Article

Decision-Theoretic Self-Entailment Gating for Adaptive Chain-of-Thought Inference on GSM8K

Firstname Lastname [1], Firstname Lastname [2] and Firstname Lastname [2,*]

[1] Affiliation 1; e-mail@e-mail.com
[2] Affiliation 2; e-mail@e-mail.com
* Correspondence: e-mail@e-mail.com; Tel.: (optional; include country code; if there are multiple corresponding authors, add author initials) +xx-xxxx-xxx-xxxx (F.L.)

Abstract

Chain-of-Thought (CoT) prompting can raise reasoning accuracy but often increases token cost and can produce fluent rationales that do not actually support the final answer. Standard self-consistency aggregates sampled CoTs as exchangeable votes, which can amplify plausible-but-unsupported traces. We study a training-free inference procedure that casts CoT use as a sequential decision problem with two metacognitive operations: deciding when deliberation is worth the cost, and discounting rationales that fail to increase support for their own answers. Our method, DT-SEACoT, first samples short direct answers to estimate an empirical belief and uncertainty; it gates deliberation based on this uncertainty. When deliberation is triggered, it samples CoT traces and weights each candidate answer by a self-entailment likelihood ratio computed via teacher-forced log-likelihood under the same model, enabling early stopping when the induced posterior becomes concentrated. We instantiate DT-SEACoT with `google/flan-t5-large` on a 200-item GSM8K slice (50 for tuning, 150 for final evaluation). The logged run shows strong adaptivity (94% early stopping; 4.57 CoT samples on average) but very low accuracy (5.33%, 8/150). We analyze this negative result, emphasizing failure modes in posterior construction, stopping calibration, and implementation mismatches between the intended Bayesian fusion rule and the executed update.

Keywords: keyword 1; keyword 2; keyword 3 (List three to ten pertinent keywords specific to the article; yet reasonably common within the subject discipline.)

1. Introduction

Chain-of-Thought (CoT) prompting has become a standard technique for eliciting multi-step reasoning from large language models (LLMs). By encouraging models to emit intermediate steps, CoT can substantially improve performance on arithmetic and symbolic benchmarks [**?**], and even a minimal trigger phrase can induce similar behavior in a zero-shot setting [**?**]. However, deploying CoT in real systems raises two intertwined concerns: cost and reliability.

First, deliberation is expensive. Generating multi-step rationales increases tokens, latency, and energy, and the extra computation is not always justified: many inputs are easy enough for a direct answer to succeed. Second, the presence of a rationale does not guarantee correctness. Models can generate explanations that are fluent yet weakly connected to the final answer, or that rationalize an answer after the fact. In such cases, aggregating multiple CoT samples can yield high confidence in an incorrect answer if many samples share correlated errors.

Received:
Revised:
Accepted:
Published:

A popular inference-time improvement is self-consistency: sample multiple CoT outputs and pick the plurality final answer. This reduces variance but implicitly treats all sampled traces as equally trustworthy evidence. In settings where unsupported but persuasive traces are common, exchangeable voting can amplify the wrong answer. Meanwhile, a complementary line of work argues that CoT can actively harm performance on tasks where added deliberation introduces noise or distracts from a correct heuristic, paralleling human overthinking [**?**]. This suggests that compute allocation should be selective rather than always-on.

This paper investigates a training-free approach to selective and quality-aware CoT at inference time. We frame inference as a sequential decision and evidence fusion problem. The method begins with a fast System-1 stage that samples short direct answers to form an empirical distribution over candidate numeric outputs. From this distribution we derive uncertainty statistics (maximum probability and entropy) and use a gate to decide whether the question warrants deliberation. If System-1 is stable, the method returns immediately; otherwise it enters System-2 and samples CoT traces. Crucially, System-2 does not treat each trace as an exchangeable vote. Instead, it estimates how much a given rationale supports its own proposed answer using a self-entailment likelihood ratio computed through teacher-forced log-likelihood with the same model. Conceptually, this operationalizes a coherence check: conditioning on the rationale should not decrease, and ideally should increase, the model's probability of the proposed answer. The method further supports early stopping when the posterior induced by accumulated evidence becomes sufficiently concentrated.

Our motivation is aligned with calls for Bayesian meta-reasoning as a unifying perspective on robust, generalizable LLM reasoning [**?**] and with recent interest in controlling inference-time "thinking speed" [**?**]. At the same time, we impose strict constraints: no fine-tuning, no additional models, and no external verifiers. This setting is attractive for broad applicability but leaves little room to correct systematic model errors.

We implement DT-SEACoT in a small, reproducible codebase and evaluate it on GSM8K, a grade-school math benchmark where CoT often helps but is costly [**?**]. The experimental design includes hyperparameter tuning on a held-out slice with Optuna and final reporting on a separate slice. The available run demonstrates that the method does allocate compute adaptively, frequently stopping after only a few CoT samples. However, the run also yields very low absolute accuracy. Rather than treating this as a mere failure, we present it as an academically useful negative result: it highlights how training-free posterior construction and stopping rules can be severely miscalibrated, and how small implementation choices (for example, how answer strings are represented for likelihood scoring) can dominate outcomes.

Contributions:

- **Adaptive training-free CoT inference.** We specify a training-free adaptive CoT inference procedure that combines uncertainty-based gating with a self-entailment likelihood ratio computed via teacher-forced scoring in the same LLM.
- **Reproducible GSM8K instantiation.** We document an end-to-end instantiation on GSM8K with `google/flan-t5-large`, including prompts, parsing rules, and an explicit early stopping criterion.
- **Negative result and analysis.** We report logged results from the provided run and analyze why strong adaptivity can coincide with poor task accuracy, emphasizing mismatches between the intended and implemented fusion rules and the need for calibrated sequential stopping.

Looking forward, our findings suggest several directions: integrating absolute answer likelihood and System-1 priors into evidence fusion (as intended by the original design),

improving answer string handling for scoring stability, and adding calibration diagnostics so that early stopping thresholds are meaningful rather than arbitrary.

2. Related Work

2.1. Chain-of-Thought prompting and zero-shot reasoning

CoT prompting showed that intermediate reasoning steps can unlock strong performance gains on arithmetic reasoning, including GSM8K, especially in large models [**?**]. Zero-shot-CoT demonstrated that even without demonstrations, a short trigger phrase can elicit step-by-step reasoning behavior [**?**]. Our work keeps the same inference-only spirit but addresses a different question: given that CoT is available, how should a system decide when to use it and how should it aggregate multiple traces?

2.2. When "thinking" hurts: overthinking and compute allocation

Prior evidence shows that CoT can reduce performance on problems where deliberation introduces noise, echoing human overthinking [**?**]. This observation motivates fast-to-slow patterns where a cheap initial policy is used unless uncertainty is high. DT-SEACoT instantiates this intuition using an empirical uncertainty estimate derived from multiple short direct samples. In contrast to approaches that always generate a rationale, our gate explicitly permits skipping CoT.

2.3. Meta-reasoning perspectives on LLM inference

A broader position advocates Bayesian meta-reasoning as a lens for robust and generalizable reasoning, emphasizing uncertainty and value of computation [**?**]. DT-SEACoT operationalizes a lightweight version of this idea at inference time: it maintains a belief over candidate answers, sequentially accumulates evidence, and provides an early stopping rule based on posterior concentration. Our contribution is not a new learned model, but rather a training-free procedure that can be layered on top of an existing LLM.

2.4. Controlling inference-time "thinking speed"

Work on controlling thinking speed argues that inference-time control of computation is important for practical deployment [**?**]. DT-SEACoT differs in three ways: it is training-free, uses a single fixed model for both generation and scoring, and grounds its adaptivity in a sequential evidence process rather than a purely prompt-level or length-level knob.

2.5. Prompt construction and evidence aggregation beyond exchangeable voting

Automatic prompt selection and construction can improve the raw quality of CoT generations [**?**]. This direction is complementary to ours. Even with a strong prompt, aggregation remains non-trivial: sampled traces can still contain correlated errors or unsupported conclusions. DT-SEACoT focuses on inference control and evidence weighting given fixed prompts, rather than searching for new prompts.

The defining contrast with standard self-consistency is that DT-SEACoT does not treat sampled rationales as equally reliable votes. It uses a self-entailment likelihood ratio to discount traces whose rationales do not increase support for their own answers. Unlike verifier-based approaches (not considered here), this signal is derived solely from the base model via teacher-forced likelihoods, preserving the training-free and single-model constraint.

3. Background

This section introduces the formal objects required to describe adaptive CoT inference and the specific GSM8K evaluation protocol used in our experiments.

3.1. Problem setting and notation

Let q denote a natural-language math word problem. A language model with parameters θ defines a conditional distribution over text completions. We focus on numeric answers y. In practice the model produces a completion t that may contain both a rationale and a final numeric answer; an answer extractor maps t to a parsed numeric prediction $\hat{y}$. The dataset provides a ground-truth numeric answer a extracted from the canonical GSM8K answer format `#### number`. A prediction is marked correct when the absolute difference $|\hat{y} - a|$ is less than 10^{-6}.

3.2. Direct vs. rationale-conditioned answer support

DT-SEACoT distinguishes two conditional answer supports. The direct support is $p_\theta(y \mid q)$, approximated by prompting the model to output only the final number. The rationale-conditioned support is $p_\theta(y \mid q, r)$, where r is a generated rationale text (in our implementation, the full CoT completion text). The core idea is to use the model's own conditional probabilities as a training-free consistency check.

3.3. Teacher-forced log-likelihood as a scoring primitive

To obtain numeric scores for candidate answers without additional models, we use teacher forcing to compute the log-likelihood of a target completion string given a prompt. Concretely, for a prompt s and target completion string u, we compute $\log p_\theta(u \mid s)$ by running the seq2seq model with u as the decoder labels and summing token log-probabilities (equivalently, negative token-level cross-entropy). This mechanism is implemented in the provided codebase for `google/flan-t5-large`.

3.4. Self-entailment likelihood ratio

For a given candidate answer y and rationale r, we define the self-entailment likelihood ratio as

$$\Delta\mathrm{LL}(y, r) = \log p_\theta(y \mid q, r) - \log p_\theta(y \mid q).$$

In our implementation, $\log p_\theta(y \mid q)$ is approximated by teacher forcing the answer string under the direct prompt, and $\log p_\theta(y \mid q, r)$ is approximated by teacher forcing the same answer string under a prompt that includes the rationale. Intuitively, if a rationale is internally coherent with its conclusion, conditioning on it should not decrease support for that conclusion.

3.5. Uncertainty from sampled direct answers

To decide whether to deliberate, System-1 samples N_0 short direct answers and forms an empirical distribution $p_0(y)$ based on counts over extracted numeric answers. Two scalar summaries used by the gate are (i) the maximum probability $\max_y p_0(y)$ and (ii) the entropy $H(p_0) = -\sum_y p_0(y) \log(p_0(y) + 10^{-10})$. These measures are simple, training-free proxies for uncertainty and disagreement.

3.6. Sequential evidence accumulation and early stopping

Adaptive CoT can be interpreted as sequentially collecting evidence about y until it is no longer worth paying for more samples. While the motivating design for DT-SEACoT is inspired by sequential testing ideas, the logged implementation uses a pragmatic rule: it maintains unnormalized answer weights derived from accumulated evidence and converts them to a normalized posterior over currently observed answers; sampling stops when the highest posterior mass exceeds a fixed threshold. This background framing clarifies both the intended behavior (selective compute) and the main risk (miscalibrated posteriors).

4. Method

We describe DT-SEACoT as implemented in the provided codebase, emphasizing the exact prompts, evidence calculations, aggregation rule, and early stopping mechanism. The method operates in two stages.

4.1. System-1: fast direct-answer sampling and gating

Given a question q, DT-SEACoT builds a direct prompt that requests only the final numeric answer:

> `Solve this math problem and give only the final numeric answer. Question: <q>. The answer is:`

It then samples n_0 independent completions at temperature `direct_temperature`, each capped by `direct_max_tokens`. Each completion is parsed into a float using a pattern-based extractor that recognizes formats such as `#### number`, `the answer is number`, or `= number` at the end of the string, with a fallback that selects the last number appearing in the text. From the multiset of parsed answers, it computes `max_prob` and `entropy` of the empirical distribution.

The gating decision is deterministic: deliberation (System-2) is triggered when `max_prob` is below `gate_confidence_threshold` or `entropy` is above `gate_entropy_threshold`. If the gate does not deliberate and at least one answer was parsed, the method returns the most frequent parsed direct answer. If no valid direct answer is parsed, the method proceeds to deliberation.

4.2. System-2: adaptive CoT sampling with self-entailment weighting

When System-2 is triggered, DT-SEACoT uses a CoT prompt that requests step-by-step reasoning and asks for the final numeric answer after `####`:

> `Solve this math problem step by step. Show your work and put your final numeric answer after ####. Question: <q>. Let's solve this step by step:`

It generates up to $K_{\max}$ samples at temperature `cot_temperature` with cap `cot_max_tokens`. For each generated completion t, it parses a numeric answer y. If parsing fails, the sample is discarded.

If `use_self_entailment` is enabled, DT-SEACoT computes a self-entailment score for this sample using teacher forcing in the same model. Let `answer_str` be the string form of the parsed float, `answer_str = str(y)` . It computes:

1. $\ell\ell_{\text{direct}} = \log p_\theta(\texttt{answer\_str} \mid \texttt{direct\_prompt})$
2. $\ell\ell_{\text{with\_rationale}} = \log p_\theta(\texttt{answer\_str} \mid \texttt{r2a\_prompt})$

where `r2a_prompt` has the form:

> `Given the reasoning below, what is the final numeric answer? Question: <q>. Reasoning: <t>. Therefore, the answer is:`

The evidence weight for the sample is $\Delta\ell\ell = \ell\ell_{\text{with\_rationale}} - \ell\ell_{\text{direct}}$. If `use_self_entailment` is disabled, `evidence_weight` is set to 0.0, yielding a uniform-weight aggregation variant.

4.3. Answer-level evidence fusion

The method maintains a dictionary mapping each observed answer value y to a log-weight. For the first observation of y, it sets `log_weight[`y`]` $\leftarrow$ `evidence_weight`. For subsequent observations of the same y, it updates `log_weight[`y`]` using `logaddexp(old, evidence_weight)`. This corresponds to summing unnormalized evidence contributions in the probability domain while maintaining numerical stability.

4.4. Early stopping

If `early_stop_enabled` is true, DT-SEACoT begins checking for early stopping after at least three CoT samples have been generated. It converts the current log-weights into a normalized posterior over the set of observed candidate answers by exponentiating after subtracting the maximum log-weight and normalizing. If the maximum posterior probability exceeds `early_stop_posterior_threshold`, sampling halts and the current top answer is returned.

4.5. Procedure summary

Algorithm 1 DT-SEACoT inference (as implemented)

Build `direct_prompt` from question q
Sample n_0 direct completions; parse numeric answers into multiset $\mathcal{Y}_0$
Form empirical distribution p_0 over values in $\mathcal{Y}_0$; compute `max_prob` and `entropy`
if $\mathcal{Y}_0 \neq \emptyset$ **and** `max_prob` $\geq$ `gate_confidence_threshold` **and** `entropy` $\leq$ `gate_entropy_threshold` **then**
  return most frequent $y \in \mathcal{Y}_0$
end if
Initialize map $w(y) \leftarrow -\infty$ for all observed y
for $k = 1$ to $K_{\max}$ **do**
  Sample CoT completion t; parse answer y; **continue** if parsing fails
  if `use_self_entailment` **then**
    Compute $\ell\ell_{\text{direct}} \leftarrow \log p_\theta(\texttt{str}(y) \mid \texttt{direct\_prompt})$
    Compute $\ell\ell_{\text{with\_rationale}} \leftarrow \log p_\theta(\texttt{str}(y) \mid \texttt{r2a\_prompt}(q, t))$
    $e \leftarrow \ell\ell_{\text{with\_rationale}} - \ell\ell_{\text{direct}}$
  else
    $e \leftarrow 0$
  end if
  $w(y) \leftarrow \log\left(\exp(w(y)) + \exp(e)\right)$ ▷ `logaddexp`
  if `early_stop_enabled` **and** $k \geq 3$ **then**
    Normalize w over observed answers to posterior π
    if $\max_y \pi(y) >$ `early_stop_posterior_threshold` **then**
      return $\arg\max_y \pi(y)$
    end if
  end if
end for
return $\arg\max_y \pi(y)$ over observed answers

4.6. Implementation note relative to the motivating design

The conceptual design motivating DT-SEACoT includes combining System-1 priors and absolute answer likelihood terms with the self-entailment ratio. The logged implementation uses $\Delta\ell\ell$ as the only evidence term and does not explicitly incorporate System-1 priors into System-2 weights. This difference matters because $\Delta\ell\ell$ captures a relative change in support under conditioning, but does not, by itself, ensure that an answer is plausible under the direct prompt.

5. Experimental Setup

We evaluate DT-SEACoT on GSM8K using a fixed pretrained model, with a protocol that separates hyperparameter tuning from final evaluation.

5.1. Dataset construction and splits

The pipeline loads the GSM8K dataset (`gsm8k`, configuration `main`) and uses the test split. It selects up to `max_samples` $= 200$ examples, shuffles them with a fixed seed 42, and

discards any examples whose gold answer cannot be parsed from the dataset-provided answer string. Gold answers are extracted using the canonical GSM8K pattern `#### number` and converted to floats. The resulting 200 examples are split into 50 validation examples (for hyperparameter tuning) and 150 test examples (for final evaluation).

5.2. Model and inference constraints

All runs use `google/flan-t5-large` as a single sequence-to-sequence model with no fine-tuning. The same model is used for (i) generation in System-1 and System-2 and (ii) teacher-forced log-likelihood computations for self-entailment scoring. The configuration supports CUDA execution with bfloat16 weights.

5.3. Methods under study

The experimental design includes two methods:

1. DT-SEACoT (proposed): direct-answer sampling, uncertainty-based gating, CoT sampling, self-entailment weighting, and posterior-threshold early stopping.
2. Fixed-K CoT self-consistency (baseline): sample k CoT completions and return the plurality-vote numeric answer.

While both implementations are present in the code, the metrics payload available for this paper includes only the proposed DT-SEACoT run, so our Results section reports only that method.

5.4. Prompts and answer extraction

Direct prompts explicitly request only the final numeric answer and end with `The answer is:`. CoT prompts explicitly request that the final numeric answer appears after `####`. Model outputs are parsed into floats using several patterns (including the GSM8K `####` pattern and a last-number fallback). Correctness is computed using absolute error less than 10^{-6}.

5.5. Hyperparameter tuning

The configuration enables Optuna with 20 trials for both DT-SEACoT and the baseline. For DT-SEACoT, the search space includes `gate_confidence_threshold`, `gate_entropy_threshold`, `early_stop_posterior_threshold`, and direct and CoT sampling temperatures. For the baseline, the search space includes the CoT sampling temperature. The evaluation payload does not include Optuna trial histories; only the final summary metrics for the DT-SEACoT run are available.

5.6. Efficiency measurement

The run logs average counts of direct samples and CoT samples used per question. It also logs `avg_tokens_per_question` computed as $(\texttt{avg\_direct\_samples} \times \texttt{direct\_max\_tokens}) + (\texttt{avg\_cot\_samples} \times \texttt{cot\_max\_tokens})$. This quantity is an upper-bound-style estimate based on configured maximum new tokens per generation, not a tokenizer-counted sum of actual generated tokens, and it does not include teacher-forced scoring costs.

6. Results

Only one run is available in the provided metrics payload: `proposed-flan-t5-large-gsm8k` (DT-SEACoT). No baseline self-consistency metrics are included, so we report absolute performance and internal behavior diagnostics for DT-SEACoT without claiming improvement over alternatives.

6.1. Overall accuracy

On the GSM8K test split of 150 examples, DT-SEACoT achieves accuracy 0.0533, corresponding to 8 correct answers out of 150. This indicates that the method, as configured and implemented in the available run, is not effective for GSM8K problem solving in absolute terms.

6.2. Adaptive compute behavior

Despite poor accuracy, the logged statistics show that DT-SEACoT does allocate computation adaptively:

- System-1 direct sampling: `avg_direct_samples` $= 5.0$, matching the configured sample count.
- Gating: `skip_rate` $= 0.0333$, implying that about 3.3% of items exit after System-1 and about 96.7% trigger System-2 deliberation.
- System-2 usage: `avg_cot_samples` $= 4.5667$, substantially below the configured maximum `k_max_cot_samples` $= 16$.
- Early stopping: `early_stop_rate` $= 0.94$, meaning early stopping triggers for most deliberative items.
- Total samples: `total_samples_per_question` $= 9.5667$ (direct plus CoT), reflecting the combination of fixed System-1 sampling and adaptive System-2 sampling.

6.3. Token-cost proxy and measurement limitations

The run reports `avg_tokens_per_question` $= 2073.6$. In the code, this quantity is computed from average sample counts multiplied by maximum new-token budgets per generation, rather than counting realized generated tokens with the tokenizer. Moreover, it excludes the additional teacher-forced likelihood computations used for self-entailment scoring. As a result, the logged token number should be interpreted only as a coarse proxy for compute, not as a precise measure of inference cost.

6.4. Summary figure

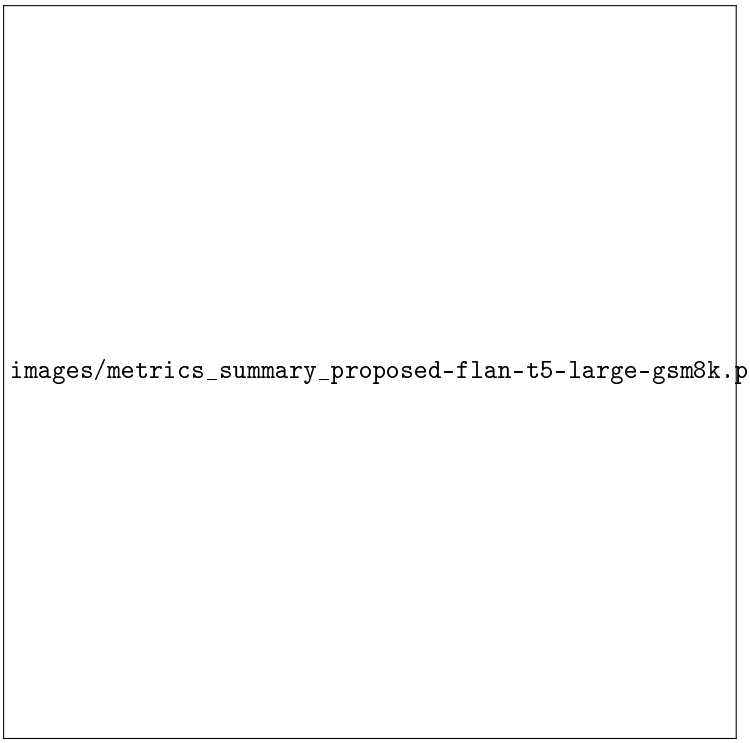

Figure 1. Summary metrics for DT-SEACoT on GSM8K for the run proposed-flan-t5-large-gsm8k; higher values indicate better performance for accuracy, while lower values indicate better performance for compute-related quantities (tokens and sample counts).

6.5. Why strong adaptivity can coexist with poor accuracy

The combination of 94% early stopping and 5.33% accuracy suggests that the internal posterior used to stop is severely misaligned with correctness: the method becomes confident quickly but is often wrong. The provided analysis points to two plausible contributors grounded in the implementation:

1. Evidence term mismatch: the intended DT-SEACoT design includes combining a System-1 prior and an absolute direct answer likelihood term with the self-entailment ratio. The implementation uses only the ratio $\Delta\ell\ell$, which measures relative change in support and does not ensure absolute plausibility of y under the direct prompt.
2. Answer string representation for scoring: teacher-forced likelihoods are computed on answer strings produced by `str(y)` where y is a parsed float. This can change formatting relative to what the model would naturally produce (for example, dropping trailing zeros), injecting noise into likelihood ratios.

6.6. Limitations of the current empirical evidence

Because baseline runs are not present, we cannot quantify a Pareto tradeoff relative to fixed-K self-consistency. We also do not have bootstrap confidence intervals, calibration metrics (such as expected calibration error), or ablation results in the provided logs. Consequently, the current evidence supports only a narrow conclusion: the logged DT-SEACoT run exhibits adaptive sampling behavior but fails to achieve meaningful GSM8K accuracy.

7. Discussion

8. Conclusions

We examined DT-SEACoT, a training-free inference procedure intended to make Chain-of-Thought prompting more metacognitively controlled by combining (i) an uncertainty-based gate derived from multiple short direct answers and (ii) self-entailment-weighted evidence aggregation computed through teacher-forced likelihoods in the same model. The broader aim is to improve reliability and efficiency simultaneously by allocating deliberation only when it is valuable and by discounting rationales that do not support their own conclusions.

On a GSM8K slice evaluated with `google/flan-t5-large` (50 items for tuning and 150 for final evaluation), the available run demonstrates the intended adaptive dynamics: deliberation is triggered for most items, but System-2 typically terminates early (94% early-stop rate) after an average of 4.57 CoT samples. However, the run achieves only 5.33% accuracy (8/150), indicating a severe failure of the overall inference procedure in this configuration.

This negative result is still informative. It highlights that training-free posterior construction from self-entailment signals can be miscalibrated and can drive confidently wrong early stopping. It also underscores the importance of aligning the implementation with the intended decision-theoretic update: incorporating System-1 priors and absolute answer likelihood terms, auditing answer-string formatting for teacher-forced scoring stability, and adding calibration diagnostics so that posterior thresholds correspond to meaningful error guarantees. These directions are necessary steps toward principled, training-free metacognitive control of CoT inference.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used: "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.", please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: This research received no external funding.

Data Availability Statement: All resources used in this study are openly available at

Acknowledgments: In this study, we automatically carried out a series of research processes—from hypothesis formulation to paper writing—using generative AI.

Conflicts of Interest: The authors declare no conflicts of interest.

. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models **2022**.
. Kojima, T.; Gu, S.S.; Reid, M.; Matsuo, Y.; Iwasawa, Y. Large Language Models are Zero-Shot Reasoners **2022**.
. Borges, Y.G.F.; Schouery, R.C.S.; Miyazawa, F.K. Mind Your Step (by Step): Chain-of-Thought can Reduce Performance on Tasks where Thinking Makes Humans Worse **2023**.
. Arvesú, J.; Ramírez-Aberasturis, A.M. Position: LLMs Need a Bayesian Meta-Reasoning Framework for More Robust and Generalizable Reasoning **2024**. https://doi.org/10.1080/10652469.2020.1830990.
. Goren, M.; Treister, E. Controlling Thinking Speed in Reasoning Models **2024**.
. Maltese, D.; Ogabi, C. Automatic Chain of Thought Prompting in Large Language Models **2023**.