External Thoughts as Behavioral Probes: Uncovering Reasoning Control in Large Reasoning Models

Anonymous ACL submission

Abstract

While recent large reasoning models separate intermediate reasoning from final responses, the mechanistic role of this exposed channel is not fully understood. Does the reasoning channel simply express logical context, or does it serve as a direct interface controlling the model's decision to stop or continue? To answer this, we introduce external thoughts, the high-level solution ideas externally generated and injected into the reasoning wrapper. Ideally, a model should evaluate this injected content and stop its own reasoning process only if the provided thought fully solves the problem.

Surprisingly, models rarely exhibit such well-calibrated early stopping. Instead, they exhibit a strong bias toward under-justified stops, bypassing further reasoning to answer directly even when the injected support is incomplete. We demonstrate that this premature termination is driven by the wrapper itself, as the same text presented outside the reasoning channel loses its stopping effect. Moreover, this reasoning-control policy diverges sharply across model families: distilled models execute almost unconditional termination, while RL-based models retain a revisable stop-and-think dynamic. Beyond these mechanistic insights, we show their practical value: they uncover the hidden risks in cost-efficient inference and provide a powerful, behavior-aware metric for screening out harmful supervision traces.

1 Introduction

Recent advances in large reasoning models (LRMs) have highlighted the value of extended thinking processes for solving complex tasks (Plaat et al., 2024; Xu et al., 2025a). By leveraging reinforcement learning (RL), these models are trained to produce step-by-step chain-of-thought (CoT) reasoning before emitting a final answer (OpenAI, 2025a; Team et al., 2025; Shao et al., 2024). A key feature of recent large reasoning models is the explicit separation between the reasoning process and the final response, often marked by special tokens such as `<think>` and `</think>`. While the presence of these traces correlates with strong performance, their exact mechanistic role is not fully understood. Does the reasoning channel merely express intermediate logic, or does its content actively dictate what the model does next? While prior studies primarily evaluate whether these traces are faithful or strictly necessary (Turpin et al., 2023; Wang et al., 2025), we investigate them as a direct control interface that governs the model's critical policies on stopping and continuation.

To investigate this, we insert an externally generated CoT directly between the model's thinking tokens. Our goal is to see how the model reacts: does it accept the provided trace and output a final answer, or does it reopen the reasoning process with a new `<think>` segment? By observing this re-entry behavior across thoughts of varying quality, we can determine whether the reasoning channel simply provides intermediate context, or if it actively controls the model's decision to stop or continue thinking.

Surprisingly, external CoTs rarely lead to well-calibrated early stopping. Models often skip further reasoning and answer directly, even when the injected thought is only partially useful. Our controlled experiments prove that this premature stopping is heavily driven by the reasoning wrapper itself. When the exact same text is placed outside the reasoning channel, the stopping effect largely disappears. This clearly indicates that the model treats the reasoning channel not just as regular text context, but as a dedicated control interface.

Interestingly, this reasoning-control policy varies significantly depending on how the model was trained. When given the same external thoughts, distilled models behave as aggressive "default stoppers," terminating their reasoning almost unconditionally. RL-based models, however, are much more cautious and retain a stronger revise-

or-continue dynamic. Both model families experience the urge to stop, but distilled models translate this urge into immediate termination, whereas RL models remain more selective. This consistent contrast suggests that a model's underlying training paradigm inherently shapes its reasoning control.

These mechanistic findings also offer immediate practical value for downstream applications. In the context of efficient inference, they explain the mechanics behind reducing generation costs: our evaluations demonstrate that while external thoughts can significantly cut reasoning length, they also amplify the risk of under-justified stopping, a trade-off that is highly sensitive to the model family. Furthermore, this behavioral probe offers a novel approach to supervision-trace screening. By quantifying when a trace's stop-inducing tendency outweighs its actual support, we construct a risk metric that successfully filters out harmful rationales and protects held-out accuracy, outperforming conventional heuristics like support quality alone.

Our contributions are as follows:

- We propose *external thoughts* as a novel behavioral probe to evaluate the reasoning-control policy of large reasoning models, testing whether they know when to stop.
- We demonstrate that explicit reasoning-channel content serves as a direct control signal. Specifically, injected thoughts frequently induce *under-justified stops*, a vulnerability that is driven almost entirely by the reasoning wrapper itself.
- We highlight a divergence in how stopping signals are executed: distilled models default to aggressive termination, whereas RL-based models preserve a revisable continuation policy. Finally, we leverage these insights to demonstrate practical improvements in cost-efficient inference and behavioral trace auditing.

2 External Thoughts as Intervention

We study *external thoughts* as a controlled intervention on reasoning-control policy in large reasoning models. By inserting a complete externally generated chain of thought into the exposed reasoning channel, we examine whether the model reopens reasoning with a new `<think>` segment or proceeds directly to the final response. This provides a functional probe of how exposed reasoning-channel content is interpreted.

After a descriptive overview of behavioral outcomes, our analysis proceeds in three phases. Phase 1 establishes that wrapped external thoughts induce stopping under partial support. Phase 2 shows that the exposed reasoning wrapper is critical for this stopping effect. Phase 3 studies how local termination signals are converted into final stopping across model families.

2.1 External-Thought Intervention Setup

To probe response policy in a controlled way, we insert externally generated thoughts into positions typically occupied by model-generated reasoning. These external thoughts are produced by separate generator models and are restricted to *high-level solution ideas* rather than detailed derivations or final answers. We use the Qwen series (Yang et al., 2024), including Qwen-Max, Qwen-Plus, Qwen-2.5-7B-Instruct, and Qwen-2.5-3B-Instruct, as thought generators so that generator capability can be varied systematically.

Regarding the probing models, we select four reasoning models, including two distillation-based models (Deepseek-distill-qwen-14b and Deepseek-distill-qwen-32b) and two RL-originated models (QwQ and Qwen-3.5-9B). Regarding the probing dataset, we select the MATH-500 dataset (Lightman et al., 2023), a widely used benchmark for evaluating the reasoning capability of LLMs. The probing results are reported using the average of three runs. Detailed generation setup is shown in Appendix A.

Inference Template. At inference time, the generated thought is enclosed within `<think>` and `</think>` and inserted into the target model's reasoning channel before generation continues, as shown below:

Inference Template

<|im_start|> **User:** **[Question]** <|im_end|>
<|im_start|> **Assistant:** <|im_end|>
<think> **[Generated Thought]** </think>

This intervention is intentionally minimal: it does not alter model parameters, decoding rules, or training data. Instead, it provides a controlled perturbation that allows us to observe how the target model regulates reasoning once an external thought has already been placed in the reasoning slot.

Table 1: **Behavioral outcome definitions in the descriptive overview.** Each outcome is defined by whether the model re-enters a thinking segment after the injected external thought and by the quality of that thought.

Outcome	Rethink?	Thought quality
Calibrated stop	No	Sufficient
Under-justified stop	No	Partially useful
Miscalibrated stop	No	Flawed / vacuous
Calibrated non-stop	Yes	Insufficient
Miscalibrated non-stop	Yes	Sufficient
Uncertain	Either	Mixed / unclear

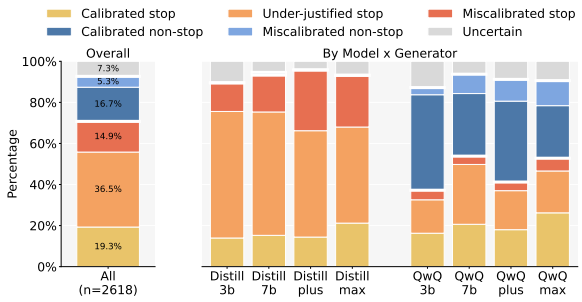

Figure 1: **Descriptive overview of behavioral outcomes under external thoughts.** The left panel shows the overall distribution across all 2,618 examples, which includes all examples having non-empty thoughts; the right panel breaks the results down by inference model family and external-thought generator strength.

2.2 Overview of Behavioral Outcomes

Before turning to controlled counterfactual probes, we first summarize the full-dataset behavioral outcomes after injecting external thoughts into the exposed reasoning channel. We classify each response by whether the model *stops* after the injected thought or *continues* reasoning, and by whether that response is justified by the actual usefulness of the injected thought. We utilize Gemini-3-Fast, GPT-5.2, and Qwen-Max for judging the usefulness and majority vote for the final decision. The resulting taxonomy is shown in Table 1, with the full judging protocol deferred to Appendix B.2.

Figure 1 provides a descriptive overview over all 2,618 examples, which includes all examples having non-empty thoughts. The dominant pattern is *under-justified stop*: models often terminate even when the injected thought offers only partial support rather than a sufficient solution. This indicates that external thoughts do not simply replace internal reasoning, but often shift the model's reasoning-control behavior toward stopping on incomplete support. The family-level breakdown further reveals a clear contrast: distilled DeepSeek models are more stop-prone, whereas RL-based QwQ models retain substantially more continuation under partial support. Motivated by this descriptive pattern, we next isolate the *partially useful* slice as a controlled probe of stopping calibration.

To isolate the mechanism more cleanly, we focus on the plus-generated partially useful slice (1,420 samples totally), which provides a controlled setting for analyzing immediate stop pressure, wrapper effects, and family-dependent signal-to-action coupling.

2.3 Phase 1: Wrapped External Thoughts Induce Stopping

We begin with a diagnostic setting for stopping calibration: external thoughts that are *partially useful* but not sufficient to fully solve the problem. In this setting, a well-calibrated model should treat the injected thought as evidence to build on, rather than as a sufficient basis for immediate termination. We therefore use this slice as a controlled probe of whether exposed reasoning-channel content pushes the model toward premature stopping.

To characterize the model's response, we measure behavior at two levels. First, we track the immediate first-step termination margin, m_1, after the injected thought:

$$m_1 = \log p(\texttt{</think>} \mid x, t_{<1}) - \log p(y_1 \mid x, t_{<1}),$$

where y_1 is the highest-probability non-termination token at the first post-injection step. Thus, m_1 measures the model's local preference for terminating the current reasoning segment relative to continuing generation. Second, we record whether the model ultimately *stops* after the injected thought, rather than reopening a new thinking segment before answering.

Figure 2 shows a clear family-level dissociation between the immediate first-step signal and final stopping behavior. Across all four models, wrapped external thoughts sharply increase final stopping under partial support. For Distill-32B and Distill-14B, final stop rate rises from 0.0% to 100.0%. For QwQ, it rises from 0.0% to 62.9%, and for Qwen3.5-9B, from 43.6% to 98.2%.

The immediate m_1 response, however, is not uniform across families. In the RL-side models, wrapped external thoughts increase mean m_1,

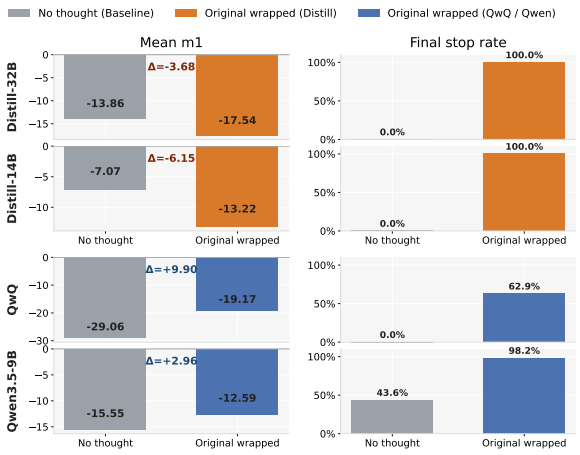

Figure 2: **Immediate and final effects of wrapped external thoughts under partial support.** Left: mean first-step termination margin (m_1); right: final stop rate. Wrapped external thoughts sharply increase final stopping across all four models, while the direction of the immediate m_1 shift differs by model family.

by $+9.90$ in QwQ and $+2.96$ in Qwen3.5-9B. In the distilled models, the same intervention shifts mean m_1 in the opposite direction, by -3.68 in Distill-32B and -6.15 in Distill-14B, even though final stopping becomes universal. This shows that wrapped external thoughts do not produce a single family-invariant local signature.

Taken together, wrapped external thoughts sharply increase final stopping under partial support across all four models, but their immediate first-step signature differs by family. In RL-side models, the intervention shifts m_1 and final stopping in the same direction. In distilled models, by contrast, final stopping becomes nearly default even though mean m_1 shifts in the opposite direction. One possible interpretation is that m_1 reflects active self-termination at the first post-injection step, whereas final stopping in the wrapped condition reflects whether the model accepts an already-closed reasoning state rather than reopening `<think>`. This dissociation motivates the next phases, where we isolate the role of the wrapper.

Takeaway. Wrapped external thoughts reliably induce final stopping, but their first-step signature is family-dependent: RL-side models shift toward termination at both levels, whereas distilled models stop almost by default once the wrapped thought is present.

Table 2: Phase-2 conditions used to isolate the effect of the reasoning wrapper.

Condition	Prompt shown to target model for generation
`no_thought`	`[chat template for Question] <think>`
`no_wrapper`	`[chat template for Question] {original_thought}`
`wrapped`	`[chat template for Question] <think> {original_thought} </think>`

2.4 Phase 2: The Exposed Reasoning Wrapper Is Critical for Final Stopping

Phase 1 showed that wrapped external thoughts sharply increase final stopping under partial support. We next ask whether this effect is driven by the external-thought content itself, or by placing that content inside the exposed reasoning channel. To isolate this factor, we compare three conditions on the same partially useful examples (see Table 2).

Figure 3 shows a clear and consistent pattern across both model families. Wrapped external thoughts sharply increase final stopping, but the same content has little effect once removed from the exposed reasoning wrapper. For the two distilled models, final stopping is near-zero in both `no_thought` and `no_wrapper`, but rises to 100.0% in `wrapped`. For QwQ, final stopping stays near-zero without the wrapper (0.9%) and rises to 62.9% when wrapped. For Qwen3.5-9B, the contrast is similarly strong: `original_no_wrapper` yields only 4.6% stopping, versus 98.2% in `original_wrapped`. This result shows that external thoughts do not act like ordinary supporting context. Their stopping effect depends primarily on being placed inside the exposed reasoning channel, indicating that the wrapper functions as a control interface rather than a neutral container for the same content.

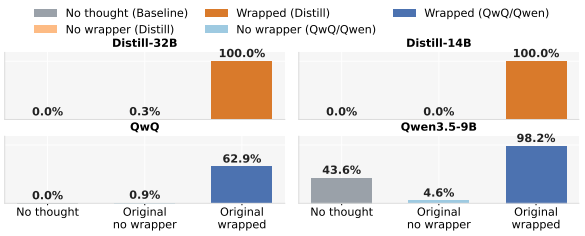

Figure 3: **Final stopping depends on the exposed reasoning wrapper.** On the partially useful slice, holding external-thought content fixed, removing the wrapper nearly abolishes the stopping effect across all models.

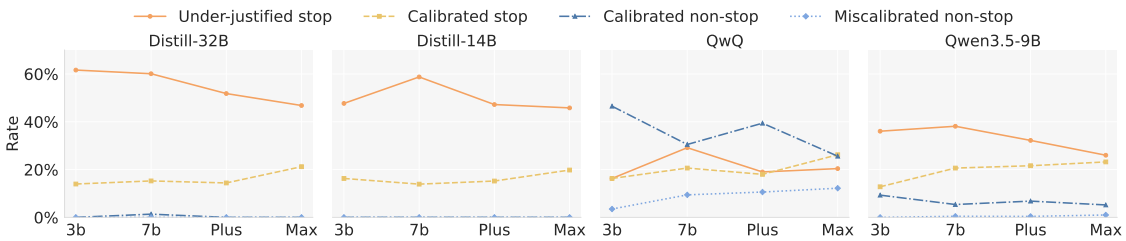

Figure 4: **Broader family-level stopping under external-thought interventions.** Across generator strengths, the distilled models are close to always-stop once an external thought is present. The RL-side models, by contrast, retain a substantial stop/non-stop trade-off and show clearer variation in calibrated continuation across model variants.

Takeaway. Holding external-thought content fixed, removing the reasoning wrapper nearly abolishes final stopping across all four models.

2.5 Phase 3: From Local Signal to Final Stopping

We now ask how local termination-preference signals are translated into final behavior across model families. The central question is not only whether external thoughts alter local signals, but how tightly those signals are coupled to eventual stopping. We utilize Δm_1, which is defined as a paired within-sample difference relative to the `no_thought` condition for the same model, to measure the stop pressure.

Figure 5 summarizes this coupling on the partially useful slice. Each point corresponds to a model-condition pair, with the x-axis showing the mean shift in first-step termination margin and the y-axis the corresponding final stop rate. Two patterns stand out. First, the same local shift can map to very different final outcomes across families. In the RL-side models, positive Δm_1 does not by itself imply termination: for example, removing the wrapper can leave the local signal elevated while

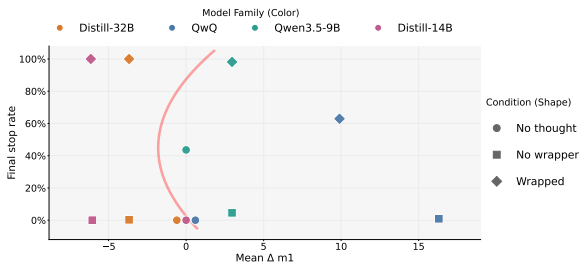

Figure 5: **Mapping from local stop pressure to final stopping.** The x-axis shows the mean first-step stop-pressure shift (Δm_1) and the y-axis shows the corresponding final stop rate.

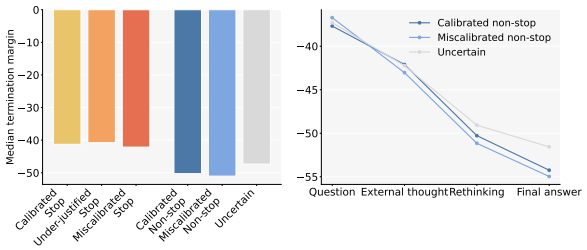

Figure 6: **Termination-margin signals as a proxy for local stop tendency.** Higher `</think>` margins indicate a stronger local preference for closing the current reasoning segment. Stop outcomes generally exhibit stronger termination preference than non-stop outcomes, and the right panel shows how this signal continues to evolve across generation stages for non-stop trajectories.

final stopping remains near zero. Second, the distilled models exhibit much tighter conversion from wrapped reasoning-channel input to stopping. Under the wrapped condition, they reach nearly saturated final stop rates even when the first-step signal is weak or shifts in the opposite direction.

This dissociation suggests that the first-step margin and final stopping do not reflect the same decision. A plausible interpretation is that the first-step signal captures local self-termination preference immediately after the injected thought, whereas final stopping in the wrapped condition depends on whether the model accepts an already closed reasoning state rather than reopening `<think>`. Under this view, distilled models behave more like default stoppers once a wrapped external thought is present, while RL-side models preserve a larger revise/continue region.

Figure 6 provides a complementary view of the local signal itself. Across outcome types, higher `</think>` margins are generally associated with stop-like behavior, supporting the use of termination-margin signals as an observable proxy for local stop tendency. At the same time, the cou-

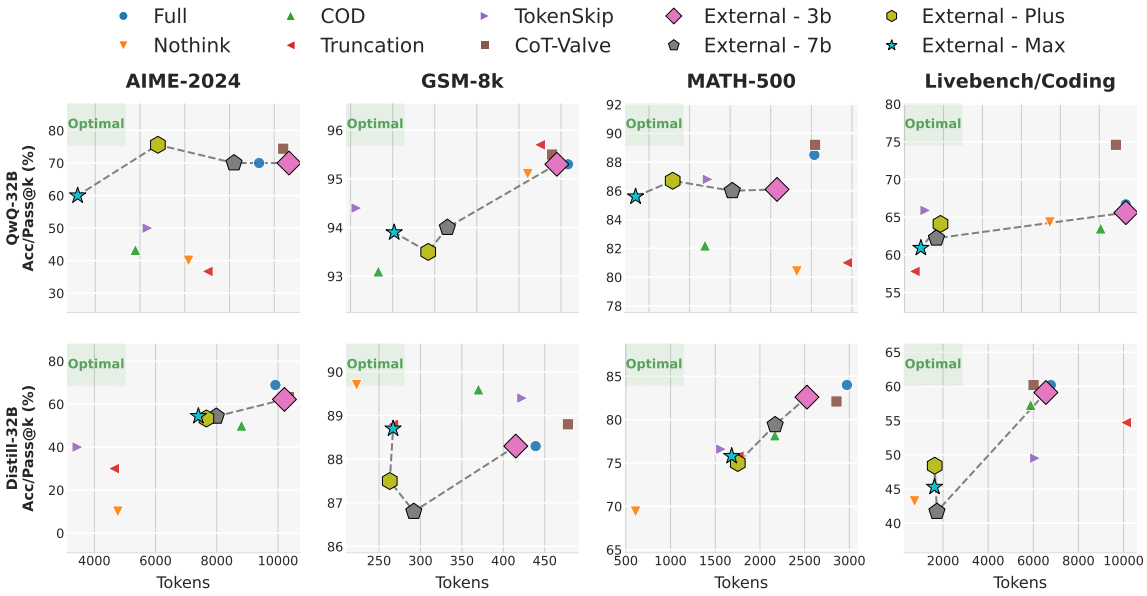

Figure 7: **Efficient inference as a mechanism-informed case study.** We compare external-thought inference with several reasoning-reduction baselines across four benchmarks, reporting accuracy against total generated tokens. The dashed line connects external-thought variants with different generator strengths.

pling analysis above makes clear that this proxy should not be equated with final stopping: both model family and wrapper condition shape whether a local stop-leaning state is ultimately executed.

Finally, Figure 4 shows that this contrast extends beyond a single comparison pair. Across generator strengths, distilled variants remain close to always-stop once an external thought is present, whereas RL-side variants retain a substantial stop/non-stop trade-off and much more calibrated continuation. Taken together, the main family-level difference is therefore not just overall stop propensity, but how directly local termination signals are converted into action.

Takeaway. Termination-margin signals are informative but not decisive: the mapping from local stop-leaning state to final stopping is strongly family-dependent. Distilled models behave like near-default stoppers once a wrapped external thought is present, whereas RL-side models retain a more selective and revisable control policy.

3 Implications

The analyses above are primarily mechanistic. Our main contribution is to use external thoughts to reveal reasoning-control policy rather than to introduce a new inference method. Nevertheless, the resulting findings also have practical implications. In this section, we discuss two such implications: efficient reasoning as a mechanism-informed case study, and supervision-trace screening as a diagnostic application of the same probe.

3.1 Efficient Inference as a Policy-Aware Case Study

Our mechanistic findings help explain why external thoughts can sometimes improve inference efficiency, but also why these gains are uneven across model families. If an injected external thought pushes the model toward termination, the model may skip a substantial portion of explicit reasoning and thus reduce generation cost. At the same time, the earlier phases show that such stopping is not always well calibrated: when the injected support is incomplete, external thoughts can also induce *under-justified stop*. Under this view, efficient reasoning is best understood not as the primary objective of external-thought prompting, but as a downstream consequence of the model's stopping policy.

Evaluation Setup. We evaluate this implication on four benchmarks spanning math and coding: AIME-2024 (Maxwell-Jia, 2024), GSM-8K (Cobbe et al., 2021), MATH-500 (Lightman et al., 2023), and the coding subset of LiveBench (White et al., 2024). For utility, we use answer matching on the math benchmarks and

pass@1 on coding. For efficiency, we count both the tokens generated by the reasoning model and the additional tokens produced by the external-thought generator, and report the total generation cost as their sum. To contextualize these costs, we also report the standalone performance of the thought generators in Table 3, showing that these generators are not strong enough to solve the tasks by themselves.

Baselines. We compare against representative methods for reducing or reshaping reasoning cost, including *Nothink* (Ma et al., 2025a), *Truncation*, *COD* (Xu et al., 2025b), *Tokenskip* (Xia et al., 2025), and *CoT-Valve* (Ma et al., 2025b). Additional implementation details are provided in Appendix C.1.

Efficiency-utility Trade-off. The main results are shown in Figure 7. External-thought prompting can achieve competitive utility while reducing reasoning cost, but the gains depend strongly on the model's stopping behavior. Models with a more aggressive stopping policy realize larger reductions in reasoning length, yet they also face a greater risk of premature and insufficiently grounded termination. This is consistent with the earlier phases: external thoughts often induce *under-justified stop* under partial support, and this tendency is substantially stronger in distillation-based models than in RL-based models.

By contrast, models with a more revisable continuation policy tend to preserve answer quality more reliably, but often realize smaller efficiency gains because they are less willing to terminate on incomplete support. The resulting efficiency–utility frontier is therefore best understood as a property of reasoning-control policy rather than of external thoughts alone. In particular, the same intervention that reduces reasoning cost can also create avoidable error when the induced stopping tendency is stronger than the support the thought actually provides.

Additional Evaluation. We further report real execution-time measurements in Appendix C.3. We also test task-specific prompts for generating external thoughts on the coding benchmark in order to assess generalization (Appendix C.2), but do not observe substantial additional gains.

Insight. Its practical value comes from exploiting a model's existing stopping policy: in models that are eager to terminate, it can reduce reasoning cost substantially, but in the same models it can also amplify the risk of under-justified stop. This suggests a broader opportunity for *policy-aware inference*: interventions on reasoning traces may be most effective when paired with explicit estimates of stopping risk, rather than applied uniformly across models or tasks.

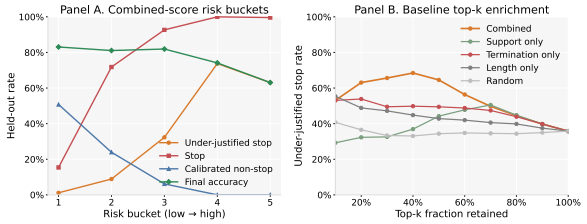

Figure 8: **Held-out ranking evaluation for supervision-trace screening. Panel A:** Traces are grouped into risk buckets using the combined score from support quality and trajectory-level termination tendency. Higher-risk buckets exhibit more under-justified stopping and overall stopping, less calibrated non-stop behavior, and lower final accuracy on disjoint held-out problems. **Panel B:** In the high-risk regime, the combined score enriches under-justified stopping more effectively than simple baselines.

3.2 Supervision-Trace Screening as a Diagnostic Application

Beyond efficient inference, external-thought probing also has diagnostic value for behavioral auditing: it can reveal cases where a trace provides limited support yet disproportionately pushes the model toward stopping. This suggests a concrete use for supervision-trace screening. When intermediate rationales are reused as supervision signals, not all superficially plausible traces should be treated equally. In particular, traces that are weakly supportive but strongly stop-inducing may be undesirable targets, because they risk transferring aggressive stopping behavior rather than reliable reasoning support. Rather than using the first-step metric m_1 directly, we combine support quality with a trajectory-level termination score, obtained by aggregating late-trace `</think>` margins, to rank candidate traces by behavioral risk (see Appendix C.4 for more information).

To test this idea, we pool traces across all available generator conditions with termination-signal scores and perform a lightweight held-out evaluation using a trajectory-level risk score derived from

external-thought probing. We use a grouped held-out protocol at the question level: all traces from the same problem are assigned to the same split, score direction, and any threshold is calibrated on one subset of questions, and all final screening results are reported on disjoint held-out questions. We then evaluate how higher-risk traces generalize.

Figure 8 shows that the resulting score has clear ranking power. As risk increases, held-out traces exhibit substantially more under-justified stopping and overall stopping, sharply less calibrated non-stop behavior, and lower final accuracy. Thus, the probe is not merely identifying traces that tend to stop more often; it is identifying traces whose stopping behavior becomes increasingly poorly calibrated and increasingly harmful to final performance. Panel B compares this combined score against simple baselines based on support only, termination signal only, trace length only, and random ranking, detailed information is shown in Appendix C.4. The combined score provides the strongest enrichment of under-justified stopping in the high-risk regime. In particular, among the top 20% highest-risk traces, the combined score yields an under-justified stop rate of 0.631, outperforming termination-only (0.538), length-only (0.488), support-only (0.323), and random ranking (0.365). This indicates that risky traces are not best characterized by weak support or strong termination tendency alone, but by their combination.

These conclusions are also robust to reasonable choices of the trajectory-level termination score (see Table 6 in Appendix C.5). Taken together, these results show that external-thought probing yields a behaviorally meaningful criterion for identifying risky supervision traces.

Insight. Traces become particularly risky when weak support is paired with strong stop-inducing tendency: such traces do not merely fail to help, but can transfer a poorly calibrated inclination to terminate reasoning. This points to a broader opportunity for behavior-aware trace curation, where candidate rationales are screened not only for what they say, but also for how they steer the model's reasoning-control policy.

4 Related Work

Large Reasoning Model. Recent studies focus on improving the capability of language models by utilizing more inference-time computation instead of simply scaling model parameters (Snell et al., 2024). This line of research produces powerful LRMs such as DeepSeek-R1 (DeepSeek-AI, 2025), OpenAI-o1/o3 (OpenAI, 2025a,b), and QwQ (Team, 2025b), which shift from fast, intuitive processing to structured, step-by-step reasoning. Deeper reasoning capabilities enable these LRMs to make remarkable improvements when tackling challenging tasks like advanced mathematics and logical reasoning (Team, 2025a).

Chain-of-Thought Compression. Despite the improved capabilities, introducing intermediate CoTs brings additional token overhead. For traditional LLMs, lots of efforts have been put into reducing redundant steps (Ding et al., 2024; Liu et al., 2024; Ma et al., 2025b), skipping less-important tokens (Han et al., 2024; Xia et al., 2025), and speculative decoding (Leviathan et al., 2023). Another line is to represent the CoT using latent space compression (Deng et al., 2024; Cheng and Van Durme, 2024), which utilizes embeddings instead of tokens to serve as the CoT.

One similar method is NoThink (Ma et al., 2025a), which inserts a placeholder sentence to accelerate the model generation. While similar in spirit, we focus on providing mechanical understanding of the LRM behavior.

Thinking Probes. Prior work has questioned whether CoT faithfully reflects the reasoning that drives a model's answer (Turpin et al., 2023; Lanham et al., 2023). A related line of work probes the necessity of intermediate reasoning more directly: Chain-of-Probe suggests that models often appear to have an answer before completing the full CoT (Wang et al., 2025). Our work shifts the focus from faithfulness to reasoning control: we use external thoughts as a controlled intervention to study whether models stop, continue, or revise, and how such policies differ across training paradigms.

5 Conclusion

We show that explicit reasoning-channel content in large reasoning models plays a dual role: it serves not only as support for further reasoning, but also as a control signal that shapes stopping behavior. We further uncover a consistent family-level contrast in this reasoning-control policy, with distilled models exhibiting a stronger stopping bias than RL-based models. These findings suggest implications for efficient inference and supervision-trace screening.

Limitations

Model Accessibility and Scope. Our external-thought intervention fundamentally relies on direct access to the exposed reasoning wrapper (e.g., the <think> channel) to inject text before final generation. Consequently, our empirical evaluation is restricted to open-weight reasoning models such as the Qwen and DeepSeek-Distilled families. Leading closed-source models (e.g., OpenAI's o1 or o3 series) do not currently expose their raw reasoning streams for arbitrary injection.

Reliance on LLM-as-a-Judge.. To categorize behavioral outcomes (such as identifying an under-justified stop), we rely on a panel of strong proprietary models (Gemini-3-Fast, GPT-5.2, and Qwen-Max) to evaluate whether an injected thought is partially useful or sufficient. Although we employ a majority-vote mechanism to mitigate individual model bias, evaluating the sufficiency of intermediate reasoning is inherently subjective. Future work could incorporate human expert annotations to further validate these behavioral boundaries.

References

Jeffrey Cheng and Benjamin Van Durme. 2024. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Yuntian Deng, Yejin Choi, and Stuart Shieber. 2024. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*.

Mengru Ding, Hanmeng Liu, Zhizhang Fu, Jian Song, Wenbo Xie, and Yue Zhang. 2024. Break the chain: Large language models can be shortcut reasoners. *arXiv preprint arXiv:2406.06580*.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2024. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, and 1 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2024. Can language models learn to skip steps? *arXiv preprint arXiv:2411.01855*.

Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. 2025a. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025b. Cot-valve: Length-compressible chain-of-thought tuning. *arXiv preprint arXiv:2502.09601*.

Maxwell-Jia. 2024. AIME 2024 Dataset. https://huggingface.co/datasets/Maxwell-Jia/AIME_2024.

OpenAI. 2025a. Introducing openai o1. https://openai.com/o1/. Accessed: 01-April-2025.

OpenAI. 2025b. Openai o3-mini. https://openai.com/index/openai-o3-mini/. Accessed: 01-April-2025.

Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. 2024. Reasoning with large language models, a survey. *arXiv preprint arXiv:2407.11511*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

M.-A-P. Team. 2025a. Supergpqa: Scaling LLM evaluation across 285 graduate disciplines. *CoRR*, abs/2502.14739.

Qwen Team. 2025b. Qwq-32b: Embracing the power of reinforcement learning.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

Zezhong Wang, Xingshan Zeng, Weiwen Liu, Yufei Wang, Liangyou Li, Yasheng Wang, Lifeng Shang, Xin Jiang, Qun Liu, and Kam-Fai Wong. 2025. Chain-of-probe: Examining the necessity and accuracy of cot step-by-step. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2586–2606.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, and 1 others. 2024. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*.

Heming Xia, Yongqi Li, Chak Tou Leong, Wenjie Wang, and Wenjie Li. 2025. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, and 1 others. 2025a. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025b. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

A CoT Generation

Specifically, the generation process is prompted to focus solely on high-level reasoning steps without delving into detailed calculations or producing final answers. The prompt is shown as follows:

Thought Generation

"If you are a teacher, you are listing the important key points for solving the problem, and no calculation details should be included. You are not allowed to produce any final answer. Add <STOP> when the key points are finished. You may provide **only very high-level ideas** for solving the problem; no calculation details should be included. If you feel that you cannot solve it, output <STOP> and return."

B Supplement for Effectiveness of External Thoughts

B.1 Sample Output of Replacing Thoughts

Thinking is Not Skipped (Rethink)

<|im_start|> **User:** A regular hexagon can be divided into six equilateral triangles. If the perimeter of one of the triangles is 21 inches, what is the perimeter, in inches, of the regular hexagon? <|im_end|>
<|im_start|> **Assistant:** <|im_end|>
<think>
[External Thoughts]
To solve this problem, consider the following key points:

- A regular hexagon can be divided into six equilateral triangles.
- The perimeter of one of these triangles is given as 21 inches.
- Since the triangle is equilateral, all three sides are equal. Thus, each side of the triangle is 7 inches (21 inches / 3).
- Each side of the equilateral triangle is also a side of the hexagon.
- A regular hexagon has six equal sides.
- Therefore, the perimeter of the hexagon is 6 times the length of one side of the triangle.

</think>

[Rethinking] Okay, let me try to figure this out. So, the problem is about a regular hexagon divided into six equilateral triangles. The perimeter of one of those triangles is 21 inches, and

I need to find the perimeter of the hexagon. Hmm, let's start breaking it down step by step. First, I know that a regular hexagon has all sides equal, and when it's divided into six equilateral triangles, those triangles must be using the sides of the hexagon as their sides. So, each triangle is equilateral, meaning all three sides are the same length.

The problem says the perimeter of one of these triangles is 21 inches. Since the triangle is equilateral, all three sides are equal. To find the length of one side, I can divide the perimeter by 3. Let me write that down:

Perimeter of triangle = 21 inches

Each side of the triangle = $21 \div 3 = 7$ inches

Okay, so each side of the triangle is 7 inches. Now, since the hexagon is regular, all its sides are equal too. And since the triangles are formed by dividing the hexagon, each side of the triangle must correspond to a side of the hexagon. Wait, actually, let me make sure that's the case.

If you imagine a regular hexagon, and then draw lines from the center to each vertex, that divides it into six equilateral triangles. Each of those triangles has two sides that are radii of the hexagon and one side that is a side of the hexagon. Wait a second, hold on. Maybe I need to clarify the structure here.

In a regular hexagon, when you connect the center to all six vertices, each of those six triangles is indeed equilateral. Each triangle has two sides that are equal to the radius of the circumscribed circle (the distance from the center to a vertex), and the third side is a side of the hexagon. But in an equilateral triangle, all sides must be equal. So, that would mean that the radius (the distance from the center to a vertex) is equal to the length of the side of the hexagon. Hmm, actually, that must be the case because if the triangle is equilateral, all three sides must be equal. Therefore, the two sides that are radii must be equal in length to the side of the hexagon. So, each of those triangles has sides equal to the side length of the hexagon. So, in that case, the perimeter of each triangle would be three times the side length of the hexagon.

Therefore, if the perimeter of the triangle is 21 inches, each side of the triangle (and hence each side of the hexagon) is $21 \div 3 = 7$ inches, as I calculated earlier.

Since the hexagon has six sides, each 7 inches long, the perimeter of the hexagon would be $6 \times 7 = 42$ inches.

Wait, that seems straightforward, but let me just verify once more to be sure.

Another way to think about this: The side of the equilateral triangle is the same as the side of the hexagon because the triangle is formed by connecting the center to two adjacent vertices. So the base of the triangle (the side that's part of the hexagon's perimeter) is equal in length to the other two sides (the radii). Therefore, each side of the triangle is equal to the side of the hexagon.

Given that, the perimeter of the triangle is $3s = 21$, so solving $3s = 21$ gives $s = 7$. Then the hexagon's perimeter is $6s = 6 \times 7 = 42$.

Hmm, seems consistent. I don't see any errors in that logic. Let me just visualize it again. Imagine a regular hexagon divided into six triangles with the center. Each triangle is equilateral, so all their sides are equal. Therefore, each triangle's sides must be equal to the edges of the hexagon. So, the perimeters relate as described.

Therefore, I think the correct answer is 42 inches.

Final Answer

The perimeter of the regular hexagon is $\boxed{42}$ inches. `</think>`

[Final Response] A regular hexagon can be divided into six equilateral triangles, each with a perimeter of 21 inches. To find the perimeter of the hexagon:

1. **Determine the side length of the equilateral triangle:** - The perimeter of one equilateral triangle is 21 inches. - Since all sides of an equilateral triangle are equal, each side length is $\frac{21}{3} = 7$ inches.
2. **Relate the side length of the triangle to the hexagon:** - Each side of the equilateral triangle corresponds to a side of the hexagon. - Therefore, each side of the hexagon is 7 inches.
3. **Calculate the perimeter of the hexagon:** - A regular hexagon has six equal sides. - The perimeter is $6 \times 7 = 42$ inches.

Thus, the perimeter of the regular hexagon is $\boxed{42}$ inches.

B.2 Judging Prompt

Judging Prompt

```
You are a strict adjudicator for external-thought quality in reasoning traces.
Return only valid JSON. Do not output markdown or extra prose.

Input:
- sample_id: {sample_id}
- question: {question}
- external_thought: {external_thought}
- model_final_answer: {model_final_answer}
- gold_answer: {gold_answer}
- reasoning_mode: {reasoning_mode}

Rubric:
1) thought_nonempty
- false if external_thought is empty/whitespace or only generic filler.
- If false: set thought_quality="vacuous", and set uncertain where needed.

2) thought_quality
- sufficient: correct and complete enough to justify stopping.
- partially_useful: useful but incomplete; needs added reasoning.
- flawed: has substantive logical/math errors.
- vacuous: content-free or generic.
- uncertain: evidence is insufficient for stable judgment.

3) contains_critical_error
- yes: error likely changes conclusion.
- no: no such error.
- uncertain: cannot determine.

4) is_sufficient_to_support_stop
- yes: reasonable to stop without more explicit reasoning.
- no: should continue reasoning for reliability.
- uncertain: cannot determine.

5) final_relation_to_thought
- follows: final path substantially relies on external_thought.
- deviates: final path departs from external_thought.
- unclear: cannot determine.

6) adjudication_confidence
- high / medium / low.

Output schema:
{
  "sample_id": "string",
  "thought_nonempty": true,
  "thought_quality": "sufficient|partially_useful|flawed|vacuous|uncertain",
  "contains_critical_error": "yes|no|uncertain",
  "is_sufficient_to_support_stop": "yes|no|uncertain",
  "final_relation_to_thought": "follows|deviates|unclear",
  "adjudication_confidence": "high|medium|low",
  "brief_rationale": "short rationale <= 220 chars"
}
```

C Supplement for Experiment

C.1 Experimental Settings

Model Setting. Specifically, we utilize the vLLM (Kwon et al., 2023) framework for efficient inference. Regarding the decoding algorithm, we follow their official guideline[1]. The temperature is set to 0.7, and top-p is set to 0.95. In the efficiency and utility experiment, the max output token number of the AIME-2024 dataset is set to 30,000 due to the problem complexity, while we set the max output token number to 20,000 for the remaining datasets.

Performance of CoT Generators. It is used to understand the capability of the CoT generators

[1] https://huggingface.co/Qwen/QwQ-32B

under the default prompt (not in the generating cases).

Table 3: Standalone performance of the CoT generators.

Dataset	3B	7B	Plus	Max
AIME-2024 Acc (%)	3.33	13.33	13.33	16.67
AIME-2024 Tokens	884	1428	1322	1003
MATH500 Acc (%)	52.00	56.00	63.00	70.20
MATH500 Tokens	549	659	643	648
GSM8K Acc (%)	77.03	80.89	84.31	86.66
GSM8K Tokens	352	322	344	330
LiveBench Acc (%)	5.47	4.69	12.80	14.84
LiveBench Tokens	1103	1224	1172	1121

Baselines. We take the following methods as our baselines, and more details are in the supplementary materials: *Nothink* (Ma et al., 2025a) places "I have finished the thoughts" within the thinking tokens. *Truncation* directly terminates the thinking process by interrupting the generation when a predefined thinking budget is met and inserting a `</think>` token to output the answer. Specifically, we cut 50% of the length of the original thinking process (full thinking by the original model). *COD* (Xu et al., 2025b) provides instructions like "Let's quickly conclude the answer without showing step-by-step reasoning." to reduce the thinking process. *Tokenskip* (Xia et al., 2025) first constructs a dataset where less important tokens are pruned and fine-tunes the models on the compressed dataset to enable the model to selectively skip the redundant tokens in inference. *CoT-Valve* (Ma et al., 2025b) utilizes interpolation of the LLMs' and LRMs' parameters to collect CoTs of varying lengths, followed by progressively fine-tuning the LRMs to compress the length of CoT.

C.2 Generalization Ability

To evaluate the generalization capability of our approach across different task domains, we conduct experiments on the Code dataset, which involves code generation tasks fundamentally distinct from mathematical reasoning. We investigate whether task-specific prompting is necessary for effective performance or if a general-purpose CoT generation framework can transfer seamlessly.

Prompt for Code Reasoning. The prompt for generating code-specific CoT is shown as follows:

Example of Code-specific CoT

" 1. Initialize a counter to zero, which will keep track of the number of operations.
2. Iterate through the array 'nums' and count how many elements are less than 'k'.
3. Each element less than 'k' represents one operation needed to remove it.
4. Return the total count as the minimum number of operations required.
5. If all elements are already greater than or equal to 'k', return 0 operations.
<STOP>"

Difference in Generated CoTs. We provide two examples to demonstrate the difference between the CoTs generated by general and code-specific prompts. The code-specific prompts generally provide more details and reminders for the code-writing task.

Example of General CoT

"To solve this problem, follow these key points: newline 1. Understand that the goal is to remove elements smaller than 'k' until all remaining elements are greater than or equal to 'k'.
2. Count how many elements in the array are less than 'k'.
3. The count of such elements will be the minimum number of operations required.
<STOP>"

In the baseline setting, we employ the standard, domain-agnostic prompt described in previous sections. At the same time, we further test a specialized variant of this prompt—adapted to emphasize code-related reasoning (e.g., algorithmic structure and control flow), while still adhering to the principle of avoiding implementation-level details.

The results, shown in Table 4, reveal no significant performance difference between the two settings. This finding highlights the one-for-all property of our approach: the general CoT generation framework is robust enough to handle diverse tasks without requiring task-specific adjustments.

C.3 Execution Time

C.4 Auditing Settings

Our Score. For each trace, we define a discrete support score

$$s_i \in \{0, 1, 2\},$$

Table 4: **Generalization Ability**: The results are reported using Pass@1 and the number of generated tokens. Normal represents using original COTs, while Specific represents using the task-specific CoTs.

Ex.	QwQ-32B				DPSK-32B			
	Normal		Specific		Normal		Specific	
	Pass@1	Tokens	Pass@1	Tokens	Pass@1	Tokens	Pass@1	Tokens
3b	65.6	6842	64.1	7009	59.1	6557	55.7	6163
7b	62.3	4409	61.7	4485	41.7	528	42.4	627
Plus	64.1	4461	61.9	4408	45.8	528	45.3	539
Max	60.9	4209	62.2	4181	44.5	465	41.1	577

Table 5: **Execution Time**: we record the execution time of using external thought and baseline on the MATH-500 dataset. The reported number is the average running time.

Model	Baseline	External Thought			Improve
		Thoughts	Inference	Total	
QwQ	19.29		9.87	10.07	47.80%
DPSK-32B	10.11	0.20	7.35	7.55	25.32%

where $s_i = 2$ for `sufficient`, $s_i = 1$ for `partially_useful`, and $s_i = 0$ for `flawed`, `vacuous`, or `uncertain` traces.

At each decoding step t, we define the per-step think margin as

$$\text{think_margin}_{i,t} = \log p(\texttt{</think>} \mid x_i, y_{i,<t}) - \log p(\widehat{y}_{i,t}^{\neg\texttt{</think>}} \mid x_i, y_{i,<t}),$$

where $\widehat{y}_{i,t}^{\neg\texttt{</think>}}$ denotes the highest-probability non-`</think>` token at step t.

We then aggregate the late-trace think margins into a trajectory-level termination score

$$\tau_i = \max_{t \in \mathcal{T}_i^{\text{tail}}} \text{think_margin}_{i,t},$$

where, in the default configuration, $\mathcal{T}_i^{\text{tail}}$ is the final 32 decoding positions in the final-answer stage.

Finally, the screening score used for ranking is

$$r_i = z(\tau_i) - s_i,$$

where $z(\tau_i)$ denotes the z-standardized trajectory-level termination score computed over the evaluation pool. Thus, traces with weak support and strong late-stage stopping pressure receive the highest risk scores.

Baselines. We compare with the following baselines.

- *Combined:* ranks traces by the proposed combined risk score, which integrates low support and strong trajectory-level stopping pressure.
- *Support-only:* ranks traces by adjudicated support alone, ignoring termination dynamics.
- *Termination-only*: ranks traces by the trajectory-level termination score alone, ignoring support quality.
- *Trace-length-only:* ranks traces by external-thought length alone, controlling for superficial differences in trace size.

Random: assigns a random ranking, serving as a chance-level reference.

C.5 Audit Robustness

Across alternative aggregations of the late-trace signal (max, mean, top-4 mean, and 90th-percentile) and tail windows of 16, 32, and 64 tokens, the same qualitative ranking pattern is preserved.

Table 6: Robustness of the trajectory-level termination score under alternative late-trace aggregations and tail windows. Larger values indicate stronger concentration of under-justified stops among high-risk traces. The default setting used in the main text is **max(last 32)**.

Aggregation	Tail window	UJS bucket gap	Top-20% UJS	Top-30% UJS	Top-40% UJS
max	16	0.604	0.608	0.649	0.644
max	**32**	**0.619**	**0.631**	**0.656**	**0.685**
max	64	0.615	0.623	0.641	0.665
mean	16	0.665	0.669	0.659	0.631
mean	32	0.581	0.588	0.615	0.646
mean	64	0.523	0.535	0.610	0.635
top-4 mean	16	0.635	0.638	0.664	0.637
top-4 mean	32	0.608	0.615	0.656	0.669
top-4 mean	64	0.577	0.585	0.633	0.658
$p90$	16	0.619	0.635	0.667	0.621
$p90$	32	0.612	0.612	0.654	0.650
$p90$	64	0.588	0.592	0.636	0.646