

DeepLook: Uncertainty-Guided Branching with Lookahead Confidence for Efficient Reasoning

Tingxin Yang^{1*}, Zefeng Wang^{2*}, Mengyue Wang¹, Jinhe Bi²,
Xingcheng Zhou¹, Volker Tresp², Yunpu Ma^{2,3,4}

¹Technical University of Munich, ²LMU Munich, ³MCML, ⁴MemAgents Lab

Correspondence: cognitive.yunpu@gmail.com

Abstract

Test-time scaling improves large language model reasoning, but spends compute uniformly along a reasoning trace, even in stretches that are already stable. Motivated by the observation that reasoning failures show an early onset of uncertainty, we introduce DeepLook, a training-free decoding framework that monitors segment-level confidence, branches only at uncertainty bottlenecks, and ranks candidate continuations by *Average Lookahead Confidence* (ALC)—their mean confidence over fixed-horizon rollouts—before completing and voting over a pruned subset. On four competition mathematics benchmarks across four open models, DeepLook matches or exceeds the accuracy of DeepConf-low in 12 of 16 settings while generating 87.3% fewer tokens, showing that a model’s own confidence is informative enough to decide *where* extra decoding compute is worth spending. On knowledge-intensive GPQA-Diamond the accuracy advantage disappears even though the token saving does not: coherent hallucinations also receive high ALC, so internal confidence localizes instability, not falsehood. Code is available at <https://github.com/txyang1/deeplook>.

1 Introduction

Test-time scaling has become a central design axis for large language model (LLM) reasoning: instead of relying only on larger pretrained models, a growing family of methods allocates additional compute during decoding to sample, search, verify, or refine candidate solutions (Wei et al., 2022; Wang et al., 2023; Yao et al., 2023a; Lightman et al., 2024; Madaan et al., 2023; Snell et al., 2025). The key question is no longer whether extra inference-time compute helps, but how to spend it effectively when reasoning traces are long, error-prone, and expensive to generate.

Current approaches leave this question only partly answered. Full-trajectory aggregation such as self-consistency regenerates long high-confidence prefixes that are largely shared across samples (Wang et al., 2023; Brown et al., 2024; Snell et al., 2025). Confidence-aware methods such as DeepConf filter low-quality traces using model-internal signals (Fu et al., 2025), yet these remain local or trace-level signals: without an explicit future check, an apparently plausible next step can still lead to a globally poor continuation (Ma et al., 2025; Xu et al., 2025a). Search and planning frameworks do evaluate alternatives beyond the next token, but their tree or graph expansion is ill-suited to intervening at a few uncertain points within an otherwise single reasoning trace (Yao et al., 2023a; Besta et al., 2024; Zhou et al., 2024). This gap motivates our central question:

How can test-time reasoning compute be spent only where a trace becomes uncertain, without falling prey to the myopia of local confidence?

In response to this question, we conduct an empirical investigation into the model’s internal uncertainty dynamics. As shown in Figure 2, correct and incorrect DeepSeek-R1-8B traces exhibit markedly different uncertainty profiles. Writing n_{unc} for the number of uncertain segments in a trace and r_{first} for the normalized position of the first one, incorrect traces contain many more uncertain segments on average (Figure 2a) and encounter their first uncertain segment earlier in the generation (Figure 2b). Such drops mark where additional compute is likely to be most useful: rather than resampling complete trajectories, inference can intervene near persistent low-confidence bottlenecks. However, triggering alone only identifies *when* to branch. It does not determine which continuation will remain reliable, motivating a future check before committing further compute.

Guided by this principle, we introduce

*Equal contribution.

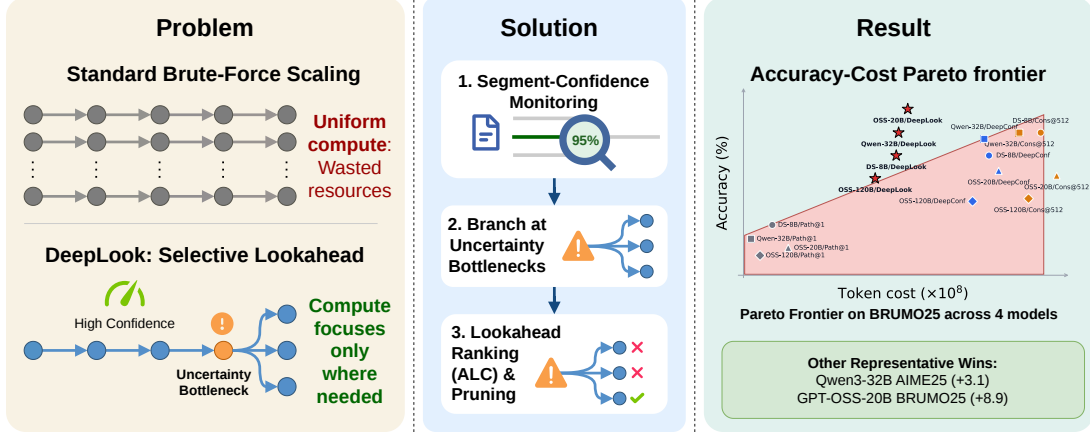


Figure 1: **DeepLook conceptual overview and accuracy-token-cost frontier.** Left: uniform brute-force scaling vs. selective lookahead at uncertainty bottlenecks. Middle: the DeepLook pipeline of segment monitoring, branching, and ALC ranking. Right: DeepLook on a higher-accuracy, lower-token-cost Pareto frontier, with an 87.3% average generated-token reduction relative to DeepConf-low (details in §4.2).

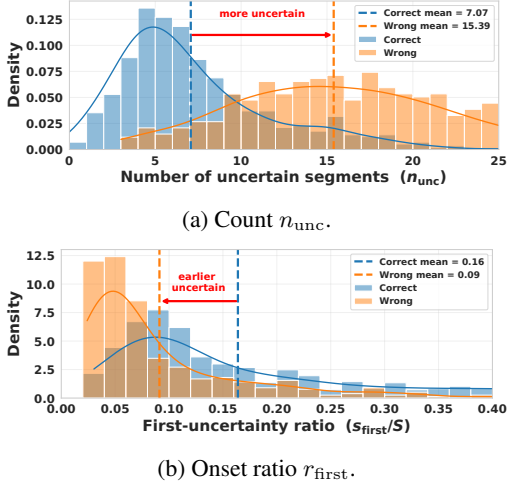


Figure 2: **Uncertainty diagnostics correlate with errors on DeepSeek-R1-8B.** Wrong traces (orange) exhibit more uncertain segments (a) and an *earlier uncertainty onset* (b) than correct traces (blue), motivating early confidence drops as triggers for lookahead exploration.

DeepLook, a confidence-triggered lookahead framework that first detects where reasoning becomes uncertain and then uses future confidence to select which continuation merits further computation. DeepLook monitors segment-level confidence during decoding, triggers intervention only at local uncertainty bottlenecks, and ranks candidate continuations by *Average Lookahead Confidence* (ALC), the mean segment-level confidence over fixed-horizon rollout continuations; selective triggering, lookahead evaluation, and adaptive pruning together preserve the robustness benefits of test-

time scaling while sharply reducing wasted compute (Figure 1).

We evaluate DeepLook on four competition-level benchmarks, AIME24, AIME25, BRUMO25, and HMMT25 (Maxwell-Jia, 2024; MathArena, 2025a,b,c), using four open-weight models: DeepSeek-R1-8B, Qwen3-32B, GPT-OSS-20B, and GPT-OSS-120B (Guo et al., 2025; Team, 2025; OpenAI, 2025). Across these settings, DeepLook **shifts the accuracy-token-cost trade-off toward substantially lower inference cost**: it matches or exceeds the accuracy of DeepConf-low in 12 of 16 settings while generating on average $8.5\times$ fewer tokens ($5.1\text{--}16.0\times$ across settings)—an average generated-token reduction of **87.3%**. The advantage persists against much larger sampling budgets such as Cons@512 (matched or exceeded in 13 of 16 settings). Targeted lookahead and pruning thus recover test-time exploration benefits without uniformly generating complete trajectories.

Our contributions are threefold:

- **Selective intervention.** DeepLook, a training-free framework that branches only at persistent uncertainty bottlenecks, leaving stable stretches of a trace untouched.
- **Non-myopic ranking.** *Average Lookahead Confidence* (ALC), which ranks continuations by their mean confidence over fixed-horizon rollouts rather than by local confidence alone.
- **Accuracy-cost evidence.** An evaluation over four benchmarks and four models, plus a boundary analysis on knowledge-intensive GPQA-Diamond, where internal-confidence interven-

tion stops helping.

2 Related work

Efficient test-time scaling. Inference-time reasoning improves LLM reliability through chain-of-thought (CoT) prompting (Wei et al., 2022; Kojima et al., 2022), repeated sampling (Wang et al., 2023; Brown et al., 2024), and compound inference systems (Chen et al., 2024), but these gains often come with large token costs (Feng et al., 2025; Sui et al., 2025; Qu et al., 2025). Recent work spends test-time compute more economically via adaptive compute control (Aggarwal and Welleck, 2025; Zhang et al., 2025; Al-Khalili et al., 2026; Scalena et al., 2026), adaptive sample budgets for self-consistency (Aggarwal et al., 2023; Li et al., 2024), reasoning compression (Xia et al., 2025; Xu et al., 2025b), pruning (Hou et al., 2026), verifier-guided inference (Lightman et al., 2024), and feedback-based refinement (Madaan et al., 2023). Yet many effective strategies still operate over complete trajectories, regenerating long shared prefixes for traces that differ only around a few pivotal steps. DeepLook targets this redundancy by moving from full-trajectory scaling to selective completion, spending extra compute only where an online signal flags a segment worth revisiting.

Confidence-aware computation. Model-internal and online generation signals provide a natural way to decide *when* extra inference should be spent, supporting confidence-based early exit, partial-trace pruning (Yang et al., 2025; Tu et al., 2026), answer selection, and trace filtering (Kang et al., 2025; Fu et al., 2025). Beyond broad evidence that model probabilities, semantic uncertainty, and self-evaluation track generation quality (Kadavath et al., 2022; Kuhn et al., 2023), this uncertainty is concentrated rather than diffuse: a small minority of high-entropy tokens act as forks that steer the trace toward different reasoning paths (Wang et al., 2025), and resampling at those specific positions—but not others—changes the outcome (Bigelow et al., 2025). This concentration is what makes a selective, trigger-driven intervention viable. Closest in signal is DEER (Yang et al., 2025), which monitors similar confidence dynamics but makes the *opposite*, complementary intervention: it exits reasoning early when the model becomes confident, whereas DeepLook branches where con-

fidence drops. Because token-level probabilities are noisy and whole-trace scores too coarse to locate where a solution destabilizes, DeepLook uses confidence primarily as a *trigger*: persistent local uncertainty determines where to branch, and a separate lookahead signal evaluates which continuations should be kept.

Exploration in sampling. Exploration-based inference addresses the complementary question of *how* to evaluate alternatives once branching is allowed. Work on non-myopic decoding directly targets the limits of next-token likelihood by using short-horizon foresight to score candidates by their downstream behavior (Ma et al., 2025; Xu et al., 2025a); Bachmann and Nagarajan (2024) trace the same difficulty to teacher-forced training itself. Search-based reasoning methods instead expand larger spaces of thoughts, trajectories, or rollouts (Yao et al., 2023a; Li et al., 2026), an idea that recurs in RL training (Xing et al., 2025); related planning and agentic frameworks explore thought graphs, simulated states, and action trajectories (Besta et al., 2024; Hao et al., 2023; Zhou et al., 2024; Yao et al., 2023b), while latent deliberation internalizes future-aware computation before visible generation (Zelikman et al., 2024). Likelihood-guided resampling has also been applied globally: Karan and Du (2025) sharpen entire traces with an MCMC-style procedure driven by the model’s own likelihoods, whereas DeepLook applies confidence *locally*, deciding where within a single trace to branch and what to keep. These methods motivate future-sensitive evaluation, but broad search can introduce substantial branching overhead; DeepLook adopts the foresight principle in a narrower form, performing fixed-budget lookahead only at confidence-triggered uncertainty bottlenecks, without expanding a full search tree.

Process reward models and step verifiers. A complementary line trains process reward models on step-level supervision to score intermediate reasoning (Lightman et al., 2024) and uses these verifiers to guide beam search or best-of- N selection; Xie et al. (2023) instead take the model’s own step-wise self-evaluation as the guidance signal within stochastic beam search. Such verifiers offer strong signals but require task-specific annotation and additional trained components. DeepLook’s ranking signal is derived entirely from the model’s own confidence distributions—no supervision, no extra models—at the cost of being blind to confidently

Algorithm 1 DeepLook (compact). Full pseudocode with all bookkeeping: Appendix Algorithm 2.

```

1: Input: prompt  $x$ ; width  $K$ ; horizon  $H$ ; rounds  $R_{\max}$ 
2:  $P \leftarrow x$ ; completed traces  $\mathcal{T} \leftarrow \emptyset$ ;  $r \leftarrow 0$ 
3: while no final answer do
4:   decode segment  $S$ ; score  $\bar{C}(S), \tau$  /* Eqs. (1)–(3) */
5:   if uncertain per Eq. (4) and  $r < R_{\max}$  then
6:      $r \leftarrow r + 1$ ; discard  $S$ ; sample  $K$  branches
7:     roll  $H$  segments ahead; rank by ALC /* Eq. (5) */
8:     keep top  $\lfloor K\rho_i \rfloor$  branches /* Eq. (6) */
9:     best branch  $b^*$  resumes as main path:  $P \leftarrow P \oplus b^*$ 
10:    complete other kept branches; add to  $\mathcal{T}$ 
11:   else
12:     accept segment:  $P \leftarrow P \oplus S$ 
13:   end if
14: end while
15: return majority vote over  $\mathcal{T} \cup \{P\}$  /* Eq. (7) */

```

stated factual errors, a boundary we characterize empirically in §4.6.

3 DeepLook

DeepLook is a monitor-and-intervene decoding framework organized around the five stages shown in Figure 3 and summarized in Algorithm 1. It converts token-level top- k log-probability statistics into segment-level confidence and intervenes only when the current segment becomes uncertain relative to its recent history. At an intervention point, DeepLook performs lookahead over K candidate continuations, ranks them by Average Lookahead Confidence (ALC), adaptively retains a subset according to the observed uncertainty, and synthesizes the final answer from the completed retained traces.

3.1 Confidence in reasoning and segment-level triggering

DeepLook monitors confidence at the segment level to decide when additional computation is needed. Following DeepConf (Fu et al., 2025), we define token confidence C_t from the top- k next-token distribution:

$$C_t = -\frac{1}{k} \sum_{\ell=1}^k \log p_t(v_t^{(\ell)}), \quad (1)$$

where $v_t^{(1)}, \dots, v_t^{(k)}$ are the top- k tokens at decoding step t .

However, token-level metrics are inherently volatile: individual token probabilities fluctuate due to local lexical ambiguity even when the underlying logic is correct (visualized in Appendix

Figure 10). To reduce this noise, DeepLook decodes the main trajectory in fixed-length segments and aggregates token confidence over each segment S_i of length L_{main} :

$$\bar{C}(S_i) = \frac{1}{L_{\text{main}}} \sum_{t \in S_i} C_t. \quad (2)$$

Rather than use a fixed confidence cutoff, DeepLook compares the current segment against a local history window of recent segment confidences. This local comparison makes the trigger depend on relative confidence changes within the current trace, reducing sensitivity to the absolute calibration of different models. For segment S_i , we define

$$\tau_i = \text{Percentile}(\mathcal{H}_i, q), \quad (3)$$

where $\mathcal{H}_i = \{\bar{C}(S_j)\}_{j=i-W+1}^i$

is the sliding window of the most recent W segment confidences (including the current segment); triggering is disabled during a short warmup while the window first fills. Lookahead exploration is governed by an uncertainty state with hysteresis:

$$\text{enter: } \bar{C}(S_i) \leq \tau_i, \quad \text{exit: } \bar{C}(S_i) \geq (1 + \delta) \tau_i. \quad (4)$$

Entering the uncertain state triggers exploration, and the state persists across subsequent segments—each further segment triggering again while the branch budget lasts—until confidence recovers above the release bound; the margin δ prevents oscillating re-triggers from minor fluctuations near the threshold, focusing branching on local instability rather than on uniformly difficult regions. The design operationalizes the early-onset observation of Figure 2 directly: what distinguishes incorrect traces is not their absolute confidence level, which varies across models and problems, but the arrival of segments markedly less confident than the trace’s own recent history. This is exactly the event the relative percentile test detects, and the earlier it fires, the more of the trace remains correctable. We treat this as a correlational signal for allocating compute, not a causal account of why reasoning fails; §4.3 reports the correct-vs-incorrect separation of the trigger statistic under all component variants. In our experiments, we use a single fixed configuration ($q = 0.10$, $W = 8$, $\delta = 0.02$) across all four model families without model-specific recalibration.

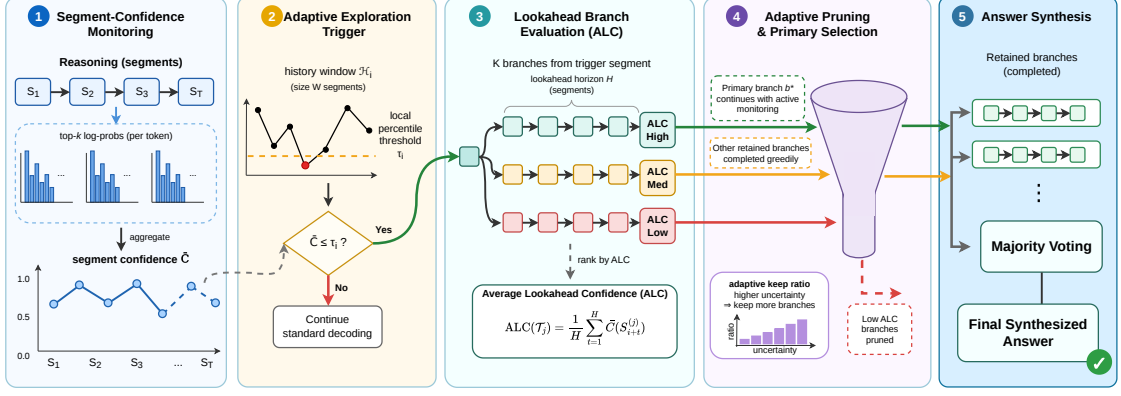


Figure 3: **DeepLook pipeline.** (1) Token-level top- k log-probabilities are aggregated into a segment confidence $\bar{C}(S_i)$, which (2) is compared against a percentile threshold τ_i over a local history window $\mathcal{H}_i$, so exploration fires only on a drop relative to the trace’s own recent history. (3) At a trigger, K branches are rolled forward over a horizon H and scored by Average Lookahead Confidence (ALC). (4) Branches are ranked and adaptively pruned, the best branch b^* resuming as the primary trajectory while the other retained branches are completed. (5) All completed traces are majority-voted into the final answer.

3.2 Lookahead exploration and Average Lookahead Confidence

Once a confidence drop triggers exploration, DeepLook suspends main-path decoding and compares several continuations before committing further computation. Because a continuation can have reasonable immediate probability yet lead to an unstable subsequent trace, DeepLook follows the non-myopic decoding principle (Ma et al., 2025) and evaluates candidates by their near-future behavior rather than by the trigger segment alone.

Specifically, DeepLook discards the trigger segment S_i and spawns K parallel candidate branches $\{b_1, \dots, b_K\}$ from the prefix preceding it, using seeded nucleus sampling to encourage diversity among local continuations. Each branch b_j first generates a rollout segment of length L_{roll} —the text that replaces S_i if the branch is adopted—and is then extended for a lookahead horizon of H further segments of length L_{look} , which are used only for scoring and then discarded. We score a branch by Average Lookahead Confidence (ALC), the mean segment confidence observed over this lookahead window:

$$\text{ALC}(b_j) = \frac{1}{H} \sum_{h=1}^H \bar{C}(S_h^{(j)}), \quad (5)$$

where $\bar{C}(S_h^{(j)})$ denotes the confidence of the h -th lookahead segment of branch b_j . ALC is an internal stability heuristic, not an external correctness verifier: branches that return to confident generation score higher, a future-sensitive ranking that avoids

committing to a locally plausible but unstable continuation. Branches are ranked by descending ALC and passed to the pruning stage.

3.3 Lookahead adaptive pruning and final answer synthesis

After ALC ranking, DeepLook keeps only a subset of branches for completion. Let $\Delta_i = \max(0, \tau_i - \bar{C}(S_i))$ denote the confidence gap at the trigger point. The keep ratio ρ_i increases with this gap so that stronger uncertainty preserves more candidate answers:

$$\begin{aligned} \rho_u &= \rho_{\min} + \text{clip}\left(\frac{\Delta_i}{s}, 0, 1\right) (\rho_{\max} - \rho_{\min}), \\ \rho_i &= \text{clip}\left(\frac{\rho_u + \rho_{\text{base}}}{2}, \rho_{\min}, \rho_{\max}\right), \end{aligned} \quad (6)$$

where $[\rho_{\min}, \rho_{\max}]$ bounds the keep ratio, s controls sensitivity, and ρ_{base} is the default retention rate; averaging with ρ_{base} damps the gap-driven term, and when $\Delta_i = 0$ (possible while the trigger remains latched) the keep ratio falls back to $\rho_{\min}$.

Among branches satisfying $\text{ALC}(b) \geq \tau_i$ and $\text{ALC}(b) > \bar{C}(S_i)$, DeepLook selects the highest-ALC branch as the primary trajectory b^* (falling back to the highest rollout confidence—the mean token confidence of a branch’s rollout segment—if none qualifies). The primary trajectory resumes segmented decoding under active monitoring and remains eligible for future interventions; the other retained branches—the top- $\lfloor K\rho_i \rfloor$ ALC-ranked candidates (at least one; ties broken by rollout confidence), minus b^* when it ranks among them—are

Model	Dataset	Path@1	Cons@512		DeepConf-low		DeepLook@128 (Ours)			
			Tok	Acc	Tok	Acc	Tok	$\Delta\% \downarrow$ (C/DC)	Acc	Δ (C/DC)
DeepSeek-R1-8B	AIME24	83.0%	3.55	86.7%	0.78	92.5%	0.121	(-96.6%/-84.5%)	93.3%	(+6.6/+0.8)
	AIME25	76.9%	4.01	82.3%	1.24	86.4%	0.127	(-96.8%/-89.8%)	86.7%	(+4.4/+0.3)
	BRUMO25	80.0%	3.56	93.3%	1.07	90.0%	0.124	(-96.5%/-88.4%)	90.0%	(-3.3/+0.0)
	HMMT25	58.1%	4.49	69.8%	1.60	77.6%	0.213	(-95.3%/-86.7%)	73.3%	(+3.5/-4.3)
Qwen3-32B	AIME24	80.6%	2.00	84.8%	0.66	89.5%	0.102	(-94.9%/-84.5%)	90.0%	(+5.2/+0.5)
	AIME25	71.7%	2.43	80.1%	1.14	80.2%	0.140	(-94.2%/-87.7%)	83.3%	(+3.2/+3.1)
	BRUMO25	78.0%	2.17	93.3%	0.96	92.4%	0.131	(-94.0%/-86.4%)	93.3%	(+0.0/+0.9)
	HMMT25	51.9%	2.76	63.4%	1.55	64.5%	0.097	(-96.5%/-93.7%)	63.3%	(-0.1/-1.2)
GPT-OSS-20B	AIME24	92.1%	5.57	96.7%	1.11	95.7%	0.168	(-97.0%/-84.9%)	96.7%	(+0.0/+1.0)
	AIME25	91.7%	6.26	95.4%	1.21	96.1%	0.235	(-96.2%/-80.6%)	96.7%	(+1.3/+0.6)
	BRUMO25	76.7%	5.16	87.1%	1.34	87.8%	0.162	(-96.9%/-87.9%)	96.7%	(+9.6/+8.9)
	HMMT25	86.7%	8.16	89.9%	2.17	89.4%	0.189	(-97.7%/-91.3%)	90.0%	(+0.1/+0.6)
GPT-OSS-120B	AIME24	91.9%	2.66	96.7%	0.53	97.0%	0.079	(-97.0%/-85.1%)	96.7%	(+0.0/-0.3)
	AIME25	91.8%	3.23	97.1%	0.49	97.9%	0.072	(-97.8%/-85.3%)	96.7%	(-0.4/-1.2)
	BRUMO25	75.6%	2.68	83.8%	0.73	83.4%	0.076	(-97.2%/-89.6%)	86.7%	(+2.9/+3.3)
	HMMT25	78.9%	4.09	92.8%	0.97	92.0%	0.094	(-97.7%/-90.3%)	93.3%	(+0.5/+1.3)

Table 1: **Main results.** Accuracy (%) and total generated-token cost (Tok, $\times 10^8$, all inference stages). For DeepLook@128, $\Delta\% \downarrow$ and Δ give token and accuracy change vs. (C/DC) = (Cons@512/DeepConf-low). Shading marks the **best** and **second-best** accuracy per row. All accuracies are measured on $N=30$ problems per benchmark.

completed with the main-path sampling parameters without recursive triggering, and the rest are discarded.

Finally, DeepLook extracts an answer a_r from the primary trajectory and from each completed retained trace, skipping traces without a parseable boxed answer, and returns the majority vote over the resulting pool $\mathcal{T}$:

$$\hat{a} = \arg \max_a \sum_{r \in \mathcal{T}} \mathbb{I}(a_r = a). \quad (7)$$

Ties are resolved by the average confidence of the traces supporting each answer, keeping the voting rule standard and lightweight.

4 Experiments

4.1 Experimental settings

Benchmarks and metrics. We evaluate DeepLook on four competition-level mathematics benchmarks: AIME24 (Maxwell-Jia, 2024), AIME25 (MathArena, 2025a), BRUMO25 (MathArena, 2025b), and HMMT25 (MathArena, 2025c), reporting **Accuracy (Acc)** of the final synthesized answer and **Token cost (Tok)**, the dataset-level total of generated tokens over all inference stages (main-path decoding, branch and lookahead rollouts, retained-branch completions), plus relative reduction ($\Delta\%$). Token counts measure

generated-token work only; they exclude wall-clock latency and the KV-cache memory needed to evaluate K branches in parallel (Appendix C). Separately from this accuracy–cost evaluation, we probe the knowledge-intensive regime on GPQA-Diamond (Rein et al., 2023) in §4.6.

Models. We use four open-weight models spanning 8B–120B: DeepSeek-R1-8B¹, a distilled reasoning model; Qwen3-32B (Team, 2025), a dense model; and GPT-OSS-20B/120B (OpenAI, 2025).

Baselines. We write DeepLook@ B for a total branch budget of $B=K \times R_{\max}$ candidate continuations per problem; DeepLook@128 uses $K=64$ branches over $R_{\max}=2$ rounds (Appendix A.2). We compare against (1) Path@1, a single sampled trajectory; (2) Cons@512, computation-heavy self-consistency (512 samples; temperature 0.6, top- p 0.95, matching DeepLook’s rollout parameters) as a brute-force ceiling; and (3) DeepConf-low (Fu et al., 2025), an adaptive confidence-guided method tuned for efficiency, our primary resource-allocation baseline.

4.2 Main results

Table 1 reports accuracy and token cost across model scales; Appendix Figure 9 plots the corresponding trade-off per benchmark.

¹The Qwen3-8B model distilled from DeepSeek-R1 (0528): <https://huggingface.co/deepseek-ai/DeepSeek-R1-0528-Qwen3-8B>.

Surgical precision over brute force. Relative to DeepConf-low, DeepLook@128 matches or exceeds accuracy in 12 of 16 settings (11 wins, 1 tie, 4 losses) while generating **87.3%** fewer tokens (a mean per-setting ratio of **8.5×**); on BRUMO25 with GPT-OSS-20B it gains **+8.9 points** ($87.8\% \rightarrow 96.7\%$) at about **12%** of the baseline’s tokens. A two-sided sign test over the 15 non-tied settings (11 wins vs. 4 losses) does not distinguish the two methods on accuracy ($p=0.118$), whereas the token reduction holds in **16 of 16** settings ($80.6\%–93.7\%$, $p<10^{-4}$). The result we defend is therefore an order-of-magnitude cheaper decode at statistically indistinguishable accuracy. Table 1 also shows where that saving costs something: on HMMT25 with DeepSeek-R1-8B and Qwen3-32B a modest accuracy drop accompanies the compute reduction, shrinking as the branch budget grows (66.7% at @32 vs. 73.3% at @128 on DeepSeek-R1-8B; Appendix Table 4), consistent with a completion-budget limitation on especially difficult instances.

The branch budget is a well-behaved dial. Accuracy is non-decreasing in the budget in all 16 settings ($@32 \leq @64 \leq @128$), with @128 strictly above @32 in 15 of them ($p<10^{-4}$, two-sided sign test), so DeepLook exposes a monotone accuracy–cost knob rather than a single tuned operating point. The cheaper settings trade accuracy for cost as expected: @32 uses $32.6\times$ fewer tokens than DeepConf-low at -3.9 points on average, @64 $15.8\times$ at -1.6 . We report @128 as the main configuration because it is the cheapest budget that reaches the baseline’s accuracy level; Appendix Table 4 gives the full curve.

Outperforming the compute ceiling. The comparison with Cons@512 highlights the diminishing returns of blind test-time scaling: on AIME25 with Qwen3-32B, DeepLook improves over Cons@512 by **+3.2 points** ($80.1\% \rightarrow 83.3\%$) at roughly **17× less compute**. Figure 4 summarizes one such case: the consensus baseline repeatedly commits to a locally plausible heuristic, while branching at detected instability recovers the correct answer—consistent with the benefit coming from intervening where the trace becomes unstable rather than from candidate count (§4.4).

4.3 Does the trigger signal track reasoning fragility?

§4.2 shows that acting on segment-level confidence pays off. Whether the gain comes from the signal itself or merely from the extra sampling it triggers

Problem (HMMT’25): Maximize door traversals in 11×11 grid loop (Edges=220; Odd V=36).

Cons@512 $\rightarrow$ Incorrect (202)

Blind Heuristic: Remove = $\text{Odd}/2 = 18$
Failure: Assumes local pairings always possible.
Final: $220 - 18 = 202$ (Topologically impossible).

DeepLook $\rightarrow$ Correct (200)

Lookahead Signal: Detects boundary parity mismatch.
Correction: Forced corner pairing adds +2 cost.
Outcome: $220 - (18 + 2) = 200$.

Figure 4: **Illustrative case summary (not a verbatim trace).** Hand-condensed decisive steps of an HMMT25 problem, contrasting the consensus failure mode of Cons@512 with the DeepLook run that recovered the correct answer; the summarization convention is described in Appendix F.

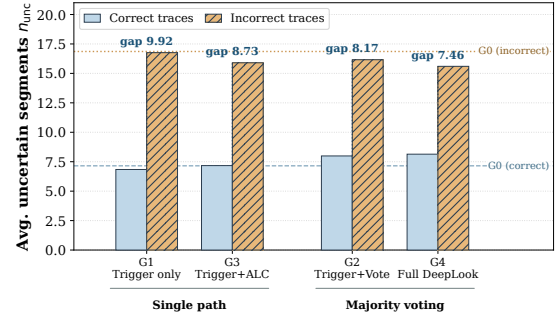
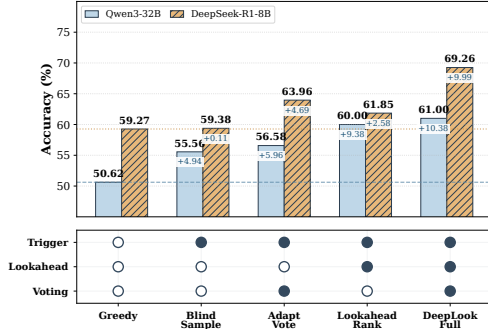


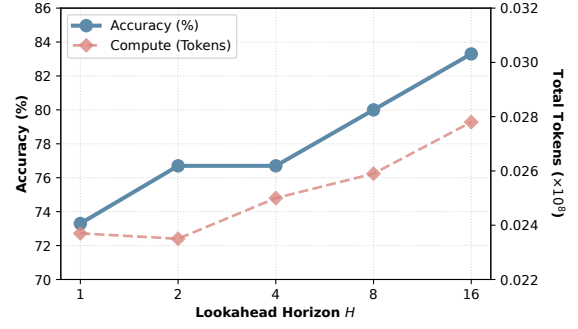
Figure 5: **Average n_{unc} for correct vs. incorrect traces,** DeepSeek-R1-8B on HMMT25, across four component variants under both readout regimes; G0 (greedy decoding) is shown as the two reference lines. The correct–incorrect gap ranges over 7.46–9.92 segments and never closes.

is a separate question; two checks address it.

The correct–incorrect separation survives every component configuration. Figure 5 reports the mean uncertain-segment count n_{unc} for correct and incorrect traces across the greedy baseline G0 (*Greedy*) and four component variants—G1 (*Blind-Sample*: trigger, random branch), G2 (*Adapt-Vote*: trigger + vote, no ranking), G3 (*Lookahead-Rank*: trigger + ALC ranking), and G4 (full DeepLook)—on HMMT25 with DeepSeek-R1-8B, under both single-path and majority-voting readouts. The gap never closes: incorrect traces carry between 7.46 and 9.92 more uncertain segments than correct ones, in every configuration. These variants differ precisely in what happens *after* triggering—whether candidates are ranked by ALC, whether outputs are voted—so a separation that is stable across all of them is a property of the trigger statistic itself, not of the machinery built on top of it.



(a) Component ablation on HMMT25.



(b) Impact of lookahead horizon H on AIME25.

Figure 6: **Ablation and scaling analysis.** (a) Accuracy as components are integrated (variants G0–G4 of §4.3); Δ is the gain over Greedy. (b) Accuracy as a function of the lookahead horizon H .

The pattern is not specific to one model family.

The diagnostics that motivate DeepLook in Figure 2 replicate on Qwen3-32B: incorrect traces contain roughly twice as many uncertain segments (11.60 vs. 5.68) and an earlier first uncertain segment (position ratio 0.24 vs. 0.41; Appendix D).

Together the two checks supply the method’s premise: n_{unc} is a correlate of trace-level correctness, stable across both model families examined, making *where to intervene* decidable at decoding time—though not a correctness certificate for any individual continuation, a distinction that matters on knowledge-intensive tasks (§4.6).

4.4 Component ablation

Figure 6a isolates the three components—triggering, ALC ranking, and voting—on HMMT25, over the configurations G0–G4 of §4.3.

Ranking and aggregation are complementary; which one wins alone is model-dependent. On Qwen3-32B a single ALC-selected trajectory (Lookahead-Rank, 60.0%) beats a consensus over unranked candidates (Adapt-Vote, 56.6%), so aggregation alone does not drive the gain. On DeepSeek-R1-8B neither suffices alone—64.0% for voting, 61.9% for ranking—against 69.3% for the full framework: lookahead prunes low-ALC branches, voting resolves what survives.

Triggering alone is not enough. Greedy vs. Blind-Sample gives inconsistent gains (+4.9 on Qwen, +0.1 on DeepSeek): locating uncertainty does not by itself separate viable from deceptive continuations.

4.5 Scaling analysis

On AIME25 with DeepSeek-R1-8B (Figure 6b), accuracy is non-decreasing in the lookahead horizon

H , which controls how much future context scores a branch: extending from $H=1$ to $H=16$ yields a **+10.0-point** gain (73.3% to 83.3%) at nearly unchanged token cost (right axis). The sweep also partly isolates the ranking signal: at $H=1$ ALC reduces to local-confidence ranking, so the H axis contrasts myopic against future-aware ranking at a near-matched budget.

4.6 When internal confidence stops helping

Both the trigger and ALC are internal-consistency signals, raising the question of knowledge-intensive tasks, where a trace can be *confidently* wrong. On GPQA-Diamond the mathematics advantage does not transfer: DeepConf-low reaches 71.7% with DeepSeek-R1-8B and 73.0% with Qwen3-32B, while DeepLook reaches 64.6% and 72.7%—at a much lower token budget but with no accuracy gain and a regression on the smaller model (Appendix Table 5). A case analysis (Appendix F) illustrates why: a hallucinated fact yields an internally consistent continuation that receives *high* ALC, which lookahead cannot veto. Internal confidence localizes instability, not falsehood.

5 Conclusion

DeepLook allocates test-time reasoning compute at uncertainty bottlenecks: it monitors segment-level confidence, branches at detected drops, ranks candidates by ALC, and votes over a pruned subset. Across four mathematics benchmarks and four models it matches or exceeds DeepConf-low in 12 of 16 settings at 87.3% fewer generated tokens. Test-time scaling can thus be made compute-aware by intervening at uncertain decisions rather than repeating whole traces—subject to the knowledge-intensive boundary observed on GPQA-Diamond.

Limitations

DeepLook needs white-box access to token-level log-probabilities, so it does not apply to API-only models. Our accuracy–cost comparison covers only competition mathematics with verifiable `\boxed{}` answers, with 30 problems per benchmark; at that scale a sign test does not separate DeepLook from DeepConf–low on accuracy ($p=0.118$), so what we defend is parity at an 87.3% lower token budget. Token counts also exclude latency and KV-cache memory, both of which grow with the number of parallel branches. Finally, ALC scores distributional stability rather than correctness, and can prefer confidently stated but factually wrong reasoning (§4.6).

Acknowledgments

Zefeng Wang is supported by the Helmholtz Association under the joint research school “Munich School for Data Science – MUDS”. The authors gratefully acknowledge the computing time provided on the high-performance computer HoreKa by the National High-Performance Computing Center at KIT (NHR@KIT). This center is jointly supported by the Federal Ministry of Research, Technology and Space and the Ministry of Science, Research and the Arts of Baden-Württemberg, as part of the National High-Performance Computing (NHR) joint funding program. HoreKa is partly funded by the German Research Foundation (DFG).

References

- Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. 2023. [Let’s sample step by step: Adaptive-consistency for efficient reasoning and coding with llms](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12375–12396. Association for Computational Linguistics.
- Pranjal Aggarwal and Sean Welleck. 2025. [L1: controlling how long A reasoning model thinks with reinforcement learning](#). *CoRR*, abs/2503.04697.
- Zena Al-Khalili, Rafi Hakim, Dietrich Klakow, and Ji-Ung Lee. 2026. [Fork-think with confidence](#). *Preprint*, arXiv:2606.31484.
- Gregor Bachmann and Vaishnavh Nagarajan. 2024. [The pitfalls of next-token prediction](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 2296–2318. PMLR / OpenReview.net.
- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefer. 2024. [Graph of thoughts: Solving elaborate problems with large language models](#). In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 17682–17690. AAAI Press.
- Eric J. Bigelow, Ari Holtzman, Hidenori Tanaka, and Tomer David Ullman. 2025. [Forking paths in neural text generation](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Bradley C. A. Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. [Large language monkeys: Scaling inference compute with repeated sampling](#). *CoRR*, abs/2407.21787.
- Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. 2024. [Are more LLM calls all you need? towards scaling laws of compound inference systems](#). *CoRR*, abs/2403.02419.
- Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. [Efficient reasoning models: A survey](#). *Trans. Mach. Learn. Res.*, 2025.
- Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. [Deep think with confidence](#). *CoRR*, abs/2508.15260.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. [Deepseek-r1 incentivizes reasoning in llms through reinforcement learning](#). *Nat.*, 645(8081):633–638.
- Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. [Reasoning with language model is planning with world model](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 8154–8173. Association for Computational Linguistics.
- Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. 2026. [Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning](#). *Trans. Mach. Learn. Res.*, 2026.

- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. [Language models \(mostly\) know what they know](#). *CoRR*, abs/2207.05221.
- Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. [Scalable best-of-n selection for large language models via self-certainty](#). *CoRR*, abs/2502.18581.
- Aayush Karan and Yilun Du. 2025. [Reasoning with sampling: Your base model is smarter than you think](#). *CoRR*, abs/2510.14901.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. [Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Jiaxi Li, Yucheng Shi, Xiao Huang, Jin Lu, and Ninghao Liu. 2026. [MITS: enhanced tree search reasoning for llms via pointwise mutual information](#). In *Advances in Knowledge Discovery and Data Mining - 30th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2026, Hong Kong, China, June 9-12, 2026, Proceedings, Part II*, volume 16598 of *Lecture Notes in Computer Science*, pages 288–300. Springer.
- Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. 2024. [Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- 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.
- Chang Ma, Haiteng Zhao, Junlei Zhang, Junxian He, and Lingpeng Kong. 2025. [Non-myopic generation of language models for reasoning and planning](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-refine: Iterative refinement with self-feedback](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- MathArena. 2025a. [AIME 2025: Matharena dataset \(aime_2025\)](#). Hugging Face Datasets. Accessed: 2026-05-06.
- MathArena. 2025b. [BRUMO 2025 \(BRUMO25\): Matharena dataset \(brumo_2025\)](#). Hugging Face Datasets. Accessed: 2026-05-06.
- MathArena. 2025c. [HMMT February 2025 \(HMMT25\): Matharena dataset \(hmmt_feb_2025\)](#). Hugging Face Datasets. Accessed: 2026-05-06.
- Maxwell-Jia. 2024. [AIME 2024 Dataset \(AIME_2024\)](#). Hugging Face Datasets. Accessed: 2026-05-06.
- OpenAI. 2025. [gpt-oss-120b & gpt-oss-20b model card](#). *CoRR*, abs/2508.10925.
- Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, Peng Li, Wei Wei, Jing Shao, Chaochao Lu, Yue Zhang, Xian-Sheng Hua, Bowen Zhou, and Yu Cheng. 2025. [A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond](#). *CoRR*, abs/2503.21614.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. [GPQA: A graduate-level google-proof q&a benchmark](#). *CoRR*, abs/2311.12022.
- Daniel Scalena, Leonidas Zotos, Elisabetta Fersini, Malvina Nissim, and Ahmet Üstün. 2026. [EAGer: Entropy-aware generation for adaptive inference-time scaling](#). *Preprint*, arXiv:2510.11170.
- Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. [Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. 2025. [Stop overthinking: A survey on efficient reasoning for large language models](#). *Trans. Mach. Learn. Res.*, 2025.
- Qwen Team. 2025. [Qwen3 technical report](#). *CoRR*, abs/2505.09388.

- Shangqing Tu, Yaxuan Li, Yushi Bai, Lei Hou, and Juanzi Li. 2026. [Deeprune: Parallel scaling without inter-trace redundancy](#). In *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pages 13389–13403. Association for Computational Linguistics.
- Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. 2025. [Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. 2025. [Tokenskip: Controllable chain-of-thought compression in llms](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 3351–3363. Association for Computational Linguistics.
- Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Qizhe Xie. 2023. [Self-evaluation guided beam search for reasoning](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Shangyu Xing, Siyuan Wang, Chenyuan Yang, Xinyu Dai, and Xiang Ren. 2025. [Lookahead tree-based rollouts for enhanced trajectory-level exploration in reinforcement learning with verifiable rewards](#). *CoRR*, abs/2510.24302.
- Fangzhi Xu, Hang Yan, Chang Ma, Haiteng Zhao, Jun Liu, Qika Lin, and Zhiyong Wu. 2025a. [\$\varphi\$ -decoding: Adaptive foresight sampling for balanced inference-time exploration and exploitation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 13214–13227. Association for Computational Linguistics.
- Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025b. [Chain of draft: Thinking faster by writing less](#). *CoRR*, abs/2502.18600.
- Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Zheng Lin, Li Cao, and Weiping Wang. 2025. [Dynamic early exit in reasoning models](#). *CoRR*, abs/2504.15895.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. [Tree of thoughts: Deliberate problem solving with large language models](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023b. [React: Synergizing reasoning and acting in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. 2024. [Quiet-star: Language models can teach themselves to think before speaking](#). *CoRR*, abs/2403.09629.
- Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025. [Adaptthink: Reasoning models can learn when to think](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 3716–3730. Association for Computational Linguistics.
- Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2024. [Language agent tree search unifies reasoning, acting, and planning in language models](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 62138–62160. PMLR / OpenReview.net.

A Algorithm and experimental setup

This section provides the complete DeepLook algorithm (Algorithm 2), the default hyperparameter configuration (Table 2), and the prompt templates (Figures 7–8) used in experiments.

A.1 DeepLook algorithm

Algorithm 2 gives a self-contained pseudocode summary of DeepLook, formalizing the five-stage pipeline described in §3: fixed-length segment decoding, segment-level confidence scoring, local adaptive uncertainty triggering, ALC-ranked branching with adaptive pruning, and majority-vote answer synthesis.

Notation. x is the input prompt and P denotes the mutable decoding prefix/trace initialized from x . L_{main} is the segment length used for main-path confidence monitoring, L_{roll} is the length of the rollout segment each candidate branch first generates, L_{look} is the shorter segment length used inside lookahead rollouts, K is the branch width (number of candidate continuations sampled at each intervention), R_{max} is the maximum number of branching interventions permitted along the main decoding path, and $B = K \times R_{\text{max}}$ is the *total branch budget*. These quantities determine the compute cost of the branching stage; all other hyperparameters govern the uncertainty trigger and are listed in Table 2.

A.2 Generation hyperparameters

Table 2 lists the default hyperparameters shared across all four model families. The three experimental variants differ only in their total branch budget $B = K \times R_{\text{max}}$ (see Algorithm 2): with $R_{\text{max}} = 2$ fixed, DeepLook@32, DeepLook@64, and DeepLook@128 set K to 16, 32, and 64, giving $B = 32, 64$, and 128, respectively. Sensitivity to the trigger parameters (q, W, δ, k) is analyzed in Appendix E.2, and sensitivity to R_{max} and the keep ratio in Appendix E.3.

A.3 Prompt templates

We use two prompt templates depending on the model family. DeepSeek and Qwen models receive a plain chat-format instruction (Figure 7), while GPT-OSS models additionally configure reasoning_effort via the tokenizer template (Figure 8). Both templates enforce chain-of-thought reasoning with a final `\boxed{}` answer, which the voting logic in §3.3 depends on.

Algorithm 2 DeepLook

```

1: Input: prompt  $x$ ; lengths  $L_{\text{main}}, L_{\text{roll}}, L_{\text{look}}$ ; width  $K$ ;
   rounds  $R_{\text{max}}$ ; horizon  $H$ ; margin  $\delta$ ; token budget  $T_{\text{max}}$ 
2: Output: final answer  $\hat{a}$ 
3:  $P \leftarrow x, \mathcal{T} \leftarrow \emptyset, \mathcal{H} \leftarrow \emptyset, r \leftarrow 0$ 
4: while no EOS or boxed answer, and  $< T_{\text{max}}$  tokens generated do
5:   Segment Monitoring:
6:    $S \leftarrow$  decode a candidate main-path segment from  $P$ 
     with length  $L_{\text{main}}$ 
7:    $\bar{C}(S) \leftarrow$  segment confidence of  $S$ ;  $\tau \leftarrow$ 
     Percentile( $\mathcal{H}, q$ ) ▷ Eqs. (1)–(3)
8:   Uncertainty Trigger:
9:   if uncertain per Eq. (4) and  $r < R_{\text{max}}$  then ▷
     hysteresis state; budget remaining
10:     $r \leftarrow r + 1$ 
11:   Lookahead Evaluation:
12:    $\mathcal{B} \leftarrow$  sample  $K$  rollout segments of length  $L_{\text{roll}}$ 
     from prefix  $P$ 
13:   for all  $b \in \mathcal{B}$  do
14:     Extend  $b$  by  $H$  lookahead segments of length
      $L_{\text{look}}$  (scoring only)
15:     Compute ALC( $b$ ) and rollout confidence
      $c_{\text{rollout}}(b)$ 
16:   end for
17:   Rank  $\mathcal{B}$  in descending order of ALC ▷ Eq. (5)
18:   Primary Branch Selection:
19:    $\mathcal{Q} \leftarrow \{b \in \mathcal{B} : \text{ALC}(b) \geq \tau \wedge \text{ALC}(b) >$ 
      $\bar{C}(S)\}$ 
20:   if  $\mathcal{Q} \neq \emptyset$  then
21:      $b^* \leftarrow \arg \max_{b \in \mathcal{Q}} \text{ALC}(b)$ 
22:   else
23:      $b^* \leftarrow \arg \max_{b \in \mathcal{B}} c_{\text{rollout}}(b)$ 
24:   end if
25:   Adaptive Pruning:
26:   Compute confidence gap  $\Delta_i \leftarrow \max(0, \tau -$ 
      $\bar{C}(S))$  and keep ratio  $\rho_i$  ▷ Eq. (6)
27:    $\mathcal{B}_{\text{keep}} \leftarrow \text{top-}\lfloor K\rho_i \rfloor$  ALC-ranked branches of  $\mathcal{B}$ ,
     minus  $b^*$ 
28:   Complete each  $b \in \mathcal{B}_{\text{keep}}$  with main-path sam-
     pling; add to  $\mathcal{T}$ 
29:    $P \leftarrow P \oplus b^*$  ▷  $b^*$ ’s rollout replaces  $S$ 
30:   else
31:      $P \leftarrow P \oplus S$  ▷  $\mathcal{H}$  receives  $\bar{C}(S)$  in both cases
32:   end if
33: end while
34: Answer Synthesis:
35:  $\mathcal{T} \leftarrow \mathcal{T} \cup \{P\}$ 
36:  $\hat{a} \leftarrow \arg \max_a \sum_{t \in \mathcal{T}} \mathbb{I}(\text{Answer}(t) = a)$  ▷ Eq. (7)
37: return  $\hat{a}$ 

```

A.4 Existing asset licenses

Table 3 lists the third-party datasets and model checkpoints used in our experiments, together with the license metadata reported by the corresponding public asset pages at the time of access. We use these assets only for inference-time evaluation and do not redistribute, modify, or repack the datasets or model weights. Baseline methods are credited through the cited papers in §4.1; we implement the inference procedures ourselves rather than incorporating external baseline code.

Hyperparameter	Value
Main segment length L_{main}	512
Confidence top- k	20
Trigger window W	8 segments
Low quantile q	0.10
Warmup	4 segments
Hysteresis δ	0.02
Branch width K	16
Rollout segment length L_{roll}	512
Lookahead horizon H	16
Lookahead segment length L_{look}	32
Keep ratio ρ_i	dynamic in [0.10, 0.25]
Base keep ratio ρ_{base}	0.10
Keep-gap scale s	0.03
Max branch rounds R_{max}	2

Table 2: **Default DeepLook hyperparameters** (corresponding to DeepLook@32). The @64 and @128 variants increase only K (to 32 and 64); all other values are unchanged.

Prompt A.1: Pure Segment Decoding (DeepSeek/Qwen)

Purpose. This prompt is prepended to each math query to enforce step-by-step reasoning and a final `\boxed{}` answer format.

[System]

(DeepSeek only) This assistant is DeepSeek-R1, created by DeepSeek.

[User]

{question}

Please reason step by step, and put your final answer within `\boxed{}`.

Figure 7: Prompt logic for `model_type=deepseek/qwen`.

B Additional experimental results

We report complete accuracy and token-cost results across four benchmarks and four model families (Guo et al., 2025; Team, 2025; OpenAI, 2025). Table 4 extends the main-text comparison by adding the DeepConf-high configuration and three DeepLook budgets. The DeepLook@32, DeepLook@64, and DeepLook@128 variants differ only in branch width K ; all other hyperparameters are held fixed at the values in Table 2. Table 5 reports the GPQA-Diamond comparison discussed in §4.6.

C Token cost breakdown

Table 6 decomposes DeepLook’s generated-token budget on AIME25 with DeepSeek-R1-8B into four inference stages: main-path decoding, branch rollouts, ALC lookahead evaluation, and comple-

Prompt A.2: Pure Segment Decoding (GPT Template)

Purpose. Same instruction as Figure 7, with optional `reasoning_effort`.

[User]

{question}

Please reason step by step, and put your final answer within `\boxed{}`.

[Template Configuration]

```
tokenizer.apply_chat_template(
    messages, tokenize=False,
    reasoning_effort={high|medium|low},
    add_generation_prompt=True)
```

Figure 8: Prompt logic for GPT-OSS models with `reasoning_effort`.

tion of retained branches. ALC scoring accounts for 23.58% of generated tokens, while retained-branch completion is the largest component at 39.52%. The lookahead mechanism is therefore not the dominant cost; most computation is spent after pruning, on branches that ALC has judged worth completing.

This breakdown also clarifies the scope of our efficiency metric: token count captures generated-token work, not wall-clock latency. In a parallel implementation, evaluating K branches simultaneously can reduce elapsed time, but increases KV-cache memory proportionally with the number of active branches—leaving latency-memory trade-offs as a deployment-level concern.

D Additional uncertainty analysis

Figure 11 confirms that Qwen3-32B follows the same uncertainty pattern as DeepSeek-R1-8B reported in the main text: incorrect traces contain more uncertain segments on average (11.60 vs. 5.68) and exhibit an earlier first confidence drop (position ratio 0.24 vs. 0.41). The signal is therefore consistent across both model families examined, which are architecturally distinct, supporting segment-level confidence as a trigger signal.

E Additional ablation studies

We include three ablations that extend the component and scaling analysis in §4: (i) a comparison against recursive tree expansion (§E.1); (ii) a one-at-a-time sensitivity study over trigger hyperparameters (§E.2), showing that the threshold parameters are robust while segment granularity is not; and (iii) a sensitivity study over branching rounds and keep ratio (§E.3), quantifying the

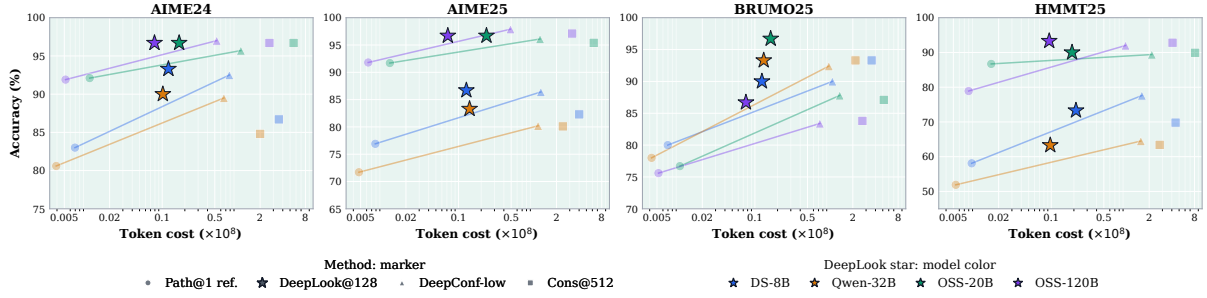


Figure 9: **Accuracy vs. token cost trade-offs across benchmarks.** Each panel plots dataset-level token cost against accuracy for one benchmark. Colors identify the base model, with DeepLook@128 shown as prominent stars and the corresponding Path@1, DeepConf-low, and Cons@512 references shown with lighter markers. Faint colored segments connect Path@1 to DeepConf-low within the same model, visualizing the baseline accuracy-cost frontier that DeepLook@128 pushes beyond. Across AIME24, AIME25, BRUMO25, and HMMT25, DeepLook@128 consistently lies in the low-cost, high-accuracy region: it uses far fewer tokens than DeepConf-low and Cons@512, while matching or exceeding their accuracy in most model-dataset settings. Path@1 is shown as the no-search reference point, plotted at the mean per-trajectory cost implied by Cons@512 (its token total divided by 512), which is the cost matching a single sampled trajectory.

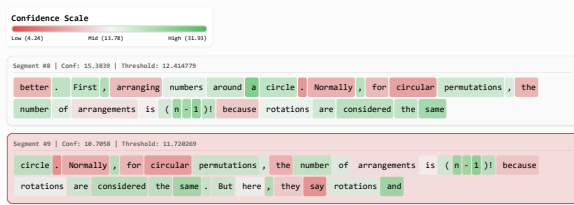


Figure 10: **Token- vs. segment-level confidence.** Each segment spans two lines for readability (segment length 32 tokens for illustration). The segment background encodes segment-level confidence $\bar{C}$, where red backgrounds indicate uncertain segments. Individual token colors (C_t) reveal local confidence fluctuations, motivating segment-level aggregation for robust detection of reasoning failures.

accuracy-cost trade-off as the completion budget grows. The signal-level analysis of the confidence trigger appears in the main text (§4.3).

E.1 Search strategy: DeepLook vs. DeepLook-Recursive

DeepLook uses asymmetric exploration: only the main path remains under active confidence monitoring, while side branches are treated as non-recursive lookahead rollouts. We compare this design with DeepLook-Recursive, a recursive variant of DeepLook that monitors and expands every generated branch.

Table 7 shows that DeepLook achieves a better accuracy-token-cost trade-off across both Qwen3-32B benchmarks: at essentially matched token budgets, DeepLook is more accurate on both. Expanding uncertainty-driven interventions onto already-

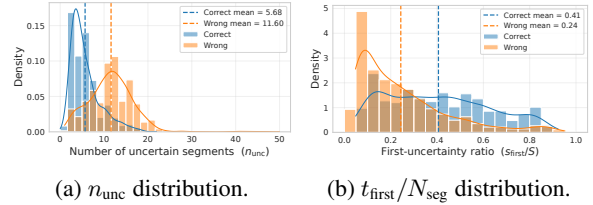


Figure 11: **Motivating analysis on Qwen3-32B.** Incorrect traces show both more frequent uncertainty and earlier uncertainty onset than correct traces.

speculative side paths appears to add noise rather than correct it; DeepLook’s asymmetric design concentrates active monitoring on the primary trajectory and treats side branches as fixed-horizon rollouts.

E.2 Trigger parameter sensitivity

A key concern for an uncertainty-driven method is whether performance depends critically on how the trigger is tuned. Table 8 reports a one-at-a-time sensitivity analysis on AIME25 (DeepSeek-R1-8B), varying one parameter at a time around a fixed ablation baseline.

The four trigger parameters are robust. Across q , W , δ and top- k , accuracy stays within one problem of the baseline: what these parameters buy is compute, not accuracy. (1) q is a *compute-accuracy dial*. Reducing q to 0.05 causes occasional missed early errors; raising it to 0.25 over-explores minor lexical hesitations, inflating token cost by 50% with no accuracy benefit. Behavior degrades smoothly within $[0.05, 0.25]$ —never

Asset type	Asset	Source/version used	License or terms
Dataset	AIME24 (Maxwell-Jia, 2024)	https://huggingface.co/datasets/Maxwell-Jia/AIME_2024 , 2024 dataset card	MIT License
Dataset	AIME25 (MathArena, 2025a)	https://huggingface.co/datasets/MathArena/aime_2025 , 2025 MathArena dataset card	CC BY-NC-SA 4.0
Dataset	BRUMO25 (MathArena, 2025b)	https://huggingface.co/datasets/MathArena/brumo_2025 , 2025 MathArena dataset card	CC BY-NC-SA 4.0
Dataset	HMMT25 (MathArena, 2025c)	https://huggingface.co/datasets/MathArena/hmmtFeb_2025 , 2025 MathArena dataset card	CC BY-NC-SA 4.0
Model	DeepSeek-R1-8B (Guo et al., 2025)	https://huggingface.co/deepseek-ai/DeepSeek-R1-0528-Qwen3-8B	MIT License
Model	Qwen3-32B (Team, 2025)	https://huggingface.co/Qwen/Qwen3-32B	Apache License 2.0
Model	GPT-OSS-20B (OpenAI, 2025)	https://huggingface.co/openai/gpt-oss-20b	Apache License 2.0
Model	GPT-OSS-120B (OpenAI, 2025)	https://huggingface.co/openai/gpt-oss-120b	Apache License 2.0

Table 3: **Licenses and terms for existing assets.** Dataset and model licenses are taken from the linked public asset pages.

catastrophically. (2) W sets how much history the threshold remembers. A large window ($W=32$) keeps high-confidence scores from early, easy segments in the reference set, which raises the low quantile and makes the trigger fire *more* often later (8.5 vs. 5.5 firings per problem) at 38% higher cost, without improving accuracy. (3) δ debounces recovery. With $\delta=0$ the uncertain state releases as soon as confidence returns to the threshold, and minor fluctuations immediately re-trigger (6.8 firings per problem versus 5.5); a large margin ($\delta=0.20$) instead holds the trigger latched far longer (9.2 firings). $\delta=0.02$ suppresses both failure modes without dulling sensitivity to genuine drops. (4) *top- k barely matters*. $k \in \{10, 20\}$ gives near-identical accuracy and firing rate, so the confidence estimate is insensitive to candidate count.

Segmentation itself is not robust. The two parameters that define what a “segment” is behave differently: quadrupling the main segment length ($L_{\text{main}}=1024$) or the warm-up (16 segments) each costs two problems (76.7%). Both act by suppressing the trigger—firings drop to 3.1 and 3.2 per problem—so the method loses its intervention points rather than mis-ranking branches. The practical reading is that DeepLook tolerates a wide range

of *threshold* settings but needs the segment granularity fine enough for a confidence drop to be localized.

E.3 Branching-round and keep-ratio sensitivity

Table 9 varies the maximum number of branching rounds R_{max} on AIME25 with DeepSeek-R1-8B ($K = 16$). Across $R_{\text{max}} \in \{2, 4, 6, 8\}$, accuracy is flat at 83.3% while token cost grows by roughly $2.4\times$: within this range the extra branches do not change the voted answer. We therefore use $R_{\text{max}} = 2$ by default—the cheapest point in this range—matching the low-budget setting reported in the main text.

Table 10 varies the static keep ratio ρ on AIME25 with Qwen3-32B ($K = 16$, $R_{\text{max}} = 2$). Accuracy improves once $\rho \geq 0.50$, where enough branches survive to benefit voting; below this threshold, over-aggressive pruning discards potentially correct candidates. The operative quantity is the retained-branch count rather than the ratio: at $K=16$, $\rho=0.50$ retains 8 branches, comparable to the 6–16 branches that the dynamic keep ratio of the main configuration retains at $K=64$. The dynamic rule adapts ρ to the observed confidence gap Δ_i (Eq. 6), providing high retention only when uncertainty is large and limiting cost elsewhere.

F Examples (illustrative summaries)

The examples below are *illustrative summaries*, not verbatim model traces: complete traces are far too long for readable presentation, so we hand-condense the decisive portions of each answer—the key reasoning step where the baseline falls into an error mode and the corresponding DeepLook correction that changes the final answer. Each figure is labeled accordingly; these summaries convey how the intervention plays out in the logged runs but should not be read as reproduced model output.

Model	Dataset	Cons@512		DeepConf-high		DeepConf-low		DeepLook@32		DeepLook@64		DeepLook@128	
		Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc
DeepSeek-R1-8B	AIME24	3.55	86.7%	1.45	86.7%	0.78	92.5%	0.036	86.7%	0.064	86.7%	0.121	93.3%
	AIME25	4.01	82.3%	2.37	81.4%	1.24	86.4%	0.027	83.3%	0.072	83.3%	0.127	86.7%
	BRUMO25	3.56	93.3%	2.17	93.3%	1.07	90.0%	0.033	86.7%	0.065	90.0%	0.124	90.0%
	HMMT25	4.49	69.8%	3.43	70.0%	1.60	77.6%	0.040	66.7%	0.082	70.0%	0.213	73.3%
Qwen3-32B	AIME24	2.00	84.8%	0.88	86.4%	0.66	89.5%	0.018	86.7%	0.046	86.7%	0.102	90.0%
	AIME25	2.43	80.1%	1.61	80.2%	1.14	80.2%	0.038	80.0%	0.068	80.0%	0.140	83.3%
	BRUMO25	2.17	93.3%	1.37	93.3%	0.96	92.4%	0.021	86.7%	0.063	90.0%	0.131	93.3%
	HMMT25	2.76	63.4%	2.24	63.6%	1.55	64.5%	0.028	56.7%	0.047	60.0%	0.097	63.3%
GPT-OSS-20B	AIME24	5.57	96.7%	3.07	96.7%	1.11	95.7%	0.044	93.3%	0.086	93.3%	0.168	96.7%
	AIME25	6.26	95.4%	3.18	95.3%	1.21	96.1%	0.054	86.7%	0.102	93.3%	0.235	96.7%
	BRUMO25	5.16	87.1%	3.49	87.2%	1.34	87.8%	0.042	93.3%	0.095	96.7%	0.162	96.7%
	HMMT25	8.16	89.9%	6.03	90.3%	2.17	89.4%	0.072	90.0%	0.126	90.0%	0.189	90.0%
GPT-OSS-120B	AIME24	2.66	96.7%	1.20	96.7%	0.53	97.0%	0.022	93.3%	0.042	96.7%	0.079	96.7%
	AIME25	3.23	97.1%	1.42	97.0%	0.49	97.9%	0.034	90.0%	0.045	96.7%	0.072	96.7%
	BRUMO25	2.68	83.8%	1.81	84.0%	0.73	83.4%	0.019	83.3%	0.049	86.7%	0.076	86.7%
	HMMT25	4.09	92.8%	2.78	93.0%	0.97	92.0%	0.036	86.7%	0.072	86.7%	0.094	93.3%

Table 4: **Full benchmark results.** Accuracy (%) and token cost (Tok, $\times 10^8$) for **Cons@512**, DeepConf baselines, and DeepLook budgets.

Model	Path@1 Acc	Cons@512		DeepConf-low		DeepLook@128	
		Tok	Acc	Tok	Acc	Tok	Acc
DeepSeek-R1-8B	62.8%	10.9	66.2%	3.46	71.7%	0.198	64.6%
Qwen3-32B	68.9%	7.44	72.2%	3.21	73.0%	0.376	72.7%

Table 5: **GPQA-Diamond results** (§4.6). Accuracy (%) and token cost (Tok, $\times 10^8$).

Component	% of Total
Main-path decoding	13.25%
Branch rollouts	23.65%
Lookahead evaluation	23.58%
Retained-branch completions	39.52%
Total	100.00%

Table 6: **Token cost breakdown on AIME25 (DeepSeek-R1-8B).** Stage-wise share of DeepLook’s dataset-level generated-token budget. The lookahead evaluation (ALC scoring) accounts for under 24% of total tokens.

Dataset	DeepLook-Recursive		DeepLook	
	Tok	Acc	Tok	Acc
AIME24	0.019	83.3	0.018	86.7
AIME25	0.039	76.7	0.038	80.0

Table 7: **Search strategy: DeepLook-Recursive vs. DeepLook.** Tok ($\times 10^8$) and Acc (%) on Qwen3-32B.

Hyperparameter	Setting	Acc (%)	Tok (M)	Trig.
<i>Ablation baseline</i>		83.3	5.0	5.5
<i>Robust: accuracy within one problem of baseline</i>				
Low quantile q	0.05	80.0	4.5	4.2
	0.25	80.0	7.5	7.3
Window size W	4	80.0	5.5	6.4
	32	80.0	6.9	8.5
Hysteresis δ	0	83.3	5.8	6.8
	0.20	80.0	7.5	9.2
Confidence top- k	20	83.3	4.9	5.3
<i>Sensitive: what defines a segment</i>				
Segment length L_{main}	1024	76.7	4.6	3.1
Warm-up (segments)	16	76.7	4.3	3.2

Table 8: **Trigger parameter sensitivity on AIME25 (DeepSeek-R1-8B).** One parameter varied at a time; all others held at the ablation baseline ($q=0.10$, $W=8$, $\delta=0.02$, $k=10$, $L_{\text{main}}=512$, warm-up 4, $K=16$, $R_{\text{max}}=10$). This ablation baseline differs from the default configuration of Table 2 in k and R_{max} , so its absolute token counts are not comparable to those of Table 9. Tok in millions; “Trig.” is the mean number of trigger firings per problem.

Rounds	Tok ($\times 10^8$)	Acc (%)
2	0.0270	83.3
4	0.0437	83.3
6	0.0545	83.3
8	0.0649	83.3

Table 9: **Branching-round sensitivity on AIME25 (DeepSeek-R1-8B).** $K = 16$. Bold marks the default R_{max} .

ρ	Tok ($\times 10^8$)	Acc (%)
0.10	0.0180	80.0
0.25	0.0438	80.0
0.50	0.0850	83.3
0.75	0.1204	83.3

Table 10: **Keep-ratio sensitivity on AIME25 (Qwen3-32B)**. $K = 16$, $R_{\max} = 2$; static keep ratio ρ .

The Heuristic Trap in Combinatorial Optimization — Illustrative Summary

Problem (HMMT 2025, Combinatorics): In an 11×11 grid, maximize the number of doors Karthik can traverse in a closed loop without repeating edges.
Mathematical Context: Total Edges = 220. The grid contains 36 vertices with odd degrees. A closed loop (Eulerian circuit) requires all vertices to have even degrees, necessitating edge removal to pair up odd vertices.

Cons@512 (Baseline Failure Path)

Model applies a standard heuristic without verifying global constraints.

Step 1: Identifies 36 vertices with odd degrees. Applies the standard formula: Edges to Remove = $\frac{36}{2} = 18$.

Step 2: “Since the grid is symmetric, local pairings are always possible regardless of boundary parity.”

Step 3: $220 - 18 = 202$.

Step 4: Multiple paths converge on 202 because they all rely on the same incomplete heuristic.

Path Selected.

Final Answer: 202 (Incorrect)

DeepLook (Ours — Correct Path)

Lookahead mechanism identifies topological constraints.

Step 1: Initially attempts to remove 18 edges based on the $N/2$ odd-vertex rule.

Step 2: Detects instability: “Wait, each side has 9 odd vertices. 9 is odd, so they cannot all be paired locally.”

Step 3: Pairing the ‘left-over’ vertices across corners increases the cost by 2 additional edges. Remove = $18 + 2 = 20$.

Step 4: $220 - 20 = 200$. The lookahead mechanism ensures the pairing strategy is globally consistent.

Final Answer: 200 (Correct)

Figure 12: **Illustrative summary: DeepLook exposes global constraints that self-consistency misses.** In this HMMT combinatorics example, Cons@512 repeatedly selects the locally plausible odd-vertex heuristic and converges to 202. By intervening at the unstable pairing step, DeepLook uses lookahead to test whether the proposed edge removals remain globally consistent, detects the boundary-parity obstruction, and redirects the solution to the correct answer 200.

The Overcounting Trap in Symmetrical Combinatorics — Illustrative Summary

Problem (AIME 2024, Combinatorics/Probability): Each vertex of a regular octagon is independently colored red or blue. The probability that the octagon can be rotated so that all blue vertices end up at originally red positions is $\frac{m}{n}$. What is $m + n$?

Mathematical Context: The problem requires finding subsets B such that $B \cap (B + k) = \emptyset$. Subsets of size $|B| \leq 3$ trivially work. For $|B| = 4$, exact inclusion-exclusion over rotational symmetries (S_1, S_2, S_4, S_6) is required to avoid overcounting invariant subsets.

Cons@512 (Baseline Failure Path)

Model loses track of set intersections in long-chain enumeration.

Step 1: Correctly identifies that only subsets $|B| \leq 4$ can satisfy the condition and correctly counts sizes 0 to 3 ($1 + 8 + 28 + 56 = 93$).

Step 2: For $|B| = 4$, it groups invariant subsets by rotational periods but fails to properly intersect S_2 and S_6 with S_4 .

Step 3: Erroneously concludes there are 24 valid subsets of size 4. $N = 93 + 24 = 117$.

Step 4: Blind scaling converges on 117/256 because the flawed inclusion-exclusion step is mechanically plausible.

Final Answer: 373 (Incorrect)

DeepLook (Ours — Correct Path)

Lookahead mechanism resolves combinatorial overlaps.

Step 1: Follows the same path to $|B| = 4$. Confidence drops when attempting to sum the subsets $2 + 6 + 16$, sensing potential double-counting.

Step 2: Lookahead rollouts simulate specific subsets (e.g., $\{0, 2, 4, 6\}$). The model detects that subsets in $S_1/S_3/S_5/S_7$ are entirely subsumed by S_2/S_6 .

Step 3: Prunes the duplicate subsets, correcting the size 4 count from 24 to 22.

Step 4: $N = 93 + 22 = 115$. $m + n = 115 + 256 = 371$.

Final Answer: 371 (Correct)

Figure 13: **Illustrative summary: DeepLook concentrates compute on fragile enumeration steps.** In this AIME 2024 counting example, Cons@512 preserves a mechanically plausible inclusion-exclusion error and overcounts the rotational cases, giving 373. DeepLook identifies the low-confidence overlap calculation as the decision point, uses lookahead rollouts to reveal duplicated symmetry classes, and prunes the overcounted branch to recover the correct answer 371.

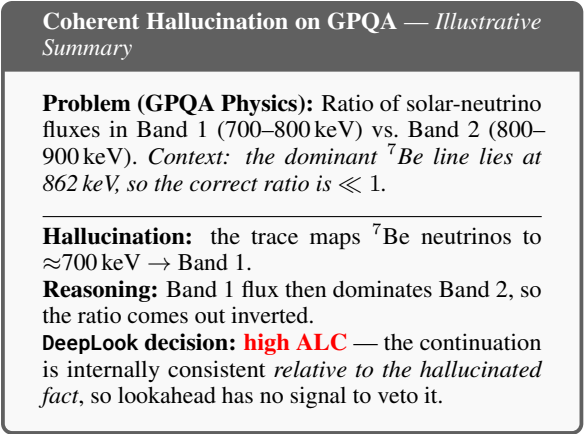


Figure 14: **Illustrative summary: the knowledge-boundary failure mode of §4.6.** A hallucinated physical fact yields a coherent continuation that scores high ALC; internal confidence localizes instability, not falsehood.

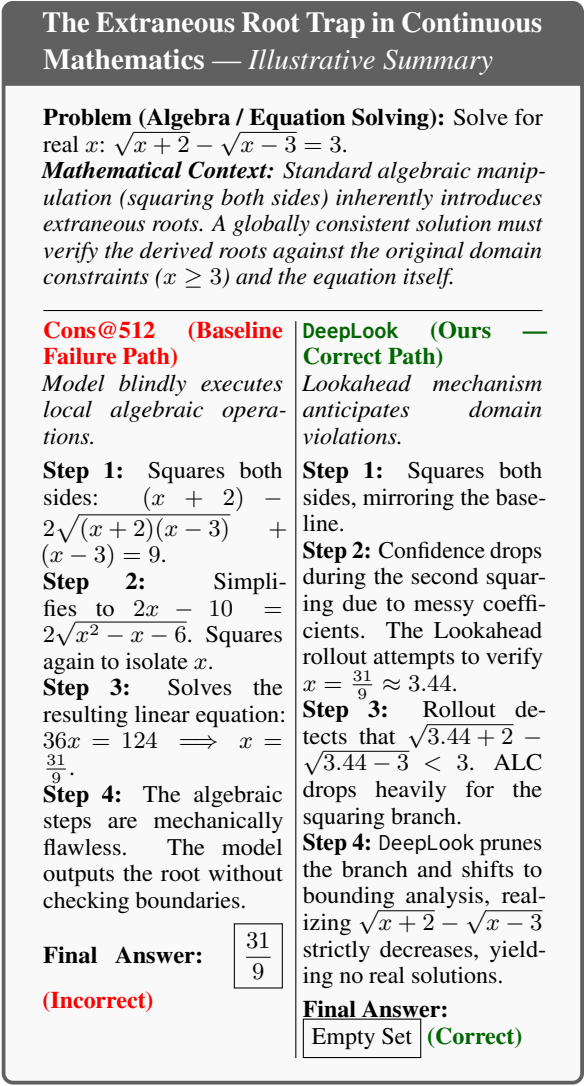


Figure 15: **Illustrative summary: DeepLook rejects deceptive algebraic branches through lookahead verification.** In this continuous-math example, Cons@512 treats the result of repeated squaring as reliable and outputs an extraneous root. DeepLook instead evaluates the downstream stability of that branch, checks the candidate against the original equation and domain behavior, and switches to the globally valid conclusion that no real solution exists.