delta_{\text{acc}}(q; x) > 0$ and *harmful* if $\Delta_{\text{acc}} < 0$. The expected accuracy gain $\Delta_{\text{acc}}(q; x)$ involves two sources of stochasticity: (i) the teacher response $a \sim p_T(\cdot \mid x, q)$ and (ii) the student's decoding $y \sim p_\theta(\cdot \mid x, q, a)$. This Δ_{acc} formulation is meant to capture the benefit of question-asking compared to the counterfactual (no intervention), but it is measured post-hoc. We do not assume access to the baseline answer at inference time. In other words, an optimal intervention is one that maximizes this shift in the answer distribution, not necessarily one where the right answer is reachable but rarely sampled. We measure this by reporting majority vote accuracy.

4 Experimental Settings

Models We use Qwen3 Yang et al. (2025) at 1.7B, 4B, and 14B parameters. For the PRM, we use Qwen2.5-Math-PRM-7B Zhang et al. (2025b). We theorize Qwen3 models are well-calibrated to math correctness, and have a stronger signal to be surfaced from their hidden state. To analyze the transferability to another model family, we also include results on Olmo-3-7B-Think Olmo et al. (2025).

We mostly test math reasoning to study the correlation between proficiency and recovery — Qwen3 models already achieve strong baseline performance on these datasets, which allows us to test self-correction on known capability. Also, process reward models are more calibrated on verifiable reasoning, so they are an effective external signal.

Datasets We evaluate on three math datasets:

- GSM8k Cobbe et al. (2021) — 1,319 samples of grade school level word problems in math. We take 100 samples of the train split for training our hidden state probe. oh
- Math500 Hendrycks et al. (2021) — 500 samples of competition-level math. We take the first 15 samples as a training set for the hidden state probe, and report results on the 485 remaining samples.
- AIME24 Zhang and Math-AI (2024) — A held-out set of 30 Math Olympiad questions from 2024.

Hyperparameters For our gating threshold, we choose γ to be 0.6 after a hyperparameter sweep. We set n to be 5 reasoning steps (split by newlines, not individual sentences), and m to be 5, as we find that there are minimal gains beyond 5 interventions. Unless otherwise specified, the default metric is Majority Vote accuracy over $K = 16$ rollouts. Since we report majority vote, we use bootstrapping to approximate confidence intervals in our results. Please see Appendix A for more details.

5 Results

5.1 Hidden state probe accuracy

Correlation to actual correctness. We take the predicted correctness rate from our probe in each model, and measure it against the actual correctness rate of the final rollouts. Our probe is reasonably calibrated to actual correctness; we achieve Brier scores of 0.157 for Qwen3-1.7B to 0.121 for Qwen3-14B *without the teacher answer*. The calibration improves linearly with model size; the full list of scores along with p-value significance can be found in Appendix B. We tested various formulations — including just the pre-question embedding $h_\theta(\tau_t)$ or just the post-question embedding $h_\theta(\tau_t, q)$ — but the combination of pre+post was the most predictive of final correctness rate. This exemplifies the concept of hidden state shift; the question itself induces a change in the hidden state representation that is indicative of final performance.

Table 1: Our main results as compared to a zero-shot baseline. (nt) indicates Non-thinking and (t) indicates Thinking. We report Mean Questions as questions that reached the teacher, accuracy under majority vote over 5 and 16 samples (maj@5, maj@16), and mean tokens per generation. *Adaptive*: PRM-weak steps and early stopping; *Fixed*: fixed interval. The subscript reports the 95% confidence interval as offsets from the point estimate.

			Accuracy, maj@5 (%)			Mean Questions			Mean Tokens		
Model	**Setting**	**Mode**	**GSM**	**MATH**	**AIME**	**GSM**	**MATH**	**AIME**	**GSM**	**MATH**	**AIME**
Qwen3-1.7B	Baseline	(nt)	$84.9_{[-2.0,+2.0]}$	$76.5_{[-3.9,+3.9]}$	$16.7_{[-13.4,+13.3]}$	—	—	—	265	871	2,933
	Fixed	(nt)	$84.3_{[-1.9,+2.0]}$	$75.9_{[-3.7,+3.7]}$	$13.3_{[-10.0,+13.4]}$	0.3	2.1	3.8	1,356	5,391	16,772
	Adaptive	(nt)	$86.1_{[-1.9,+1.9]}$	$76.9_{[-3.7,+3.9]}$	$13.3_{[-10.0,+13.4]}$	0.1	0.5	1.2	318	1,260	4,584
	Baseline	(t)	$91.4_{[-1.4,+1.5]}$	$92.8_{[-2.5,+2.3]}$	$55.6_{[-14.1,+14.2]}$	—	—	—	1,380	5,074	13,570
	Adaptive	(t)	$91.6_{[-3.1,+2.8]}$	$92.0_{[-2.3,+2.2]}$	$56.8_{[-15.3,+14.8]}$	0.1	1.0	2.0	840	4,932	14,388
Qwen3-4B	Baseline	(nt)	$90.6_{[-2.1,+1.9]}$	$86.6_{[-3.1,+2.9]}$	$36.7_{[-16.7,+16.6]}$	—	—	—	270	698	1,271
	Fixed	(nt)	$92.6_{[-1.5,+1.3]}$	$86.0_{[-2.3,+3.7]}$	$28.6_{[-14.3,+17.8]}$	0.0	1.0	2.3	1,374	4,885	20,523
	Adaptive	(nt)	$93.1_{[-1.4,+1.4]}$	$85.6_{[-2.6,+3.2]}$	$30.0_{[-16.7,+16.7]}$	0.0	0.2	1.0	296	1,104	5,317
	Baseline	(t)	$95.2_{[-1.2,+1.2]}$	$96.7_{[-1.8,+1.8]}$	$76.7_{[-16.7,+13.3]}$	—	—	—	2,327	6,897	12,259
	Adaptive	(t)	$96.1_{[-2.1,+0.3]}$	$95.8_{[-1.8,+1.6]}$	$80.8_{[-14.4,+12.2]}$	0.0	0.6	0.7	568	3,207	5,863
Qwen3-14B	Baseline	(nt)	$96.0_{[-2.2,+2.0]}$	$88.5_{[-3.1,+3.1]}$	$43.3_{[-16.6,+16.7]}$	—	—	—	192	840	4,324
	Fixed	(nt)	$95.2_{[-1.6,+1.3]}$	$87.8_{[-3.1,+2.7]}$	$40.0_{[-16.7,+20.0]}$	0.1	3.9	3.1	1,115	4,058	17,730
	Adaptive	(nt)	$95.4_{[-1.3,+1.1]}$	$89.1_{[-2.9,+2.7]}$	$36.7_{[-16.7,+16.6]}$	0.0	0.2	0.9	200	979	3,915
	Baseline	(t)	$96.6_{[-1.1,+1.0]}$	$97.1_{[-2.9,+2.3]}$	$80.0_{[-30.0,+20.0]}$	—	—	—	2,699	4,611	11,871
	Adaptive	(t)	$96.6_{[-1.9,+1.8]}$	$96.0_{[-1.7,+1.5]}$	$81.8_{[-14.0,+12.3]}$	0.1	0.8	1.1	533	3,595	10,370
Olmo-3	Baseline	(t)	$94.9_{[-2.5,+2.3]}$	$91.2_{[-4.1,+3.6]}$	$76.7_{[-16.7,+13.3]}$	—	—	—	2,066	5,026	15,429
	Adaptive	(t)	$95.5_{[-2.3,+2.0]}$	$97.0_{[-1.6,+1.4]}$	$76.7_{[-16.7,+13.3]}$	0.0	0.3	1.0	2,087	7,668	33,243
	Fixed	(t)	$95.7_{[-2.2,+2.1]}$	$97.3_{[-1.9,+1.7]}$	$73.3_{[-16.6,+13.4]}$	0.3	1.0	2.0	10,103	31,212	93,340

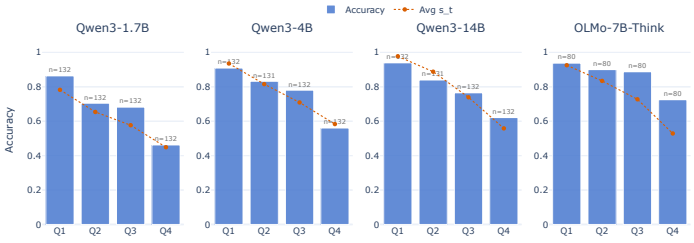


Figure 2: Majority vote accuracy vs. probe score range quartile across all belief-gated settings from Table 1, stratified by model. s_t indicates the minimum predicted probe score, which we use to gate the QA intervention.

Predictive power of probe score. For our fixed setting, we track the range of predicted probe scores as the diff between max. and min. scored correctness. The results are available in Figure 2. Binning the result into four quartiles, we observe that accuracy is negatively correlated with range; that is, the wider the range of sampled values, the lower the final model accuracy. This indicates that answer mass is detectable from the questions asked by the model, even before incorporating the teacher's answer, and this is strongly correlated with the student's final correctness. We also analyze the semantic diversity of the probe scores with a cheap TF-IDF metric, as shown in Figure 4. There is some weak relationship between semantic diversity and probe spread in the Qwen3 models — we measured ρ scores between 0.258–0.619 — but the correlation is weaker in Olmo-3-7B-Think ($\rho = -0.042$). This indicates the hidden state information learned by the probe is not strongly related to the semantics of the question.

5.2 Effectiveness of Question-Asking as an Intervention

Next, we use this hidden state probe score as a threshold value for our gating policy $\pi(\tau_t)$ in an LLM's reasoning trajectory. We seek to understand if this signal can be translated into a test-time intervention that can consistently detect and correct weaker trajectories. Our main findings are summarized in Table 1. maj@16 is reported for the baselines and Adaptive (t) on MATH and AIME; all other cells only report maj@5 due to computational constraints. We hypothesize that GSM8k's problem set is more suited to question-asking; there is, for example, a socratic setting of the original dataset where the step-by-step solution is paired with guided questions. However, after accounting for

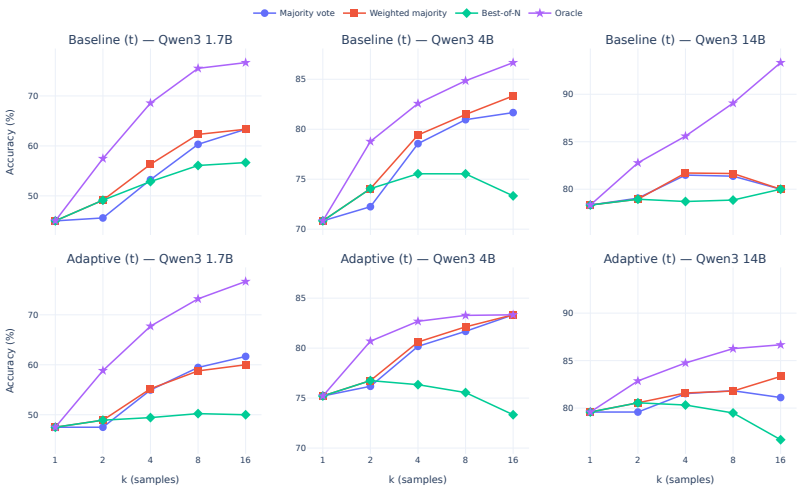


Figure 3: The effect of increasing sample counts over zero-shot and our question-asking pipeline on Qwen3 models in the AIME24 dataset.

confidence intervals, this gain is only significant on Qwen3-4B. Comparing Baseline (t) and Adaptive (t), we make consistent gains on most models except for Qwen3-1.7B on "hard" (Olympiad) math benchmarks, and Qwen3-14B on "easy" (grade-school) benchmarks. For Qwen3-1.7B we believe the model is too small to think of any effective question, whereas on Qwen3-14B, the model is already saturated on GSM8k and more likely to get confused on easier samples. To study Δ_{acc}, the rest of our ablations only use non-thinking mode. We conduct further analysis below.

Question-asking vs. increased thinking. Overthinking is well-studied in previous works Chen et al. (2024); Sui et al. (2025) — in thinking mode, Qwen3-14B uses 20x more tokens than non-thinking mode on GSM8k for only +0.6pp improvement. However, as shown in Table 1, our adaptive setting achieves greater token efficiency for the gained accuracy (+4.7 points accuracy compared to the non-thinking baseline for ≈2x the tokens in our adaptive setting on Qwen3-1.7B, compared to thinking which is +10.7 points accuracy for ≈10x the tokens). The gain is more apparent with the Adaptive (t) setting. Our QA pipeline allows the model to complete generation faster than zero-shot prompting (10x reduction in Qwen3-4B and Qwen3-14B). While there are no dramatic performance improvements (the Confidence Interval at 30 samples is $\approx \pm 15$pp), our adaptive setting observes consistent improvements.

We also observe our gating policy is weakly aligned to difficulty and supports the concept of token budgeting by naturally gating more for uncertain or weak reasoning steps. In Table 1, the mean number of backtracks typically increases with dataset difficulty. AIME24, our most challenging dataset, has the most mean backtracks per sample. For Math500 in our adaptive setting, the number of iterations is *weakly correlated* to the level of difficulty (ρ between 0.143-0.179 per model), which indicates that the samples models struggle with are only weakly correlated to human-perceived difficulty of the questions. we also find the probe score s_t is even more strongly correlated to the difficulty level. This indicates that s_t tracks difficulty more closely than the probe score range, even though these measures are correlated.

Relationship to scaling We additionally increase the sample count k in the adaptive setting to evaluate the effect of scaling, shown in Figure 3. We measured over standard metrics in test-time scaling such as weighted majority and Best-of-N (BoN), as determined by the PRM score. As shown by the Oracle setting (whether or not the correct answer was in any of the sampled trajectories), there is still a benefit to sampling more from the same model in both zero-shot and question-asking settings. It is also interesting to note that the BoN most preferred trajectory according to the PRM is often not the one preferred by majority vote, and the latter often produces a better outcome. Finally, we observe the scaling is more stable in our Question-asking setting than in the zero-shot setting. While the oracle scoring can achieve impressive scaling, especially in Qwen3-14B, the answer is not easily identified by majority vote or PRM-based scoring, indicating there is some randomness in how the trajectory is found. In contrast, the weighted majority in our question-asking setting increases

Table 2: Comparison of belief-state gating (with PRM-adaptive vs. fixed scheduling) vs. no gating for fixed intervention intervals across all models. The No-gating setting is only reported on Math500, while other settings use all datasets. We report the number of samples (n), number of samples that flipped to a different prediction (nf), and $\hat{\Delta}_{acc}$. We also report the Correct (C)/ Wrong (W) samples fired, the total pool of samples, and the marginal Δ_{acc} relative to the portion fired and the entire pool.

		Adaptive, Gating			Fixed, Gating			Fixed, No gating		
Direction	Bin	n	nf	**Δacc**	n	nf	**Δacc**	n	nf	**Δacc**
Correct	≤ 2/5	30	14	**+4.7**	60	29	**+1.3**	48	23	**+0.0**
	3/5	39	11	**+2.6**	88	30	**+2.3**	69	17	**+5.8**
	4/5	58	12	**-5.5**	137	22	**-7.0**	129	17	**-3.2**
	5/5	135	15	**-13.8**	582	25	**-6.4**	1,084	13	**-2.8**
Wrong	≤ 2/5	168	136	**+5.6**	204	170	**+6.4**	189	157	**+6.9**
	3/5	45	27	**+6.2**	69	40	**+9.3**	62	28	**+8.1**
	4/5	36	14	**+8.3**	48	17	**+10.0**	42	14	**+10.5**
	5/5	28	8	**+19.3**	45	7	**+8.0**	63	8	**+3.2**
Split (fire rate %)		262 C / 277 W (51.4% W)			867 C / 366 W (29.7% W)			1,330 C / 356 W (21.1% W)		
Pool n (fire rate %)		3,839 (14.0%)			4,188 (29.4%)			1,708 (98.7%)		
Δacc (fired)		**+0.22%**			−1.31%			−0.35%		
Δacc (pool)		**+0.031%**			−0.39%			−0.34%		

monotonically. This demonstrates that there is more stability in the trajectories sampled after our question-asking pipeline, which is desirable in applications that require a consistent output.

Measuring $\hat{\Delta}_{acc}$ and Δ_q in Question-asking. In Table 2, we isolate the benefit of our question-asking intervention from pure resampling by taking the subset of all samples and models where the gating policy triggered. We define a wrong sample as one where the baseline's majority-vote accuracy was wrong, and divide the results by correct vs. incorrect at the zero-shot baseline. The No Gating setting is where we intervene on every sample for the full m iterations, where wrong samples only represent 23.3% of the samples; belief-state gating with fixed scheduling fires on wrong samples even less frequently at 17%. In our adaptive setting, the gating policy uses the internal hidden state probe and an external PRM; of the trajectories caught by this policy, 51.4% were wrong in their measured baseline. This demonstrates the policy itself is efficient at catching samples that are likely wrong.

In confident baselines (all correct or all incorrect), δ_{acc} increases monotonically with selectivity (+3.2 to +8.0, then +19.29). We observe samples that were incorrect can recover to a correct answer, but our intervention can also *harm* confidently correct samples; for the fixed trigger scheduling, $|\hat{\Delta}_{acc}|$ is *greater* in magnitude on confidently correct samples than on confidently incorrect ones. While there are many samples that flip predictions in less confident bins (e.g. 136/168 had some shift in the adaptive setting, ≤2/5 bin), the magnitude of $\hat{\Delta}_{acc}$ is low because the trajectory does not shift towards the right answer. The marginal Δ_{acc} is relatively even, within 1% pp of the baseline.

This asymmetry reflects the same mechanism identified by Kumaran et al. (2025), where a model is more susceptible to changing its opinion when it is underconfident. Our question-asking intervention implicitly signals doubt in the current trajectory and the model treats this as necessary feedback, updating its distribution even when the original answer was correct. Additionally, the PRM threshold filter also catches samples where, even though the trajectory is confidently correct, might contain reasoning steps where the model is less confident.

5.3 What makes a good question?

Quantifying question diversity We measure diversity along two dimensions: sample diversity, i.e. how many of the $m = 8$ questions are unique, and semantic diversity, i.e. how unique is the semantic content of the documents. We use TF-IDF embeddings for fast calculation.

Effect of the teacher's answer In Figure 5, we additionally analyze the effect of the teacher answer on the performance of our weakest model, Qwen3-1.7B. We compare self-play to a stronger model, Qwen3-14B, on GSM8k and Math500, and examine the dynamics of the fixed setting over early stopping at the first iteration, vs. the final output after five iterations. Without an adaptive scheduler, the fixed setting of our pipeline actively removes all benefits of the strong teacher and *results in worse performance than self-play*. However, the adaptive setting slightly improves the

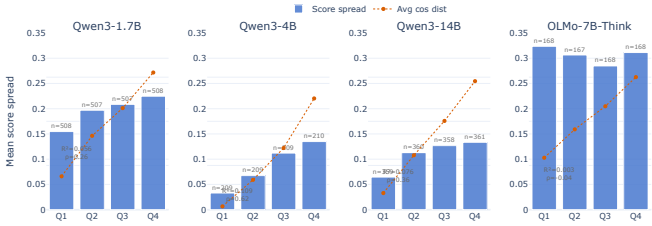
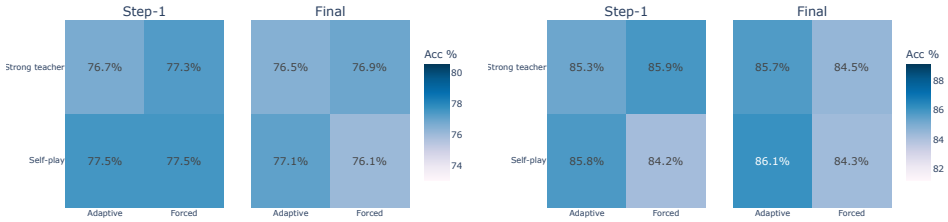


Figure 4: Correlation between semantic diversity (measured as cosine similarity over TF-IDF embeddings) versus probe score on the Adaptive setting, averaged across all datasets. The semantic diversity is divided into four quartiles and plotted against the mean range in each quartile.



(a) Math500. (b) GSM8k.

Figure 5: Effect of teacher model and multiple question-asking iterations on Qwen3-1.7B, measured as Majority vote accuracy on Math500 and GSM8k. Self-play is where the student and teacher model share weights, and *Strong teacher* is where we use Qwen3-14B as the teacher.

best teacher performance. This suggests there exists an ideal timing where an intervention helps steer an otherwise incorrect trajectory towards the correct answer. We theorize the adaptive setting has slightly more positive shifts because it is selective and allows more targeted interventions, even when we observe the PRM is poor at selecting the most correct answer in the BoN metric. However, these are not significant considering the confidence intervals of these experiments. The performance improvements are still bounded by the quality of the questions asked by the student, and the baseline model uncertainty observed in Table 2. We then use simple regex string matching to categorize different types of teacher answers. We observe four categories from our MATH500 experiments:

Category	(t) MATH, $n = 1136$	(nt) MATH, $n = 472$
Confirm	22.0%	24.6%
Correct	3.3%	5.7%
Hedge	1.3%	2.8%
other	73.4%	66.9%

6 Conclusion

This work investigates the dynamics of question-asking as a test-time intervention. Our study shows that (1) models partially encode their own correctness in the hidden state shift induced by asking a question; (2) this signal is weakly correlated to difficulty in a way that external PRM signals are not; and (3) there is a gap between detecting this signal and recovering the trajectory. While interrupting a reasoning model with a question prompt can change the overall trajectory, there is no consistent shift towards correctness. Instead, we discover that the rate of distribution shift is more correlated with the model's confidence, where targeting uncertainty is also likely to cause the model to second-guess correct trajectories. We demonstrate that comparing logprob-based selection to our probe-based scoring, the best question is often not the most probable one. These results demonstrate a gap between hidden state detection and recovery in the reasoning trace. LLMs still struggle to identify flaws in their own logic, even though the signal exists in the hidden state. This gap has broader implications on self-refinement, especially when the model is uncertain, and we encourage further research to improve the model's inherent ability to self-diagnose or understand its own actions.

REFERENCES

M. Aliannejadi, H. Zamani, F. Crestani, and W. B. Croft. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 475–484, 2019.

R. B. Ambati, T. Niu, A. Singh, S. Mishra, S. Chaturvedi, and S. Srivastava. Socratic students: Teaching language models to learn by asking questions. *arXiv preprint arXiv:2512.13102*, 2025.

Y. Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, Mar. 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/.

K. Chang, Y. Shi, C. Wang, H. Zhou, C. Hu, X. Liu, Y. Luo, Y. Ge, T. Xiao, and J. Zhu. Step-level verifier-guided hybrid test-time scaling for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18473–18488, 2025.

X. Chen, J. Xu, T. Liang, Z. He, J. Pang, D. Yu, L. Song, Q. Liu, M. Zhou, Z. Zhang, et al. Do not think that much for 2+3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Z. Hu, C. Liu, X. Feng, Y. Zhao, S.-K. Ng, A. T. Luu, J. He, P. W. Koh, and B. Hooi. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=CVpuVe1N22.

A. Huang, A. Block, Q. Liu, N. Jiang, A. Krishnamurthy, and D. J. Foster. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment. *arXiv preprint arXiv:2503.21878*, 2025.

J. Huang, X. Chen, S. Mishra, H. S. Zheng, A. W. Yu, X. Song, and D. Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.

S. Kambhampati, K. Stechly, K. Valmeekam, L. P. Saldyt, S. Bhambri, V. Palod, A. Gundawar, S. R. Samineni, D. Kalwar, and U. Biswas. Stop anthropomorphizing intermediate tokens as reasoning/thinking traces! In *NeurIPS 2025 Workshop on Bridging Language, Agent, and World Models for Reasoning and Planning*, 2025.

M. Kazemi, B. Fatemi, H. Bansal, J. Palowitch, C. Anastasiou, S. V. Mehta, L. K. Jain, V. Aglietti, D. Jindal, Y. P. Chen, et al. Big-bench extra hard. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26473–26501, 2025.

D. Kumaran, S. M. Fleming, L. Markeeva, J. Heyward, A. Banino, M. Mathur, R. Pascanu, S. Osindero, B. De Martino, P. Velickovic, et al. How overconfidence in initial choices and underconfidence under criticism modulate change of mind in large language models. *arXiv preprint arXiv:2507.03120*, 2025.

X. Li, E. Callanan, A. Ghassel, and X. Zhu. Entropy-gated branching for efficient test-time reasoning. In V. Demberg, K. Inui, and L. Marquez, editors, *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5054–5069, Rabat, Morocco, Mar. 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.235. URL https://aclanthology.org/2026.eacl-long.235/.

A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

N. Muennighoff, Z. Yang, W. Shi, X. L. Li, L. Fei-Fei, H. Hajishirzi, L. Zettlemoyer, P. Liang, E. Candès, and T. Hashimoto. s1: Simple test-time scaling. In C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20275–20321, Suzhou, China, Nov. 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

T. Olmo, A. Ettinger, A. Bertsch, B. Kuehl, D. Graham, D. Heineman, D. Groeneveld, F. Brahman, F. Timbers, H. Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, 2023.

S. Rao and H. Daumé III. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In I. Gurevych and Y. Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2737–2746, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1255. URL https://aclanthology.org/P18-1255/.

N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

C. Snell, J. Lee, K. Xu, and A. Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Y. Sui, Y.-N. Chuang, G. Wang, J. Zhang, T. Zhang, J. Yuan, H. Liu, A. Wen, S. Zhong, N. Zou, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

Z. Tao, T.-E. Lin, X. Chen, H. Li, Y. Wu, Y. Li, Z. Jin, F. Huang, D. Tao, and J. Zhou. A survey on self-evolution of large language models. *arXiv preprint arXiv:2404.14387*, 2024.

G. Tyen, H. Mansoor, V. Cărbune, Y. P. Chen, and T. Mak. Llms cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13894–13908, 2024.

X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Z. Wang and C. Xu. ThoughtProbe: Classifier-guided LLM thought space exploration via probing representations. In C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6018–6039, Suzhou, China, Nov. 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.307. URL https://aclanthology.org/2025.emnlp-main.307/.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

E. Zelikman, Y. Wu, J. Mu, and N. Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Q. Zhang, F. Lyu, Z. Sun, L. Wang, W. Zhang, W. Hua, H. Wu, Z. Guo, Y. Wang, N. Muennighoff, et al. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*, 2025a.

Y. Zhang and T. Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Z. Zhang, C. Zheng, Y. Wu, B. Zhang, R. Lin, B. Yu, D. Liu, J. Zhou, and J. Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10495–10516, 2025b.

Y. Zhu, D. Liu, Z. Lin, W. Tong, S. Zhong, and J. Shao. The llm already knows: Estimating llm-perceived question difficulty via hidden representations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1160–1176, 2025.

LIMITATIONS

We would like to highlight several limitations of this study. Due to computational limitations, we only sample 5 trajectories in our core comparisons table. While we believe the principles discussed are applicable to non-verifiable domains, the contents of this study are intentionally limited to mathematical reasoning, over a model that was optimized to the test sets we report. We leave exploration of other domains, including logical reasoning, for future work.

A EXTENDED EXPERIMENTAL DETAILS

For the hidden-state probe, we use a 64-dim MLP architecture, trained with a regression objective over self-consistency for the belief state prediction. We trained each probe for 10 epochs and a learning rate of 1e-3, over offline trajectories sampled from a mix of 100 samples of GSM8k, 15 from Math500, and 15 from Big-Bench Extra Hard Kazemi et al. (2025). We used a special data collection setting where, on the fixed intervention schedule, we sample self-play teacher answers, and five unique rollouts per question. See Table 5 for the validation accuracy results.

For our gating threshold, we choose γ to be 0.6 after an initial hyperparameter sweep on the training data at the intervals [0.1, 0.3, 0.5, 0.7, 0.9]. We set n to be 5 reasoning steps (split by newlines, not individual sentences), and m to be 5. We experimented with $m = 15$, but find that there are minimal gains beyond 5 interventions in one reasoning trajectory. We measure the efficiency of the "natural" generation for a model — in other words, we allow the model to generate until the EOS token.

All experiments are performed on a cluster of Nvidia H100 80GB GPUs. Experimental time depends on the model and experimental setting, ≈6-20 GPU hours per experiment run. All experiments were run on vLLM with fp16 precision, including the hidden state probe - we modified vLLM to host the hidden state and feed a dual-head classifier and inference endpoint, so the probe and the token generation shared the same weights and KV cache. For the sake of efficiency in the KV cache we reserved one GPU per model, so one experiment uses 1-3 GPUs depending on the student-teacher model configuration and the presence of a PRM.

For Qwen3, except for the thinking baseline, we use non-thinking mode and the recommended decoding settings: temperature at 0.7, top-p at 0.9, and top-k at 20. Olmo-3-7B-Think has thinking mode on by default, and we also use the recommended sampling settings (temperature=0.6, top-p=0.95). We also include the full system prompts in Table 3 and user message templates in Table 4. For evaluation, we used standard math evaluation libraries available through lm-evaluation-harness[2]. To estimate confidence intervals on our main results we used bootstrapping — we resample with replacement 1000 times, recalculate the majority-vote accuracy, and report the 2.5th and 97.5th percentile as the general confidence interval.

All artifacts used in this work are released in the public domain. Qwen3, Olmo-3, and AIME24[3] all use Apache 2.0 licenses, while Math500[4] and GSM8k[5] were released under MIT licenses. Coding agents were used to write code and monitor and re-submit experiments, and a human manually inspected and hand-corrected the generated code. LLMs were not used to formulate the core methodology or inform experimental decisions.

B ADDITIONAL EXPERIMENTAL RESULTS

Intervention strategies We compare two intervention policy choices for $\pi(\cdot \mid \tau_t)$ to analyze the effect of available information on the final success:

- **Backtrack:** Framing the intervening step as a "bad" step, we prompt the model to re-generate the step based on the teacher's feedback but *without seeing the original question.*
- **Rewrite:** Our full QA intervention. The student has access to their original question, the teacher's answer, and they re-generate the step with full information.

[2] https://github.com/EleutherAI/lm-evaluation-harness
[3] https://huggingface.co/datasets/math-ai/aime24
[4] https://github.com/hendrycks/math/blob/main/LICENSE
[5] https://github.com/openai/grade-school-math/blob/master/LICENSE

Table 3: System prompts used in Pipeline v5. Each prompt is assigned to a specific role in the student–teacher QA loop.

Prompt Name	Content
Default System Prompt (Qwen3 recommended setting)	You are solving a problem step by step. Show your reasoning clearly. Separate each reasoning step with a blank line. Put your final answer within `\boxed{}`.
Continuation	You are continuing your reasoning after receiving feedback on a previous attempt. Incorporate the feedback and continue from where you left off. Separate each reasoning step with a blank line. Put your final answer within `\boxed{}`.
Student Question	You are working through a logical reasoning problem step by step. You have access to a knowledgeable teacher (oracle) who can answer questions about the problem's facts and rules. You are reviewing a specific step in your reasoning — ask the teacher a question about that step that will help you answer the overall problem. DO NOT answer the question yourself. Do not ask the teacher for the final answer or whether an answer is correct — the teacher will not respond to such questions. Ask about specific facts, rules, or intermediate steps instead.
Teacher	You are a knowledgeable oracle for a logical reasoning problem. Answer the student's question using ONLY the facts and rules provided. Do not speculate or add information not present in the problem. Your task is to answer any questions the student may have, but DO NOT PROVIDE THE CORRECT ANSWER DIRECTLY. If the student asks you whether an answer is correct, do not respond. Keep your responses short and concise — address only the specific fact, rule, or intermediate step the student raised.
Full Answer	You have been exploring a problem step by step and received feedback from a teacher. Based on your exploration and the feedback, produce a complete solution. Show your reasoning clearly and put your final answer within `\boxed{}`.
Rewrite Step†	You are solving a problem step by step. Show your reasoning clearly. Separate each reasoning step with a blank line. Put your final answer within `\boxed{}`. Important: `{teacher_response}`

†Dynamic prompt: `{teacher_response}` is replaced with the teacher's answer, stated as a bare fact so the model naturally incorporates it.

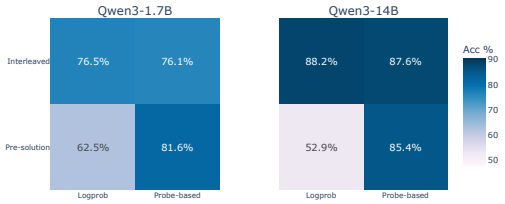


Figure 6: Comparing available model context vs. question selection on Math500 results across Qwen3-1.7B and Qwen3-14B. Pre-solution is where we ask all questions before the student attempts to solve the problem, and interleaved is where the student first generates n reasoning steps before we sample a question. Finally, we compare sampling by highest logprob vs. our probe-based probable correctness scoring.

These results are available in Table 6.

Table 4: User message templates in Pipeline v5. Placeholders in `braces` are filled at runtime. Each template is paired with the corresponding system prompt from Table 3.

System Prompt	User Message Content
Baseline	`{task_question}`

Continuation	Problem:
	`{task_question}`
	
	Feedback from previous attempt:
	`{prior_feedback}`
	
	Continue solving the problem, incorporating the feedback above.

Student Question	Problem: `{task_question}` Your reasoning so far: `{partial_solution}` The step you are reviewing: > `{uncertain_step}` Ask a probing question that stress-tests this step.

Teacher	Full problem:
	`{problem_text}`
	
	Student's work so far:
	`{partial_solution}`
	
	Student's question: `{question}`
	Answer using only the facts and rules above.

Full Answer	Problem:
	`{task_question}`
	
	Your exploration so far:
	`{exploration}`
	
	Teacher feedback:
	Q1: `{q}` / A1: `{a}` ...
	
	Based on the exploration and feedback above, determine the answer. Show your complete reasoning and put your final answer within `\boxed{}`.

Rewrite Step	`{problem_text}` (Prior clean steps appended as generation prefix)

B.1 Scaling on Math500

B.2 Qualitative analysis of Questions

While we originally had some clustering analysis of the questions, which is where the semantic diversity graphs from Figure 3 were derived from, we omitted them from the final draft for brevity. Here are some example clusters from that study:

Cluster 0 ($n = 263$)**:**

- What if the number of junior programmers is not $2/5$ of the total number of employees, but instead $2/5$ of the total number of senior programmers? Would that change the calculation?

Table 5: Belief-state regressor training results. The new regressor uses concatenated pre-question and post-question hidden states with a cons-only prediction target, compared to the old regressor which used pre-question hidden states only with a dual cons + Δcons target. Variance explained is $1 - \text{MSE}/\text{Var}(y)$, where $\text{Var}(y) = \bar{y}(1 - \bar{y})$ is the Bernoulli variance of the cons labels.

	Model	Pairs	Frac. wrong	Val MSE	Target var	Var. explained
Old	Qwen3-1.7B	511	0.881	0.049	0.122	60%
	Qwen3-4B	926	0.773	0.073	0.175	58%
	Qwen3-14B	39	0.949	0.272	0.049	0%
New	Qwen3-1.7B	988	0.598	0.048	0.243	80%
	Qwen3-4B	1533	0.639	0.037	0.184	80%
	Qwen3-14B	556	0.536	0.113	0.249	55%

Table 6: Backtrack vs. rewrite correction (PRM-triggered, belief-gated, adaptive). Acc and MV are accuracy of last-step answer and majority vote (%); Bt is mean backtracks per sample; Tok is mean tokens per sample.

		Backtrack				Rewrite			
Model	Dataset	Acc	MV	Bt	Tok	Acc	MV	Bt	Tok
Qwen3-1.7B	GSM8K	93.2	85.6	0.0	359	91.0	86.1	0.1	318
	MATH500	80.0	73.2	0.2	1201	77.5	72.4	0.5	1260
	AIME24	23.3	20.0	0.8	5180	23.3	13.3	1.2	4584
Qwen3-4B	GSM8K	95.2	92.5	0.0	286	95.4	92.9	0.0	296
	MATH500	86.0	80.2	0.1	1152	90.8	85.6	0.2	1104
	AIME24	36.7	23.3	0.0	5439	33.3	30.0	1.0	5317
Qwen3-14B	GSM8K	97.7	96.6	0.0	207	96.6	95.2	0.0	200
	MATH500	85.6	84.7	0.1	1338	86.0	83.7	0.2	979
	AIME24	50.0	30.0	0.0	4600	40.0	33.3	0.9	3915

- How does the principle of inclusion-exclusion apply when we are given the total number of students, the number in each club, and the number in neither? What assumptions are being made that could potentially affect the result?
- What if the total number of students in the hall was not 1000, but a different number, say 1001? Would the reasoning still hold?

Cluster 1 ($n = 71$)**:**

- What if the cost of lemons or sugar was not per gallon but per glass? Would that affect the calculation of total profit?
- *Probing Question:* Assuming the ratio of sugar to water is 7:13, and the total amount of sugar is 120 teaspoons, does the calculation correctly account for the fact that the amount of water is in cups, not teaspoons? What if the conversion from cups to teaspoons is not fixed or known?

Cluster 2 ($n = 641$)**:**

- Is there any possibility that the fraction $2/5$ refers to something other than the proportion of boys among the total Grade 5 students, and could this affect the calculation?
- Is there any possibility that the first 3 customers did not each buy exactly 1 DVD, and if so, how would that affect the calculation?

Cluster 3 ($n = 62$)**:**

- *Question:* Is there a possibility that the assumption that is isosceles with leads to an incorrect conclusion about the angle , and does this affect the validity of the final answer?
- Is there a guarantee that the given side lengths satisfy the triangle inequality, ensuring that the triangle (and hence the tetrahedron) can exist?

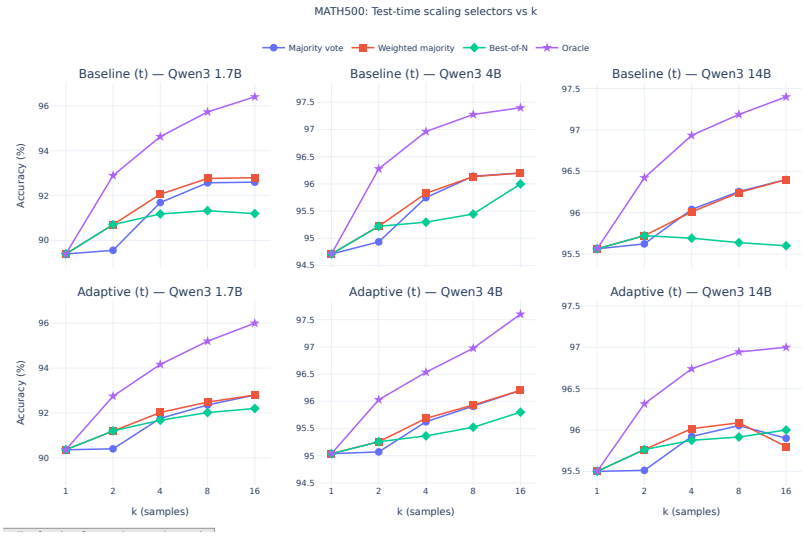


Figure 7: The effect of increasing sample counts over zero-shot and our question-asking pipeline on Qwen3 models in the AIME24 dataset.

Cluster 4 ($n = 166$)**:**

- What if there were more than one type of mutant invertebrate, each with a different number of legs? Would the calculation still hold?
- What if there were more than one type of mutant invertebrate, and some had 8 legs while others had 6? Would the calculation still hold?

Cluster 5 ($n = 139$)**:**

- Is the "half as long" referring to half the time he spends watching TV, or half the total time spent on both TV and reading in each session?
- Is there a possibility that "half as long" could refer to half the total time spent on both TV and reading in a single session, rather than half the time spent watching TV?

Cluster 6 ($n = 166$)**:**

- *Question:* Given the expression , is it possible that the minimum value of this expression is 100? Why or why not?

Table 7: Accuracy (%) across models, datasets, and intervention strategies compared against a zero-shot baseline. The best results are in **bold**, and *strong* results are within 1 point of accuracy of the best performance. — indicates results are not available as there is no equivalent non-thinking setting for Olmo-3-Think. ⋆ indicates we obtained the value from Yang et al. (2025)

			Baseline		Front-loaded			Interleaved		
			non-thinking	thinking	logprob	probed	random	Fixed	Adaptive	Adaptive (t)
Qwen 3-1.7B	GSM8K	P@5	92.6	**95.2**	25.2	88.0	78.8	87.9	91.0	*94.2*
		Maj	84.8	*91.5*	24.6	85.5	76.3	84.2	86.1	**92.2**
	MATH500	P@5	81.6	*95.2*	67.6	77.9	70.9	76.5	75.9	**96.0**
		Maj	74.8	**92.8**	59.6	76.3	61.9	70.1	70.9	*92.6*
	AIME24	P@5	30.0	48.3⋆	10.0	–	0.0	20.0	23.3	**66.7**
		Maj	20.0	48.3⋆	10.0	–	0.0	13.3	13.3	**53.3**
Qwen 3-4B	GSM8K	P@5	*96.7*	*96.7*	69.9	93.7	46.1	**97.4**	95.4	*96.9*
		Maj	92.7	*95.2*	61.3	90.3	42.2	*95.3*	93.0	**96.1**
	MATH500	P@5	87.6	*98.0*	43.1	80.8	12.6	83.1	82.6	**98.5**
		Maj	82.1	96.7	34.6	73.6	10.9	77.8	77.6	**98.5**
	AIME24	P@5	33.3	**80.0**	66.7	–	64.3	39.3	33.3	**80.0**
		Maj	33.3	76.7	60.0	–	57.1	28.6	30.0	**80.0**
Qwen 3-14B	GSM8K	P@5	95.0	*97.6*	51.0	95.0	–	95.8	96.6	**97.9**
		Maj	91.6	**96.6**	47.2	91.6	–	95.4	95.2	**96.6**
	MATH500	P@5	88.8	**98.8**	59.9	84.3	12.8	86.9	83.9	*98.5*
		Maj	87.0	*97.1*	50.6	79.8	12.0	81.9	81.4	**97.3**
	AIME24	P@5	46.4	80.0	66.7	–	–	50.0	40.0	**90.0**
		Maj	32.1	80.0	61.9	–	–	40.0	33.3	**86.7**
Olmo3-Think	GSM8K	P@5	–	**98.0**	–	–	–	96.9	–	*97.7*
		Maj	–	*94.9*	–	–	–	**95.7**	–	*95.5*
	MATH500	P@5	–	94.3	–	–	–	**98.3**	–	*98.0*
		Maj	–	91.2	–	–	–	*97.2*	–	**97.6**
	AIME24	P@5	–	76.7	–	–	–	**86.7**	–	83.3
		Maj	–	**76.7**	–	–	–	73.3	–	**76.7**

Table 8: Spearman and Pearson correlation between MATH500 difficulty (level 1–5) and number of student–teacher iterations under the adaptive rewrite setting (`v4_prm_belief_rewrite`).

Model	**Spearman** ρ	p	**Pearson** r	p	**Mean iter.**	
Qwen3-1.7B	0.143	1.3×10^{-3}	0.132	3.1×10^{-3}	1.23	Difficulty parsed
Qwen3-4B	0.165	2.2×10^{-4}	0.135	2.5×10^{-3}	1.10	
Qwen3-14B	0.137	2.2×10^{-3}	0.084	6.0×10^{-2}	1.15	
Olmo-3-7B-Think	0.179	5.8×10^{-3}	0.149	2.2×10^{-2}	1.07	

from the MATH500 `level` field. All four models show a small but consistent positive correlation: harder problems trigger slightly more iterations.

Table 9: Calibration scores for the hidden state belief probe against correctness rate across three Qwen3 model sizes. *** indicates p<0.001.

Model	n	ρ (cons.)	ρ (correct)	Brier	ECE
Olmo-3-7B-Think	532	+0.203***	+0.205***	0.177	0.109
Qwen3-1.7B	1562	+0.213***	+0.247***	0.197	0.103
Qwen3-4B	1767	+0.269***	+0.237***	0.209	0.157
Qwen3-14B	857	+0.362***	+0.331***	0.195	0.154

Table 10: Spearman correlation between MATH500 difficulty (level 1–5) and the belief-state probe (cons) score range across candidate questions, comparing adaptive vs. forced budgets. *Mean step range*: per-sample mean of $(\max \text{cons} - \min \text{cons})$ across gated steps. *Trajectory range*: per-sample $(\max_t \max \text{cons}_t) - (\min_t \min \text{cons}_t)$ over all gated steps.

	Adaptive				**Forced**			
	Mean step		Trajectory		Mean step		Trajectory	
Model	ρ	p	ρ	p	ρ	p	ρ	p
Qwen3-1.7B	0.178	1.4×10^{-3}	0.171	2.2×10^{-3}	0.359	8.9×10^{-12}	0.377	6.7×10^{-13}
Qwen3-4B	0.182	1.6×10^{-3}	0.232	5.5×10^{-5}	0.347	1.6×10^{-12}	0.366	8.0×10^{-14}
Qwen3-14B	0.209	6.2×10^{-4}	0.201	1.0×10^{-3}	0.463	1.2×10^{-6}	0.477	5.1×10^{-7}
Olmo-3-7B-Think	0.193	5.3×10^{-3}	0.225	1.1×10^{-3}	0.232	1.6×10^{-2}	0.221	2.2×10^{-2}