What the Thinking Token Does in Large Reasoning Models: A Learned Commitment Marker, Not a Quality Check

Anonymous ACL submission

Abstract

Large reasoning models (LRMs) enclose chain-of-thought in a span marked by tokens such as `<think>` and `</think>`, which we call the *reasoning wrapper*. We ask whether the LRM's emission of `</think>` reflects a content-quality judgement that the reasoning suffices, or a reflex to the wrapper itself. To answer this question, we inject externally generated thoughts of three quality tiers (sufficient, partial, materially wrong) into four reasoning models on math and coding tasks, either inside the `</think>` wrapper or outside it.

We find that three of four LRMs accept the wrapped thought on $\geq 97\%$ of trials regardless of content quality, while the same content placed outside the wrapper drops to $\leq 1.5\%$ acceptance in five of six receiver-task cells (one outlier at 34% in GLM-4.7-Flash coding). When these LRMs commit to wrapped wrong content, accuracy drops up to 39% below their no-injection baseline; QwQ-32B, which reopens `<think>` on roughly half of such trials, recovers to within 1% of its baseline. The same pattern shows up at the token level: after the closing `</think>`, the open-think token sinks far down the receiver's next-token distribution while answer-continuation tokens move to the top. Aggregated across the trace and combined with content support, the same ordering predicts on held-out problems where traces produce wrong final answers ($+26.2\%$ accuracy gap). The closing wrapper is a learned commitment marker, not a content-quality judgment.

1 Introduction

Large reasoning models (LRMs) perform well on hard tasks by producing extended chain-of-thought (CoT) reasoning before they answer (Plaat et al., 2024; Xu et al., 2025a; OpenAI, 2025a; Team et al., 2025b; Shao et al., 2024). These models separate reasoning from the final answer with a pair of special tokens, `<think>` and `</think>`, and only the text after the closing tag is returned to the user. We call this pair the *reasoning wrapper*: emitting `</think>` is what tells the model to stop reasoning and write the answer.

Table 1: **Where the injected thought is placed (top) and the two receiver responses we count (bottom).** `{ext.}` is the same external thought in both placements.

Condition	**Prompt structure shown to receiver**
wrapped	`User: Q <think>{ext.}</think>`
unwrapped	`User: Q {ext.} <think>`
Receiver response	**What the receiver emits next**
accept	final answer; no further `<think>`
reject	a new `<think>`...`</think>`, then the final answer

What sits inside the wrapper correlates with reasoning quality (Xu et al., 2025b; Wei et al., 2022), but the act of closing the wrapper has had little direct attention. Prior work asks whether reasoning traces faithfully reflect a model's decision process (Turpin et al., 2023; Lanham et al., 2023; Wang et al., 2025), and one line of efficiency work skips the trace by inserting an empty placeholder inside the wrapper (Ma et al., 2025). Both treat `</think>` as a delimiter and leave open what its emission represents. We vary wrapper presence and content quality independently, so we can read off what the wrapper alone contributes.

To study this question, we intervene directly on the reasoning channel. For each problem, we get a solution sketch from one strong external generator, drop it where the receiver's own reasoning would go, and watch whether the receiver answers or reopens `<think>`. Two factors vary within each problem: the quality of the sketch (*sufficient*, *partial*, *wrong*) and where the sketch sits (*wrapped* inside the `<think>`...`</think>` channel or *unwrapped* in the user turn; Table 1 gives the prompt structures and the two response categories). Further

in-wrapper variants (provenance attribution, off-topic content, an empty wrapper, an in-wrapper reconsider instruction) are introduced in Section 3.2. This separates the wrapper from its content; we then measure what happens, behaviorally and at the token level.

We run this on four reasoning models that span the major training recipes: Distill-Qwen-32B (supervised distillation), GPT-OSS-20B (OpenAI open-weights), GLM-4.7-Flash (RL-trained mixture of experts), and QwQ-32B (RL from a base model). The tasks are MATH-500 (stratified by difficulty) and LiveBench coding. Three of four receivers stop and answer on $\geq 97\%$ of wrapped trials, whether the injected sketch is a full solution, a half-finished setup, or a materially wrong outline. Difficulty does not break this either: at MATH-500 level 5 on *wrong* content, acceptance is still $\geq 96.6\%$.

To isolate the wrapper's effect, we hold the injected thought fixed and vary only its placement or its in-wrapper contents. Acceptance drops to near zero once the wrapper is removed, and an in-wrapper "another assistant's draft thought" attribution barely changes it. A 200-word off-topic essay in place of the thought, or an explicit "let me reconsider" suffix inside the wrapper, still draws $\geq 93\%$ acceptance on three of four receivers. Even an empty `<think></think>` keeps the two non-Qwen receivers committing on $\geq 98\%$ of trials, while the two Qwen receivers reject it. What gets accepted is the wrapper, not its contents; whether the wrapper alone suffices splits along the Qwen-family axis. Accepting whatever is wrapped, with no check on the content, has a cost in final-answer accuracy. When the format-driven receivers commit to a wrapped *wrong* thought, accuracy drops well below their no-injection baselines. QwQ-32B, which re-opens `<think>` on roughly half of those trials, recovers to within 1% of baseline on the ones it re-opens: acceptance pays the cost, rejection recovers it.

To see why the wrapper pulls so hard, we look at the receiver's next-token probabilities right after the prompt's closing `</think>`. The open-think token drops far down the candidate list in three of four receivers while the close-think token stays in the top ten; remove the wrapper and the pattern reverses. The closing wrapper is a learned *commitment marker*: once `</think>` is in the prompt, the receiver tends to continue an answer rather than re-open thinking, plausibly because the training distribution overwhelmingly puts a final answer, not more reasoning, after `</think>`.

The same ordering is also a usable screening signal. We aggregate the close-vs-open margin over the receiver's continuation and combine it with a content-quality score; on held-out problems this predicts which traces produce wrong final answers, with a $+26.2\%$ accuracy gap ($p{=}2.6{\times}10^{-45}$). Content quality and the wrapper-commitment signal are largely independent, and together they isolate a top-quintile bucket whose answers are wrong more often than right.

Our contributions can be summarized as follows:

- A probe that varies the reasoning wrapper independently of the content it encloses, so we can read off what the closing `</think>` alone contributes to the receiver's stop decision.
- Across four reasoning models and two tasks, we show that acceptance tracks the wrapper rather than content quality, and that accepting wrapped wrong content costs final-answer accuracy.
- Token-level evidence that the closing wrapper is a learned commitment marker: it makes the receiver favour continuing an answer over reopening thinking. The same signal, aggregated across the trace and combined with content support, predicts on held-out problems which traces produce wrong final answers.

2 Probing the Stop Decision with External Thoughts

We study how an LRM treats content placed in its reasoning channel when deciding whether to stop reasoning and emit a final answer. Our core tool is an *external-thought* intervention: we insert an externally generated solution sketch into the position normally occupied by the model's own reasoning, then observe whether the model proceeds to answer or instead re-opens a `<think>` segment and continues to reason. The setup below holds two factors that could in principle govern this stop decision under controlled variation:

- the *quality* of the injected content – whether the thought is a correct solution, a partial setup, or a materially wrong outline; and
- the *format* in which it is presented – whether the content sits inside the `<think>. . . </think>` channel and whether it is attributed to a third party.

Separating these two factors is the goal of the analyses in Section 3.

2.1 Intervention Setup

Receivers. We probe four reasoning models that span the major public training pipelines for reasoning: DeepSeek-Distill-Qwen-32B (Yang et al., 2024), distilled from DeepSeek-R1; GPT-OSS-20B (OpenAI, 2025), OpenAI's recent open-source reasoning model; QwQ-32B (Team, 2025b), trained by reinforcement learning from a base model; and GLM-4.7-Flash (Team et al., 2025a), a mixture-of-experts reasoning model also trained with reinforcement learning. This selection lets us check whether the stopping behavior we observe is shared across architecturally and procedurally distinct reasoning models, rather than an artifact of any single training recipe or base family.

Thought generator and quality validation. All injected thoughts are produced by a single strong generator, DeepSeek-V4-Pro (DeepSeek-AI, 2026), used in a controlled way: by varying only the prompt, we elicit three quality tiers for every problem – *sufficient* (a complete and correct solution outline), *partial* (a correct setup that stops short of a full derivation), and *wrong* (a confidently stated outline that is materially incorrect). Using one generator across all tiers controls for stylistic and capability differences between generators, so that the three tiers differ chiefly in content quality. Standalone, V4-Pro reaches 90.5% on the 200 MATH-500 problems and 77% pass@1 on the 100 LiveBench coding problems (Table 2), 5 to 8% above the strongest receiver, so the sufficient-tier thoughts we inject are not bottlenecked by the generator's own capability ceiling.

Table 2: **Standalone performance of DeepSeek-V4-Pro** on the same 200 MATH-500 problems and 100 LiveBench coding problems used in our experiments.

	MATH-500 (acc.)	LiveBench (pass@1)
DPSK-V4-Pro	90.5%	77.0%

To verify that the generator hits its targeted tier, every produced thought is independently labeled by a panel of three LLM judges – `gpt-5.1-chat`, `gemini-3-flash-preview`, and `deepseek-v4-pro` – under a unified rubric. Table 3 reports inter-judge agreement: Fleiss κ is substantial on both tasks and the three judges concur unanimously on roughly 70% of thoughts. The two end tiers (sufficient, wrong) are hit by V4-Pro with $\geq$ 93% on-target rate under majority vote; non-matching slots are refilled from a substitution buffer drawn from the same problem. Per-tier on-target rates, per-judge label distributions, and per-field κ are reported in Appendix D.2.

Table 3: **Three-judge panel agreement on thought quality.** Agreement is substantial under all three measures.

	MATH	Coding
Fleiss κ (quality, 4-way)	0.65	0.68
Pairwise agreement	79.8%	79.8%
Unanimous (all 3 judges)	70.1%	71.0%

Tasks. We evaluate on two domains. For mathematics we use MATH-500 (Lightman et al., 2023), sampling 200 problems stratified across its five difficulty levels (40 per level). For coding we use the LiveBench coding subset (White et al., 2024), sampling 100 problems balanced across its two task categories (`LCB_generation` and `coding_completion`). For each problem we generate all three quality tiers from the same generator, so every contrast we report is *within-problem matched*: the same question is presented to the same receiver with thoughts that differ in quality.

Inference template and decoding. For each problem the retained thought is enclosed within `<think>` and `</think>` and inserted into the receiver's reasoning channel between the assistant's turn-start and the first generated token; the intervention is intentionally minimal and does not alter parameters, decoding rules, or training data. All receivers and the V4-Pro generator are decoded with temperature 0.7, top-p 0.95, and a 30,000-token budget, one sample per problem. The full template, the literal strings used for each in-wrapper manipulation, the no-injection baseline, and the acceptance-detection rule are given in Appendix A.

Inference Template

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

3 How Models Respond to External Thoughts

The setup of Section 2 lets us measure how each receiver actually responds when a wrapped external

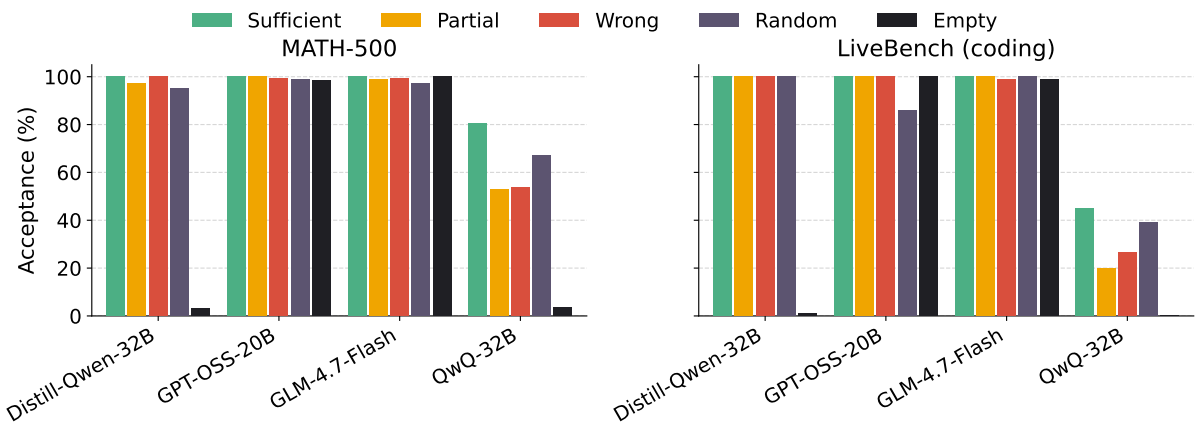

Figure 1: **Wrapped-thought acceptance rates across receivers, tasks, and content tiers.** For each receiver we report the percentage of trials on which the receiver emits a final answer without re-opening `<think>` after the wrapped thought.

thought is placed in its reasoning channel. We first establish the basic phenomenon: wrapped thoughts overwhelmingly induce stopping, across receivers, tasks, and content qualities (Section 3.1). We then ask whether what is being accepted is the content or the wrapper, and which receivers, if any, escape the pattern.

3.1 Wrapped Thoughts Overwhelmingly Stop the Receiver

We measure each receiver's *acceptance rate*: the fraction of trials on which the receiver emits a final answer after the wrapped thought without re-opening a new `<think>` segment. A high acceptance rate means the receiver treated the injected thought as a sufficient end-of-reasoning state for that problem.

Figure 1 shows acceptance across content tiers and ablations, separately for MATH-500 and the LiveBench coding subset. For three of the four receivers, namely Distill-Qwen-32B (supervised distillation), GPT-OSS-20B (open-weights), and GLM-4.7-Flash (RL-trained mixture of experts), acceptance does not vary meaningfully with the content of the wrapped thought, whether it is a complete solution outline (*sufficient*: 100% across the three receivers and both tasks), an incomplete but on-track setup (*partial*: 97 to 100%), a materially incorrect outline (*wrong*: 98.9 to 100%), or a 200-word off-topic essay drawn from a randomized topic pool (*random*: 95 to 99% on math, 86 to 100% on coding). The *empty*-wrapper case (no content at all) is a separate manipulation that exposes a model-dependent split we return to in Section 3.2.

The same flatness extends to problem difficulty within MATH-500 (Figure 2): across the five difficulty levels, the format-driven cluster never drops below 89.5% acceptance in any tier, and remains $\geq$ 96.6% on *wrong* content even at level 5.

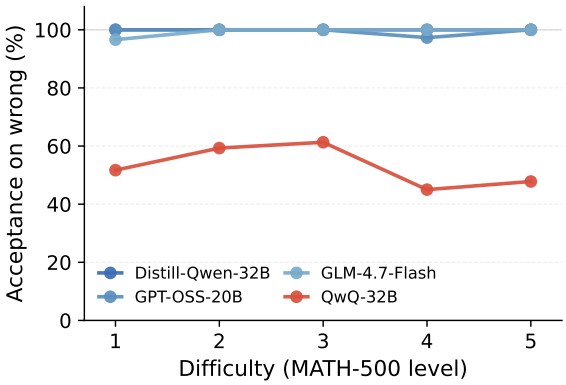

Figure 2: **Acceptance is flat across MATH-500 difficulty for the format-driven cluster.** Acceptance rate on the *wrong* content tier by difficulty level (1 = easiest, 5 = hardest).

The fourth receiver, QwQ-32B, behaves visibly differently. Acceptance is 80.6% on MATH *sufficient* thoughts (versus $\geq$ 97% for the others), and drops to 52.8% on *partial* and 53.8% on *wrong*; on coding it is uniformly lower still (44.8%, 20.0%, 26.5%). When QwQ does not accept, it re-opens `<think>` after the injected thought and reasons further; we examine the cost and recovery of this split in Section 4.

Takeaway. Wrapped external thoughts elicit near-universal acceptance ($\geq$ 97%) in three of four receivers, regardless of the relevance

Table 4: **Acceptance (%) under wrapper manipulations**, on two content bases (partial-tier reasoning and a 200-word off-topic essay drawn from a 32-topic pool) and both tasks. **W** = wrapped, **U** = unwrapped (same content placed in the user turn); **prov** = wrapped content with an in-wrapper provenance attribution; **AC** (anti-commit) = wrapped content with an in-wrapper reconsider suffix. Provenance is applied only to partial-tier content (semantically a "draft thought").

	MATH partial				*Coding partial*				*MATH random*			*Coding random*		
Receiver	W	U	prov	AC	W	U	prov	AC	W	U	AC	W	U	AC
Distill-Qwen-32B	97.0	0.5	91.4	98.5	100.0	1.0	100.0	100.0	95.0	5.0	93.5	100.0	2.0	100.0
GPT-OSS-20B	100.0	0.0	100.0	100.0	100.0	0.0	100.0	100.0	99.0	0.0	99.5	86.0	0.0	98.0
GLM-4.7-Flash	99.0	1.5	98.5	98.5	100.0	20.0	100.0	100.0	97.0	10.0	98.0	100.0	20.0	100.0
QwQ-32B	52.8	0.0	24.9	20.0	20.0	8.0	9.0	4.0	67.0	9.0	37.5	39.0	1.0	16.0

or quality of what is wrapped. QwQ-32B is the exception: its stop decision tracks content, and its rethinking on wrong thoughts recovers most of the accuracy that acceptance would cost.

3.2 The Wrapper, Not the Content, Drives Acceptance

To isolate what the three format-driven receivers stop on, we run four wrapper manipulations across two content bases (partial-tier reasoning and a 200-word off-topic essay drawn from a 32-topic pool) on both tasks (Table 4): *unwrapped* (same content outside the `<think>. . . </think>` channel), *provenance* (an attribution prefix), *anti-commit* (an in-wrapper reconsider instruction), and *empty* (`<think></think>` with no content). Exact strings and essay construction are in Appendix A.

Removing the wrapper collapses acceptance. On partial content, unwrapping drops acceptance by 80 to 100% for the three format-driven receivers (Distill-Qwen-32B 97.0% $\rightarrow$ 0.5% on MATH). The same collapse holds when the content is replaced with a 200-word off-topic essay: wrapped acceptance stays at 95–100% on math and 86–100% on coding; unwrapping drops it to 0–20%.

In-wrapper modifications do not move acceptance. *Provenance* shifts MATH acceptance by at most $\sim$6% for the three format-driven receivers (Distill-Qwen-32B 97.0% $\rightarrow$ 91.4%; GPT-OSS-20B and GLM-4.7-Flash within 1%). *Anti-commit*, an explicit reconsider suffix before the closing `</think>`, leaves acceptance within $\pm$2% of its no-suffix control in 11 of 12 cells across four content settings; on partial-tier content the same three receivers still commit at $\geq$98.5% once the suffix is added.

QwQ-32B is the only receiver reading the wrapper's content. QwQ-32B moves on every in-wrapper intervention: provenance drops MATH acceptance by $-$28% (52.8% $\rightarrow$ 24.9%); anti-commit drops it by 16 to 33% across all four content settings. The wrapper contrast itself is just as sharp (MATH partial 52.8% $\rightarrow$ 0% on unwrapping), and Section 4 shows that QwQ's rejection on wrapped *wrong* content recovers most of the accuracy that acceptance would cost.

Empty wrapper splits receivers along the Qwen-family axis. With no content between the tags, GLM-4.7-Flash and GPT-OSS-20B still commit on $\geq$ 98% of trials on both tasks; Distill-Qwen-32B and QwQ-32B reject on $\geq$ 96% and re-open a fresh reasoning channel. The split tracks chat-template structure (GPT-OSS's harmony format places a final-output marker after the empty prompt; GLM-4.7-Flash's template auto-injects a second `</think>`; the Qwen receivers see only the bare `<think></think>`).

Takeaway. Wrapper presence drives acceptance; content quality, relevance, provenance, and explicit reconsider instructions leave it essentially unchanged. Whether the wrapper alone, with no content, suffices splits along the Qwen-family axis.

4 Two Faces of the Wrapper: Cost of Acceptance, Benefit of Rejection

The wrapper makes acceptance the receiver's default, but accuracy depends entirely on what the wrapper holds (Figure 3). The accept path uses only 3 to 22% of the no-injection baseline's reasoning tokens on every cell, while QwQ's reject path pays 60 to 94% to re-derive.

Accept is cheap regardless of content; its accuracy is whatever the content delivers. On

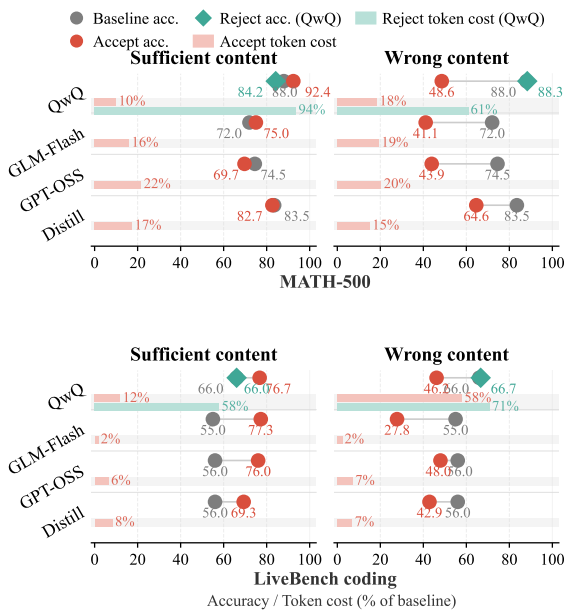

Figure 3: **Two faces of the wrapper.** 2×2 panels (task × tier). Dots = accuracy (gray: baseline, coral: accept, green: QwQ reject); light bars = token cost as % of baseline reasoning tokens.

wrapped *wrong* content the four receivers land at 64.6% / 43.9% / 41.1% / 48.6% on MATH and 42.9% / 48.0% / 27.8% / 46.2% on coding (Distill-Qwen-32B / GPT-OSS-20B / GLM-4.7-Flash / QwQ-32B), losses of 19 to 39% relative to baseline. On wrapped *sufficient* content the same accept path tracks or exceeds baseline (Figure 3, left column): same wrapper, same token cost, different accuracy. The cost grows with problem difficulty: on MATH-500, Distill-Qwen-32B drops from 75.9% at level 1 to 34.8% at level 5.

Format-driven receivers have no second branch and pay every time. ≥97% wrapped acceptance means the cost above is paid on essentially every wrong trial. This is true even when the wrapped content bears no relation to the question: Distill-Qwen-32B commits to off-topic prose on 100% of coding trials and scores 46% pass@1, while the same content delivered *unwrapped* (refused on 98% of trials) returns to 74%, a 28% swing from the wrapper alone.

QwQ recovers via rejection, but only on wrong content. QwQ's reject branch on wrapped *wrong* reaches 88.3% on MATH and 66.7% on coding, within 1% of baseline and +39.7% / +20.5% above the accept branch. The recovery is content-conditional: on *sufficient* content rejection slightly hurts (84.2 vs. 92.4% MATH; 66.0 vs. 76.7% coding), and on *partial* the two branches sit within ∼4%. Rethinking pays only where format-driven receivers pay their largest cost.

Takeaway. The wrapper is not a free signal: it makes acceptance the receiver's default, and the cost of that default is the accuracy of whatever wrapped content arrives. Format-driven receivers pay the full cost; QwQ's rethink-on-reject behavior partially escapes it, but only on the subset of trials where it actually rejects.

5 The Wrapper at the Token Level

Sections 3 and 4 establish the wrapper-driven stop behaviorally. We now ask what the next-token distribution looks like at the moment the wrapper closes.

For every saved (x, y) pair we run a teacher-forced forward pass and read off the distribution at the *decision position* $t = 1$, the first generated token after the prompt: $p_t(\cdot) := p_\theta(\cdot \mid x, y_{<t})$. The two target tokens are the *open-think* token v_{open}, which re-opens the thinking channel (`<think>` for the Qwen-based receivers, `<|channel|>` for GPT-OSS-20B), and the *close-think* token v_{close}, which closes it (`</think>` or `<|end|>`).

We summarize each receiver's distribution with two quantities. The *boundary margin*

$$m^{(c)} \;:=\; \text{median}\left[\, \log p_t(v_{\text{close}}) - \log p_t(v_{\text{open}}) \,\right]$$

over samples in condition c is positive when v_{close} has higher log-probability than v_{open}, and negative otherwise. The candidate-list rank

$$\text{rank}_t(v) \;:=\; 1 + \left|\{u \in \mathcal{V} : p_t(u) > p_t(v)\}\right| \quad (1)$$

locates each boundary token within the receiver's candidate list. All comparisons below are within-receiver: vocabulary and calibration differ across model families, so absolute ranks and probabilities are not commensurate across rows.

Boundary preference changes sign with the wrapper. For QwQ-32B and GPT-OSS-20B the boundary margin is negative under *unwrapped* (open-think has higher log-probability) and positive under *wrapped* (close-think has higher log-probability): −4.8 → +8.7 and −42.9 → +6.6 respectively (Table 5). For GLM-4.7-Flash the margin is already positive under *unwrapped* and grows further under *wrapped* (+8.0 → +12.5).

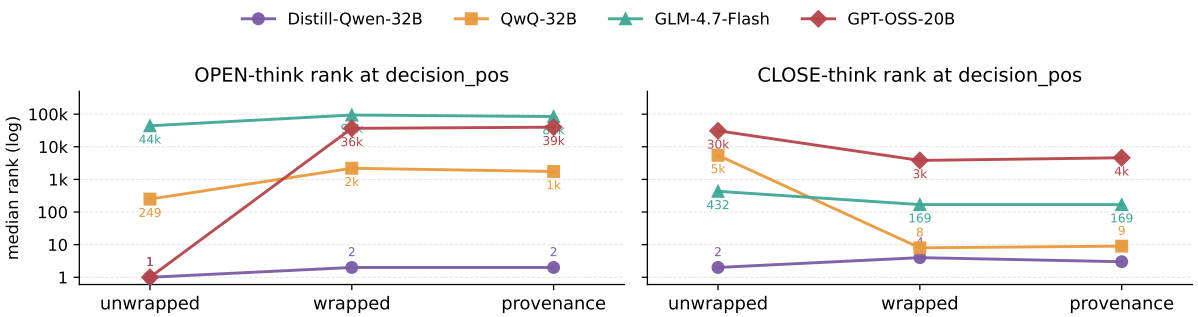

Figure 4: **Decision-position rank of the open-think (left) and close-think (right) tokens, MATH-500 partial.** Each line is one receiver; y-axis is median rank on log scale. *Wrapped* buries the open-think token (top of left panel) and keeps the close-think token in the top ten (bottom of right panel); *unwrapped* reverses the contrast; *provenance* sits on top of *wrapped* for every receiver.

The $\geq 97\%$ wrapped vs. $\leq 1\%$ unwrapped acceptance gap on these three receivers mirrors this sign pattern.

In rank space: open-think buried, close-think near the top. Figure 4 shows the two halves of this contrast. Under *wrapped* the open-think token sits deep in the candidate list for three receivers: median rank 2,196 (QwQ-32B), 36,782 (GPT-OSS-20B), and 93,970 (GLM-4.7-Flash). Removing the wrapper lifts it to rank 249, 1, and 44,039 respectively. The probability shift is largest for GPT-OSS-20B: $p_t(v_{\text{open}})$ goes from 1.00 under *unwrapped* (the next token is essentially deterministic) to 1.3×10^{-12} under *wrapped*, a drop of twelve orders of magnitude. The close-think token shows the mirror image. Under *wrapped* it sits in the top ten on three receivers (rank 4 for Distill-Qwen-32B, 8 for QwQ-32B, 169 for GLM-4.7-Flash); under *unwrapped* its probability drops by one to three orders of magnitude.

Distill-Qwen-32B: same behavior, different fingerprint. Distill-Qwen-32B does not match either summary. Its boundary margin stays negative in every condition (Table 5), and the open-think rank stays in the top two whether the wrapper is present or removed. Yet its behavior matches the format-driven cluster (97% wrapped acceptance vs. 0.5% unwrapped). The boundary contrast for this receiver shows up not as `<think>` sinking but as another answer-continuation token (typically a leading digit, a space, or the first character of the boxed answer) outranking it; `<think>` stays a viable candidate that is never sampled.

Table 5: **Boundary margin $m^{(c)}$ at the decision position**, MATH-500 partial. Positive values mean v_{close} has higher probability than v_{open}; negative values the reverse.

	unwrapped	*wrapped*	*provenance*
Distill-Qwen-32B	−24.2	−7.7	−7.4
QwQ-32B	−4.8	+8.7	+8.0
GLM-4.7-Flash	+8.0	+12.5	+12.5
GPT-OSS-20B	−42.9	+6.6	+6.6

Provenance is invisible at the logit level. Provenance tracks the wrapped distribution within $\sim 5\%$ on both panels of Figure 4, and the wrapped and provenance margins in Table 5 agree to within 0.7 for every receiver. The wrapper, not the attribution, is what the next-token distribution covaries with, in line with the behavioral finding in Section 3.2.

Takeaway. The wrapper's effect is visible in the next-token distribution: the boundary margin and candidate-list fingerprint covary with the wrapper close, either through the open-think token being buried (three receivers) or through an answer-continuation token outranking it (Distill-Qwen-32B), and explicit attribution leaves both unchanged. Acceptance is not an evaluation; it is the natural continuation of the logit state observed under the wrapper.

6 Diagnosing Risky Reasoning Traces

If the wrapper acts as a learned commitment marker, the close-vs-open ordering in Section 5 should have predictive value at the trace level: a trace is more likely to be wrong when its content is weakly supportive *and* the receiver, position-by-position, prefers closing over re-opening the wrapper. We turn this into a held-out screening

test.

Setup. For each trace from Section 3.2 we compute a *support* score from the three-judge majority label (`flawed`, `vacuous`, or `uncertain` → 0; `partially_useful` → 1; `sufficient` → 2) and a *termination* score equal to the late-trace maximum of the position-wise boundary margin $\log p_t(\texttt{</think>}) - \log p_t(\texttt{<think>})$ over the final 32 tokens. We form three rankings: *support-only*, *termination-only*, and *combined* (z-standardised sum of low support and high termination). The split is question-level; the disjoint held-out half is scored on final-answer accuracy against gold, which is independent of the LLM judgments used to build the scores (Appendix B).

Result. Across n=2,732 held-out traces (Table 6), all three rankings are individually predictive: support-only $+13.8\%$ (p=1.2×10^{-13}), termination-only $+16.3\%$ (p=1.9×10^{-18}), combined $+26.2\%$ (p=2.6×10^{-45}). The combined gap nearly doubles the better single-signal gap, and the two component dimensions overlap only marginally ($13.8 + 16.3 - 26.2 \approx 3.9$ pp), so the two signals carry largely independent information about trace risk. In the extreme quintile, combined pulls held-out accuracy from 77.0% on the safest fifth down to 41.7% on the riskiest fifth.

Which token channel carries the signal. Replacing the boundary margin with $\log p_t(\texttt{<think>})$ alone gives a non-significant termination Δ ($+1.6\%$); $\log p_t(\texttt{</think>})$ alone gives $+15.1\%$ (p=4.3×10^{-16}); the boundary contrast gives $+16.3\%$. The pressure to close the wrapper, contrasted against the pressure to re-open it, is the operative dimension, the same one Section 5 identified at t=1.

Per-receiver gaps. Within each receiver, combined Δ is uniformly positive: Distill-Qwen-32B $+10.2\%$, QwQ-32B $+9.3\%$, GPT-OSS-20B $+13.6\%$, GLM-4.7-Flash $+6.8\%$. Part of the pooled $+26.2\%$ reflects inter-receiver variance (GLM-Flash clusters in the high-risk half; QwQ and Distill in the low-risk half), so within-receiver gaps are the operative signal for deployment.

Takeaway. Content support and the close-vs-open commitment signal carry largely independent information about trace risk. Combined, they isolate a top-quintile bucket whose final answers are wrong more often than right.

Table 6: **Held-out screening across four receivers (n=2,732).** Final-answer accuracy (%) on disjoint held-out traces, screened-in (low-risk half) vs. screened-out (high-risk half) under each ranking. Termination uses the boundary margin $\log p_t(\texttt{</think>}) - \log p_t(\texttt{<think>})$, taken as the maximum over the final 32 tokens of the receiver's continuation.

Ranking	In	Out	Δ (%)	p
Support-only	69.1	55.3	$+13.8$	1.2×10^{-13}
Termination-only	70.4	54.1	$+16.3$	1.9×10^{-18}
Combined	**75.3**	**49.1**	$\mathbf{+26.2}$	$\mathbf{2.6\times10^{-45}}$

7 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 (Guo et al., 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).

Thinking probes. Prior probes of CoT-as-reasoning ask whether trace *content* faithfully reflects the model's answer (Turpin et al., 2023; Lanham et al., 2023; Wang et al., 2025); we instead ask whether the receiver's decision to *stop* reasoning evaluates content sufficiency or merely reacts to the wrapper format. Another close precedent is NoThink (Ma et al., 2025). It demonstrates that reasoning can be bypassed for efficiency gains by inserting minimal content into the reasoning channel. Instead, we directly probe whether the stopping decision evaluates content quality, varying wrapper presence and content quality as independent factors.

8 Conclusion

For present-day LRMs, the closing `</think>` acts as a learned commitment marker rather than a content-quality judgement; any pipeline that places external text inside the reasoning channel inherits this commitment by default. Whether the bias is intrinsic to the wrapper format or specific to current training pipelines is an open question.

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. Our content-tier labels (*sufficient*, *partial*, *wrong*) rely on a panel of three strong LLM judges (`gpt-5.1-chat`, `gemini-3-flash-preview`, `deepseek-v4-pro`). 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 labels.

Observational mediator analysis. The token-rank evidence in Section 5 is teacher-forced and observational: we show that the boundary margin and candidate-list fingerprint covary with the wrapper close, but we do not intervene on the logits themselves. A strict causal-mediation claim would require ablating the close-think representation and measuring the downstream acceptance shift, which we leave to future work.

References

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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*.

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*.

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

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

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*.

GLM Team, Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, Kedong Wang, Lucen Zhong, Mingdao Liu, Rui Lu, Shulin Cao, Xiaohan Zhang, Xuancheng Huang, Yao Wei, and 152 others. 2025a. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *Preprint*, arXiv:2508.06471.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025b. 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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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*.

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 Inference Setup Details

Inference template. For each problem the retained thought is enclosed within `<think>` and `</think>` and inserted into the receiver's reasoning channel between the assistant's turn-start and the first generated token. The intervention does not alter parameters, decoding rules, or training data, and only places content in the slot the receiver would normally fill itself.

Wrapper manipulations (literal strings). The in-wrapper interventions in Section 3.2 use the following exact strings:

- *Provenance:* the wrapped content is prefixed inside the `<think>...</think>` channel with "`Here is another assistant's draft thought:`".
- *Anti-commit:* the wrapped content is suffixed inside the channel with "`Wait, this is incomplete, let me reconsider.`", placed before the closing `</think>`.
- *Empty:* the channel is opened and closed with no content between the tags (`<think></think>`).
- *Unwrapped:* the same content is placed in the user turn (appended to the question) and the `<think>` that would normally open the receiver's own reasoning channel is left in place.

Off-topic essay construction (*random* tier). Essays used in the *random* tier are produced by DeepSeek-V4-Pro under the same decoding settings as the thought generator. Each essay is approximately 200 words on a topic drawn uniformly at random from a fixed pool of 32 everyday topics (e.g., marathon training, bird migration, sourdough baking, urban gardening). One unique essay is drawn per problem, yielding 200 essays on MATH-500 and 100 on the LiveBench coding subset; no essay references the corresponding question.

Decoding hyperparameters. All receivers and the V4-Pro generator are decoded with temperature 0.7 and top-p 0.95 (QwQ-32B's official recommendation (Team, 2025b), used uniformly across receivers for cross-model comparability), a maximum decoding budget of 30,000 tokens, and one sample per problem. Coding pass@1 reported in the main text is computed from this single sample per problem against the LiveBench reference test suite.

No-injection baseline. The *no-injection baseline* cell for each (receiver, task) pair runs the same receiver under the same decoding settings on the same problem set with no injected thought: the receiver produces its own `<think>...</think>` reasoning and final answer from scratch. Baseline accuracy and pass@1 numbers in the main text are computed this way.

Acceptance detection. A trial is classified as *reject* if the receiver's generated continuation (everything after the prompt) contains a `</think>` token, indicating the receiver opened and closed a new thinking segment; otherwise it is classified as *accept*. Under wrapped conditions the prompt already ends with a closing `</think>` from the injected thought, so a fresh `</think>` in the continuation is unambiguous evidence of re-opened reasoning. Under unwrapped and no-injection conditions the prompt ends with a template-driven `<think>`, so receivers normally emit `</think>` during their own reasoning; at the 30,000-token decoding budget, truncation before close is $\leq 24\%$ on every cell and $\leq 7\%$ on every wrapped cell (wrapped commitment typically occurs within $<$300 tokens).

Token-rank methodology (Section 5). For each saved (x, y) pair, where x is the receiver's prompt under the specified condition and y is the receiver's actual generated continuation under that prompt, we run a teacher-forced forward pass and read the next-token distribution at the decision position $t =$

1, the first generated token after the prompt. All comparisons are between conditions on the *same* receiver: vocabulary and calibration differ across model families, so absolute ranks and probabilities are not commensurate across rows. Rank-trajectory plots use the MATH-500 partial-tier cells with one forward pass per problem (200 forward passes per receiver per condition).

B Trace Screening Protocol

Support score. The support score for trace i is

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

where $s_i = 2$ for thoughts whose three-judge majority label is `sufficient`, $s_i = 1$ for `partially_useful`, and $s_i = 0$ for `flawed`, `vacuous`, or `uncertain` (judge protocol in Appendix D.2).

Per-step boundary margin and trajectory-level termination score. At each decoding step t in the receiver's final-answer continuation, we define the per-step boundary margin as the log-odds of closing versus re-opening the reasoning wrapper:

$$\begin{aligned}\text{think_margin}_{i,t} = \log p(\texttt{</think>} \mid x_i, y_{i,<t}) \\ - \log p(\texttt{<think>} \mid x_i, y_{i,<t}).\end{aligned}$$

This is the continuous-valued counterpart of the close-vs-open ordering documented at $t{=}1$ in Section 5, evaluated position-by-position over the final-answer span. We aggregate the late-trace margins into a trajectory-level termination score

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

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

Combined risk score. The combined risk score is

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

where $z(\tau_i)$ is the z-standardised termination score over the evaluation pool. High r_i corresponds to weak support paired with strong late-stage closing pressure.

Held-out protocol and selectivity. Traces are split at the question level so that all traces from one problem sit on one side of the split. The cutoff is calibrated on the calibration half; evaluation reports final-answer accuracy on the disjoint held-out half without retraining. Selectivity (fraction screened in) is held fixed across rankings so that gap comparisons are apples-to-apples.

Robustness. The combined ranking's accuracy gap is stable under alternative aggregations of the late-trace signal (max, mean, top-4 mean, 90th percentile) and tail windows of 16, 32, and 64 tokens. Among single-channel termination scores, $\log p_t(\texttt{<think>})$ alone is non-significant ($+1.6\%$ termination Δ); $\log p_t(\texttt{</think>})$ alone gives $+15.1\%$ ($p{=}4.3{\times}10^{-16}$); the boundary contrast used above gives $+16.3\%$ ($p{=}1.9{\times}10^{-18}$). The boundary contrast is also the cleanest match to the Section 5 mechanism, which is stated in terms of the close-vs-open ordering, not of either token's absolute probability.

C 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."

D Supplement for Effectiveness of External Thoughts

D.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.

D.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"
}
```

Inter-judge agreement. We compute three-way agreement on the panel's last label per thought (Table 7). On the headline *thought_quality* field, Fleiss κ is substantial on both tasks (0.65 on MATH, 0.68 on Coding). Agreement on *contains_critical_error* is the highest of the three labeled fields (0.82 MATH, 0.75 Coding) – detecting an actual error is more objective than judging "is this enough to justify stopping," which is the field with the lowest agreement (0.58 MATH, 0.68 Coding). We therefore use the four-way *thought_quality* label for tier assignment, since it has the strongest agreement among the fields that map directly to our three quality tiers.

Table 7: Per-field Fleiss κ on the three-judge panel. Last label per judge.

Field	MATH	Coding
thought_quality (4-way)	0.65	0.68
contains_critical_error (3-way)	0.82	0.75
is_sufficient_to_support_stop (3-way)	0.58	0.68

Per-judge label distribution. Table 8 shows the marginal label distribution of each judge over the union of all generated thoughts. The judges agree on *ordering* (`gemini` $\geq$ `deepseek-v4-pro` $\geq$ `gpt-5.1-chat` on the "is this enough" axis) but draw the sufficiency threshold in different places: `gemini` is the most lenient, `gpt-5.1-chat` the most demanding. The three-judge majority absorbs this leniency spread; we report the majority label as the operative quality tier and the per-judge distributions here for transparency.

Per-tier on-target rate and partial-tier drift. Table 9 reports the fraction of generated thoughts whose three-judge majority label matches the targeted tier. The two end tiers are hit reliably ($\geq$ 93%), while the *partial* tier has a noticeably lower on-target rate (59.3% on MATH, 66.9% on Coding) because eliciting a thought that is *useful but incomplete* from a strong generator is intrinsically harder than eliciting a correct outline or an incorrect one. Among partial-targeted thoughts in MATH, the panel's majority label distributes as: partial 59.3%, sufficient $\sim$32% (over-delivers), wrong $\sim$8% (drifts low). For Coding the corresponding split is partial 66.9%, sufficient $\sim$18%, wrong $\sim$9% (rest ambiguous). The receiver experiments in Section 3 use the generator-target label as the operative tier; the within-tier heterogeneity this introduces strengthens rather than weakens the wrapper-vs.-content contrast in Section 3, because acceptance remains uniformly high on partial-target thoughts even though roughly half of them are not in fact sufficient.

Generator/judge overlap. `deepseek-v4-pro` serves as both the thought generator and one of three judges in the panel. The three-judge majority bounds the influence of any single judge, and the per-judge distribution in Table 8 shows `deepseek-v4-pro` sits between `gemini` (lenient) and `gpt-5.1-chat` (demanding) rather than systematically endorsing its own outputs; its label distribution on the partial tier (18.6% partial on MATH, 18.1% on Coding) is closer to `gpt-5.1-chat` than to `gemini`, so the majority vote does not collapse to a self-endorsement of the generator's intended tier.

Table 8: Per-judge *thought_quality* marginals (last label per thought).

Task	Judge	sufficient	partial	flawed	vacuous
MATH	`gpt-5.1-chat`	50.2%	24.3%	25.1%	0.4%
	`gemini-3-flash-preview`	68.2%	5.0%	26.6%	0.2%
	`deepseek-v4-pro`	55.6%	18.6%	25.7%	0.2%
Coding	`gpt-5.1-chat`	39.1%	24.6%	35.1%	1.1%
	`gemini-3-flash-preview`	59.8%	7.2%	31.9%	1.1%
	`deepseek-v4-pro`	46.4%	18.1%	35.1%	0.4%

Table 9: Per-tier on-target rate of injected thoughts (three-judge majority).

Targeted tier	MATH	Coding
sufficient	97.4%	98.9%
partial	59.3%	66.9%
wrong	93.1%	93.4%